

# Asynchronous Rendering Update for VR

Robin Lobel, August 2012



Fig 1. Asynchronous rotation update applied to the rendering

## Abstract

*This paper presents an asynchronous rendering update method allowing a constant 120 frames per second suited for Virtual Reality even when the source 3D engine is only able to produce 30 frames per second.*

## 1. Introduction

3D engines such as CryEngine or Unreal Engine offer high-quality rendering in real-time, making them ideal candidates for an immersive experience in VR.

Unfortunately such « realtime », usually up to 30-60 frames per second with today's hardware, is not fast enough for the emerging generation of VR headsets such as the Oculus Rift, which target a constant 120 frames per second.

120 frames per second is critical to VR as any tiny head movement must reflect immediately in the 3D rendering shown to the user, otherwise the risk of nausea caused by cinetose (sickness of movement) is high.

The largest visible changes to the point of view are caused by head rotations : any small degree of rotation affects every pixel of the rendered image, while small translations are barely visible at all with distant objects.

## 2. Asynchronous rendering

Fortunately, rotations can be accurately gathered with on-board gyroscopes at very high-rate, and these rotations infos can be used to transform the latest rendered picture by applying pixel shifts. Therefore the rendering process and post-rendering process must be treated asynchronously, the post-rendering process being a « last-minute » update to the latest rendered picture, more accurately reflecting the real head rotation at a given time.

What about the missing pixels on the border ? Fortunately the human peripheral vision is much less accurate than central vision. The missing pixels could be extrapolated from their direct neighbors, and that small peripheral blur, if noticed at all, would only occur for a very short time.

Of course HUD informations should be uncoupled from the rendered picture, since they are linked to head rotation and position. Therefore they should be rendered during the post-rendering stage.