Spherical Skinning with Dual Quaternions and QTangents

Ivo Zoltan Frey
Crytek Research & Development
Frankfurt am Main, Germany
ivof@crytek.com

Ivo Herzeg
Crytek Research & Development
Frankfurt am Main, Germany
ivo@crytek.com

1. Introduction

Building upon previous work with Dual Quaternion skinning [KAVAN 2008], we will introduce a new format to represent tangent frames.

Vectors have commonly been the preferred format to store tangent frames, but a quaternion has many attractive features: smaller memory foot-print and native fit with spherical skinning.

We will go into detail on how we converted our vector based description of a tangent frame into what we call a QTangent.

Finally, by combining Dual Quaternions and QTangents we will present our spherical skinning solution which we employed in CryENGINE 3 and Crysis 2.

2. Exposition

We start by converting our tangent frame from vector form into a quaternion. We will showcase how this is done and what information we lose in the process: namely the mirroring.

We will take a look at different methods to encode the missing information with the quaternion which leads to the definition of our QTangent format. Special care will be taken to avoid singularities and to make sure that our encoding is suitable for vertex shaders running on the current generation of hardware.

Finally we will present our spherical skinning solution employing both Dual Quaternions and QTangents. We will go into details in how we implemented and optimized our vertex shaders to efficiently deal with quaternion math.

Following will be a comparison in performance, quality and memory foot-print with standard linear skinning.

We will also touch upon the benefits as well as the problems in adopting spherical skinning over the traditional linear solution, and what challenges this imposes on the content creation side.

Figure 1. Crysis 2 character rendered with our spherical skinning solution.

3. Results

By employing Dual Quaternions and QTangents we were able to improve performance, quality and memory foot-print over traditional linear skinning.

It also has been proven that QTangents can be transparently integrated into an existing pipeline.

4. Conclusions

We demonstrated a novel, efficient way to handle skinning, and proved its effectiveness during production.

Further work will involve finding efficient ways to employ QTangents for non-skinned geometry as well as propagate its use to the rest of the rendering pipeline.

References