Our implementation takes advantage of this property to save the number of rays and for smooth stereo rendering. We render a single wider image from a single point of view, then split it into both L and R images during the post processing phase. Precise parallax correction is required by further research, but this can also be solved with ray tracing and will not significantly increase the number of additional rays.

Parallel OpenCL ray tracer High performance and scalability are achieved by an OpenCL ray tracer designed to take advantage of multiple OpenCL devices. Since low latency is important for VR, every frame’s computation is split across devices rather than alternate frame rendering. We developed a load balancing method computing screen split dynamically for each frame by taking the computational power of devices, image area, number of sampling points, and computational cost of samples into consideration. As all of those factors are evaluated on the fly, it can use any combination of devices without any precomputation. Sampling points which are computed by a device is selected by the k-NN information stored for each pixel in parallel.

Results Figure 1 illustrates our results. The OpenCL ray tracer was tested with Oculus Rift DK1 HMD (1280x800@60Hz). Our technique effectively reduces the number of pixels to shade by 1/20, achieving more than 75 fps while preserving the same visual quality.

Acknowledgements Rungholt scene ©kescha. Sibenik ©Marko Dabrovic. Crytek ©Frank Meinl. IBL map from Smart IBL.

References

