Reality Tooning: Fast Non-Photorealism for Augmented Video Streams

Jan Fischer Dirk Bartz Wolfgang Straßer
WSI/GRIS - VCM, University of Tübingen, Germany
e-mail: fischer@gris.uni-tuebingen.de

Abstract

Recently, we have proposed a novel approach to generating augmented video streams. The output images are a non-photorealistic reproduction of the augmented environment. Special stylization methods are applied to both the background camera image and the virtual objects. This way, the graphical foreground and the real background images are rendered in a similar style, so that they are less distinguishable from each other. Here, we present a new algorithm for the cartoon-like stylization of augmented reality images, which uses a novel post-processing filter for cartoon-like color segmentation and high-contrast silhouettes. In order to make a fast post-processing of rendered images possible, the programmability of modern graphics hardware is exploited. The system is capable of generating a stylized augmented video stream of high visual quality at real-time frame rates.

1. Stylized Augmented Reality

In conventional augmented reality systems, the graphical objects are rendered over the camera image using standard real-time graphics algorithms. The resulting renderings can look artificial, and they tend to stand out from the background image (e.g., see Fig. 2(a)). We have recently presented an alternative approach to generating augmented video streams. Our *stylized augmented reality* technique attempts to create similar levels of realism in both the camera image and the graphical objects [1]. The cartoon-like stylization mode, which we have described, produces augmented images composed of mostly uniformly colored regions, which are enclosed with black silhouette lines. Here, we introduce a new algorithm for the cartoon-like stylization of AR images. This new method is a specialized post-processing filter, which is applied to the augmented image after the overlay of virtual objects. The implementation of the new algorithm uses vertex and fragment shaders, which run on the programmable GPUs of recent graphics cards. This allows for a straightforward and efficient design of the algorithm. The new method produces a stylized augmented video stream of a better visual quality at a significantly higher frame rate compared to the previously described approach. In particular, it all but eliminates flickering silhouettes in the processed video, which were often generated by the previous algorithm.

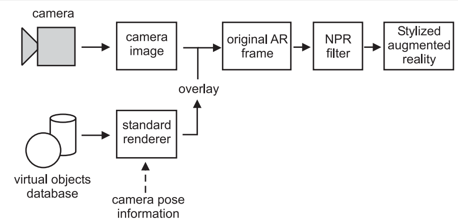


Figure 1. Overview of the new *stylized augmented reality* pipeline.

2. Description of the Algorithm

We present a new algorithm for the stylization of an augmented video stream. For each frame, a standard augmented reality pipeline first generates an output image containing the camera image with overlaid virtual objects. A post-processing filter is then applied to it, which is executed by the graphics processing unit (GPU). An overview of the approach is shown in Figure 1. The stylization filter consists of two steps. In the first step, a simplified color image is computed from the original AR frame. The simplified color image is made up of mostly uniformly colored regions. A non-linear filter using a photometric weighting of pixels is the basis for this computation. Several filtering iterations are consecutively applied to the image. The repetition of the filter operation is necessary in order to achieve a sufficiently good color simplification. The second stage of the non-photorealistic filter is an edge detection step. The simplified color image is the primary input for this operation. This way, the generated silhouette lines are located between similarly colored regions in the image, which is an approximation of a cartoon-like rendering style. Finally, the simplified color image is combined with the edge detection results. The simplified color image is enlarged to the size of the original input image. The combined responses of the edge detection filters are drawn over the enlarged image as black lines. A specific weight function is used for computing a transparency for the detected edge pixels, which produces a smooth blending over the color image.

