電機四 B00901016 張宇方

Rasterization

3DMM HW1 2015

A. Describe your rasterization algorithm

Wireframe : line rasterization

I implement incremental algorithm (Digital Differential Analyzer) for line rasterization. To deal with the gap issue in scanning line, the incremental step axis depends on the slope. If the slope magnitude is less than 1, I iterate in x-direction in each step otherwise in y-direction. Additionally, to realize the depth visualization(the nearer the deeper), I draw the pixel color with 1/depth.

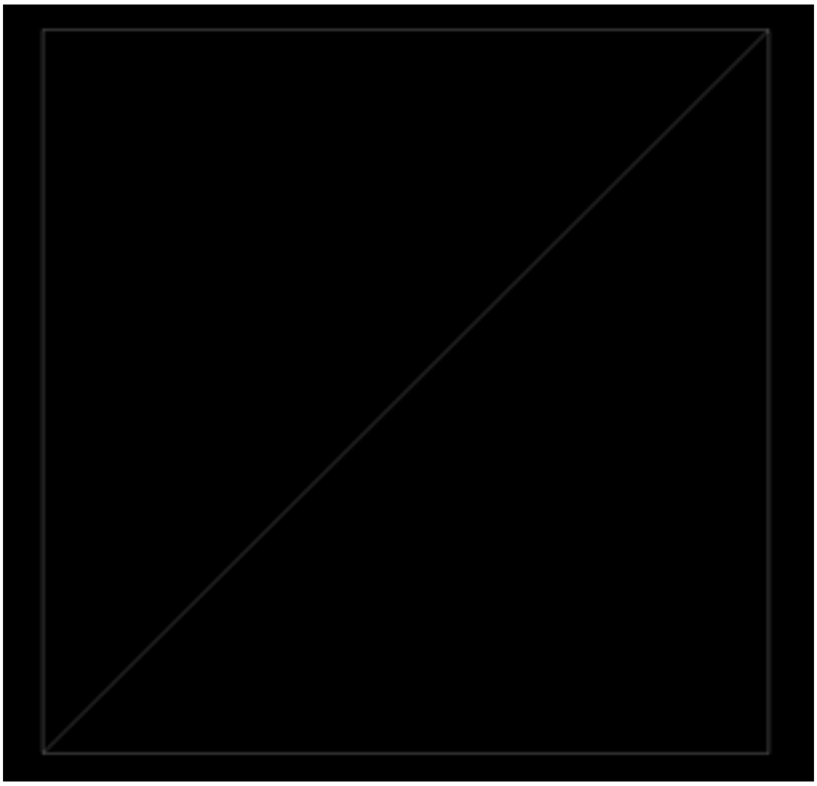
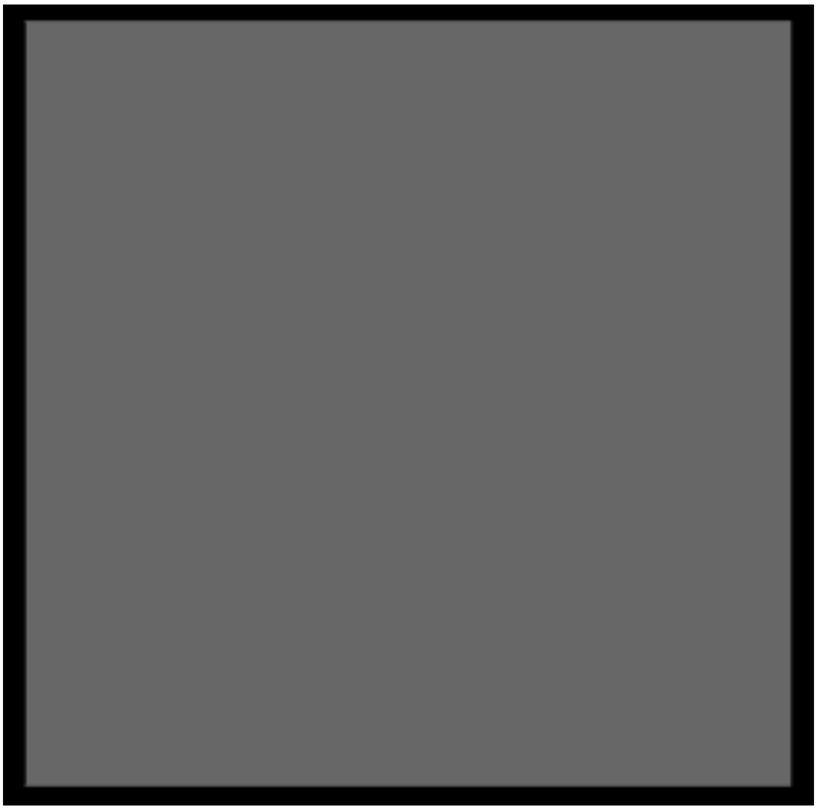
Solid : triangle rasterization

