Real-Time Watercolor for Animation*

Thomas Luft and Oliver Deussen
Department of Computer and Information Science, University of Konstanz, Konstanz, Germany
E-mail: {luft, deussen}@inf.uni-konstanz.de
Revised December 10, 2005.

Abstract We present algorithms that allow for real-time rendering of 3D-scenes with a watercolor painting appearance. Our approach provides an appropriate simplification of the visual complexity, imitates characteristic natural effects of watercolor, and provides two essential painting techniques: the wet-on-wet and the wet-on-dry painting. We concentrate on efficient algorithms based on image space processing rather than on an exact simulation. This allows for the real-time rendering of 3D-scenes. During an animation a high frame-to-frame coherence can be achieved due to a stable segmentation scheme. Finally, we seamlessly integrate a smooth illumination into the watercolor renderings using information from the 3D-scene.

Keywords non-photorealistic rendering, watercolor painting, real-time rendering

1 Introduction

The complex nature of watercolor is characterized by soft, flowing, multifaceted, and random patterns that occur due to the motion of water and color pigments during the painting and drying processes. The interaction of a large number of natural effects make watercolor a unique drawing medium that can be exploited by a skilled artist to achieve a wide variety of artistic results. In the field of non-photorealistic computer graphics, the production of watercolor paintings is one of the most intricate and complex processes.

In contrast to physically based simulations, our aim is to create convincing watercolor renderings of 3D-scenes in real-time. We introduce several algorithms that allow the imitation of the most important drawing techniques and natural effects, which give the appearance of natural watercolor paintings (see Fig.1). The underlying 3D-scenes provide the necessary information for the creation of individual watercolor washes and for the production of a smooth lighting and shadowing. The real-time ability is achieved due to the potentials of hardware accelerated shaders.

We focus on watercolor drawing and also animation as described by the following aspects:

Simplification and Abstraction. We first create a number of abstract watercolor layers that are each based on an individual intensity image. Each intensity image contains the simplified shape of a single object or a group of objects that have a more or less uniform content or color, such as the plate and the particular fruits in Fig.1. Finally, these semi-transparent watercolor layers are composited on the screen.

Watercolor Effects. Our approach imitates significant effects of watercolor, such as the edge darkening, the flow pattern or the pigment granulation that occur due to the paper structure. This allows us to incorporate two important painting techniques: the wet-on-wet painting and the wet-on-dry painting.

Lighting and Shadowing. We incorporate a lighting model that follows the idea of the Phong lighting. This allows us to integrate a smooth illumination that provides very dynamic results. We further describe different methods for the application of the lighting information, such as a color modulation or a masking of the watercolor layers. Additionally, we integrate shadows as an additional watercolor layer.

Real-Time Animation. Since the computation of the watercolor effects is based on an efficient image space processing rather than on an exact simulation, we easily achieve real-time results. During animation a high frame-to-frame coherence can be achieved due to a stable segmentation scheme and the subsequent image space processing.

* A preliminary version of this paper appeared in Proc. Pacific Graphics 2005, Macau.
1.1 Related Work

There are related work that explicitly treat the generation of watercolor paintings. A physically based approach is the work of Curtis et al.\textsuperscript{[1]}, which is directly related to the cellular automaton approach of Small\textsuperscript{[2]}. Curtis et al. give a detailed description of the most important effects that occur during the watercolor painting and drying process. The key idea was to use a three-layer model that consists of a shallow-water layer, a pigment-deposition layer and a capillary layer. While the simulation of these three layers produce very convincing results, it is computationally expensive, and thus cannot be considered as real-time graphics. Another work based on this three-layer model is described by Van Laerhoven et al.\textsuperscript{[3]} Their approach allows an interactive simulation of the watercolor painting process. Other work that simulates a virtual canvas is the IM-PaSto framework of Baxter et al.\textsuperscript{[4]} and the work of Chu and Tai\textsuperscript{[5]}. Here the user acts as the artist and draws the scene using special input devices.

