

Lab 1 Report

Esfandiar Manii (CSC 572)
Mitchell Rosen (CSC 458)

Experiment 1: The first experiments was on rasterizing the Bunny10k.m mesh file. Due to not having colors for each vertex, we decided to use a constant color value for all vertices. Figure 1 is a 2000x2000 pixels image. The total runtime of the rasterizer on Tesla was 5.38 seconds. Also, the rasterizer running profile is presented below which shows the percentages and runtime of each function inside the main code.

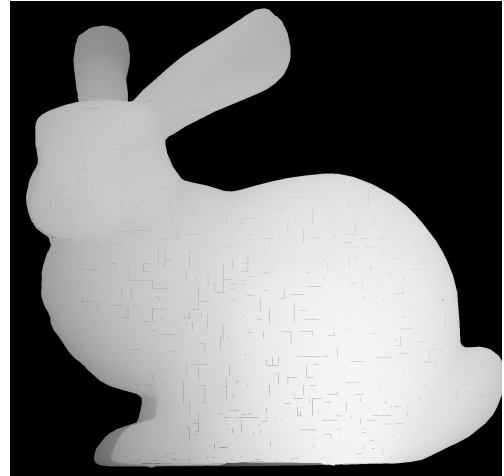


Figure 1: Rasterized Bunny, 2000x2000 pixels

Each sample counts as 0.01 seconds.

	%	cumulative	self		self	total	
time	seconds	seconds	calls	s/call	s/call		name
47.42	2.55	2.55	66287628	0.00	0.00		normalize(Vector3*)
22.50	3.76	1.21	1	1.21	5.18		calculateBarycentric(int, int)
12.09	4.41	0.65	33143814	0.00	0.00		computeColor(Vector3, float)
11.16	5.01	0.60	180574719	0.00	0.00		Vector3::Vector3(float, float, float)
1.86	5.11	0.10	33143814	0.00	0.00		dotProduct(Vector3, Vector3)
1.49	5.19	0.08	1	0.08	0.08		imageWrite()
1.30	5.26	0.07	3712646	0.00	0.00		convertPointTo2d(Vector3, Vector3)
1.12	5.32	0.06	5065	0.00	0.00		operator new(unsigned long, void*)
0.37	5.34	0.02	10000	0.00	0.00		calculateBoundingBox(Tri*)
0.37	5.36	0.02	1	0.02	5.28		Parse(int)
0.37	5.38	0.02					main
0.00	5.38	0.00	15065	0.00	0.00		readLine(char*)
0.00	5.38	0.00	2	0.00	0.00		dim3::dim3(unsigned int, unsigned int, unsigned int)
0.00	5.38	0.00	1	0.00	0.00		global constructors keyed to Triangles
0.00	5.38	0.00	1	0.00	0.00		global constructors keyed to parImage
0.00	5.38	0.00	1	0.00	0.00		Rasterizer(Tri*, float*, float*, int)
0.00	5.38	0.00	1	0.00	0.08		readStream(std::istream&)
0.00	5.38	0.00	1	0.00	0.08		ReadFile(char*)

Experiment 2: The second experiment was on generating 25 bunnies inside a 2000x2000 pixels image. This experiment was done by a sequential rasterizing algorithm which each triangle of the bunny model is rasterized one by one. The total runtime of the rasterizer on Tesla was 2.67 seconds. The reason that the running time is less than the first experiment is that in this experiment, one bunny is generated and then 25 copies will be written to the file (without rasterizing). The output of the rasterizer is presented in the figure 2. Also, the rasterizer running profile is presented below which shows the percentages and runtime of each function inside the main code.

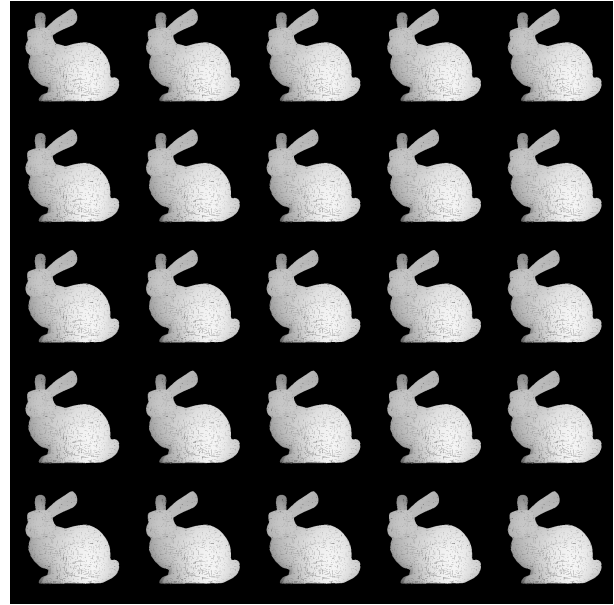


Figure 2: 25 Rasterized Bunnies. Sequential

Rasterizer.

