NRC Publications Archive
Archives des publications du CNRC

Augmenting Non-Rigid Objects with Realistic Lighting

Bradley, D.; Roth, Gerhard

For the publisher’s version, please access the DOI link below./ Pour consulter la version de l’éditeur, utilisez le lien DOI ci-dessous.

https://doi.org/10.4224/5763100

NRC Publications Record / Notice d'Archives des publications de CNRC:
https://nrc-publications.canada.ca/eng/view/object/?id=cfdce6a8-06a2-41c2-a3fc-f770f528ebcc
https://publications-cnrc.canada.ca/fra/voir/objet/?id=cfdce6a8-06a2-41c2-a3fc-f770f528ebcc

Access and use of this website and the material on it are subject to the Terms and Conditions set forth at
https://nrc-publications.canada.ca/eng/copyright
READ THESE TERMS AND CONDITIONS CAREFULLY BEFORE USING THIS WEBSITE.

L’accès à ce site Web et l’utilisation de son contenu sont assujettis aux conditions présentées dans le site
https://publications-cnrc.canada.ca/fra/droits
LISEZ CES CONDITIONS ATTENTIVEMENT AVANT D’UTILISER CE SITE WEB.

Questions? Contact the NRC Publications Archive team at
PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca. If you wish to email the authors directly, please see the first page of the publication for their contact information.

Vous avez des questions? Nous pouvons vous aider. Pour communiquer directement avec un auteur, consultez la première page de la revue dans laquelle son article a été publié afin de trouver ses coordonnées. Si vous n’arrivez pas à les repérer, communiquez avec nous à PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca.

National Research Council Canada
Conseil national de recherches Canada

National Research
Council Canada

Institute for
Information Technology

Conseil national
de recherches Canada

Institut de technologie
de l'information

Augmenting Non-Rigid Objects with Realistic Lighting *

Bradley, D,. Roth, G.
October 2004

* published as NRC/ERB-1116. October 28, 2004. 19 Pages. NRC 47398.

National Research Council Canada
Institute for Information Technology

Conseil national de recherches Canada
Institut de technologie de l'information

ERB-1116

Augmenting Non-Rigid Objects with Realistic Lighting

Bradley, D., Roth, G.
October 2004

NRC 47398

TECHNICAL REPORT

NRC 47398/ERB 1116
Printed October 2004

Augmenting Non-Rigid Objects with Realistic Lighting

Derek Bradley
Gerhard Roth

Computational Video Group
Institute for Information Technology
National Research Council Canada
Montreal Road, Building M-50
Ottawa, Ontario, Canada K1A 0R6

NRC 47398/ERB 1116
Printed October 2004

Augmenting Non-Rigid Objects with Realistic Lighting

Derek Bradley
Gerhard Roth
Computational Video Group
Institute for Information Technology
National Research Council Canada

Abstract

In augmented reality, virtual objects are added to the real world by superimposing them onto a video stream or a head-mounted display in real-time. Typical augmented reality applications track 2D patterns on rigid planar objects in order to acquire the pose of the camera in the scene. Rigid augmentations are then performed on the planar objects. We present a method to track non-rigid objects such as cloth, and to perform flexible augmentations on the cloth while stretching and rippling it in real-time, using a single camera. In addition, we present a simple technique to apply real-world illumination to the augmentation, which greatly improves perception and realism. Results show believable real-time augmentations of a non-rigid object, even under extreme lighting conditions.

Contents

List of Figures

1 Introduction

One of the main crossovers between computer vision and computer graphics is the field of augmented reality (AR). Augmented reality is the concept of adding virtual objects to the real environment. Virtual objects can be as simple as 2D textual labels to identify real objects, or as complicated as a full 3D scene.

There exists a wide range of AR applications that benefit military operations, tourism, education, medical procedures and the entertainment industry, just to name a few. Azuma et al. give a good summary of recent AR research and applications [2].

The AR process consists of determining the camera pose (its location and orientation in space) and then rendering virtual objects in the scene. In many AR applications, augmentations are aligned with a 2D plane. This plane can be determined by rigid planar patterns on real objects, or in the case of markerless AR, planes aligned with rigid real objects in the scene [4, 27, 9]. We present a method to perform real-time 2D augmentations on non-rigid objects such as a flexible piece of cloth using a single camera.

The second contribution of this paper involves the use of real scene lighting to shade the virtual augmentation. Often, virtual object shading is performed statically. Static object shading deters from realism in AR systems, especially when real-world lighting conditions are variable. The addition of realistic lighting to AR scenes increases perception of spatial relations between real and virtual objects [22]. Previous research has provided some techniques for establishing realistic shading between real and virtual objects, however these techniques are over-complicated for our AR application. We present a simple method to illuminate the non-rigid augmentation using the lighting from the physical environment. Our method uses the intensity map of the input image without any extra processing, and it makes use of the Gouraud shading algorithm [12]. This technique greatly improves the realism of the augmentation, especially under variable lighting conditions.

The remainder of this paper is organized as follows. In section 2 we discuss the related work in non-rigid object tracking and common illumination between real and virtual scenes. In section 3 we present our method for tracking non-rigid objects for AR. Section 4 describes our technique to render augmentations on cloth using realistic lighting. Section 5 concludes with future improvements that can be made to our system.

