Augmenting Deformable Objects in Real-Time

Julien Pilet Vincent Lepetit Pascal Fua
Computer Vision Laboratory
École Polytechnique Fédérale de Lausanne (EPFL) 1015 Lausanne, Switzerland
{Julien Pilet, Vincent.Lepetit, Pascal.Fua}@epfl.ch
http://cvlab.epfl.ch/research/augm/deformable.html

Abstract

We present a real-time system that can draw virtual patterns or images on deforming real objects by estimating both the deformations and the shading parameters. We show that this is what is required to render the virtual elements so that they blend convincingly with the surrounding real textures.

The whole process of uncompressing the video stream, measuring the deformations, estimating the lighting parameters, and realistically augmenting the input image takes about 100 ms on a 2.8 GHz PC. It is fully automated and does not require any manual initialization or engineering of the scene. It is also robust to large deformations, lighting changes, motion blur, specularities, and occlusions. It can therefore be demonstrated live on a simple laptop.

1. Introduction

In this paper, we show that highly deformable objects, such as clothes or sheets of paper, can be augmented in real-time to create a convincing illusion, such as the one depicted by Fig. 1. This is done automatically on an ordinary PC without *any* manual intervention or engineering of the environment.

This level of performance is achieved by integrating two different technologies. The first one, which we developed in earlier work [6], lets us quickly register *deformable* and non-instrumented surfaces to a reference image. It works on all incoming images of a video stream independently and does not require any hand-supplied initialization. Given this registration, the second one, which is the original contribution of this paper, is a dynamic approach to estimating the amount of light that reaches individual image pixels by comparing their gray levels to those of the reference image. This lets us either erase patterns from the original images and replace them by blank but correctly shaded areas, which we think of as *Diminished Reality*, or to replace them by virtual ones that convincingly blend-in because they are properly lighted. As illustrated by Fig. 1(e,f), this is important because adequate lighting is key to realism.

Not only is our approach real-time and fully automated but it also handles complex lighting effects, such as cast shadows, specularities, and multiple light sources of different hues and saturation. Even though realistic augmentation of rigid objects is by now a well-researched topic, we believe that achieving similar levels of performance for deformable objects and without either markers, manual intervention or light probe goes beyond the state-of-the-art.

In the remainder of the paper, we first discuss related work and briefly describe our automated registration technique. We then introduce our approach to realistically rendering the deformed patterns and present our results.

2. Related Work

Our approach relies on computing surface deformations in real-time and estimating shading parameters to produce realistic renderings. Since the main topic of this paper is lighting, we refer the interested reader to [6] for related work about surface registration.

In addition to geometric registration, shading parameters also have to be recovered if photo-realistic rendering of the inserted virtual objects is to be achieved. In [1], a white deformable surface with black markers is realistically augmented. The white part between markers is sampled on the input image to estimate lighting. By contrast, our approach does not need any marker and is able to recover lighting even on a textured surface showing no white part.

Another class of approaches [2, 3] focuses on interactive rendering of illumination changes caused by virtual objects in the real scene. However, both the illumination and geometry of the scenes are typically assumed to be static.

Unlike these earlier appraches, we can handle illumination changes and do not require calibration objects or lighting probes since we directly use the target objects to recover the required illumination parameters. Our approach is related to the ratio-images of [5], which are used off-line to locally change the illumination of human faces and produce