Each sample counts as 0.01 seconds.

%	cumulative	self	self	total		
time	seconds	seconds	calls	s/call	s/call	name
44.59	1.19	1.19	26505504	0.00	0.00	normalize(Vector3*)
23.98	1.83	0.64	25	0.03	0.10	calculateBarycentric(int, int)
12.74	2.17	0.34	13252752	0.00	0.00	computeColor(Vector3, float)
10.12	2.44	0.27	72278937	0.00	0.00	Vector3::Vector3(float, float, float)
3.37	2.53	0.09	1	0.09	0.09	imageWrite()
2.44	2.60	0.07	13252752	0.00	0.00	dotProduct(Vector3, Vector3)
0.75	2.62	0.02	1502528	0.00	0.00	convertPointTo2d(Vector3, Vector3)
0.75	2.64	0.02	5065	0.00	0.00	operator new(unsigned long, void*)
0.75	2.66	0.02				main
0.37	2.67	0.01	1	0.01	2.56	Parse(int)
0.19	2.67	0.01	10000	0.00	0.00	calculateBoundingBox(Tri*)
0.00	2.67	0.00	15065	0.00	0.00	readLine(char*)
0.00	2.67	0.00	2	0.00	0.00	dim3::dim3(unsigned int, unsigned int, unsigned int)
0.00	2.67	0.00	1	0.00	0.00	global constructors keyed to Triangles
0.00	2.67	0.00	1	0.00	0.00	global constructors keyed to parImage
0.00	2.67	0.00	1	0.00	0.00	Rasterizer(Tri*, float*, float*, int)
0.00	2.67	0.00	1	0.00	0.03	readStream(std::istream&)
0.00	2.67	0.00	1	0.00	0.03	ReadFile(char*)

Experiment 3: The third experiment was on generating 25 bunnies inside a 2000x2000 pixels image. This experiment was done by a parallelized rasterizing algorithm which all triangles of the bunny model are rasterized in parallel. The total runtime of the rasterizer on Tesla was 0.15 seconds. . The output of the rasterizer is presented in the figure 3. Also, the rasterizer running profile is presented below which shows the percentages and runtime of each function inside the main code.

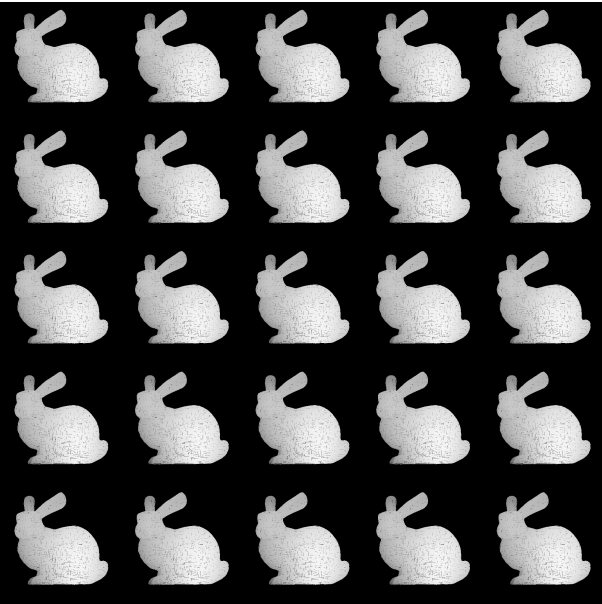


Figure 3: 25 Rasterized Bunnies. Parallelized Rasterizer.

%	cumulative	self		self	total	
time	seconds	seconds	calls	ms/call	ms/call	name
53.36	0.08	0.08	1	80.04	80.04	imageWrite()
46.69	0.15	0.07				main
0.00	0.15	0.00	300000	0.00	0.00	std::vector<Vector3*, std::allocator<Vector3*> >::operator[](unsigned long)
0.00	0.15	0.00	125065	0.00	0.00	Vector3::Vector3(float, float, float)
0.00	0.15	0.00	30000	0.00	0.00	convertPointTo2d(Vector3, Vector3)
0.00	0.15	0.00	15065	0.00	0.00	readLine(char*)
0.00	0.15	0.00	10000	0.00	0.00	calculateBoundingBox(Tri*)
0.00	0.15	0.00	2	0.00	0.00	dim3::dim3(unsigned int, unsigned int, unsigned int)
0.00	0.15	0.00	1	0.00	0.00	Rasterizer(Tri*, float*, float*, int)
0.00	0.15	0.00	1	0.00	0.00	readStream(std::istream&)
0.00	0.15	0.00	1	0.00	0.00	Parse(int)
0.00	0.15	0.00	1	0.00	0.00	ReadFile(char*)

Finally: The average error between the CPU calculations and GPU calculations was very close to zero.