What is claimed is:

1. A method comprising: accumulating a reduced representation of geometry, input data, and state for a tile when a primitive list is created; before per tile rasterization, determining whether the reduced representation for the current tile in the current frame is the same as the reduced representation of a corresponding tile in the previous frame by testing if there is a single primitive that overlays the tile completely and whether all other primitives in the primitive list are occluded by that primitive in the primitive list; and if so, then reusing pixels from the tile in the previous frame as pixels for the tile in the current frame when rendering the current tile.

2. The method of claim 1 including using a hash function to produce the reduced representation.

3. The method of claim 1 including using a bloom filter to produce the reduced representation.

4. The method of claim 3 including using two bloom filters, one for the current tile and one for a previous tile.

5. The method of claim 4 including using one pair of bloom filters for each of geometry, texture addresses, and shaders.

6. The method of claim 4 wherein after processing the current frame, using the bloom filter for the current as the bloom filter for the prior frame and obtaining and using the bloom filter for the previous frame as a bloom filter for the next frame.

7. The method of claim 1 including accumulating in a sort-middle architecture.

8. The method of claim 1 wherein reusing includes reusing the color values.

9. The method of claim 1 wherein reusing includes reusing depth or stencil buffers.

10. The method of claim 1 including refraining from sending data for display when the compact representations are the same.

11. The method of claim 1 including changing the current compact representation to the previous compact representation.

12. The method of claim 1 including a draw call associated with a tile in a first frame, a draw call associated with a corresponding tile in a second frame, and determining if the same draw call is used for the same tile in successive frames to determine if the reduced representations are the same.

13. One or more non-transitory computer readable media storing instructions to implement a sequence comprising: accumulating a reduced representation of geometry, input data, and state for a tile when a triangle list is created; before per tile rasterization, determining whether the reduced representation for the current tile in the current frame is the same as the reduced representation of a corresponding tile in the previous frame by testing if there is a single primitive that overlays the tile completely and whether all other primitives in the primitive list are occluded by that primitive in the primitive list; and if so, then reusing pixels from the tile in the previous frame as pixels for the tile in the current frame when rendering the current tile.

14. The media of claim 13, said sequence including using a hash function to produce the reduced representation.

15. The media of claim 13, said sequence including using a bloom filter to produce the reduced representation.

16. The media of claim 15, said sequence including using two bloom filters, one for the current tile and one for a previous tile.

17. The media of claim 16, said sequence including using one pair of bloom filters for each of geometry, texture addresses, and shaders.

18. The media of claim 13, said sequence including accumulating in a sort-middle architecture.

19. The media of claim 13 wherein reusing includes reusing the color values.

20. The media of claim 13 wherein reusing includes reusing depth or stencil buffers.

21. The media of claim 13, including a draw call associated with a tile in a first frame, a draw call associated with a corresponding tile in a second frame, and said sequence including determining if the same draw call is used for the same tile in successive frames to determine if the reduced representations are the same.

22. The media of claim 13, said sequence including wherein after processing the current frame, using the bloom filter for the current as the bloom filter for the prior frame and obtaining and using the bloom filter for the previous frame as a bloom filter for the next frame.

23. An apparatus comprising: a processor to accumulate a reduced representation of geometry, input data, and state for a tile when a triangle list is created, before per tile rasterization, determine whether the reduced representation for the current tile in the current frame is the same as the reduced representation of a corresponding tile in the previous frame by testing if there is a single primitive that overlays the tile completely and whether all other primitives in the primitive list are occluded by that primitive in the primitive list, and if so, then reuse pixels from the tile in the previous frame as pixels for the tile in the current frame when rendering the current tile; and a storage coupled to said processor.

24. The apparatus of claim 23, said processor to use a hash function to produce the reduced representation.

25. The apparatus of claim 23, including a bloom filter to produce the reduced representation.

26. The apparatus of claim 25 including two bloom filters, one for the current tile and one for a previous tile.

27. The apparatus of claim 26 including one pair of bloom filters for each of geometry, texture addresses, and shaders.

28. The apparatus of claim 23, said processor to accumulate in a sort-middle architecture.

29. The apparatus of claim 23, said processor to refrain from sending data for display when the compact representations are the same.

30. The apparatus of claim 23, including a draw call associated with a tile in a first frame, a draw call associated with a corresponding tile in a second frame, said processor to determine if the same draw call is used for the same tile in successive frames to determine if the reduced representations are the same.

31. The apparatus of claim 23 including a battery.

32. The apparatus of claim 23, said processor to process the current frame, use the bloom filter for the current as the bloom filter for the prior frame and obtain and use the bloom filter for the previous frame as a bloom filter for the next frame.