2 Related work

This section outlines related work in tracking non-rigid objects such as cloth, and related work in establishing common illumination for AR scenes.

2.1 Cloth tracking

In order to render augmentations on a non-rigid object such as cloth, this object must be accurately tracked in a real-time video sequence. Cloth tracking is a very difficult problem due to its flexibility and self-occlusion properties.

The first results in cloth tracking from a video sequence were presented by Guskov as an extension to facial animation [13, 16]. In this work, a t-shirt marked with a checkerboard grid of black quads is tracked using a single camera. Quads are tracked individually to allow for occlusions. However, the process requires user interaction to initiate the tracking, which occurs offline after the entire video sequence has been captured. Unlike our work, the motivating goal of this research is 3D shape reconstruction using multiple cameras, not real-time interactive augmented reality.

Guskov improved on his initial results of non-rigid surface acquisition with a real-time tracking procedure for arbitrary quad-marked surfaces [14, 15]. Quads are tracked individually in each frame, using spatial prediction and spatial coherence. Results show efficient tracking of spheres, gloves and t-shirts for the purposes of motion capture, however no attempts at non-rigid augmentations were performed.

Pritchard and Heidrich show impressive results over Guskov's initial method of motion capture [26]. Their system recovers geometry using stereo vision and then applies the Scale Invariant Feature Transform (SIFT) [21] to identify a pattern printed on the cloth. However like Guskov, their goal is to obtain cloth motion capture, so the system is not real-time and does not include augmentations.

2.2 Common illumination

Establishing common illumination between the real and virtual environments is a problem that is generally divided into four categories. Shadows must be cast from real objects onto real objects, from virtual objects onto virtual objects, from virtual objects onto real objects, and from real objects onto virtual objects [17]. Shadows from real objects cast onto other real objects come for free, provided that the natural scene lighting is not modified virtually. Shadows from virtual objects cast onto other virtual objects can be dealt with using standard graphical techniques for real-time shading. However, the latter two cases are the ones that are necessary in order to achieve a high degree of realism in a mixed reality environment.

A substantial amount of research has been performed by Drettakis et al. [7] and Loscos et al. [20, 19] in the field of common illumination for AR systems and interactive virtual relighting of real scenes. Based on the work of Fournier et al. [8], Loscos and Drettakis virtually modify the lighting and geometry of a static scene by constructing a model from multiple photographs and then using a hierarchical radiosity algorithm with shaft data structures for illumination. Virtual objects and lights can be added and updated, and real objects and lights can be removed at less than real-time frame rates.

Debevec developed an algorithm to add virtual objects to a real photograph using a high dynamic range light-based model to illuminate the new objects [6]. A mirrored sphere is used as a light probe to gather the light model near the location of the synthetic objects to be inserted. The reflectance model and geometric model of the scene are estimated. Virtual objects are rendered using global illumination and then composited onto the photograph using differential rendering.

Stauder estimates the intensity and direction of a distant point light and the ambient light in a video frame by examining the two succeeding images [28]. Scene objects are modeled using ellipsoid-like 3D models, and the illumination parameters are estimated from the shape, motion and displacement of the models in the image sequence. Results show moving virtual and real objects with realistic lighting, however the system is restricted to non-occluding, simple, moving, rigid objects.

Another solution to adding dynamic virtual objects to a static real scene was proposed by Gibson and Murta. [11]. In this system, distant real world illumination is captured by an omni-directional image and then basis radiance maps are pre-computed. Virtual objects are added to the scene and illuminated using sphere mapping, which yields much faster frame-rates than global illumination algorithms. Shadows are cast from virtual objects to real objects, from real objects to virtual objects, and from virtual objects onto themselves in real-time. Gibson et al. [10] improve upon previous results by removing the assumption that light sources are distant. In the new system, hierarchical shaft data structures are used to subdivide light transport paths and determine sources of light occluded by synthetic objects. Standard graphics hardware is used to render shadows, and the algorithm is capable of trading shadow accuracy for rendering speed. However, the system is restricted such that the only objects that can move are the synthetic ones.

Wittkämper et al. simulate complex realistic lighting for virtual objects on a mixed reality stage [32]. Their idea is that the real lighting environment must be completely duplicated virtually in order for the virtual objects to look realistic. However, complex lighting with hundreds of lights on a stage is too costly to render in real-time. Therefore, the authors derive a method to merge virtual light sources together in an effort to minimize the number of required virtual lights while maximizing realism.

Naemura et al. [24] introduce the concepts of a “Virtual Light” and a “Virtual Shadow” for mixed reality environments. In this context, a virtual light is a hand-held flashlight-like device that casts shadows of real objects onto virtual objects, virtual objects onto real objects and virtual objects onto other virtual objects. The authors use three different virtual light devices, namely a 3D sensor, a sensor with an attached camera, and a light projector. In the latter case, users must wear an HMD to see the virtual objects. The system is based on the fact that the virtual light position and orientation is known.