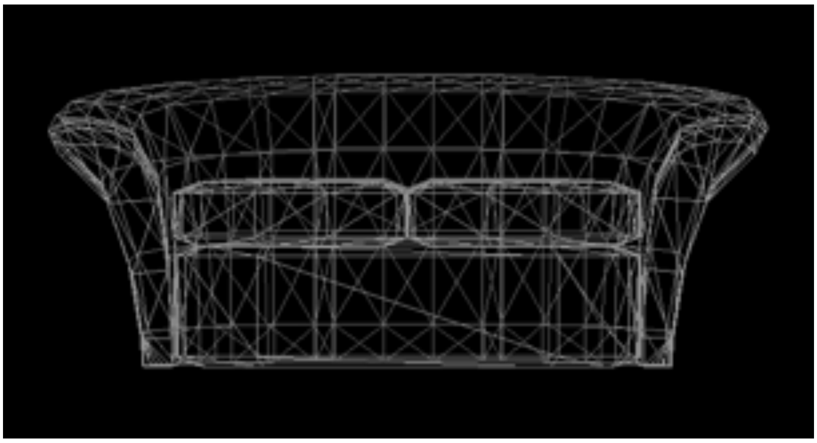
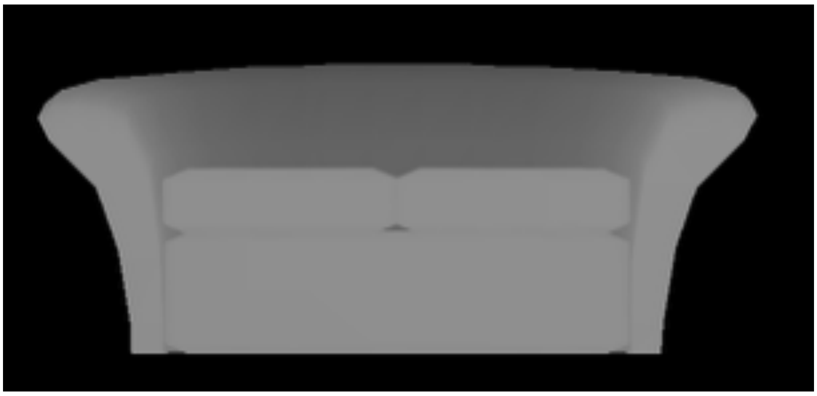
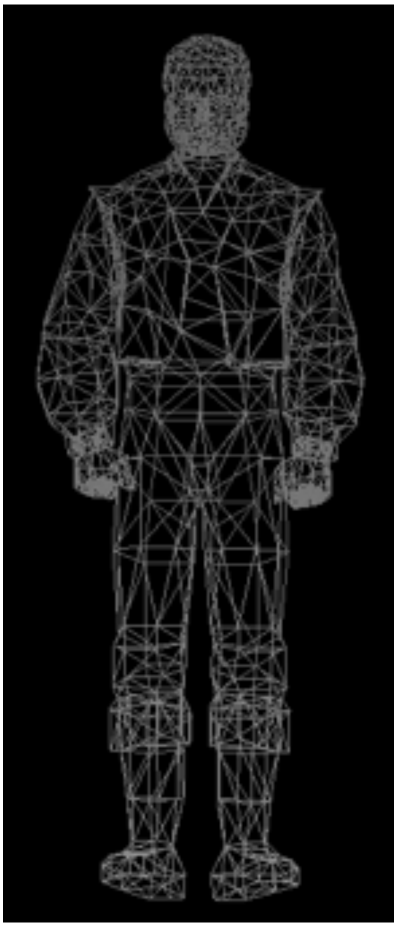
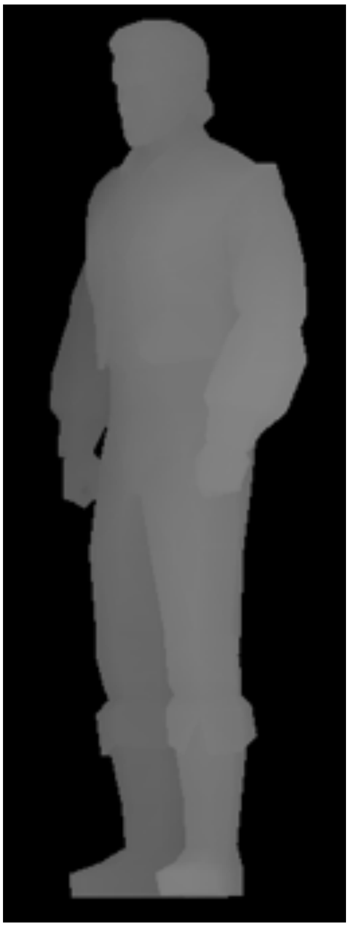
The main idea of the triangle(V1-V2-V3) rasterization is to split the triangle to flat bottom or top triangles to generalize the work. For both flat triangles, I find the prominent point(V1) and draw V1-V2 but stop if the algorithm moves one pixel in y-direction. Furthermore, I draw also the line V1-V3 and stop if the algorithm moves one pixel in y-direction. At this point we are on the same y-coordinate for line V1V2 as well as for line V1V3. Finally, I draw the horizontal lines between both current line points.