Some work is produced that does not yield an exact simulation, but rather use image processing algorithms to reproduce a watercolor like finish. The work of Lum and Ma\textsuperscript{[6]} was inspired by watercolor paintings. They describe a raytracing based rendering pipeline that creates procedural textures based on LIC (line integral convolution), which produce a watercolor like appearance. Here the combination of several semi-transparent color layers forms the final result. Their work allows rendering of 3D-scenes, however it does not allow real-time rendering. Burgess et al.\textsuperscript{[7]} describe a similar system using Wyvill noise to create a brush like appearance. The work of Lei and Chang\textsuperscript{[8]} is based on simple image space filters to imitate some key effects of watercolor, such as using a Sobel filter to create the edge darkening.

There are also numerous stroke based approaches, such as Strasser\textsuperscript{[9]}, or Hertzmann\textsuperscript{[10]}. The approach of Hertzmann places strokes automatically in an iterative process on the basis of an input image. Thereby large strokes are drawn first and then smaller ones are added iteratively, in order to capture the details of the input image. The results are more similar to acrylic or oil than to watercolor painting. Other stroke based approaches that discuss Chinese watercolor paintings are those of Chu and Tai\textsuperscript{[11]} or Su et al.\textsuperscript{[12]} A milestone based on particle systems was the painterly rendering for animation introduced by Meier\textsuperscript{[13]}. Her work combines a brush stroke like appearance with stable, frame-to-frame coherent 3D-particle sets. However, the results also resemble more acrylic or oil paintings than watercolor.

2 Imitating Watercolor Paintings

The nature of watercolor paintings is characterized by the influence of several essential effects, such as diffusion, backruns, edge darkening or pigment granulation. We concentrate on the imitation of the most significant effects by using simple algorithms instead of an exact, but nevertheless complex simulation. Furthermore, watercolor is a transparent medium, a suspension of water, binder, and color pigments, which results in a transparent and luminous impression.

Our pipeline produces a number of individual watercolor layers that are composited as semi-transparent layers on the rendering device. This procedure follows the natural process of painting with watercolor and also allows for watercolor glazing, which is an important characteristic of watercolor. Glazing describes the composition of several thin washes that are painted on top of each other after each layer has dried thoroughly. As a consequence, the color pigments are blended optically instead through a physical mixing.

In the following, we describe our approach for reproducing a watercolor appearance.

Simplification and Abstraction. The process of drawing is always an abstraction and simplification of the motif through the use of different brushes, special techniques, and a careful selection of watercolors, the artist is able to create a convincing simplification that is still rich of information.

Following this principle, our final result is a composition of several abstract watercolor layers each containing a more or less uniform content or color, such as the branches and the foliage in Fig.2. To attain this result, the abstraction step of our pipeline begins with the segmentation of the 3D-scene on basis of unique identifiers that are assigned to objects or a group of objects. This segmentation scheme can be improved by using textures containing ID-information, which produce a higher granularity, and thus a higher level of detail. As a result of the segmentation, we obtain a number of intensity images $\rho : \mathbb{N}^2 \rightarrow \mathbb{R}$ with $\rho(x,y) \in [0,1]$.

A low-pass filter is then applied to the intensity images using a Gaussian filter kernel. Here the kernel size represents the abstraction level of the watercolor layers. This simplification step results in abstract smooth shapes that directly describe the shape of the individual watercolor layers. Additionally, each watercolor wash is comprised of a user specified color and transparency thereby reproducing the ratio between water and color pigments. Consequently, our watercolor layers $\lambda : \mathbb{N}^2 \rightarrow \mathbb{R}$ are initialized with an overall color vector $c_{ob}$ and an overall transparency $c_e$. In the following, we outline the imitation of the watercolor effects due to an additional modulation of the alpha channel $\lambda_a(x,y) \in [0,1]$ of each watercolor layer.

Shape Extraction and Flow Pattern. The shape of a watercolor layer arises from the intensity values of the corresponding intensity image $\rho(x,y)$. To produce a first, hard edged shape of the watercolor layer, we threshold the intensity image by applying a step function and compute an initial transparency (see Fig.3(a)):

$$\lambda_a(x,y) = c_a \cdot \text{step}(\kappa_p, \rho(x,y)).$$