Computers & Graphics 50 (2015) 61–70

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Technical Section

A multiscale model for rain rendering in real-time ☆

Yoann Weber *, Vincent Jolivet, Guillaume Gilet, Djamchid Ghazanfarpour

XLIM - UMR CNRS 7252, University of Limoges, France

ARTICLE INFO

Article history:
Received 24 December 2014
Received in revised form
13 April 2015
Accepted 23 April 2015
Available online 28 May 2015

Keywords:
Natural phenomena
Rain
Physically-based modeling
Real-time rendering
Attenuation of visibility
Extinction

ABSTRACT

This paper presents a coherent multiscale model for real-time rain rendering which takes into account local and global properties of rainy scenes. Our aim is to simulate visible rain streaks close to the camera as well as the progressive loss of visibility induced by atmospheric phenomena. Our model proposes to correlate the attenuation of visibility, which is due in part to the extinction phenomenon, and the distribution of raindrops in terms of rainfall intensity and camera's parameters. Furthermore, this method proposes an original rain streaks generation based on spectral analysis and sparse convolution theory. This allows an accurate control of rainfall intensity and streaks appearance, improving the global realism of rainy scenes.

1. Introduction

Rain rendering has been widely studied in computer graphics. Rainy scenes include several effects such as streaks due to the velocity of raindrops and camera's exposure time or retinal persistence. It also includes among other features a progressive loss of visibility, due to suspended raindrops in the air spreading light in a specific way. Thus, the realism of a rainy scene in computer graphics depends on the raindrops appearance as well as the global properties of this meteorological phenomenon. These problems can be tackled by simulating the mesoscopic and macroscopic views of a rainy atmosphere. In this paper, we take into account these two scales.

The existing methods mainly focus on streaks rendering. The most recent ones, such as [1,2], use a particle system and a textures database to render streaks. Nevertheless, such a database has a high memory footprint and also requires complex mechanisms to control particles, in order to preserve a constant and physical distribution of raindrops in space during the rain simulation. Another technique for rain rendering [3] consists in using one or several rainfall textures positioned at different distances from the observer. These textures are defined in a finite space. Such methods cause a repetition of rain patterns as textures are moving and repeated over and over again. All of these methods usually use fog models in an empirical way to simulate the loss of visibility naturally present in rainy scenes. Their attenuation is neither based on physically realistic extinction coefficients nor correlated with rainfall intensity. This paper addresses these issues and proposes a real-time multiscale method relying on image-space post-processing (i.e. independent of scene complexity) computations. The main contributions of our method are

- A procedural generation of streaks in the image space, based on spectral analysis of real rain patterns. The sparse convolution noise technique is used to control precisely the rainfall intensity and to manage only visible streaks. This generation also enables a constant and physical distribution of raindrops during the simulation.
- A density function to quantify the visible streaks in the view frustum in terms of rainfall intensity and camera's parameters.
- A physical correlation between the density of visible streaks and the attenuation of visibility. Only once distribution is used to evaluate the density of raindrops and also to render the loss of visibility in the scene.

2. Previous works

We will focus on methods which propose streaks generation and attenuation of visibility. These two phenomena are the key elements of our multiscale model.

2.1. Rain streaks rendering

We can classify papers into two categories: the ones based on scrolling rainfall textures and the others based on a particle system.

The papers [3,4] fall into the first category of methods. The advantage lies in its simplicity and reduction of the computation

☆This article was recommended for publication by Holly Rushmeier.

* Corresponding author.
E-mail address: y_weber@orange.fr (Y. Weber).

http://dx.doi.org/10.1016/j.cag.2015.04.007
0097-8493/

time. Nevertheless there are two recurrent problems with the rainfall textures. The repetition of the patterns prevents the rain effect from being visually realistic while the simulation is running. Indeed, those textures are finite and usually artistically-generated. This results in an empirical distribution of raindrops. Moreover the user cannot change the rainfall intensity in real-time during the simulation and it is impossible to evaluate collisions between raindrops and objects in the scene.

Meanwhile, the papers [1,2,5–9] achieve streaks rendering by using a particle system. Those methods are more computationally expensive. Generally, they either extract textures from video which are then mapped onto particles or use some precomputed ones. In this regard, Garg and Nayar [2] are the first to propose a complete precomputed streaks database. This method has been used by Tariq [7] and more recently by Creus and Patow [1], yielding really convincing results. Textures are classified in terms of three different angles depending on the position of the observer and the light sources. The main downside is that such a database has a very important memory footprint. Moreover a particle system costs a lot of GPU resources, being fill rate based, and rain animation is not a trivial task, managing only visible particles in the view frustum while avoiding repetitive behavior and having an invariant distribution of raindrops. The random respawn of particles does not guarantee a physically realistic spatial distribution of raindrops. This implies an adapted implementation.