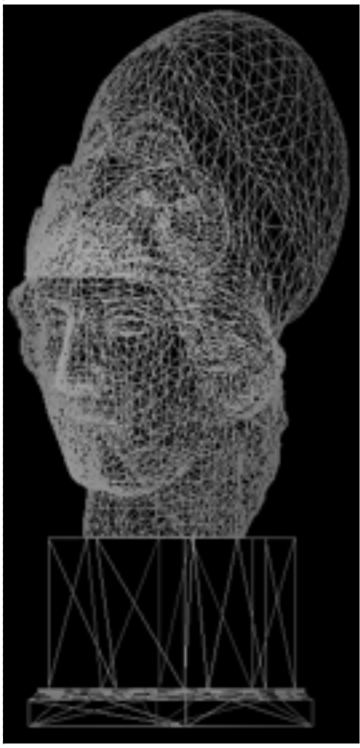
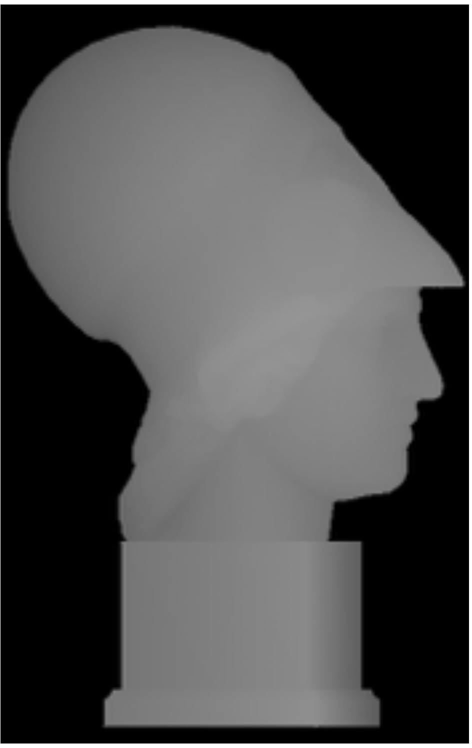
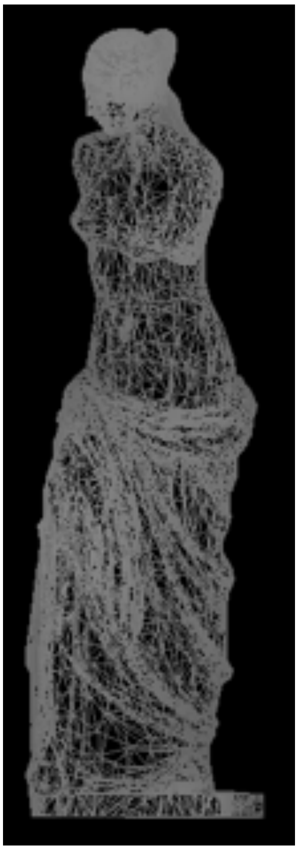
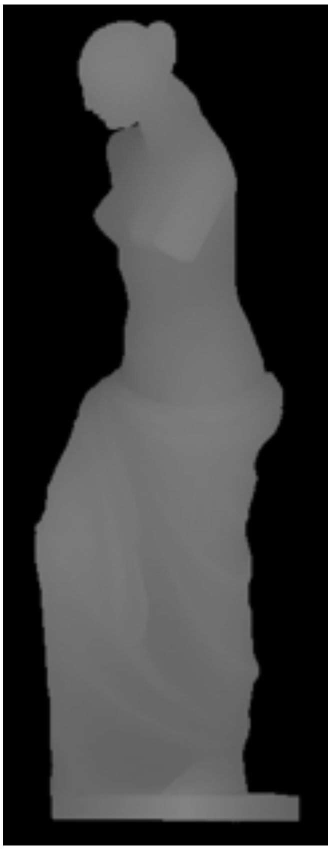
B. Evaluate performance of rasterization with/without culling