The most recent work on establishing common illumination for mixed reality scenes was performed by Haller at al. [17]. In this research, the authors modify a real-time shadow volume algorithm that is typically used for computer generated scenes to work in mixed reality applications. Real objects are represented by virtual "phantom" objects, so authoring a new scene must be done manually and is limited to a single light source. Real objects require markers to determine their position and orientation, and if moved, the shadow volume must be recalculated each frame. However, real-time results of shadow casting from all four shading categories mentioned above are demonstrated.

Since our goal is to perform augmentations on cloth, common illumination can be achieved by solving only one of the four shading problems mentioned above, that of casting shadows from real objects onto the virtual augmentation. Therefore, we present a simpler approach than previous methods of establishing common illumination. Our method, which requires only intensity values of the input image, takes into account multiple real light sources, requires no pre-processing or 3D model building, creates no shadow volumes or virtual lights, and yet still displays augmentations with realistic lighting in a real-time interactive application.

3 Non-rigid tracking

Tracking a non-rigid object, such as a piece of cloth, is a difficult task for real-time applications. Cloth is very flexible and it has the ability to stretch, ripple and self-occlude. Most cloth tracking methods are performed offline. However, for the purpose of non-rigid augmentations, a dense representation of the cloth model is not required. We propose that realistic non-rigid 2D augmentations can be performed in real-time with only a sparse representation of the cloth. Our augmentations take into account the flexibility of the cloth, including stretching and rippling; however correctly augmenting with self-occlusions has yet to be solved.

Transitioning from augmentations on simple rigid objects to a full non-rigid augmentation on a piece of cloth is a large step. For this reason, we first establish the tracking method and flexible augmentation on a non-rigid piece of paper. The paper can bend and ripple similar to cloth, so it is a logical starting point.

3.1 Tracking method

In order to obtain a sparse representation of the object to be tracked, a small number of self-identifiable markers are placed on the paper or cloth. We use 16 small square patterns arranged in a four by four grid (see Figure 1). Each pattern is unique and the topology of grid is known apriori. The markers consist of a self-identifiable grayscale pattern surrounded by a small black border, which allows us to use ARToolKit [1] to quickly find the four corners of each marker in the scene. The markers are small enough that they can still be found under some flexible object conditions.

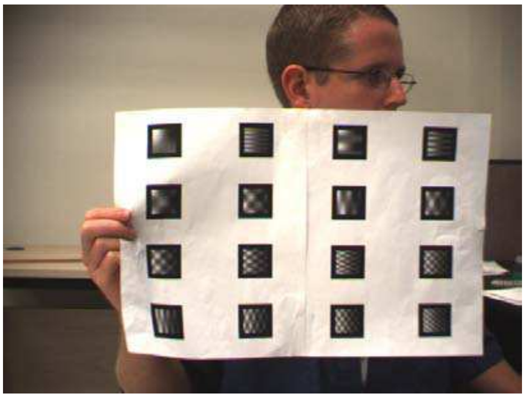
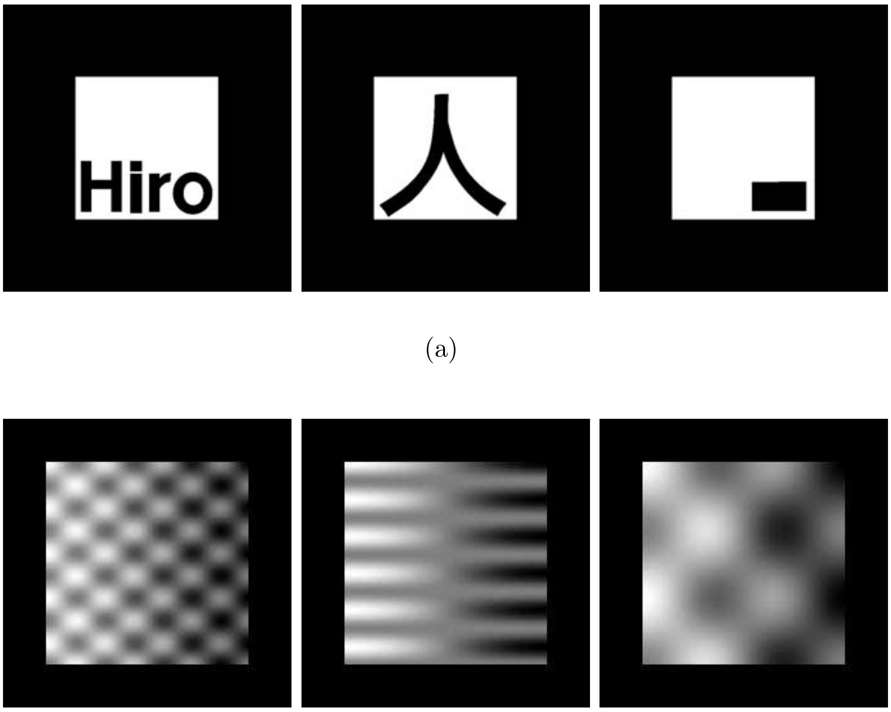


Figure 1: Markers to track on non-rigid object

The use of self-identifying patterns is common in AR applications, however it is not so common to require the correct identification of 16 patterns in one video frame simultaneously. Some example ARToolKit markers are shown in Figure 2 (a). These patterns are suitable when there are only a small number of markers to be detected. However, generating 16 patterns of this type can lead to false-positives and mis-identification in the tracking process. To establish more robust identification, we use patterns constructed from orthogonal Discrete Cosine Transform (DCT) basis images similar to Owen et al. [25]. Some examples of our patterns are shown in Figure 2 (b). To choose the 16 patterns that are most easily identifiable simultaneously, we attempted to track 47 different DCT-generated patterns and then we chose the 16 patterns with the best tracking confidences, as defined by ARToolKit.