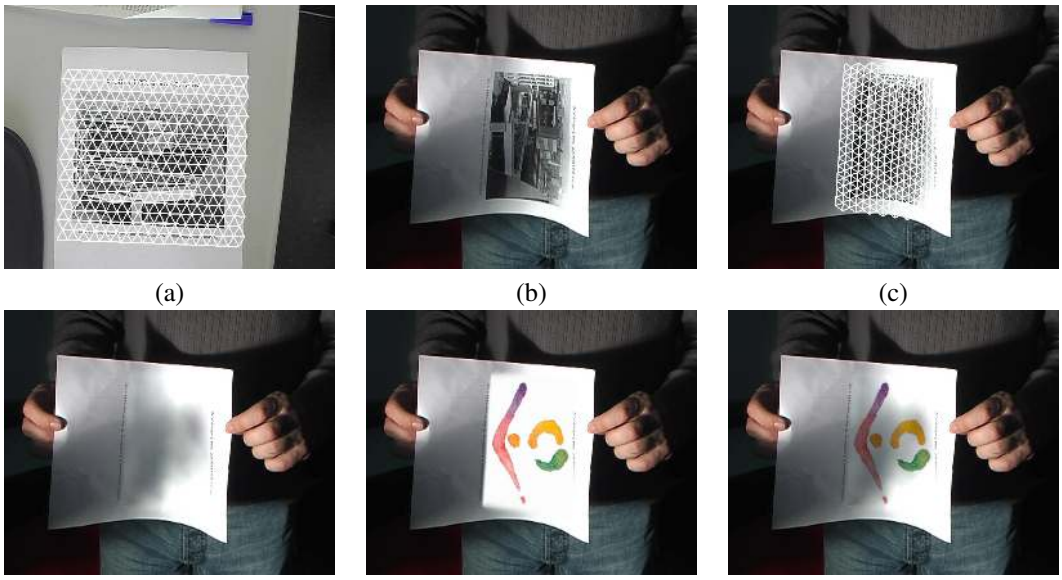


Figure 1: (a) The reference image of the target surface with the model mesh overlaid. (b) An input image. (c) The mesh is correctly deformed and registered to the input image. (d) The original pattern has been erased and replaced by a blank but correctly shaded image. (e) A virtual pattern replaces the original one. It is correctly deformed but not yet relighted. (f) The virtual pattern is deformed and relighted.

convincing new expressions. We also consider for the surface points the ratio of their colors before and after illumination changes. We show here that these ratios can be used in real-time to handle dynamic illumination and render complex illumination effects, such as cast shadows, saturation, and specularities.

3. Non-Rigid Surface Augmentation

In order to augment a deformable surface, the first step is to register the model on the input image, that is to compute object deformation between a reference image such as the one of Fig. 1(a) and an input image such as the one of Fig. 1(c).

We achieve this by first establishing point-to-point correspondences between reference and input images using a classification-based approach to wide-baseline matching [4] that is fast enough for real-time use. Given such correspondences, if the target object were rigid, detecting it and estimating its pose could be implemented using a robust estimator such as the random sample consensus. However, for a deformable object, the problem becomes far more complex because not only pose but also a large number of deformation parameters must be estimated. Deformable 2–D meshes and a well designed robust estimator is the key to dealing with this large number of parameters. Fitting then amounts to minimizing a criterion that is the sum of two terms. The first is a robust estimate of the square distances of the keypoints in the model image to that of the corresponding ones in the input image. The second is a quadratic deformation energy. This quadratic term allows the use of a semi-implicit minimization scheme that converges even when the initial estimate is very far from the solution, which, in our context, is what happens when the object is severely deformed. For additional details, we refer the interested reader to our earlier publication [6].

As a result, for surface points that project within the triangulated mesh that serves as a model, we can establish a one to one mapping between pixel locations in the two images. In this section, we show that this is the only information we need to realistically relight the virtual patterns we wish to draw on the surface by virtually altering the surface diffuse albedo visible on the input image.

We first introduce our rendering approach in the Lambertian case. We then argue that it remains valid in the presence of specularities and conclude the section with some implementation details.

3.1. The Lambertian Case

In practice, if we wish to build a versatile system that can be demonstrated in uncontrolled environments, we cannot make strong assumptions about light sources that are present when acquiring the input video. There can be many and their respective intensities and spectral properties are unknown, which can result in complex shading, shadowing, and color effects. To avoid the latter, we work independently on the red, green, and blue bands of color images.

However, it is easy to control the acquisition of the reference image. With no loss of generality, we can therefore

assume that it has been acquired when the surface was both undeformed and lighted uniformly, which means that every surface point receives the same amount of light in the color band we are working with.

Under this assumption, let $\mathbf{p}_r$ and $\mathbf{p}_i$ be the projections of the same surface point p in the reference and input image respectively, and let A_p be the corresponding surface albedo. In the Lambertian case, the contributions of all the light sources seen at $\mathbf{p}_r$ and $\mathbf{p}_i$ add linearly. We can therefore write

$$I_{r,p} = L_r A_p \ , \tag{1}$$

$$I_{i,p} = L_{i,p} A_p \ , \tag{2}$$

where $I_{r,p}$ and $I_{i,p}$ are the pixel intensities in the reference and input image respectively, $L_{i,p}$ the total irradiance in the input image at $\mathbf{p}_i$, and L_r the total irradiance in the reference image assumed to be the same at all surface points. In general, the values of $I_{r,p}$ and $I_{i,p}$ are different due to changes in both normal orientations and lighting conditions. However, the geometric registration we have established between the two images tells us that they correspond to the same physical point, which we exploit as follows.