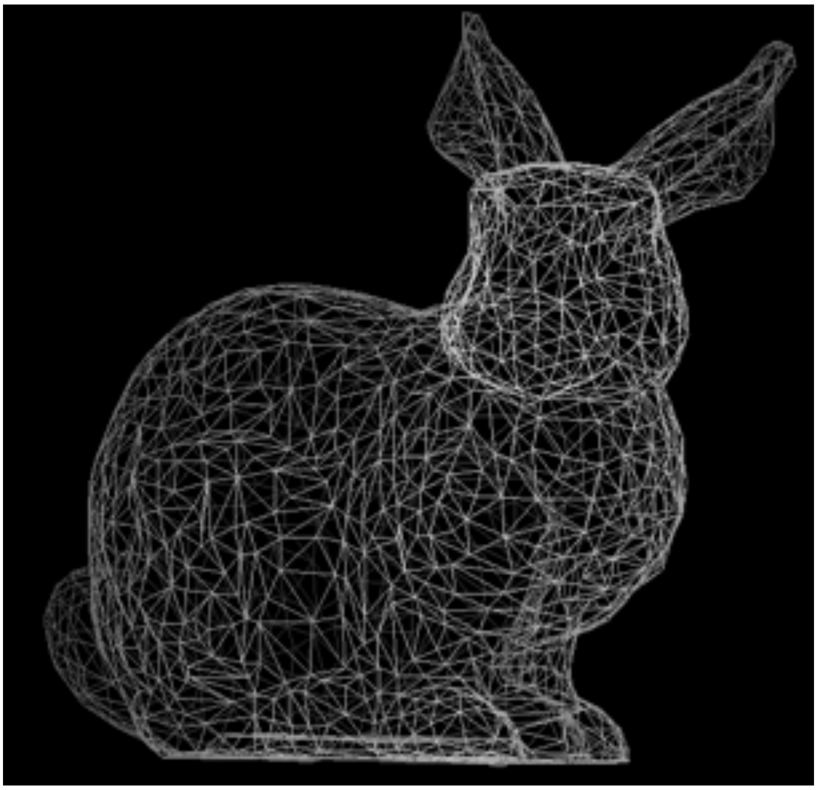
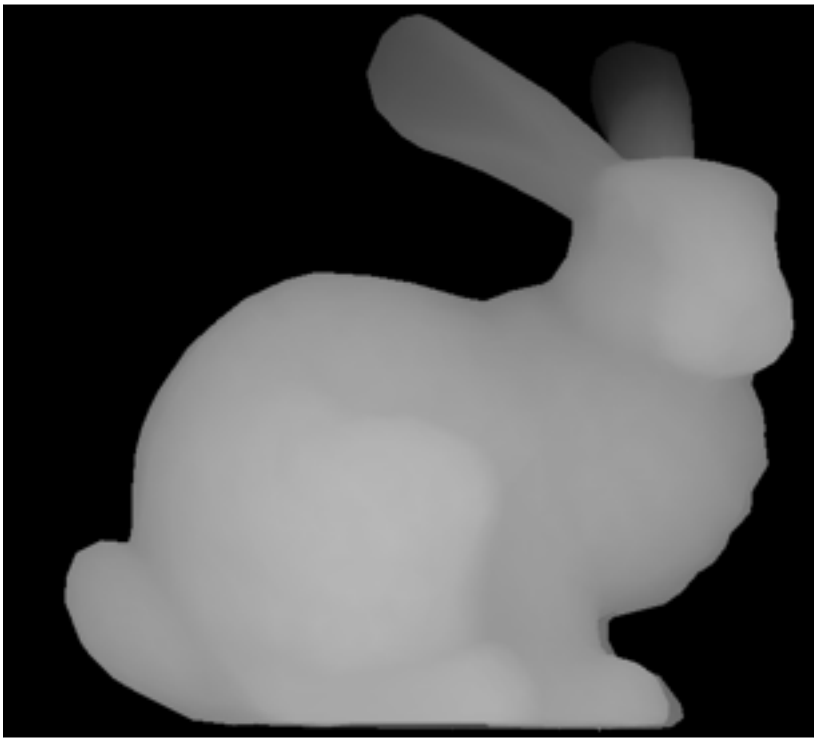
Model Name	Time (with culling)	Time (without culling)
quad	54.98	44.47
couch	43.63	33.75
blaze	59.79	37.84
ateneal	30.46	20.08
venusm	20.40	12.46
bunnyC	31.82	17.92
duck4KN	29.35	18.69
happy10KN	29.24	16.91
dragon10KN	27.97	15.46
elephant16KN	25.72	13.18
Statue_of_Liberty	15.62	7.65
Nissan_Pathfinder	6.74	3.40