The process of tracking the cloth starts by locating possible markers in the scene, based on their black square borders. The input image from the video stream is binarized by thresholding the intensity map, and then connected regions of pixels are segmented into blobs. This is accomplished by examining the neighbourhood around each non-zero pixel and establishing a list of connected components. For each blob, the contour of the external boundary is extracted, and then a filter is applied to remove regions that do not have four-sided external boundaries. The result of this first processing stage is a set of possible marker locations. Figure 3 illustrates these steps to locate possible markers on a flexible sheet of paper.

Now each possible marker is examined individually. Markers are identified based on their unique interior pattern. The first step is to remove the perspective distortion



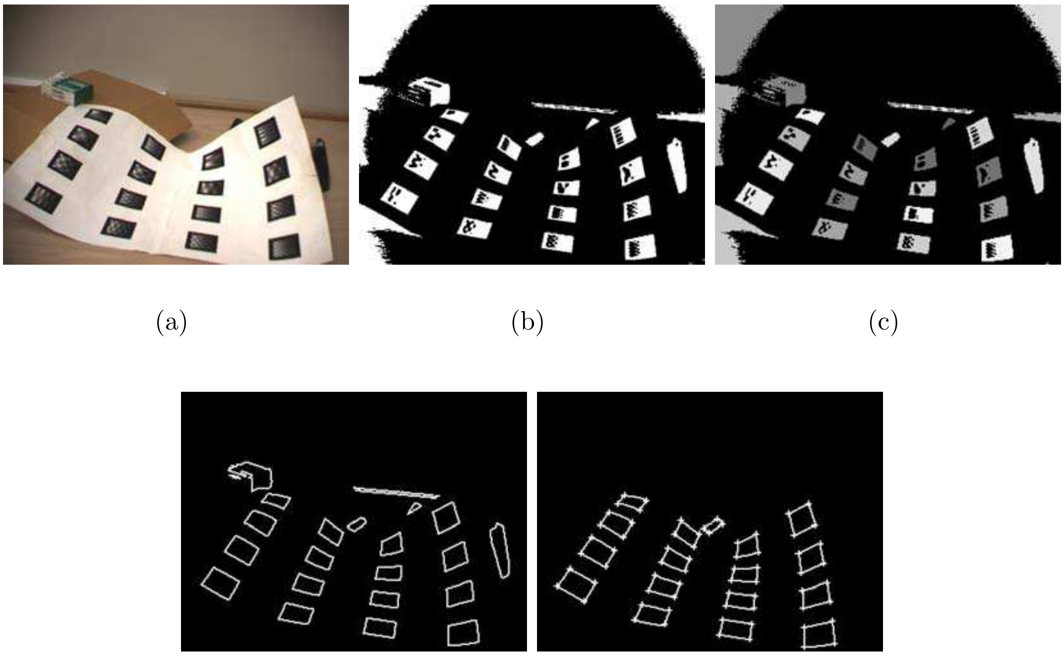
(a)

(b)

Figure 2: Example marker patterns

of the pattern by applying a transformation to a 32 x 32 sample grid of the region. Specifically, the transformation is a one-to-one mapping called a *homography*, which is defined by a 3 x 3 matrix [18]. The homography is computed by mapping the four corners of the marker in screen coordinates to the four corners in *normalized pattern-space*. Normalized pattern-space is the coordinate space where the lower left corner of the marker has a value of (0,0) and the upper right corner has a value of (1,1). Standard mathematical techniques are used to compute the homography. Once computed, each pixel on the screen maps to exactly one point in pattern-space and vice-versa, providing an orthogonal view of the marker pattern. Each possible pattern is then compared to a set of stored images of the known patterns to be detected. The patterns are compared at all four possible orientations, and a confidence value for each comparison is computed as the normalized dot product between the 32 x 32 vector and the stored image. If any comparison generates a confidence value greater than a certain threshold then it is claimed to be a match and the four corners of the marker in screen coordinates are returned.

Since the cloth contains 16 unique markers, the tracking procedure can find up to



(a) (b) (c)

(d) (e)

Figure 3: Establishing possible marker locations. a) input image; b) binarized image; c) connected components (shaded differently for visualization); d) region contours; e)four-sided regions.

64 pixel locations that map to known locations on the cloth. These points are then passed to the augmentation process, described in section 4.

