

CLAIMS:

1. An apparatus comprising:

logic, the logic at least partially comprising hardware logic, to determine a primitive intersection cost value for each atomic tile of an image at least partially based on a vertex element size, a vertex shader length, and a number of vertices of a primitive of the image.

2. The apparatus of claim 1, wherein the atomic tile is to comprise an undividable rectangle pixel block.

3. The apparatus of claim 1 or 2, wherein the logic is further to generate a super tile based on the primitive intersection cost value for each atomic tile value.

4. The apparatus of claim 3, wherein the logic is further to perform one or more vertex shading operations on the super tile.

5. The apparatus of claim 3 or 4, wherein the logic is further to perform one or more raster operations on the super tile.

6. The apparatus of any one of claims 1 to 5, wherein the logic is further to load vertex data and perform screen position calculations, wherein the vertex data is to comprise a vertex position, vertex normal, and texture coordinators.

7. The apparatus of any one of claims 1 to 6, wherein the logic is to determine the primitive intersection cost value at least partially based on the vertex element size, the vertex shader length, the number of the vertices, a weight of the vertex element size, and a weight of the vertex shader length.
8. The apparatus of any one of claims 1 to 7, wherein a processor is to comprise the logic.
9. The apparatus of any one of claims 1 to 8, wherein a processor having one or more processor cores, the logic, or memory are on a single integrated circuit die.
10. A method comprising:

determining a primitive intersection cost value for each atomic tile of an image at least partially based on a vertex element size, a vertex shader length, and a number of vertices of a primitive of the image.
11. The method of claim 10, further comprising generating a super tile based on the primitive intersection cost value for each atomic tile value.
12. The method of claim 11, further comprising performing one or more vertex shading operations on the super tile.
13. The method of claim 11 or 12, further comprising performing one or more raster operations on the super tile.

14. The method of any one of claims 10 to 13, further comprising loading vertex data and performing screen position calculations, wherein the vertex data is to comprise a vertex position, vertex normal, and texture coordinators.
15. The method of any one of claims 10 to 14, further comprising determining the primitive intersection cost value at least partially based on the vertex element size, the vertex shader length, the number of the vertices, a weight of the vertex element size, and a weight of the vertex shader length.
16. A program which makes a processor execute the method of any one of claims 10 to 15.
17. A computer-readable storage medium storing the program as recited in claim 16.
18. A system comprising:
 - a processor having one or more processor cores;
 - memory to store data, corresponding to at least one frame of a scene, to be accessed by at least one of the one or more processor cores;
 - a display device to present the at least one frame of the scene;
 - logic to determine a primitive intersection cost value for each atomic tile of the frame at least partially based on a vertex element size, a vertex shader length, and a number of vertices of a primitive of the frame.

19. The system of claim 18, wherein the atomic tile is to comprise an undividable rectangle pixel block.
20. The system of claim 18 or 19, wherein the logic is further to generate a super tile based on the primitive intersection cost value for each atomic tile value.
21. The system of claim 20, wherein the logic is further to perform one or more vertex shading operations on the super tile.
22. The system of claim 20 or 21, wherein the logic is further to perform one or more raster operations on the super tile.
23. The system of any one of claims 18 to 22, wherein the logic is further to load vertex data and perform screen position calculations, wherein the vertex data is to comprise a vertex position, vertex normal, and texture coordinators.
24. The system of any one of claims 18 to 23, wherein the logic is to determine the primitive intersection cost value at least partially based on the vertex element size, the vertex shader length, the number of the vertices, a weight of the vertex element size, and a weight of the vertex shader length.
25. The system of any one of claims 18 to 24, wherein the processor is to comprise the logic.

26. The system of any one of claim 18 to 25, wherein one or more of the one or more processor cores, the logic, or the memory are on a single integrated circuit die.