

More Sketchy, More AR, More Fun!

Michael Haller*, Florian Landerl†
Upper Austria University of Applied Sciences, Hagenberg - AUSTRIA

Mark Billinghurst‡
HITLab NZ



1 Introduction

Augmented Reality (AR) based entertainment applications are becoming more and more attractive and popular. A lot of effort has been put in improving AR interfaces, finding novel interaction techniques, but less work has been spent on improving the rendering quality. Is there a way to make AR installations more visually appealing and fun? Although a lot of computer games (including AR games/installations) still aim to create real-time, photorealistic rendered environments, there seems to be a new trend in the direction of making games stylistically more believable and/or more enjoyable. Consequently, non-photorealistic rendered (NPR) scenes are not only becoming more and more interesting for artists and game designers, but also for people who understand the content easier if the objects are rendered in a simple sketchy style. The aim is to create a world where augmented objects are expressive, clear and look aesthetically perfect. Furthermore, there are some unique advantages to using NPR techniques. For example hand-drawn sketches can often communicate complex object relationships in a better way than photorealistic pictures. We recognized that our approach was highly accepted by people, because viewer's mind completed the overall picture and they liked the artistic pen-and-ink rendering method a lot.

In this sketch, we present *sketchy-ar-us*, a modified, real-time version of the Loose and Sketchy algorithm [Curtis 1999] in combination with an AR environment using programmable GPU hardware. Curti's approach was mainly designed for offline renderings. In contrast, we optimized the algorithm by extensively using modern 3D hardware (e.g. direct rendering to textures using PBuffers, optimized particle system by using a reference image, filter operation using the pixel shader etc.). Equipped with a handheld visor, visitors can look at a real still life setup, where the augmented virtual objects are combined with the real environment and drawn in a non-photorealistic style.

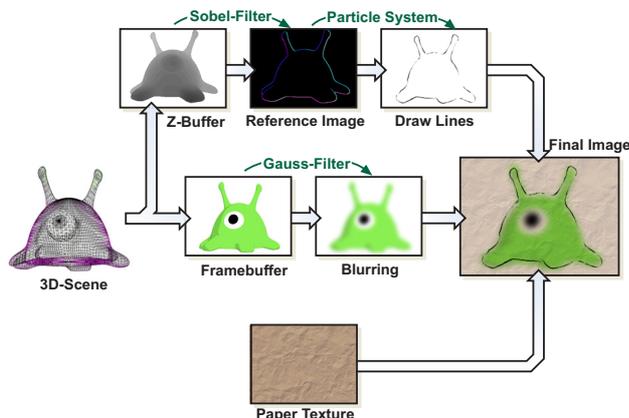
2 Method and Results

Three steps have to be realized to achieve the Loose and Sketchy rendering result in real-time: calculating a sketched silhouette, blurring the image, and combining the image with a paper texture.

*e-mail: haller@fh-hagenberg.at

†e-mail: florian.landerl@fh-hagenberg.at

‡e-mail: mark.billinghurst@hitlabnz.org



The pipeline can be described as follows: an object silhouette is drawn based on particles that are generated around the edges which appear and disappear randomly. The corresponding particle positions are calculated by a reference image that is used as a template for both the position and a "force"-mask that shows where the particles tend to move to. The particles themselves are the starting point and the ending point for the brushes that will be placed accordingly. The reference image itself has to be calculated by a sobel filter that is applied to the depth buffer of the virtual (augmented) 3D geometry and the red portion of the real image. The seamless combination of both the real and the virtual objects is achieved by blurring the image with a Gauss filter realized by a fragment shader. Finally, we blend a paper texture to get the final Loose and Sketchy result. *sketchy-ar-us* allows users to place virtual/real objects inside the scene - the virtual objects are attached to real fiducial markers, which are tracked by the ARToolKit software library. The original Loose and Sketchy algorithm was mainly designed for offline rendering, thus it took 10-60 seconds for rendering each frame. In our AR environment, we achieve up to 11 fps including tracking, rendering, and object interaction. The algorithm does not depend on the complexity of geometry. The randomized appearance/disappearance of the silhouette guarantees highly dynamic animation and results in an attractive image.

References

CURTIS, C. 1999. Non-photorealistic animation. In *Proc. ACM SIGGRAPH 1999*, ACM Press.