3.2 Adaptive thresholding

The first step in the tracking process described above is to binarize the input image by thresholding the intensity map. This can be accomplished by choosing a global threshold value, **T**, and then setting all pixels in the intensity image that are greater than **T** to 1 and all pixels that are less than **T** to 0. This is known as *global thresholding*. However, global thresholding can be problematic in many computer vision applications, especially under variable or extreme lighting conditions. For example, consider Figure 4. The input image (Figure 4(a)) shows a particular illumination condition where part of the cloth to be tracked is brightly lit and another part is dimly lit. Using global thresholding, the darker markers can be segmented correctly with a threshold value of $\mathbf{T_1} = \mathbf{54}$, however the brighter markers are lost completely

(Figure 4(b)). Similarly, the brighter markers can be correctly segmented using a threshold value of $\mathbf{T_2 = 134}$, however the darker markers are then lost (Figure 4(c)). This illustrates the need for a technique with better performance than global thresholding. The answer is *adaptive thresholding.*

(a) (b) (c)

Figure 4: Global thresholding. a) input image; b) threshold for dimly lit markers ($T_1 = 54$); c) threshold for brightly lit markers ($T_2 = 134$).

Adaptive thresholding is similar to global thresholding except that the threshold value changes dynamically over the image. This method can accommodate variable lighting conditions such as strong illumination gradients or shadows. A number of different techniques exist to perform adaptive thresholding [3, 23, 31, 33]. In our system, we extend the method of Wellner [31]. In Wellner's algorithm, a moving average of the last **s** pixels is calculated while traversing the image. If the value of the current pixel is **t** percent lower than the average then it is set to black, otherwise it is set to white. The advantage of this method is that only a single pass through the image is required. Wellner uses 1/8th of the image width for the value of **s** and 15 for the value of **t**. However, this technique yields different results depending on whether the image is scanned from left to right, from right to left or alternating from the left and from the right, often resulting in unwanted artifacts. Taking the previous line of approximate averages into account when processing each scan line will help remove the artifacts without requiring additional passes through the image. However the process becomes more complex and different outcomes will still be seen depending on whether the image is scanned from the top to the bottom or vice versa. We chose to extend this solution to a very simple approach that produces the same output independent of how the image is processed, and we sacrifice only one additional pass through the image. Instead of computing a running average of the last **s** pixels seen, we pre-compute the average of an **s** x **s** window of pixels around each pixel, which is equivalent to blurring the input image. This average computation can be accomplished in linear time by using the *integral image* of the input [5, 29, 30]. The integral image technique can be used whenever we have a function from pixels to real numbers $\mathbf{f(x, y)}$ (for instance, pixel intensity), and we wish to compute the sum of this function over some rectangular region of the image. Without an integral image, this can be accomplished in linear time per rectangle by calculating the value of the function for each pixel individually. However, if we need to compute the sum over multiple different rectangular windows, we can use an integral image and achieve a constant number of operations per rectangle with only a linear amount of pre-processing to compute the integral image. To compute the integral image, we store at each location, $\mathbf{I(x, y)}$, the sum of all $\mathbf{f(x, y)}$ terms to the left and above the pixel $\mathbf{(x, y)}$. Figure 5 illustrates the computation of the integral image. The integral image can be computed in linear time using the following equation for each pixel, taking into account the image border cases;

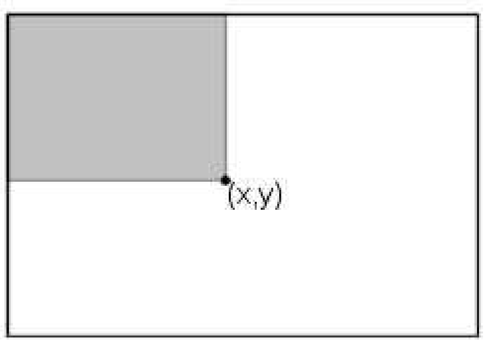


Figure 5: Computing the integral image

$$\mathbf{I(x, y) = f(x, y) + I(x - 1, y) + I(x, y - 1) - I(x - 1, y - 1)}$$

Once we have the integral image, the sum of the function for any arbitrary rectangle $\mathbf{(x_1, y_1)}$ to $\mathbf{(x_2, y_2)}$ can be computed in constant time using the following equation, again taking into account the image border cases;

$$\sum_{x_1 <= x <= x_2, y_1 <= y <= y_2} \mathbf{f(x, y) = I(x_2, y_2) - I(x_2, y_1 - 1) - I(x_1 - 1, y_2) + I(x_1 - 1, y_1 - 1)}$$

Figure 6 illustrates that the above equation to compute the sum of $\mathbf{f(x, y)}$ over the rectangle D is equivalent to computing the sums over the rectangles (A+B+C+D)-(A+B)-(A+C)+A.

During the adaptive thresholding algorithm, the first pass through the input image calculates the integral image. During the second pass, the $\mathbf{s}$ x $\mathbf{s}$ average around each

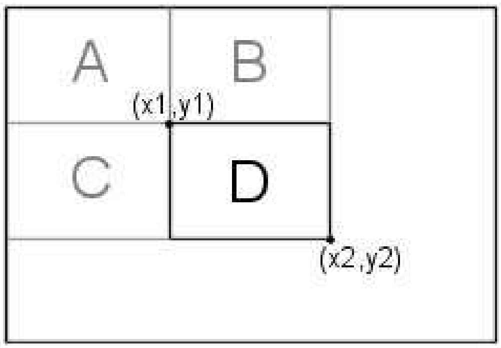


Figure 6: Using the integral image to calculate sums

pixel is calculated in constant time using the integral image. If the value of the current pixel is **t** percent less than this average it is set to black, otherwise it is set to white. This results in a two-pass adaptive thresholding algorithm that is simple and independent of the scanning order. Figure 7 illustrates the result of our adaptive thresholding method on the input image from Figure 4.

