What is claimed is:

1. An apparatus comprising:

a graphics processor to generate an identification code corresponding to one or more of uniforms, vertex data, texture coordinates or graphics state for a tile in a current frame, determine whether there is a draw call that fully covered the tile, store only the identification code for the draw call that fully covers the tile ~~an image of pixels~~ for a corresponding tile in a previous frame before the tile in the current frame is to be rendered, determine whether the code for the tile in the current frame is the same as a code for the corresponding tile in the previous frame, if the code for the corresponding tile in the previous frame and the code for the tile in the current frame are the same then reuse the image from the previous frame, as pixels for the tile in the current frame to avoid rendering the tile in the current frame; and

a frame buffer coupled to said processor.

2. The apparatus of claim 1 wherein the code for the corresponding tile in the current frame includes at least two of uniforms, triangle data, texture data or state for a tile.

3. The apparatus of claim 1 wherein the code for the corresponding tile in the current frame includes at least three of uniforms, triangle data, texture data or state for a tile.

4. The apparatus of claim 1 wherein the code for the corresponding tile in the current frame includes uniforms, triangle data, texture data and state for a tile.

5. The apparatus of claim 1 wherein said graphics processor uses sort middle processing.

6. The apparatus of claim 5, said processor to generate the code for the corresponding tile in the current frame before or at ~~during~~ a natural break point in sort middle processing.

7. The apparatus of claim 1 wherein the code for the corresponding tile in the current frame is a hash.

8. The apparatus of claim 1 wherein said processor to generate the code for the corresponding tile in the current frame using a list of primitives.

9. A method comprising:

generating an identification code corresponding to one or more of uniforms, vertex data texturing coordinates or graphics state for a tile in a current frame;

determine whether there is a draw call that fully covered the tile,

storing ~~an image of pixels~~ only the identification code for the draw call that fully covers the tile for a corresponding tile in a previous frame before the tile in the current frame is to be rendered;

determining whether the code for the tile in the current frame is the same as a code for the corresponding tile in the previous frame; and

if the code for the corresponding tile in the previous frame and the code for the tile in the current frame are the same then reusing the image from the previous frame, as pixels for the tile in the current frame to avoid rendering the tile in the current frame.

10. The method of claim 9 wherein the code for the corresponding tile in the current frame includes at least two of uniforms, triangle data, texture data or state for a tile.

11. The method of claim 9 wherein the code for the corresponding tile in the current frame includes at least three of uniforms, triangle data, texture data or state for a tile.

12. The method of claim 9 wherein the code for the corresponding tile in the current frame includes uniforms, triangle data, texture data and state for a tile.

13. The method of claim 9 including using sort middle processing.

14. The method of claim 13 including generating the code for the corresponding tile in the current frame before or at ~~during~~ a natural break point in sort middle processing.

15. The method of claim 9 wherein the code for the corresponding tile in the current frame is a hash.

16. The method of claim 9 wherein said processor to generate the code using a list of primitives.

17. An apparatus comprising:

means for generating an identification code corresponding to one or more of uniforms, vertex data, texture coordinates or graphics state for a tile in a current frame;

determine whether there is a draw call that fully covered the tile,

means for storing ~~a corresponding image of pixels~~ only the identification code for the draw call that fully covers the tile for a tile in a previous frame before the tile in the current frame is to be rendered;

means for determining whether the code for the tile in the current frame is the same as a code for the corresponding tile in the previous frame; and

means for reusing the image from the previous frame, as pixels for the tile in the current frame to avoid rendering the tile in the current frame if the code for the corresponding tile in the previous frame and the code for the tile in the current frame are the same.

18. The apparatus of claim 17 wherein the code for the corresponding tile in the current frame includes at least two of uniforms, triangle data, texture data or state for a tile.

19. The apparatus of claim 17 wherein the code for the corresponding tile in the current frame includes at least three of uniforms, triangle data, texture data or state for a tile.

20. The apparatus of claim 17 wherein the code for the corresponding tile in the current frame includes uniforms, triangle data, texture data and state for a tile.

21. The apparatus of claim 17 wherein said generating means uses sort middle processing.

22. The apparatus of claim 21 including means for generating the code for the corresponding tile in the current frame before or ~~during~~at a natural break point in sort middle processing.

23. The apparatus of claim 17 wherein the code for the corresponding tile in the current frame is a hash.

24. The apparatus of claim 17 wherein said generating means includes means for generating the code using a list of primitives.

25. A system comprising:

a graphics processor to generate an identification code corresponding to one or more of uniforms, vertex data, texture coordinates or graphics state for a tile in a current frame, determine whether there is a draw call that fully covered the tile, store ~~an image of pixels~~ only the identification code for a draw call that fully covers the tile for a corresponding tile in a previous frame before the tile in the current frame is to be rendered, determine whether the code for the tile in the current frame is the same as a code for the corresponding tile in the previous frame, if the code for the corresponding tile in the previous frame and the code for the tile in the current frame are the same then reuse the image from the previous frame, as pixels for the tile in the current frame to avoid rendering the tile in the current frame; and

a display coupled to said processor.

26. The system of claim 25 wherein the code for the corresponding tile in the current frame includes at least two of uniforms, triangle data, texture data or state for a tile.

27. The system of claim 25 wherein the code for the corresponding tile in the current frame includes at least three of uniforms, triangle data, texture data or state for a tile.

28. The system of claim 25 wherein the code for the corresponding tile in the current frame includes uniforms, triangle data, texture data and state for a tile.

29. The system of claim 25 wherein said graphics processor to use sort middle processing.

30. The system of claim 25, said processor to generate the code for the corresponding tile in the current frame before or at~~during~~ a natural break point in sort middle processing.

31. The system of claim 25 wherein the code for the corresponding tile in the current frame is a hash.

32. The system of claim 25 wherein said processor to generate the code for the corresponding tile in the current frame using a list of primitives.

33. A system comprising:

means for generating an identification code corresponding to one or more of uniforms, vertex data, texture coordinates or graphics state for a tile in a current frame;

determine whether there is a draw call that fully covered the tile,

means for storing ~~a corresponding image of pixels~~ only the identification code for the draw call that fully covers the tile for a tile in a previous frame before the tile in the current frame is to be rendered;

means for determining whether the code for the tile in the current frame is the same as a code for the corresponding tile in the previous frame;

means for reusing the image from the previous frame, as pixels for the tile in the current frame to avoid rendering the tile in the current frame, if the code for the corresponding tile in the previous frame and the code for the tile in the current frame are the same; and

means for displaying images.

34. The system of claim 33 wherein the code for the corresponding tile in the current frame includes at least two of uniforms, triangle data, texture data or state for a tile.

35. The system of claim 33 wherein the code for the corresponding tile in the current frame includes at least three of uniforms, triangle data, texture data or state for a tile.

36. The system of claim 33 wherein the code for the corresponding tile in the current frame includes uniforms, triangle data, texture data and state for a tile.

37. The system of claim 33 wherein said generating means uses sort middle processing.

38. The system of claim 37 including means for generating the code for the corresponding tile in the current frame before or at~~during~~ a natural break point in sort middle processing.

39. The system of claim 33 wherein the code for the corresponding tile in the current frame is a hash.

40. The apparatus of claim 33 wherein said generating means includes means for generating the code for the corresponding tile in the current frame using a list of primitives.