

A New Retexturing Method for Virtual Fitting Room Using Kinect 2 Camera

Andres Traumann, Gholamreza Anbarjafari
iCV Group, IMS Lab, Institute of Technology,
University of Tartu, Tartu 50411, Estonia
{andres.traumann, sjafari}@ut.ee

Sergio Escalera
Computer Vision Center,
Universitat de Barcelona, Spain
sergio@maia.ub.es

Abstract

This research work proposes a new method for garment retexturing using a single static image along with depth information obtained using the Microsoft Kinect 2 camera. First the garment is segmented out from the image and texture domain coordinates are computed for each pixel of the shirt using 3D information. After that shading is applied on the new colours from the texture image by applying linear stretching of the luminance of the segmented garment. The proposed method is colour and pattern invariant and results in to visually realistic retexturing. The proposed method has been tested on various images and it is shown that it generally performs better and produces more realistic results compared to the state-of-the-art methods. The proposed method can be an application for the virtual fitting room.

1. Introduction

Retexturing is the process of replacing the texture of an object on an image or video with a new texture while considering the deformations and shading of the object. This process has several applications in entertainment, advertising, computer graphics and virtual reality. Retexturing enables to modify existing objects in movies and computer games. Increase interest in virtual fitting room has boost the amount of research work in this field [1, 2, 3, 4].

In the last decade, garment retexturing has attracted the interest of many researchers [5, 6, 7, 8, 9]. Many of these research works were using multi-view systems for cloth tracking and retexturing [9, 10, 11, 12, 13]. Optical flow has been widely used in current garment tracking and retexturing [14, 5, 15]. Scholz and Magnor [16] used optical flow to calculate 3D scene flow in a multi-view system and they improved their method by using colour-code with more code-words [17].

A method for retexturing non-rigid objects from a single view using colour markers was proposed by White and Forsyth [18]. However the aforementioned work was lim-

ited to recover irradiance to screen printing technique where the number of colours is finite. Pilet et al. in [19] were using a wide baseline matching algorithm and taking shadows and illumination into account. Guo, Wang et. al. [20]. 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 minimization problem. They solve the lighting problem by converting to $YCbCr$ colour space and using the Y component as the intensity value for the new texture colour.

Hilsmann and Eisert [15] used a dynamic texture overlay method from monocular images. They retexture garments by estimating the elastic deformations of cloth from a single camera and recovering illumination of the textured surface of a shirt in real time. Cushen and Nixon proposed a fusion of augmented reality and virtual try on [21]. They tried to recover a rectangular surface from a deformed arbitrarily shaped texture. Khan et al. also proposed a method for replacing one material with another by exploiting the fact that human vision is tolerant of certain physical inaccuracies [22].

The method we propose here has been developed with the focus on a virtual fitting room application. It replaces the texture of a shirt with a new custom texture from an image file. It is achieved by using the Microsoft Kinect 2 camera to obtain depth information about the scene.

In this work the segmentation of a shirt has been done by using the depth information of Kinect to effectively remove the background and manually marking some borders of the garment and the body. The texture coordinates are calculated using the information about the real world coordinates of an object corresponding to the pixel coordinates on the image. By considering only the X and Y real world coordinates when creating a map, it creates a very realistic perspective effect and the overall result looks very realistic. A lighting estimation is achieved by using equalization of the luminance information of the original image. The lighting estimation is used to shade the new colours of the texture image and obtain a realistic result where some parts of the

surface are in shadow while other parts are very bright.

Our results in comparison to state-of-the-art methods show significantly better results especially regarding the shading of the surface. As the visual quality of the retextured images greatly depends on how realistic the lighting effects like shadows and reflections are, it means that the proposed method often results in retextured images that have a higher visual quality than the results of the state of the art methods.

The rest of the paper is organized as follows: Next section gives a detailed description of the proposed retexturing method. Experimental results and visual comparisons are reported in section 3. Section 4 concludes the paper.

2. Proposed Method

The proposed method uses single images taken from the Microsoft Kinect 2 camera along with the depth information and real-world coordinates it provides. The method works on images that contain a person wearing a shirt, it replaces the texture of that shirt with a new texture from an input image file. The pseudocode of the retexturing algorithm is shown in Algorithm 1. The main application of the proposed technique is the virtual fitting room software where the person can change the texture of the garment of the 3D model.

