Eurographics Symposium on Rendering - Experimental Ideas & Implementations (2016)
E. Eisemann and E. Fiume (Editors)

Fast Shadow Map Rendering for Many-Lights Settings

[SUPPLEMENTAL MATERIAL]

K. Selgrad, J. Müller, C. Reintges, M. Stamminger

Computer Graphics Group, University of Erlangen-Nuremberg, Germany

In this supplemental document we provide full tabulation of the results presented in the diagrams of our paper, as well as further results that were only given in summary.

- Table 1 lists the breakdown of render times underlying Figure 5 in the paper and also the total render times given in Figure 6 in the paper.
- Table 2 backs up our summary that performance drops for grid resolutions finer than $32 \times 32 \times 32$, but that lower resolutions could be used.
- Table 3 shows that our method works over a wide range of shadow map resolutions.
- Table 4 lists culling time and efficiency (reduction in scene primitives) and resulting render times when comparing the old line traversal found in previous work with our new warp-parallel traversal. Regarding culling time this optimization is negligible, but the resulting reduction in scene primitives provides a noticeable speed-up.
- Table 5 further shows how fast the culling step of our method is by listing the impact of intra-voxel light clustering on render times.
- Finally, Table 6 shows render times when using clipping versus keeping a separate list of overly large primitives.

© 2016 The Author(s)
Eurographics Proceedings © 2016 The Eurographics Association.

		Per light						
Scene	Method	Culling	Build draw call	Render SM	Shade	Sum	Total frame time	Tris after culling
Parabolic Shadow Maps								
GRILLE	No Culling	0.0000	0.0000	1.2700	0.1500	1.4500	146	100.0 %
	Frustum Culling	0.0380	0.0230	0.1700	0.1500	0.3900	45	6.0 %
	Voxel-Culling	0.0100	0.0210	0.1100	0.1500	0.3400	40	2.9 %
ROOM	No Culling	0.0000	0.0000	1.4000	0.1400	1.5900	311	100.0 %
	Frustum Culling	0.0490	0.0320	0.7300	0.1500	0.9900	198	43.7 %
	Voxel-Culling	0.0110	0.0240	0.2000	0.1500	0.4500	94	9.6 %
DOOR	No Culling	0.0000	0.0000	1.4400	0.1500	1.6300	316	100.0 %
	Frustum Culling	0.0440	0.0270	0.4200	0.1500	0.6800	130	21.8 %
	Voxel-Culling	0.0075	0.0190	0.0850	0.1500	0.3100	66	1.7 %
FLAG	No Culling	0.0000	0.0000	1.4100	0.1500	1.6100	626	100.0 %
	Frustum Culling	0.0460	0.0310	0.6100	0.1500	0.8900	348	34.4 %
	Voxel-Culling	0.0100	0.0250	0.4100	0.1500	0.6600	262	23.1 %
VILLAGE	No Culling	0.0000	0.0000	1.6100	0.1500	1.8000	993	100.0 %
	Frustum Culling	0.0390	0.0230	0.7200	0.1500	0.9800	542	34.0 %
	Voxel-Culling	0.0076	0.0180	0.5000	0.1500	0.7300	405	27.7 %
CELLAR	No Culling	0.0000	0.0000	1.3600	0.1500	1.5500	156	100.0 %
	Frustum Culling	0.0450	0.0260	0.7500	0.1500	1.0200	105	42.9 %
	Voxel-Culling	0.0100	0.0210	0.1700	0.1500	0.4100	45	6.9 %
Cube Shadow Maps								
GRILLE	No Culling	0.0000	0.0000	6.3900	0.4800	6.8700	665	100.0 %
	Frustum Culling	0.0390	0.0230	0.7300	0.4200	1.2700	124	6.0 %
	Voxel-Culling	0.0100	0.0210	0.4000	0.4100	0.9000	92	2.9 %
ROOM	No Culling	0.0000	0.0000	8.3600	0.4600	8.8200	1702	100.0 %
	Frustum Culling	0.0700	0.0270	5.0500	0.4200	5.6000	1084	43.7 %
	Voxel-Culling	0.0110	0.0250	1.1500	0.4200	1.6500	325	9.6 %
DOOR	No Culling	0.0000	0.0000	8.9600	0.4600	9.4800	1804	100.0 %
	Frustum Culling	0.0680	0.0230	2.5400	0.4200	3.0800	593	21.8 %
	Voxel-Culling	0.0075	0.0190	0.2900	0.4200	0.7800	154	1.7 %
FLAG	No Culling	0.0000	0.0000	8.0600	0.4600	8.5600	3290	100.0 %
	Frustum Culling	0.0790	0.0260	3.9200	0.4200	4.4800	1729	34.4 %
	Voxel-Culling	0.0100	0.0270	2.5200	0.4200	3.0200	1171	23.1 %
VILLAGE	No Culling	0.0000	0.0000	7.7900	0.4500	8.3100	4540	100.0 %
	Frustum Culling	0.0630	0.0200	4.0500	0.4200	4.6000	2532	34.0 %
	Voxel-Culling	0.0076	0.0190	2.4300	0.4200	2.9200	1612	27.7 %
CELLAR	No Culling	0.0000	0.0000	7.5700	0.4500	8.0400	782	100.0 %
	Frustum Culling	0.0690	0.0350	4.8800	0.4200	5.4500	531	42.9 %
	Voxel-Culling	0.0100	0.0220	0.7500	0.4200	1.2400	127	6.9 %

