RPR-SORS: An Authoring Toolkit for Photorealistic AR

Guilherme de S. Moura, Saulo A. Pessoa, João Paulo S. do M. Lima, Veronica Teichrieb, Judith Kelner
Centro de Informática
Universidade Federal de Pernambuco (UFPE)
Cidade Universitária S/N, 50670-901, Recife, PE, Brasil
gsm@cin.ufpe.br, sap@cin.ufpe.br, jpsml@cin.ufpe.br, vt@cin.ufpe.br, jk@cin.ufpe.br

Abstract

This paper presents a solution for photorealistic rendering of virtual objects into dynamic real scenes. It consists of a toolkit for Augmented Reality, named RPR-SORS, which is composed of two artifacts: an API that handles various computer graphics techniques in a novel pipeline approaching illumination, reflectance model, shadowing, composition, and camera effects; and a material editor that accelerates the creation of complex photorealistic materials. An experimental application was developed with the proposed solution, where both the API and material editor were used to accomplish photorealistic effects in an application of Augmented Reality in architecture. The results obtained show that the API provides a flexible infrastructure for the photorealistic Augmented Reality pipeline, and that the material editor facilitates the creation of photorealistic materials and their use for Augmented Reality.

1. Introduction

Photorealistic Rendering (PR) is responsible for creating synthetic images of a given scene that are as close as possible to the visual response obtained from a real scene. A specific problem in this context is the Photorealistic Rendering of Synthetic Objects into Real Scenes (PR-SORS), which is commonly faced by the cinematographic industry when inserting digital characters into movies. Although there are several methods that try to solve this problem, they are mostly designed to run offline. There is nonetheless a growing demand for real-time techniques that can achieve similar results. This relatively new problem has been named Real-Time Photorealistic Rendering of Synthetic Objects into Real Scenes (RPR-SORS) and appears in the context of Augmented Reality (AR) applications.

Some AR systems (e.g. industrial applications [1]) demand functional realism, where the main concern is providing useful information about the task represented by the scene [2], and thus opt for non-photorealism. However, many AR application domains take profit from photorealism, such as games [3], cultural heritage [4], and interior design [5]. The somehow utopian goal of these AR applications is to create a scene in real-time where a user believes that the presented objects are real.

The rendering of photorealistic objects involves many considerations and techniques, such as illumination and occlusion, to name only a few. In addition, the integration of these techniques generates a large amount of parameters to be set for each photorealistic object in the scene, making their manipulation a complex task. The use of toolkits promotes an easier and more effective application authoring process, since they usually provide a set of helpful resources to developers.

This paper presents the RPR-SORS authoring toolkit, a solution for photorealistic AR which consists of two artifacts: the RPR-SORS Application Programming Interface (API) and the RPR-SORS Material Editor. The RPR-SORS API is mainly targeted towards programmers and consists of an implementation of the photorealistic AR techniques described in [6] and in [7]. It allows developers to support a multitude of available photorealistic effects in its AR application, as well as combining just some of them. The chaining of the different techniques is handled in a user transparent way by the API. The RPR-SORS Material Editor is well suited for artists/designers and aims to increase the productivity and quality in the creation of photorealistic materials for virtual objects. It is capable of handling the material parameters interactively, offering an additional feature to visualize the photorealistic objects in AR mode. The final result produced by the editor can then be saved and exported in order to be loaded by an application that was developed using the RPR-SORS API.

This paper is structured in the following way. Section 2 reviews authoring tools related to the one developed in this work. Section 3 gives details about the RPR-SORS API and Material Editor. Section 4 presents the obtained results and