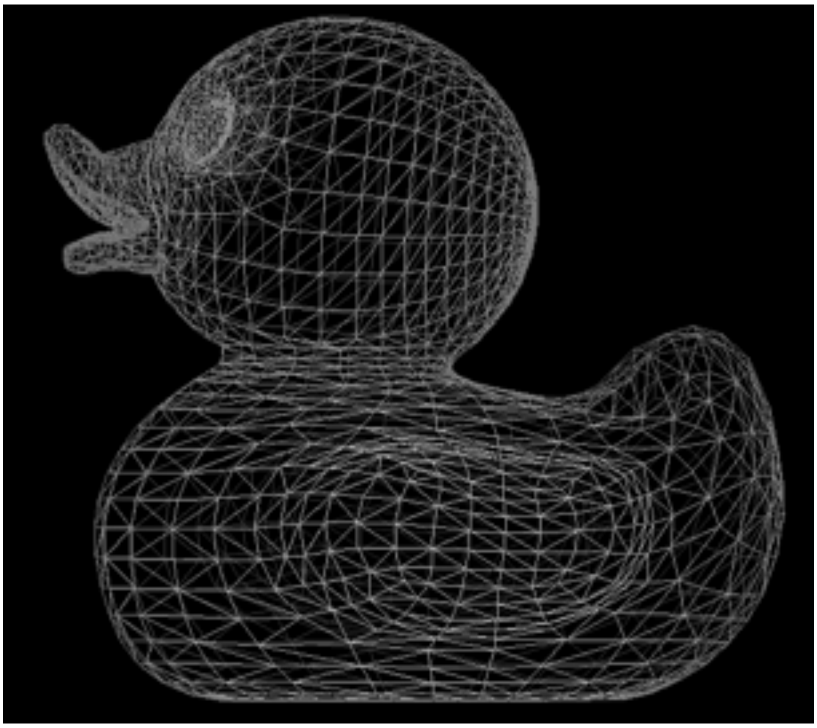
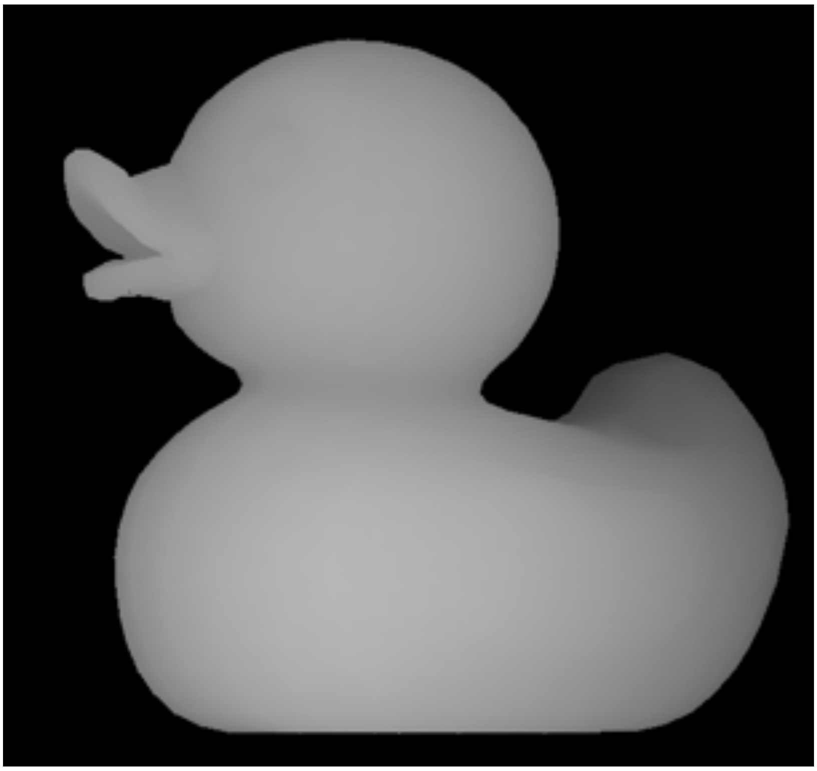
C. rasterized results of wireframe / solid mode