The proposed method consists of three steps: the segmentation step, the retexturing step and the shading step. In the segmentation step a person is segmented out from the image using depth information. Prior information is used to segment out the correct part of the person corresponding to the garment. In the retexturing step the texture domain coordinates are calculated for each pixel in the area of interest. In the shading step the texture image is sampled and shading is applied to the colour information to make it look more realistic.

2.1. Segmentation

A person on the image can be segmented out very effectively using the depth information provided by Kinect. It is assumed that the person on the image is standing in front of the camera at a distance of up to 2.5 meters. That means that it is possible to disregard all the objects that are farther or closer, i.e. it is possible to threshold based on the depth information. As the person who is standing near the camera generally forms the largest object in the given distance range, the algorithm automatically chooses the largest object in the provided distance range and all the smaller objects in the same distance range are ignored. As a result of this procedure an image with removed background is obtained where the only object is the person who is used for garment retexturing.

The next step is to use the segmented part of the image to apply more accurate segmentation in order to obtain the

Algorithm 1: Proposed Retexturing Algorithm

Input: ColourImage, TextureImage,
CoordinateMapper, LumLow, LumHigh

Output: RetexturedImage

```

1 SegmIm ← segmentImage(ColourImage)
2 W ← ColourImage.width
3 H ← ColourImage.height
4 LumImage ← getLumImage(ColourImage)
5 HistogramEqualize(LumImage)
6 Lummin ← min{LumImage(x, y) :
   SegmIm(x, y) = 1}
7 Lummax ← max{LumImage(x, y) :
   SegmIm(x, y) = 1}
8 foreach (x, y) ∈ LumImage do
9   LumImage(x, y) ←
   LumLow +  $\frac{LumHigh - LumLow}{Lum_{max} - Lum_{min}} \cdot$ 
   (LumImage(x, y) - Lummin)
10 foreach (x, y) ∈ ColourImage do
11   i ← 0
12   if SegmIm(x, y) = 1 then
13     (X, Y, Z) ← CoordinateMapper(x, y)
14     XCoords[i] ← X
15     YCoords[i] ← Y
16     i ← i + 1
17 foreach (x, y) ∈ ColourImage do
18   if SegmIm(x, y) = 1 then
19     (X, Y, Z) ← CoordinateMapper(x, y)
20     u ← W  $\frac{X - \min(XCoords)}{\max(XCoords) - \min(XCoords)}$ 
21     v ← H  $\frac{Y - \min(YCoords)}{\max(YCoords) - \min(YCoords)}$ 
22     ColourImage(x, y) ←
      $\frac{1}{255} LumImage(x, y) \cdot TextureImage(u, v)$ 

```

exact area corresponding to the garment that is later going to be retextured. This step is generally much more complicated. In simpler cases it would be possible to use thresholding based on colour. However, our aim is to create a retexturing algorithm that would work on garments with different colours which might also contain a complicated texture on them. For that reason simple thresholding techniques can not be used.

To do the final segmentation we use prior information about which part of the previously segmented out person corresponds to the garment that needs to be retextured. The segmentation is performed by running a BFS-based algorithm starting from a point that is given in the prior information. The algorithm is able to move to each pixel in

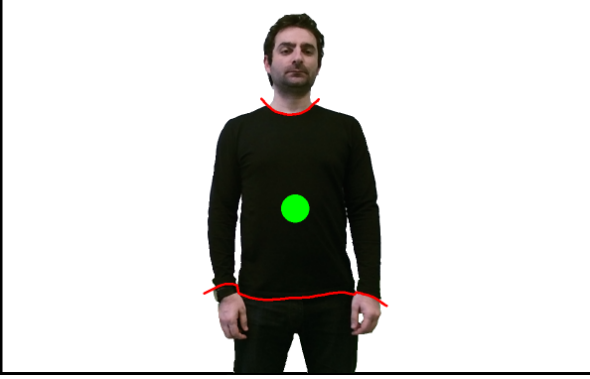


Figure 1. Interactive garment selection system enables the user to select the garment and to mark boundaries i.e. restricted areas. The green dot denotes the starting point of the BFS-based segmentation algorithm and the red lines are the restricted pixels that make sure the algorithm includes only the selected garment to the final segmentation.

the 4-neighbourhood of any given pixel that was previously reached while running the BFS-search, unless the neighbourhood pixel is marked as restricted according to the prior information. Finally the algorithm adds all the pixels that it was able to reach to the final segmented object, as shown in Figure 1.