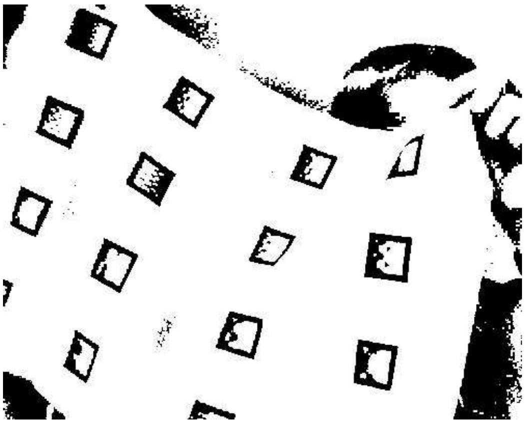


Figure 7: Adaptive thresholding

4 Flexible augmentation

In this section we describe how augmentations are performed in the scene, based on the results of the non-rigid tracking procedure described in section 3. The first step is to create a mesh for the virtual object, and then a unique illumination process is applied to make the augmentation more realistic.

4.1 Mesh creation

The non-rigid tracking method in section 3 returns the pixel locations of the four corners of each of the targets that are successfully found in the video image. In order to draw an augmentation on a flexible object like cloth, we require a mesh representation of the object. Since each of the targets are self-identifiable we can predetermine the connectivity of the corner points such that an appropriate triangulation of the object can be built. Figure 8 illustrates the mesh that is created from the target corner points.

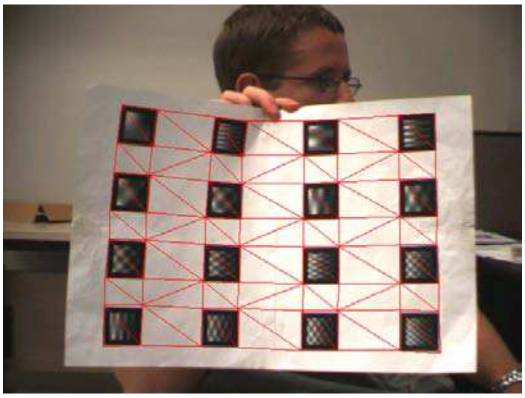


Figure 8: Virtual object mesh

The remaining step to render an augmentation is to apply a texture to the mesh. This is accomplished by using the fact that the targets are at fixed locations on the cloth and then applying static texture coordinates to each corner point. In this way by using OpenGL to perform the rendering, the mesh will contain a smooth texture without any unwanted artifacts. Figure 9 shows the textured mesh augmentation on the non-rigid object.

4.2 Realistic lighting

Thus far we have accomplished our goal of rendering an augmentation on a non-rigid object. However, as mentioned earlier, the addition of realistic lighting to AR scenes increases perception of spatial relations between real and virtual objects [22]. In our case, the virtual object is bound to the real cloth, and therefore the addition of realistic lighting will increase the perception that the cloth is truly exhibiting the virtual texture. Since the virtual object is rendered directly onto the real object, realistic lighting can be achieved by solving only one of the four shadow problems described in section 2. That is, the problem of casting shadows from real objects onto the virtual object. In our system, we exploit the fact that the augmentations are rendered onto a

Figure 9: Textured augmentation on the non-rigid object

white object. This means that the real-world illumination environment, including all of the shadows that should be cast onto the virtual object, is directly visible on the real object that we are already tracking. The intensity image of the non-rigid object, minus the targets, can be considered an illumination map for the scene. We propose a simple method to realistically acquire soft shadows from the real object, by sampling the illumination map. Each vertex in our final mesh corresponds to exactly one target corner on the cloth. Our idea is to determine a light intensity at each vertex and then use the Gouraud shading algorithm [12] to blend the illumination across the triangles. In order to determine the light intensity for each vertex, we compute the average of a 3 x 3 rectangle in the illumination map, just outside the target marker in the direction of the vector from the target center to the corner of interest. For example, Figure 10 shows the locations on the illumination map from where the light intensity values are acquired for the four corners of a target. The specific intensities are also displayed for visualization. Now the computed intensity values can be used during rendering to realistically display soft shadows on the augmentation. Figure 11 shows a series of illumination conditions for a flexible sheet of paper, including extreme light contrast, where the non-rigid augmentation demonstrates impressive realism through the use of realistic lighting.

Figure 12 shows the flexibility and realistic illumination of the augmentation on a piece of cloth while stretching and rippling.