2.1. Generation of Simplified Color Image

A shrunk version of the original AR frame is rendered into the frame buffer of the graphics card. The non-linear filter is then applied iteratively to this AR frame. Our non-linear filter is inspired by bilateral filtering, which is a widespread method for creating uniformly colored regions in an image [2]. This filter combines geometric and photometric weights when adding up pixels in the neighborhood of the currently regarded pixel. However, we have found the photometric weight to be sufficient for the purpose of creating a stylized augmented reality video stream. We denote the original RGB image function as $\mathbf{f}$ and the corresponding color coordinates in YUV space as $\mathbf{f}_{UV}$. The non-linear filter computes the simplified RGB image $\mathbf{h}$ using the following equation:

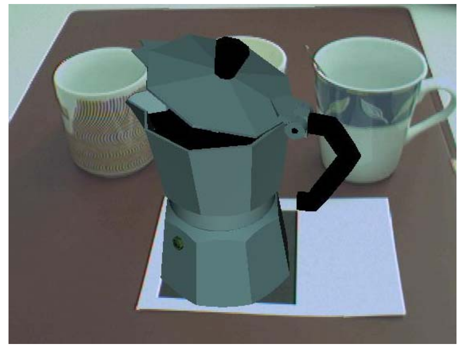
$$\mathbf{h}(\mathbf{x}) = k^{-1}(\mathbf{x}) \sum_{\xi \in \Omega_{\mathbf{x}}} \mathbf{f}(\xi)\, s(\mathbf{f}_{UV}(\xi), \mathbf{f}_{UV}(\mathbf{x})) \quad (1)$$

In Equation 1, $\mathbf{x}$ is the currently regarded point in the output image. A weighted sum is computed over image points ξ in the neighborhood $\Omega_{\mathbf{x}}$ of $\mathbf{x}$. The weight $s(\mathbf{f}_{UV}(\xi), \mathbf{f}_{UV}(\mathbf{x}))$ depends on the similarity of values in the color channels $\mathbf{f}_{UV}(\xi) - \mathbf{f}_{UV}(\mathbf{x})$. In our algorithm, s is a Gaussian function of the absolute value of the color difference. In order to maintain the overall brightness of the image, the weighted sum is divided by the normalization factor $k(\mathbf{x})$. The effect of this non-linear filter is that an averaging of pixels only occurs in places where nearby pixels have similar colors. Only in such places in the image $s(\mathbf{f}_{UV}(\xi), \mathbf{f}_{UV}(\mathbf{x}))$ is large. Thus strong edges in the image are preserved.

3. Results

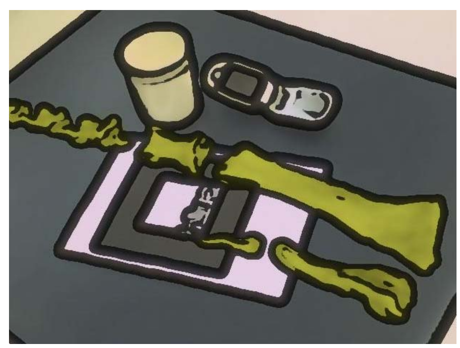
The implementation of our new algorithm uses the OpenGL Shading Language (GLSL) in order to execute the entire post-processing filter on the Graphics Processing Unit (GPU). All image processing operations have been realized as efficient GLSL shader programs which work on texture images in local framebuffer memory. We have tested our new algorithm with various AR scenes. Benchmark measurements show that our algorithm is capable of delivering frame rates of more than 25 fps for typical algorithm parameters. The generated output video stream has a significantly higher visual quality than the results described in [1], particularly due to reduced flickering in the detected silhouettes. Figure 2 shows some example images generated with the new algorithm

This work has been supported by project VIRTUE in the focus program on "Medical Robotics and Navigation" (SPP 1124) of the German Research Foundation (DFG).



(a) Virtual coffee maker in front of real cups in conventional AR

(b) Coffee maker scene in stylized AR



(c) Virtual dinosaur bone model with real cup and cell phone in stylized AR

Figure 2. Example images generated with our algorithm.

References

[1] J. Fischer, D. Bartz, and W. Straßer. Stylized Augmented Reality for Improved Immersion. In *Proceedings of IEEE Virtual Reality (VR)*, pages 195–202, March 2005.

[2] C. Tomasi and R. Manduchi. Bilateral Filtering for Gray and Color Images. In *International Conference on Computer Vision (ICCV)*, pages 839–846, 1998.