2.2. Computation of Texture Coordinates

The Kinect API enables to translate screen coordinates of an image into real world coordinates. More specifically the following map is obtained:

$$\omega : (x, y) \rightarrow (X, Y, Z), \quad (1)$$

where $x, y \in \mathbb{N}$ denote the screen coordinates of the image and $X, Y, Z \in \mathbb{R}$ denote the real-world coordinates corresponding to image coordinates (x, y) . The objective is to map the screen coordinates into texture domain coordinates (u, v) . It is achieved using the following two functions:

$$u = f_u(x, y) = W \frac{\omega_x(x, y) - X_{min}}{X_{max} - X_{min}} \quad (2)$$

$$v = f_v(x, y) = H \frac{\omega_y(x, y) - Y_{min}}{Y_{max} - Y_{min}}, \quad (3)$$

where u and v denote the new texture domain coordinates, $\omega_x(x, y)$ and $\omega_y(x, y)$ denote the X and Y coordinates respectively obtained from the map $\omega(x, y)$, X_{min} , Y_{min} , X_{max} , Y_{max} denote the smallest and largest X and Y coordinates of the pixels from the blob obtained in the last step and W and H are the width and height of the texture image respectively. Note that the texture space coordinates that are calculated in this way do not depend on the real-world Z coordinate. The way the map from (x, y) to (u, v) is defined ensures that the so-called perspective effect can be observed.

2.3. Shading

Now that the texture space coordinate for each pixel have been specified, it becomes necessary to calculate their new colour values. A quite standard approach here would be to use the already existing colour information from the original image. However, this has a drawback of not being totally colour invariant, as darker colours have a much smaller amount of intensity changes. Instead we are using a linear luminance stretching technique that works by applying it to the intensity component image of the scene.

It is clear that a black T-shirt has a very low colour intensity in most parts. To achieve a realistic lighting intensity estimation, histogram equalization is applied on the previous intensity values $(r' + g' + b')/3$ of the black shirt. Finally the resulting equalized values are used as lighting intensity values as an approximation of l to finally calculate the new colour $c_n = lc = (lr, lg, lb)$. It is also helpful to add constant upper and lower bounds α and β and to stretch the equalized histogram between these values so that $\alpha \leq c_n \leq \beta$. It avoids darker colours going absolutely black and it also avoids creating unrealistically bright areas. That way this shading method can work very well in a wide range of conditions without depending on the original colour.

3. Results and Discussion

The experimental results of the proposed method are shown in Figures 2. Images (a) – (d) are the original images used for retexturing. Images (e) – (h) are retextured using the proposed method and images (i) – (l) are retextured with the method of Khan, Reinhard et. al. [22] using the same texture image to provide some visual comparison to another state of the art retexturing method. Images (m) – (p) are also created using the proposed method to provide some more examples created with the proposed retexturing method using various texture images as input.

The method of Khan and Reinhard [22] was selected for comparison, as it operates under similar conditions as the proposed method: It uses static images as input and does not depend on any prior knowledge about the surface pattern i.e. it does not use any prepared markers for surface reconstruction.

The visual comparison shows that the method of Khan, Reinhard depends greatly on the original shirt colour, whereas the proposed method does not depend on the original colour or texture of the shirt. However, the state-of-the-art method fails to produce realistic results when the original colour of a shirt is dark and does not have much changes in colour intensity, it results in images where the texture colour is almost uniform and does not have any visible shadows on the garment. The visual results clearly show that the proposed method performs better than the other state-of-

the-art method and in various conditions it produces more stable results, it is clear that the proposed method can effectively be used to perform realistic retexturing.

4. Conclusion

In this paper a method for shirt retexturing using the Microsoft Kinect 2 camera that provides colour and depth information is proposed. A shirt was segmented out from an image based on its depth and manually added colour markers. Depth and corresponding 3D coordinates of the scene were used to create a texture map from the surface of a shirt to the image containing the new texture. Shading of the new texture was done by using the linear stretching of luminance information. In the future work we are planning to address the problem of self-occlusions.