5 Conclusions and future work

We present a method to track non-rigid objects and perform flexible augmentations, which is a new area of research in the field of augmented reality. Unlike previous AR research, our method is capable of tracking a flexible piece of cloth in real-time and

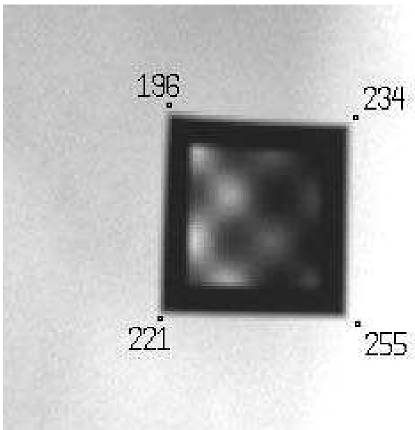


Figure 10: Locations where the illumination map is sampled for one target

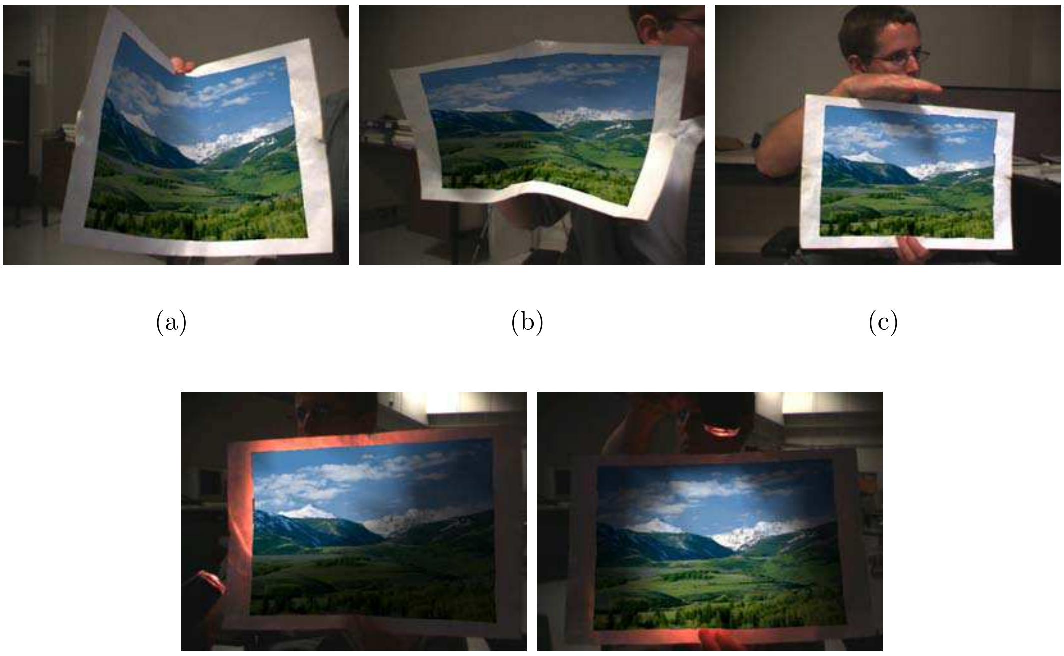
augmenting a texture onto the cloth using realistic lighting, even while stretching or rippling the cloth.

The next step in this research is to investigate better targets and tracking methods to acquire data about the cloth. Currently with the square targets, if a single target is occluded or missing then four mesh points are lost, resulting in a large hole in the augmentation. Also as the cloth bends, occasionally a target will become non-quadrilateral and tracking will fail. Performing 3D augmentations on non-rigid objects is another interesting area of future research. A difficult open problem is that of tracking self-occlusions in real time. Another area of future work would be to improve the illumination system to handle hard shadows in addition to the soft shadows that can be handled currently. This could be accomplished by using the intensity map of the input image containing the white cloth as a texture to blend with the augmentation. In that case, the actual targets would have to be removed from the texture and an impainting algorithm would be required to fill the holes where the targets existed. These ideas for continuing research are just some of the many possible areas of future work, since our proposed method of performing augmentations on flexible objects is an entirely new topic in augmented reality.

References

[1] ARToolKit. http://www.hitl.washington.edu/artoolkit.

[2] Ronald Azuma, Yohan Baillot, Reinhold Behringer, Steven Feiner, Simon Julier, and Blair MacIntyre. Recent advances in augmented reality. *IEEE Computer Graphics and Applications*, 21(6):34–47, 2001.



(a) (b) (c)

(d) (e)

Figure 11: Realistic lighting. a),b) shadows that illustrate the curves of the non-rigid object; c) shadow of a real object (a hand); d),e) extreme light contrast created by a flashlight.

[3] Francis H. Y. Chan, F. K. Lam, and Hui Zhu. Adaptive thresholding by variational method. *IEEE Transactions on Image Processing*, 7(3):468–473, March 1998.

[4] Andrew I. Comport, Eric Marchand, and Francois Chaumette. A real-time tracker for markerless augmented reality. In *IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2003)*, pages 36–45, Tokyo, Japan, October 2003.

[5] Franklin C. Crow. Summed-area tables for texture mapping. *SIGGRAPH Comput. Graph.*, 18(3):207–212, 1984.

[6] Paul Debevec. Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 189–198. ACM Press, 1998.

(a) (b)

Figure 12: Flexible augmentation. a) stretching cloth; b) sagging, rippling cloth.

[7] George Drettakis, Luc Robert, and Sylvain Bougnoux. Interactive common illumination for computer augmented reality. In *Proceedings of the Eurographics Workshop on Rendering Techniques '97*, pages 45–56. Springer-Verlag, 1997.

