Precomputed Ambient Occlusion for Character Skins

Adam G. Kirk
University of California, Berkeley

Okan Arikan
University of Texas, Austin

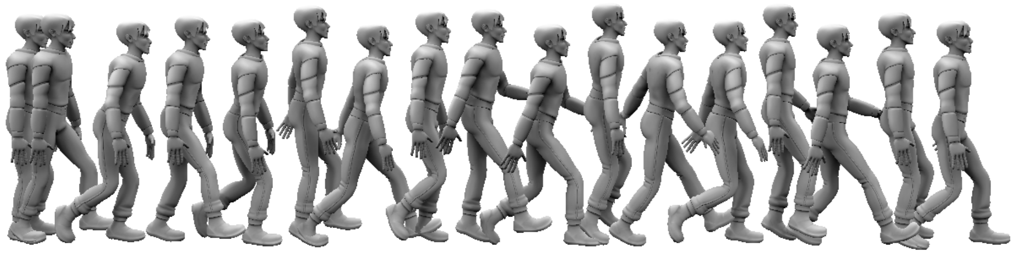

Figure 1: Time-lapse screen capture showing ambient occlusion values calculated in real-time.

We present a single-pass hardware accelerated method to compute ambient occlusion values in real-time on dynamic character skins. Ambient occlusion at a point is the cosine weighted integral of the visibility function. Assuming the scene is enclosed in a spherical area source light, ambient occlusion corresponds to the amount of light that will reach each point. Unfortunately, computing ambient occlusion is time-consuming. It requires sampling the hemisphere around each point to determine visibility. Our method takes a data driven approach to approximating ambient occlusion in real-time. It builds a set of linear functions over several subsets of poses, then produces the ambient occlusion using a blend of these functions. Because we are using linear functions our output is smooth, fast to evaluate, and easy to implement in a vertex or fragment shader.

Several researchers have examined precomputation of illumination. [Sloan et al., 2005] used spherical harmonics to compress low-frequency transfer functions at points on a static mesh. This method handles a wide range of low-frequency phenomena. However, it is limited to local changes in geometry, and does not handle the type of deformation seen in character animation. [James and Twigg, 2005] precompute illumination data at several points in state space, then blend this data as they deform the mesh. Our method is similar, but we provide a means to minimize error variance within local regions of state space. Recently, [Bunnell, 2005] proposed a multi-pass method for calculating ambient occlusion in real-time. This method makes approximations to the shape of elements, and error is dependent on the number of passes.

Ambient occlusion is a function of vertex positions, which in turn are functions of the character's joint angles. Therefore, ambient occlusion for character skins is also a function of joint angles. It is desirable to represent this function linearly, because linear functions can be computed easily in hardware. Unfortunately, ambient occlusion can change nonlinearly over a wide range of poses. The key observation of our method is that ambient occlusion changes smoothly for small changes in joint angles. Rather than building one large linear function, our method builds several linear functions, each local to a different region of pose space. These regions are defined using k-means clustering, and projected to a lower dimensional space using principle component analysis. For each vertex in the pose cluster, our method uses a least squares solver to fit a linear function in reduced pose space to the ambient occlusion values. Because these functions are fit independently in each pose cluster, there may be a discontinuity as the character transitions between clusters. Our method eliminates the discontinuity by blending between clusters using moving least squares. Computing the ambient occlusion using our method consists of evaluating a small dot product per vertex, which can be done easily in a hardware shader.

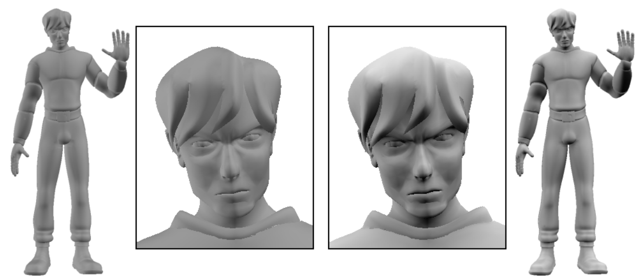

Figure 2: Comparison of standard opengl shading (left) to our method (right).

Clusters	Dimensions	Mean Error	Size(MB)
5	5	0.011	4.5
5	10	0.008	8.1
5	15	0.007	11.6
10	5	0.008	8.8
10	10	0.006	16.0
10	15	0.005	23.2
15	5	0.008	13.1
15	10	0.006	23.9
15	15	0.005	34.7

Table 1: Comparison of error and compression based on the number and dimension of pose clusters. Error is measured as the absolute value of the difference between computed values and ground truth per vertex.

Our dataset consists of 4130 frames of motion data, including walking, running, skipping, reaching, and crouching motions. In Table 1 we summarize effects on the accuracy and compression due to changes in parameters. Ambient occlusion is defined to be a number between zero and one. In general, note that the mean error between values obtained with our method and ground truth is quite low even with a function representation less than 5MB in size. For further results, please see the accompanying video.

References

Bunnell, M. 2005. *Dynamic Ambient Occlusion and Indirect Lighting*. Addison-Wesley Professional, ch. 14.

James, D. L., and Twigg, C. D. 2005. Skinning mesh animations. *ACM Transactions on Graphics (SIGGRAPH 2005)*.

Sloan, P.-P., Luna, B., and Snyder, J. 2005. Local, deformable precomputed radiance transfer. In *Proceedings of ACM SIGGRAPH 2005*, ACM Press.