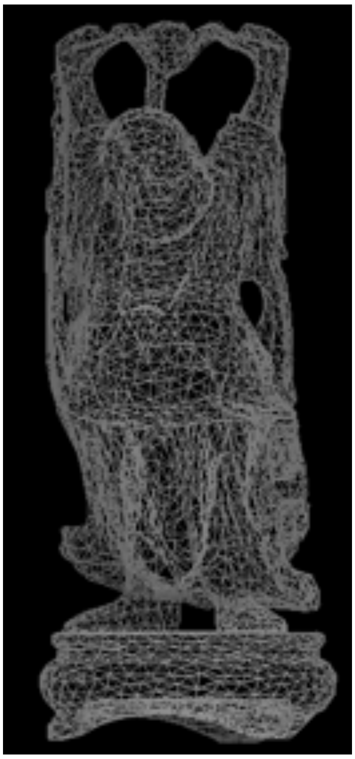
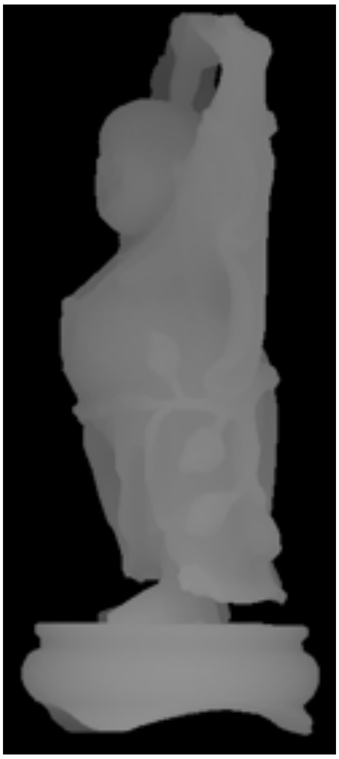
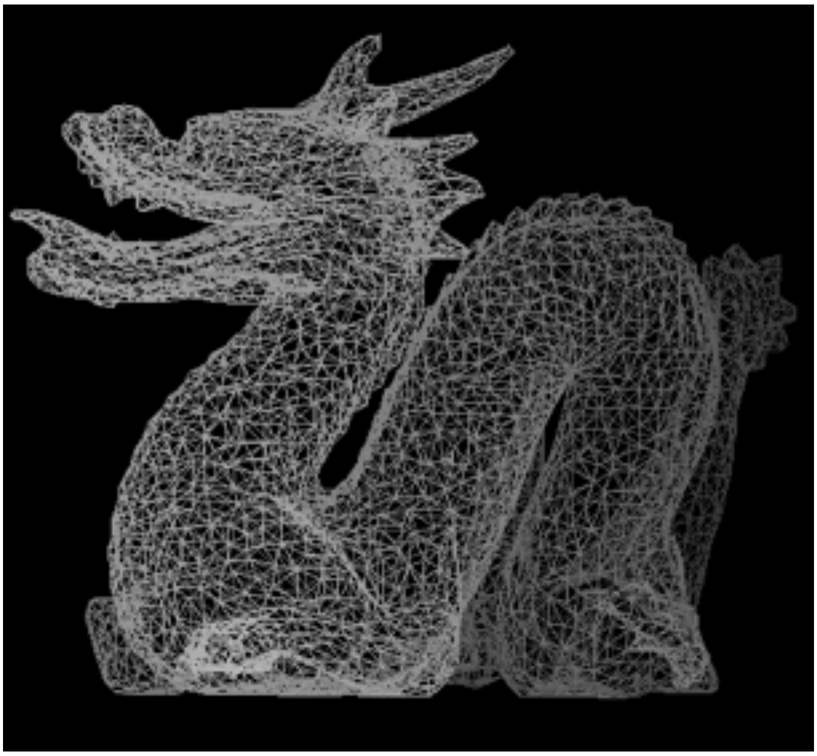
Model	**Wireframe/ Solid**	
quad	W	
	S	