2.2. Attenuation rendering

Most methods use an empirical approach to simulate the attenuation of visibility. Using an arbitrary fog model yields to a non-physically realistic rendering. For instance, Tatarchuk and Isodoro [3] and Tariq [7] propose an attenuation without using a specific value for the so-called extinction coefficient. However our research shows that this coefficient must be correlated with the rainfall intensity and the distribution of raindrops.

Changbo et al. [10] offer the possibility to simulate the attenuation of visibility taking into account different sizes of raindrops in the air. They made computations for every type of raindrops, and considered a uniform phase function to evaluate how light is scattered. It is however difficult to know which data must be precomputed and set precisely the extinction coefficient in order to obtain a correlation with the distribution of raindrops.

A different approach for snow or rain rendering is proposed by Langer et al. [11], using an infinite density function based on a sum of 2D waves cosine to simulate the density of precipitation. The density function is expressed in the spectral domain. The Fourier Transform and its inverse are computed. The evaluation of this function in the spectral domain is not based on a physical distribution of raindrops in the spatial domain.

These methods achieve an attenuation rendering by using a macroscopic approach. Indeed, increasing the number of particles does not simulate the continuity of the meteorological phenomenon. Generating more particles induces visual artifacts due to the resolution and does not naturally generate attenuation of visibility. Nevertheless, most of these methods overlook the distribution of raindrops to get a physically realistic solution. This prevents a correlation between the rainfall intensity and the attenuation of visibility.

3. New multiscale model

A rainy scene is typically composed of many raindrops – each with a size much larger than visible wavelengths – which causes the light to spread in a specific way. Closer raindrops, which have a projected size larger than a pixel, are individually visible, and further ones can only be seen in a macroscopic view as a participating media. As a result, visibility decreases until details blur into a fog like appearance. Taking into account these two rain features enables us to propose a physically based rain simulation. The main contribution of this paper is to propose a correlation between the density of close and visible raindrops and the attenuation of visibility.

According to the preceding observations, we decompose a rainy scene into two areas. The first one corresponds to the region of the scene where streaks have a projected size larger than a pixel or thus are individually visible. We will refer further at this region as the Visible Streaks Region (VSR). In this region, our rain streaks generation has to take into account the problems inherent to particle systems and finite rainfall textures, as we discussed in Section 2. Beyond this rain region, streaks cannot be individually seen because it would require a subpixel precision. Moreover, that would imply simulating a large number of raindrops. The idea is to develop a scattering method specifically for these Scattering Rain Phenomena (SRP) in order to indirectly simulate remote raindrops. This phenomenon is continuous in the whole scene from the camera to the furthest visible object in the scene. It simulates the raindrops too small to be represented even in the Visible Streaks Region. There is actually an attenuation effect in the whole scene, even in the VSR where it could be assumed to be very close to zero. All raindrops cannot be actually simulated even in the VSR. This implies rendering this phenomenon with a macroscopic approach to simulate how light behaves in contact with raindrops in the air. Such a multiscale method enables us to represent each phenomenon of rainy scenes with an adapted model and to correlate each model according to physically based parameters. This part of the paper will discuss which models to use to simulate the VSR and the SRP, and which physical parameters will allow us to correlate each model and control our rain simulation in real-time.

We propose in Section 3.1 a generation of visible rain streaks. This first model simulates only visible and individual streaks close to the camera. The second model, presented further in Section 3.2, takes into account the distribution of far raindrops in the scene with a single scattering method, to progressively attenuate the visibility.

3.1. The Visible Streaks Region (VSR)

Section 2 highlights the recurrent problems inherent to existing methods based on particle systems and finite rainfall textures. Particle systems offer the possibility to precisely control rain streaks but require complex mechanisms to manage only visible raindrops. Indeed, some of those raindrops may be visible after projection and some others may not, depending on their size and distance from the observer. Complexity also lies in both simulating a large number of particles with a constant distribution over space and usually in storing a texture database with a high memory footprint.

Meanwhile, finite rainfall textures reproduce the global appearance of these meteorological phenomena. Few calculations are necessary but the number of available rainfall textures is usually limited and therefore does not allow a precise control over the rainfall intensity in real-time. Moreover, repetitions appear due to finite resolution issues while rainfall textures are scrolling down. To tackle these problems, we propose a middle ground method to combine these two approaches. This technique is controlled by a density function in real-time, evaluated in terms of rainfall intensity and camera's parameters, and enables us to manage only visible rain streaks. It provides the following advantages:

- A control over the density of raindrops through physical parameters.
- A physically based distribution of raindrops.

We consider rain as a set of raindrops, distributed in space. Their size varies and they overlap on screen once projection is done.