5. Acknowledgement

This research is funded by ERDF program "Estonian higher education information and communications technology and research and development activities state program 2011-2015 (ICT program)", the Estonian Research Council Grant PUT (PUT638), and Spanish project TIN2013-43478-P.

References

- [1] A. Taguchi, T. Aoki, and H. Yasuda, "A study on real-time virtual clothing system based on two-dimensional plane model," in *Information and Telecommunication Technologies, 2005. APSITT 2005 Proceedings. 6th Asia-Pacific Symposium on*. IEEE, 2005, pp. 126–130.
- [2] J. Ehara and H. Saito, "Texture overlay for virtual clothing based on pca of silhouettes," in *Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality*. IEEE Computer Society, 2006, pp. 139–142.
- [3] F. Pereira, C. Silva, and M. Alves, "Virtual fitting room augmented reality techniques for e-commerce," in *ENTERprise Information Systems*. Springer, 2011, pp. 62–71.
- [4] R. Kaur, "Virtual fitting room," Dec. 18 2014, uS Patent 20,140,368,499.
- [5] I. Guskov, S. Klibanov, and B. Bryant, "Trackable surfaces," in *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 2003, pp. 251–257.
- [6] Q.-S. Le and M.-T. Tran, "Overlay upper clothing textures to still images based on human pose estimation," *Journal of Mobile Multimedia*, vol. 10, no. 1&2, pp. 43–57, 2014.
- [7] D. Pritchard and W. Heidrich, "Cloth motion capture," in *Computer Graphics Forum*, vol. 22, no. 3. Wiley Online Library, 2003, pp. 263–271.
- [8] P. Eisert and A. Hilsmann, "Realistic virtual try-on of clothes using real-time augmented reality methods," *E-Letter*, 2011.
- [9] R. White, K. Crane, and D. A. Forsyth, "Capturing and animating occluded cloth," in *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3. ACM, 2007, p. 34.
- [10] D. Bradley, T. Popa, A. Sheffer, W. Heidrich, and T. Boubekeur, "Markerless garment capture," in *ACM Transactions on Graphics (TOG)*, vol. 27, no. 3. ACM, 2008, p. 99.
- [11] V. Scholz and M. Magnor, "Multi-view video capture of garment motion," in *Proc. IEEE Workshop on Content Generation and Coding for 3D-Television*, 2006, pp. 1–4.
- [12] A. Del Bue and A. Bartoli, "Multiview 3d warps," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 675–682.
- [13] J. Shen, X. Yan, L. Chen, H. Sun, and X. Li, "Re-texturing by intrinsic video," *Information Sciences*, vol. 281, pp. 726–735, 2014.
- [14] A. Hilsmann and P. Eisert, "Optical flow based tracking and retexturing of garments," in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*. IEEE, 2008, pp. 845–848.
- [15] —, "Tracking and retexturing cloth for real-time virtual clothing applications," in *Computer Vision/Computer Graphics Collaboration Techniques*. Springer, 2009, pp. 94–105.
- [16] V. Scholz and M. A. Magnor, "Cloth motion from optical flow," in *VMV*, vol. 4, 2004, pp. 117–124.
- [17] V. Scholz, T. Stich, M. Keckeisen, M. Wacker, and M. Magnor, "Garment motion capture using color-coded patterns," in *Computer Graphics Forum*, vol. 24, no. 3. Wiley Online Library, 2005, pp. 439–447.
- [18] R. White and D. Forsyth, "Retexturing single views using texture and shading," in *Computer Vision—ECCV 2006*. Springer, 2006, pp. 70–81.
- [19] J. Pilet, V. Lepetit, and P. Fua, "Fast non-rigid surface detection, registration and realistic augmentation," *International Journal of Computer Vision*, vol. 76, no. 2, pp. 109–122, 2008.
- [20] Y. Guo, J. Wang, X. Zeng, Z. Xie, H. Sun, and Q. Peng, "Image and video retexturing," *Computer Animation and Virtual Worlds*, vol. 16, no. 3-4, pp. 451–461, 2005.
- [21] G. A. Cushen and M. S. Nixon, "Markerless real-time garment retexturing from monocular 3d reconstruction," in *Signal and Image Processing Applications (ICSIPA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 88–93.
- [22] E. A. Khan, E. Reinhard, R. W. Fleming, and H. H. Bühlhoff, "Image-based material editing," *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3, pp. 654–663, 2006.