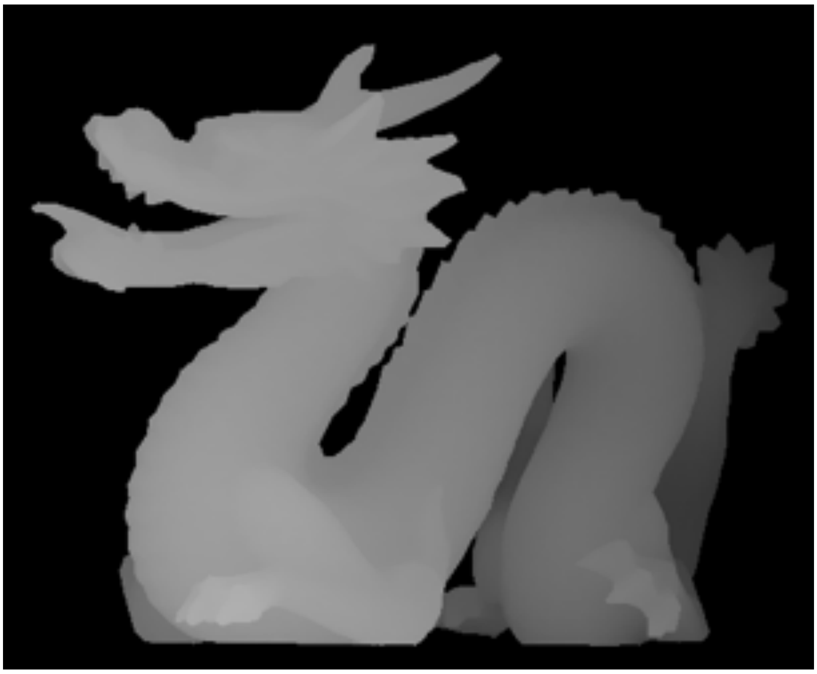
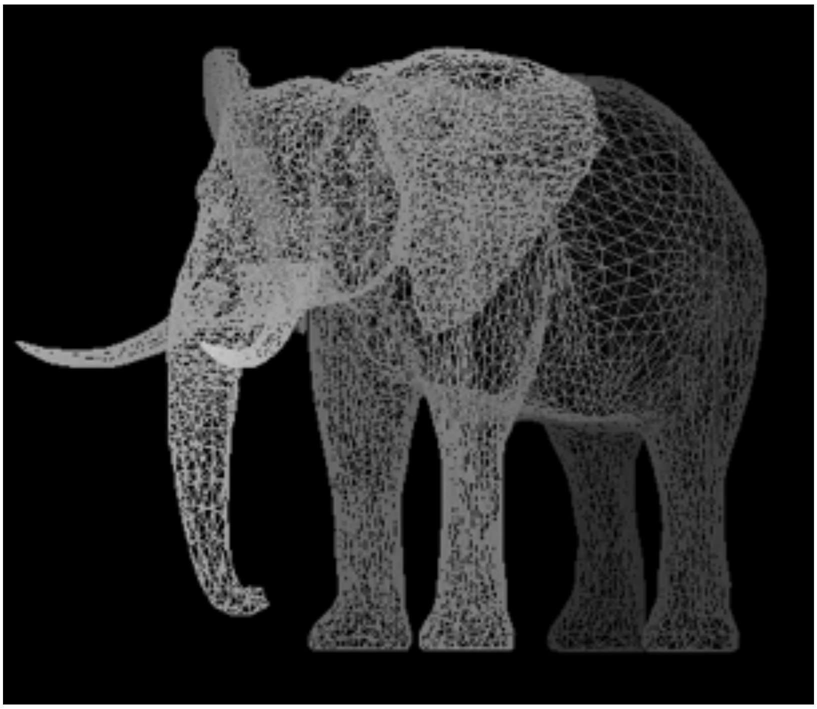
couch	W	
	S	
blaze	W/S	 