Table 1: *Detailed render times (in ms) of the different scenes shown in Figures 1 and 4 in the paper. The plot shown in Figures 5 and 6 are generated on this data.*

© 2016 The Author(s)
Eurographics Proceedings © 2016 The Eurographics Association.

Voxel Grid Res.	GRILLE	ROOM	DOOR	FLAG	VILLAGE	CELLAR
$16 \times 16 \times 16$						
Total per light	0.33	0.42	0.32	0.69	0.74	0.41
Total frame time	39.00	86.60	68.20	272.00	419.00	46.10
$32 \times 32 \times 32$						
Total per light	0.33	0.46	0.31	0.67	0.73	0.40
Total frame time	39.20	93.50	66.00	262.00	407.00	45.20
$64 \times 64 \times 64$						
Total per light	0.44	0.60	0.42	0.75	0.79	0.51
Total frame time	48.30	122.30	86.40	301.00	440.00	56.00

Table 2: *Render times (in ms) for varying grid resolutions when using parabolic shadow maps. It can be seen that the performance of the smaller resolution is similar, but for larger resolutions render times increase rapidly.*

Shadowmap Res.	GRILLE	ROOM	DOOR	FLAG	VILLAGE	CELLAR
Voxel Culling						
256×256						
Render SM	0.100	0.190	0.079	0.400	0.490	0.160
Total per light	0.330	0.450	0.300	0.650	0.700	0.400
Total frame time	38.400	92.500	63.500	257.000	398.000	44.200
512×512						
Render SM	0.110	0.200	0.085	0.410	0.500	0.170
Total per light	0.340	0.450	0.320	0.670	0.720	0.410
Total frame time	38.600	93.700	65.100	265.000	406.000	45.600
1024×1024						
Render SM	0.170	0.240	0.150	0.480	0.580	0.240
Total per light	0.420	0.490	0.380	0.750	0.830	0.500
Total frame time	45.000	101.000	79.000	296.000	462.000	54.100
2048×2048						
Render SM	0.460	0.440	0.360	0.840	0.980	0.590
Total per light	0.730	0.700	0.620	1.120	1.220	0.840
Total frame time	76.200	142.000	123.000	443.000	682.000	89.700
Frustum Culling						
256×256						
Render SM	0.160	0.720	0.420	0.620	0.720	0.750
Total per light	0.390	1.010	0.660	0.880	0.970	1.000
Total frame time	44.800	203.000	138.000	356.000	540.000	104.000
512×512						
Render SM	0.160	0.730	0.430	0.630	0.730	0.770
Total per light	0.400	1.020	0.710	0.900	0.980	1.050
Total frame time	46.200	205.000	132.000	347.000	548.000	107.000
1024×1024						
Render SM	0.240	0.780	0.500	0.680	0.830	0.870
Total per light	0.500	1.050	0.770	0.990	1.060	1.130
Total frame time	51.200	213.000	153.000	383.000	591.000	116.000
2048×2048						
Render SM	0.600	1.000	0.920	1.170	1.320	1.430
Total per light	0.880	1.290	1.220	1.480	1.610	1.780
Total frame time	91.300	254.000	237.000	577.000	884.000	175.000

Table 3: *Render times (in ms) for different shadow map resolutions with parabolic shadow maps. With increasing shadow map resolution the gap between our method and frustum culling closes, however, very slowly. The average time required to render the shadow map using our method as compared to frustum culling is 0.43% for 256×256, 0.45% for 512×512, 0.49% for 1024×1024 and 0.57% for 2048×2048.*

© 2016 The Author(s)
Eurographics Proceedings © 2016 The Eurographics Association.