Let us consider a white surface area with albedo A_w at location w on the surface. If the target surface has no white part, it is always possible to put a white object next to it while taking the reference image. We can measure on the reference image the pixel intensity $I_{r,w}$ where this white location w is projected and write $I_{r,w} = L_r A_w$, where L_r is the irradiance of Equ. 1.

Using this white normalization $I_{r,w}$, we can compute a new image, looking similar to the input one, except that the surface albedo is changed to A_w. In the input image, if there was no texture, the corresponding image intensity should be

$$I_{x,p} = L_{i,p} A_w = A_w L_r \frac{I_{i,p}}{I_{r,p}} = I_{r,w} \frac{I_{i,p}}{I_{r,p}} \ . \tag{3}$$

Note that $I_{x,p}$ is expressed exclusively in terms of image intensities, which are readily available, as opposed to albedoes or surface normals that are not.

Replacing the intensities $I_{i,p}$ of all the pixels on the object surface by $I_{x,p}$ yields images such as the one of Fig. 1(d) where the original texture has been replaced by a blank but correctly shaded surface. To draw a shaded new texture, as in Fig. 1(f), we simply multiply texture values with their corresponding white $I_{x,p}$.

Note that, because we perform the computation locally, it remains valid no matter how many sources there are and what their specific characteristics may be. The only thing that has to be true is that the contribution of the individual light sources to the pixel intensity are all modulated by the same diffuse albedo and do not depend on the viewpoint.

In some cases, $I_{x,p}$ is difficult to estimate reliably on large single-colored area. In the example of Fig. 3(a), recovering the $I_{x,p}$ blue component over the red area is hard because sensor inaccuracy on remaining blue light is amplified by a big factor. However, the visual impression given by Fig. 3(b) is still that the original painting has been erased and replaced.

3.2. Specularities and Saturation

The assumptions used to derive the formulas of Section 3.1 are clearly violated for specular materials. However, as illustrated by Fig. 2, this does not have severe consequences even in the presence of strong specularities and the illusion remains convincing.

This is because, when there is a specularity, the image intensity increases and the $\frac{I_{i,p}}{I_{r,p}}$ ratio of Equ. 3 becomes large. As a result, the $I_{x,p}$ intensity that is used to draw the synthetic patterns also increases, which is perceptually correct since it yields intensity maxima at specularities' locations. In other words, the absolute value of $I_{x,p}$ may not be correct but its magnitude relative to its neighbors remains consistent with the presence of a specularity. And since the human eye is much more sensitive to relative values than to absolute ones, this suffices.

In practice, specular peaks often saturate the camera sensor, thus making the estimation of $I_{i,p}$ unreliable. We detect such cases by simple thresholding and we handle saturation by setting $I_{x,p}$ to its maximal possible value. Since color computation is applied independently on the red, green and blue channels, one channel can saturate while the other do not. As a result, not only specular peaks but also saturated areas in the input image are correctly transcribed into the synthetic ones.

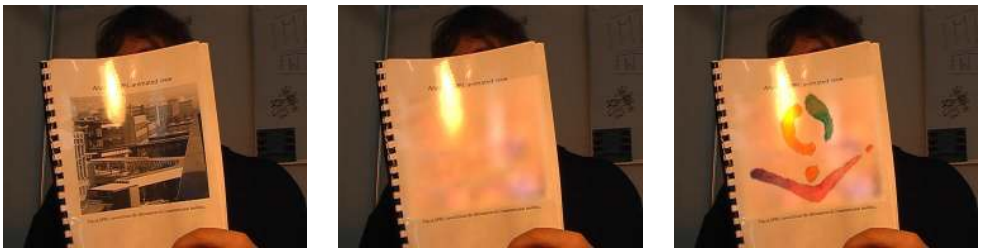
3.3. Implementation Issues

A completely straightforward implementation of the approach of Section 3.1, for example one that would directly use the formula of Equ. 3 to synthesize pixel intensities, would not be optimal because model registration does not always reach pixel accuracy. To overcome this problem, pixel values $I_{i,p}$ and $I_{r,p}$ are averaged over mesh facets while $I_{x,v}$ intensity ratios are computed at each vertex $\mathbf{v}$ by averaging the contributions of neighboring facets. The resulting smoothing allows the system to tolerate some amount of registration inaccuracy, at the cost of blurring sharp shadow transitions.