[8] Alain Fournier, Atjeng S. Gunawan, and Chris Romanzin. Common illumination between real and computer generated scenes. In *Proceedings of Graphics Interface '93*, pages 254–262, Toronto, ON, Canada, May 1993.

[9] Yakup Genc, S. Riedel, Fabrice Souvannavong, C. Akinlar, and Nassir Navab. Marker-less tracking for ar: A learning-based approach. In *IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2002)*, pages 295–304, September 2002. Darmstadt, Germany.

[10] Simon Gibson, Jon Cook, Toby Howard, and Roger Hubbold. Rapid shadow generation in real-world lighting environments. In *Proceedings of the 14th Eurographics workshop on Rendering*, pages 219–229. Eurographics Association, 2003.

[11] Simon Gibson and Alan Murta. Interactive rendering with real-world illumination. In *Proceedings of the Eurographics Workshop on Rendering Techniques 2000*, pages 365–376. Springer-Verlag, 2000.

[12] Henri Gouraud. Continuous shading of curved surfaces. *IEEE Transactions on Computers*, 20(6):623–628, 1971.

[13] Igor Guskov. Efficient tracking of regular patterns on non-rigid geometry. In *Proceedings of ICPR*, pages 1057–1060, 2002.

[14] Igor Guskov and Ben Bryant. Real-time tracking of quad-marked surfaces. In *IEEE Workshop on Motion and Video Computing*, 2002.

[15] Igor Guskov, Sergey Klibanov, and Benjamin Bryant. Trackable surfaces. In *ACM/EG Symposium on Computer Animation*, 2003.

[16] Igor Guskov and Leonid Zhukov. Direct pattern tracking on flexible geometry. In *Winter School of Computer Graphics*, 2002.

[17] Michael Haller, Stephan Drab, and Werner Hartmann. A real-time shadow approach for an augmented reality application using shadow volumes. In *ACM Symposium on Virtual Reality Software and Technology (VRST 2003)*, pages 56–65, October 2003.

[18] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049, 2000.

[19] Céline Loscos, George Drettakis, and Luc Robert. Interactive virtual relighting of real scenes. *IEEE Transactions on Visualization and Computer Graphics*, 6(4):289–305, 2000.

[20] Céline Loscos, Marie-Claude Frasson, George Drettakis, Bruce Walter, Xavier Granier, and Pierre Poulin. Interactive virtual relighting and remodeling of real scenes. In D. Lischinski and G.W. Larson, editors, *Rendering techniques '99 (Proceedings of the 10th Eurographics Workshop on Rendering)*, volume 10, pages 235–246, New York, NY, June 1999. Springer-Verlag/Wien.

[21] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[22] C.B. Madsen, M.K.D. Sørensen, and M. Vittrup. The importance of shadows in augmented reality. In *Proceedings: 6th Annual International Workshop on Presence, Aalborg, Denmark*, October 2003.

[23] Siddharth Manay and Anthony Yezzi. Anti-geometric diffusion for adaptive thresholding and fast segmentation. *IEEE Transactions on Image Processing*, 12(11):1310–1323, November 2003.

[24] Takeshi Naemura, Takuya Nitta, Atsushi Mimura, and Hiroshi Harashima. Virtual shadows in mixed reality environment using flashlight-like devices. *Trans. Virtual Reality Society of Japan*, 7(2):227–237, 2002.

[25] Charles B. Owen, Fan Xiao, and Paul Middlin. What is the best fiducial? In *IEEE International Augmented Reality Toolkit Workshop*, pages 98–105, 2002.

[26] David Pritchard and Wolfgang Heidrich. Cloth motion capture. *Computer Graphics Forum (Eurographics 2003)*, 22(3):263–271, Sep 2003.

[27] Gilles Simon, Andrew W. Fitzgibbon, and Andrew Zisserman. Markerless tracking using planar structures in the scene. In *Proc. International Symposium on Augmented Reality*, pages 120–128, October 2000.

[28] Jurgen Stauder. Augmented reality with automatic illumination control incorporating ellipsoidal models. *IEEE Transactions on Multimedia*, 1(2):136–143, 1999.

[29] Olga Veksler. Fast variable window for stereo correspondence using integral images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2003)*, pages 556–661, Madison, Wisconsin, June 2003.

[30] Paul Viola and Michael J. Jones. Robust real-time face detection. *Int. J. Comput. Vision*, 57(2):137–154, 2004.

[31] Pierre D. Wellner. Adaptive thresholding for the digitaldesk. Technical Report EPC-93-110, EuroPARC, 1993.

[32] Michael Wittkämper, Eckhard Meier, and Wolfgang Broll. Illuminating the mixed reality stage: Applying complex lighting conditions to ar. In *Proceedings of the IEEE and ACM International Symposium on Augmented Reality (ISAR'01)*, page 187. IEEE Computer Society, 2001.

[33] Xiao-Ping Zhang and Mita D. Desai. Segmentation of bright targets using wavelets and adaptive thresholding. *IEEE Transactions on Image Processing*, 10(7):1020–1030, July 2001.