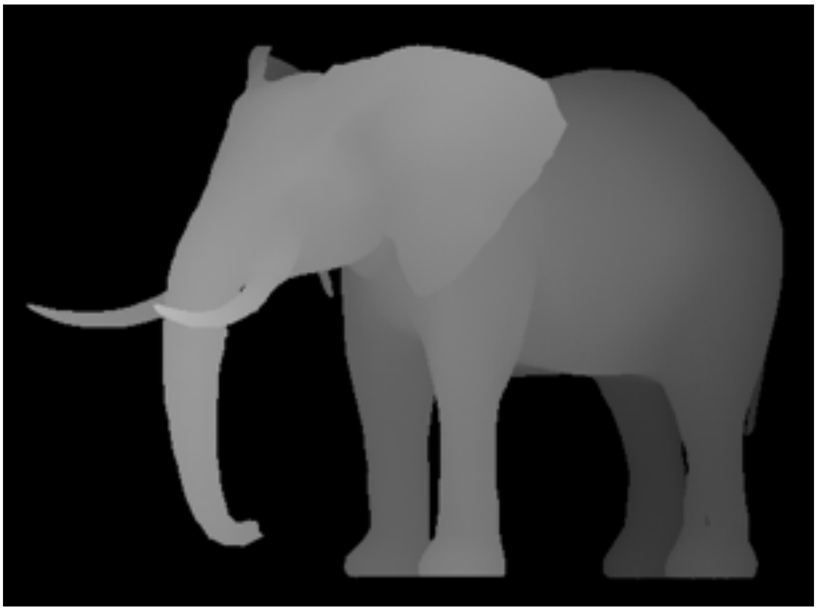
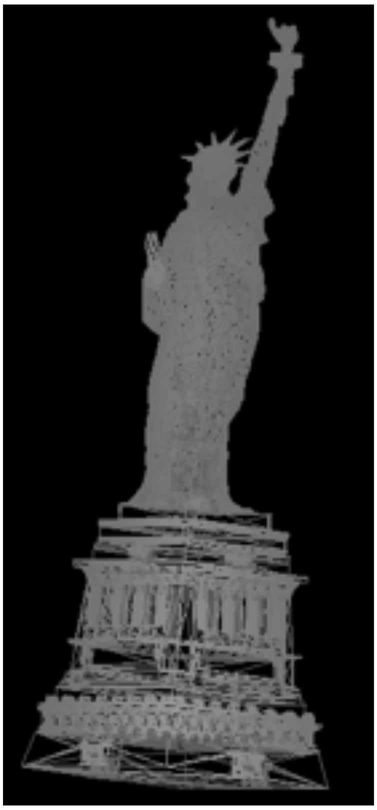
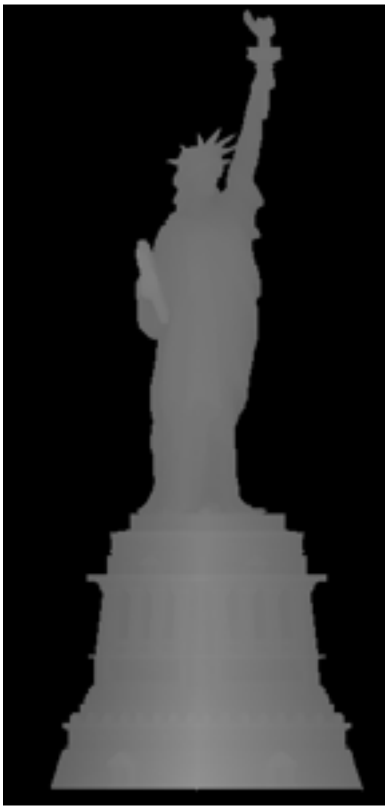
ateneal	W/S	 
venusm	W/S	 

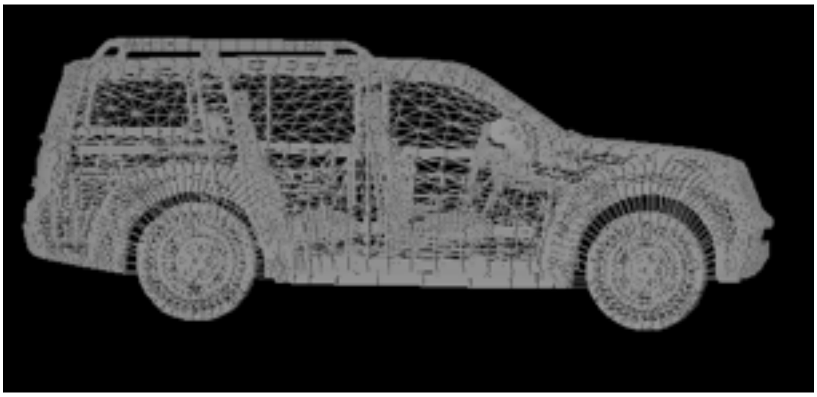
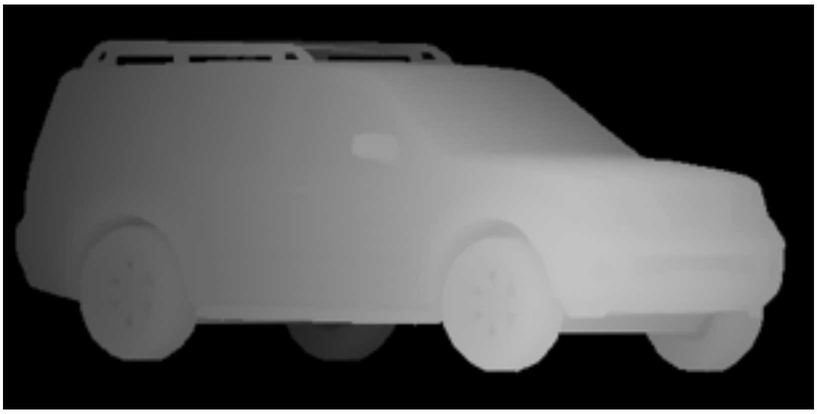
bunnyC	W	
	S	

duck4KN	W	
	S	

happy10KN	W/S	 
dragon	W	

	S	
elephant	W	

	S	
Statue_of_ Liberty	W/S	 

Nissan_ Pathfinder	W	
	S	

From the frame per second (fps) result above, we can observe that rasterization with back face culling is more efficient than without culling and more significant as increasing faces.