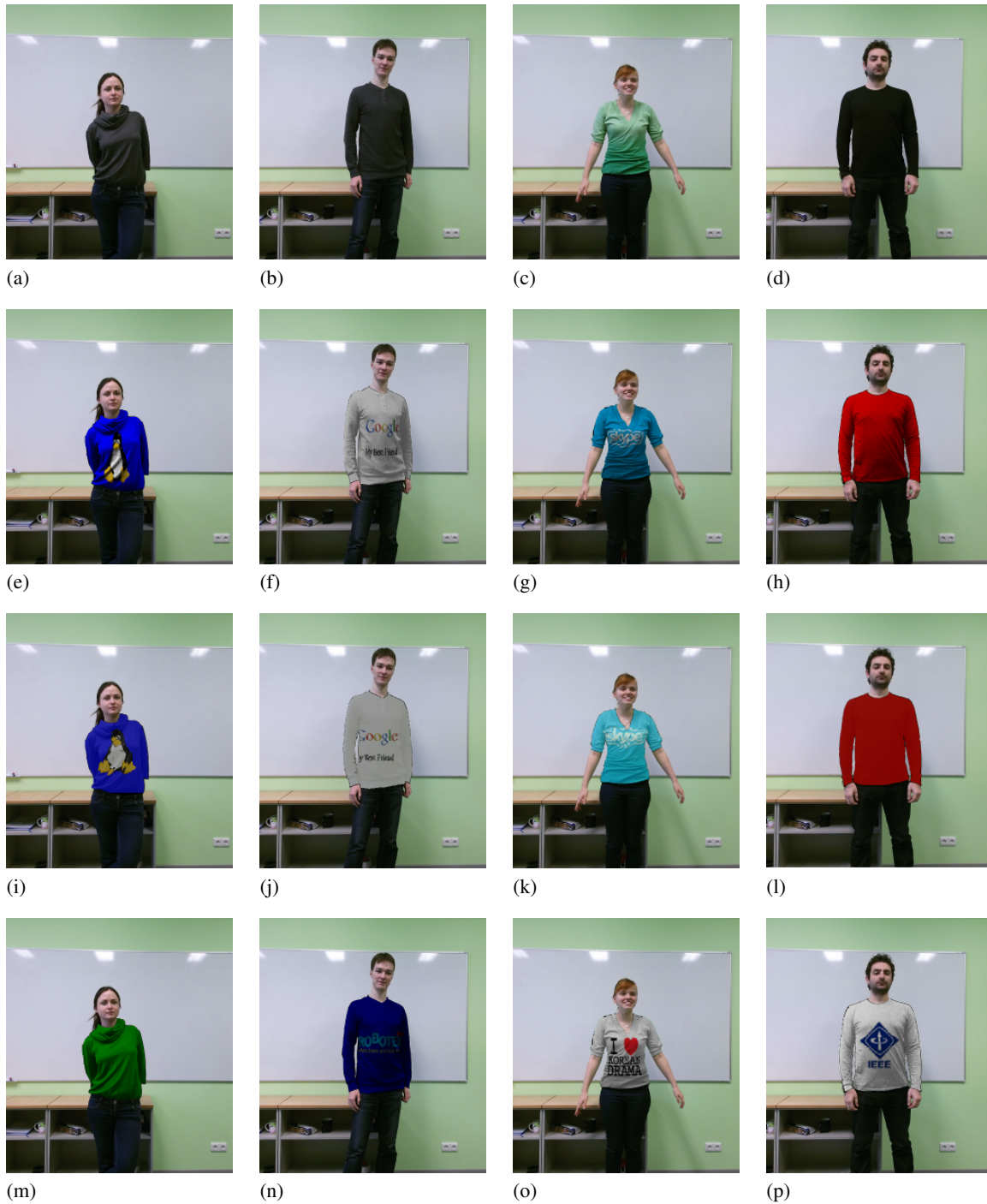


Figure 2. Images (a) – (d) are the original images used for retexturing. Images (e) – (h) are retextured using the proposed method and images (i) – (l) are retextured with the method of Khan, Reinhard et. al. [22] using the same texture image for visual comparison. The last images (m) – (p) are also obtained using the proposed method to provide a few more examples of our method.