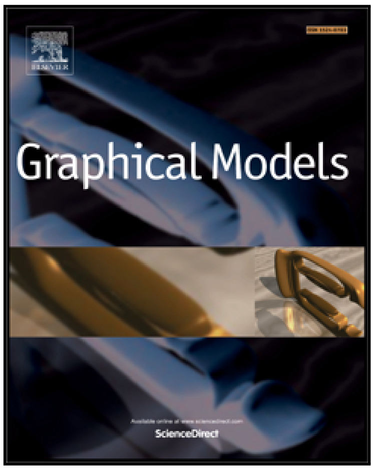
Accepted Manuscript



Revolution Mapping with Bump Mapping Support

Anouar Ragragui , Adnane Ouazzani Chahdi , Akram Halli , Khalid Satori

PII: S1524-0703(18)30039-0
DOI: https://doi.org/10.1016/j.gmod.2018.09.001
Reference: YGMOD 1011

To appear in: *Graphical Models*

Received date: 7 October 2017
Revised date: 11 July 2018
Accepted date: 3 September 2018

Please cite this article as: Anouar Ragragui , Adnane Ouazzani Chahdi , Akram Halli , Khalid Satori , Revolution Mapping with Bump Mapping Support, *Graphical Models* (2018), doi: https://doi.org/10.1016/j.gmod.2018.09.001

Revolution Mapping with Bump Mapping Support

Anouar Ragragui[a], Adnane Ouazzani Chahdi[a], Akram Halli[b], Khalid Satori[a]

[a]*LIIAN Laboratory Faculty of Science Dhar El Mahraz, Sidi Mohamed Ben Abdellah University, P.O. Box 1796, Fez, Morocco*

[b]*OMEGA-LERES Laboratory FSJES-UMI, Moulay-Ismaïl University, Meknes, Morocco*

Abstract

Per-pixel Revolution Mapping is an image-based modeling and rendering technique or IBMR (Image-Based Modeling and Rendering). It consists of creating a virtual 3D objects without polygonal meshes. This technique uses a single RGBA texture that stores the needed data to generate the revolved surface. The main problem with this technique is that the 3D revolved models are rendered without realistic surface wrinkles. In this paper, we presented an improvement to enhance the realism of the 3D revolved models by combining the revolution mapping and the bump mapping. In order to synchronize between real-time depth scaling of the microrelief and the resulting shading, we have added a scaling factor that makes it possible to have a realistic depth animation. This new technique creates very convincing 3D models with realistic looking surface wrinkles and allows rendering at interactive frame rates.

Keywords: Revolution mapping, Image-Based Modeling, and Rendering, 3D object, Revolved models, Bump mapping, Real-time, depth scaling, Frame rates.

1. Introduction

Image-based rendering is a very serious alternative to traditional rendering methods based on polygonal meshes. Because the latter will always be limited by the number of graphics primitives that the hardware will process.

Per-pixel revolution mapping [1] is based on a single RGBA texture and a simple shape box. The texture is used to store all the related data to the geometry, it is called shape map, and the shape box is created from the shape map and combined with the revolution algorithm to create the 3D object. However, the revolved geometry is rendered using a ray-tracing algorithm that operates in the programmable units of the graphics card. This technique limits considerably the number of graphics primitives constituting the complex scenes.

Contrary to the conventional techniques of per-pixel displacement mapping that are based on a grayscale image representing any relief, the revolution mapping uses a 2D binary image, where only the zero-values pixels constitute the basic shape of the 3D object that will be generated by the revolution technique. In order to find the intersection point between the viewing ray and the virtual geometry, this technique uses the Euclidean Distance Transform (EDT) [2] in order to skip the empty space and to converge more rapidly towards the intersection point.

2. Contribution

Our contribution concerns the revolution mapping technique that is a technique for the rendering of revolution surfaces without polygonal meshes and which are displayed interactively by the graphics cards. The problem with this technique is that the 3D objects are created without any microrelief effect because the technique uses the shape map to find the intersection point, so, it does not take into account the realism of the revolved models. Based on this observation, this paper presents a new revolution mapping algorithm that takes into account the microrelief effect based on the bump mapping.

We used the bump mapping in order to perturb the normals of the 3D revolved surfaces, which will produce the illusion of microreliefs. Indeed, we must calculate the partial derivatives of the depth map in the preprocessing stage, and in the rendering stage, we solve the problem of the tangent space associated with each intersection point. Which leads us to determine the tangent and the bi-normal for each found intersection point and used them during the shading phase to perturb the normal of the basic shape.

The main problem is that the three components of the tangent space are computed directly from the parametric surface where they are determined using the geometric coordinates (x, y, z) and the texture coordinates (u, v) of the vertices constituting the 3D mesh, which is not the case for revolution mapping that is an image-based method and does not use the polygonal mesh to create the revolved models. So, we propose a solution to calculate the tangent space associated with each intersection point during the rendering stage.

To manage shading correctly, we present a synchronization between real-time depth scaling and the resulting shading. Indeed, if it is desired to modify the depth of the microrelief in real time, we will observe that the shading remains constant and does not vary according to the depth scale. We propose a solution to this limitation that consists in recalculating the normal during the rendering stage as a function of the depth scale and the partial derivatives of the depth map.

E-mail addresses: anouar.ragragui@usmba.ac.ma **(Anouar Ragragui)**, AdnaneOuazzaniChahdi@gmail.com **(Adnane Ouazzani Chahdi)**, akram_halli@yahoo.fr **(Akram Halli)**, khalidsatori@gmail.com **(Khalid Satori)**.