		Per light				
Scene	Method	Culling	Render SM	Sum	Total frame time	Tris after culling
GRILLE	Warp-parallel DDA	0.0100	0.1100	0.3500	39	2.9 %
	Standard DDA	0.0170	0.1200	0.3700	41	3.7 %
	Frustum Culling	0.0390	0.1600	0.3900	44	6.0 %
ROOM	Warp-parallel DDA	0.0120	0.2000	0.4500	93	9.6 %
	Standard DDA	0.0230	0.2100	0.4700	98	10.4 %
	Frustum Culling	0.0470	0.7300	0.9800	195	43.7 %
DOOR	Warp-parallel DDA	0.0074	0.0850	0.3100	65	1.7 %
	Standard DDA	0.0130	0.1100	0.3400	71	3.2 %
	Frustum Culling	0.0350	0.4200	0.6700	131	21.8 %
FLAG	Warp-parallel DDA	0.0110	0.4200	0.6700	267	23.1 %
	Standard DDA	0.0220	0.5000	0.7600	300	28.1 %
	Frustum Culling	0.0460	0.6200	0.9000	352	34.4 %
VILLAGE	Warp-parallel DDA	0.0076	0.4900	0.7200	405	27.7 %
	Standard DDA	0.0140	0.5800	0.8200	458	32.5 %
	Frustum Culling	0.0400	0.7100	0.9500	531	34.0 %
CELLAR	Warp-parallel DDA	0.0100	0.1600	0.4100	45	6.9 %
	Standard DDA	0.0200	0.2000	0.4500	50	9.2 %
	Frustum Culling	0.0490	0.7500	1.0300	105	42.9 %

Table 4: *Comparison of using our new, warp-parallel conservative line rasterization method to employing standard DDA line traversal with conservative tagging of the one-ring of each cell traversed (all times in ms). The improved performance of the culling step itself is, in fact, negligible, however, since it is also not as overly conservative as the previous method fewer cells are tagged to be relevant and thus shadow map rendering performance increases.*

		Per light					
Scene	Method	Culling	Build draw call	Render SM	Shade	Sum	Total frame time
GRILLE	Cull each light	0.0310	0.0370	0.1100	0.1500	0.3660	42
	Cull each cluster	0.0100	0.0210	0.1100	0.1500	0.3400	40
ROOM	Cull each light	0.0300	0.0390	0.2000	0.1500	0.4800	101
	Cull each cluster	0.0110	0.0240	0.2000	0.1500	0.4500	94
DOOR	Cull each light	0.0310	0.0410	0.0850	0.1500	0.3440	72
	Cull each cluster	0.0075	0.0190	0.0850	0.1500	0.3100	66
FLAG	Cull each light	0.0300	0.0410	0.4100	0.1500	0.6970	276
	Cull each cluster	0.0100	0.0250	0.4100	0.1500	0.6600	262
VILLAGE	Cull each light	0.0300	0.0370	0.5000	0.1500	0.7640	424
	Cull each cluster	0.0076	0.0180	0.5000	0.1500	0.7300	405
CELLAR	Cull each light	0.0340	0.0390	0.1700	0.1500	0.4500	49
	Cull each cluster	0.0100	0.0210	0.1700	0.1500	0.4100	45

Table 5: *Clustering lights that lie in the same voxel speeds up rendering by 4.5% to 9%.*

© 2016 The Author(s)
Eurographics Proceedings © 2016 The Eurographics Association.

	GRILLE	ROOM	DOOR	FLAG	VILLAGE	CELLAR
With clipping						
Total render time, parabolic shadow maps	39.1	93.0	64.3	261.5	405.1	45.6
Total render time, cube maps	92.5	322.1	153.4	1170	1627	126.3
Grid construction time	301	301	301	301	328	301
Without clipping:						
Total render time, parabolic shadow maps	40.4	91.9	73.3	241.7	384.9	47.2
Total render time, cube maps	97.0	304.5	183.3	1043	1507	128.7
Grid construction time	43.8	43.8	43.8	43.8	48.1	43.8
Comparison:						
Relative render time, parabolic shadow maps	103.3 %	98.8	113.8	92.4	95.0	103.5
Relative render time, cube maps	104.8 %	94.5	119.5	89.1	92.6	101.9
Relative construction time	14.5 %	14.5 %	14.5 %	14.5 %	14.7	14.5 %

Table 6: *The initial grid construction has not yet been optimized and is a simple, sequential process implemented on the CPU. Clipping the input triangles to the grid cells generates a considerable overhead during construction, and it seems that this overhead does not consistently pay off during rendering. We are confident that both approaches lend themselves nicely to highly parallel implementation on the GPU and find that this is one of the most important issues to tackle in future work. Note however, that even as-is, the faster-to-construct version does not cause our method to be slower than frustum culling (while certainly being quite costly).*

© 2016 The Author(s)
Eurographics Proceedings © 2016 The Eurographics Association.