A standard graphic card can then very efficiently interpolate vertices illumination $I_{x,v}$ over a triangle before multiplying texels.

4. Results

In this section, we present our results. All corresponding videos can be downloaded from `http://cvlab.epfl.ch/research/augm/deformable.html`.



(a) (b) (c)

Figure 2: Handling specularities. (a) Input image with strong specularities. The main one is produced by a lamp, while the two smaller ones can be attributed to light coming through window. To produce this result, the paper has been covered by a transparent plastic sheet. (b) The picture has been erased from the surface but the specularities still appear to be at the right places. (c) The ISMAR logo has been inserted.

(a) (b)

Figure 3: (a) original image. (b) The ISMAR logo replaces the shirt print. Recovering white is hard in this image since the model has large single-colored areas, making light evaluation difficult.

In fig 2, we augment the pages of a book. To create this sequence, we first shot three very short movies. After extracting and printing in black and white the first frame of each movie we assembled the pages to make a small book. Then, we passed to our software the three printed pages as models, the three short movies, and a video sequence showing the being turned pages. In this sequence, the light is not uniform at all. Some pages are lit mainly by indirect daylight and the rest by incandescent light. Without any further interaction, the system produced a realistic augmented movie. The system does not alter the input video if it does not detect any of the three models. Upon successful detection, the appropriate artificial image replaces the picture on the paper. This is done by detecting one model per frame. If detection fails, the system looks for the next model on the next frame. Otherwise, it continues with the same model.

In Fig. 3, we superimpe the ISMAR logo on an ICCV shirt. This shirt has bright colors, making light recovery more difficult, while the very flexible material allows it to bend quite much, making detection difficult. However, our system is able to realistically change the logo on the fly. If detection fails, the input image is shown without the ISMAR logo, but without anything wrong. In other words, our method prefers not to augment the input frame rather than augmenting it in an incorrect way.

As the experimental videos available on our website show, our system has proven its ability to augment realistically flexible surfaces, despite changing lighting environment and dynamic geometry.

5. Conclusion

We have presented an approach to augmenting deformable objects that relies on estimating the deformations and shading parameters in real-time. This can be used either to convincingly erase real surface patterns or to render new ones so that they are properly shaded and appear to be glued to the real surfaces.

The system is robust and easy to use because it processes each new input frame automatically and independently. It therefore does not require manual initialization and is not subject to the kind of failures that afflicts systems that rely on knowing the object position and shape in the previous frames to compute those in the incoming ones. Moreover, no assumption is made on scene lighting and no scene instrumentation is required. Thus this system can be easily used in a very wide range of uncontrolled environments.

References

[1] D. Bradley and G. Roth. Augmenting Non-Rigid Objects with Realistic Lighting. Technical Report NRC/ERB-1116, Oct. 2004.

[2] G. Drettakis, L. Robert, and S. Bougnoux. Interactive common illumination for computer augmented reality. In *Eurographics Rendering Workshop*, pages 45–56, June 1997.

[3] S. Gibson and A. Murta. Interactive Rendering with Real-World Illumination. In *Eurographics Workshop on Rendering*, June 2000.

[4] V. Lepetit, P. Lagger, and P. Fua. Randomized Trees for Real-Time Keypoint Recognition. In *Conference on Computer Vision and Pattern Recognition*, San Diego, CA, June 2005.

[5] Z. Liu, Y. Shan, and Z. Zhang. Expressive Expression Mapping with Ratio Images. In *Computer Graphics, SIGGRAPH Proceedings*, 2001.

[6] J. Pilet, V. Lepetit, and P. Fua. Real-Time Non-Rigid Surface Detection. In *Conference on Computer Vision and Pattern Recognition*, San Diego, CA, June 2005.