What is claimed is:

1. A method comprising:

determining, using a hardware processor, Zmin, Zmax, residuals and Zmin/Zmax-index selection mask for depth offset compression; and

using bit combinations that are unused for depth offset compression to improve compression.

2. The method of claim 1 identifying and reusing redundant bit combinations that arise from overlapping Zmin and Zmax ranges.

3. The method of claim 1 including determining whether the Zmin/Zmax-pair constitutes a redundant combination or is an invalid reference value pair for depth offset compression, and if so, using a different depth offset compression scheme, or a different compression algorithm altogether with the remaining and reclaimed bits.

4. The method of claim 3 including reusing the bits otherwise allocated for an index mask to extend residuals with one bit per sample.

5. The method of claim 1 including determining whether Zmin is greater than Zmax and, if so, swapping Zmin and Zmax for an extra bit to improve precision of Zmin or Zmax.

6. The method of claim 1 including, if Zmin is greater than Zmax, then using the next WxH bits as a clear mask to indicate whether the pixel is cleared or not, where W is a tile's width and H is a tile's height.

7. The method of claim 6, including storing depths in uncompressed form depending on how many non-cleared depths exist.

8. The method of claim 1 including detecting whether an index mask is all ones or zeros, and, if so, using unused bits of Zmin or Zmax to improve precision of the reference value is actually used, or redistributing the extra bits among the residuals.

9. The method of claim 1 including determining more than two depth values per tile.

10. The method of claim 9 including breaking a tile into subtiles and assigning one or more reference depth values each of the said subtiles, and from which residuals to the actual depth values are calculated and stored.

11. The method of claim 1 including using the same technique for color data.

12. One or more non-transitory computer readable media storing instructions executed by a processor to perform a sequence comprising:

determining, using a hardware processor, Zmin and Zmax for depth offset compression;

determining residual bits per depth with respect to at least one of Zmin or Zmax that are unused for depth offset compression; and

using bits that are unused for depth offset compression to improve compression.

13. The media of claim 12, said sequence including identifying and using redundant bit combinations that arise from overlapping Zmin and Zmax ranges.

14. The media of claim 12, said sequence including determining whether Zmin is equal to Zmax, and if so, using a different depth offset compression scheme, or a different compression algorithm altogether with the remaining bits.

15. The media of claim 14, said sequence including reusing the bits otherwise allocated for an index mask to extend residuals with one bit per sample.

16. The media of claim 12, said sequence including determining whether Zmin is greater than Zmax and, if so, swapping Zmin and Zmax for an extra bit to improve precision of Zmin and Zmax.

17. The media of claim 12, said sequence including, if Zmin is greater than Zmax, then using the next WxH bits as a clear mask to indicate whether the pixel is cleared or not, where W is a tile's width and H is a tile's height.

18. The media of claim 17, said sequence including storing depths in uncompressed form depending on how many non-cleared depths exist.

19. The media of claim 12, said sequence including detecting whether an index mask is all ones or zeros, and, if so, using unused bits of Zmin or Zmax to improve precision of reference values that are actually used, or redistributing the extra bits among the residuals.

20. The media of claim 12, said sequence including determining more than two depth values per tile.

21. The media of claim 20, said sequence including breaking a tile into subtiles and assigning depth values to said subtiles, and selecting at least one subtile's depth value as a depth value of the tile.

22. The media of claim 12 including using the same technique for color data.

23. An apparatus comprising:

determine, using a hardware processor, Zmin and Zmax for depth offset compression, determine residual bits per depth with respect to at least one of Zmin or Zmax that are unused for depth offset compression, and use bits that are unused for depth offset compression to improve compression; and

a storage coupled to said processor.

24. The apparatus of claim 23, said processor to identify and use redundant bit combinations that arise from overlapping Zmin and Zmax ranges.

25. The apparatus of claim 23, said processor to determine whether Zmin is equal to Zmax, and if so, use a different depth offset compression scheme, or a different compression algorithm altogether with the remaining bits.

What is claimed is:

1. A method comprising:

determining, using a hardware processor, Zmin, Zmax, residuals and Zmin/Zmax-index selection mask for depth offset compression; and

using bit combinations that are unused for depth offset compression to improve compression.

2. The method of claim 1 identifying and reusing redundant bit combinations that arise from overlapping Zmin and Zmax ranges.

3. The method of claim 1 including determining whether the Zmin/Zmax-pair constitutes a redundant combination or is an invalid reference value pair for depth offset compression, and if so, using a different depth offset compression scheme, or a different compression algorithm altogether with the remaining and reclaimed bits.

4. The method of claim 3 including reusing the bits otherwise allocated for an index mask to extend residuals with one bit per sample.

5. The method of claim 1 including determining whether Zmin is greater than Zmax and, if so, swapping Zmin and Zmax for an extra bit to improve precision of Zmin or Zmax.

6. The method of claim 1 including, if Zmin is greater than Zmax, then using the next WxH bits as a clear mask to indicate whether the pixel is cleared or not, where W is a tile's width and H is a tile's height.

7. The method of claim 6, including storing depths in uncompressed form depending on how many non-cleared depths exist.

8. The method of claim 1 including detecting whether an index mask is all ones or zeros, and, if so, using unused bits of Zmin or Zmax to improve precision of the reference value is actually used, or redistributing the extra bits among the residuals.

9. The method of claim 1 including determining more than two depth values per tile.

10. The method of claim 9 including breaking a tile into subtiles and assigning one or more reference depth values each of the said subtiles, and from which residuals to the actual depth values are calculated and stored.

11. The method of claim 1 including using the same technique for color data.

12. One or more non-transitory computer readable media storing instructions executed by a processor to perform a sequence comprising:

determining, using a hardware processor, Zmin and Zmax for depth offset compression;

determining residual bits per depth with respect to at least one of Zmin or Zmax that are unused for depth offset compression; and

using bits that are unused for depth offset compression to improve compression.

13. The media of claim 12, said sequence including identifying and using redundant bit combinations that arise from overlapping Zmin and Zmax ranges.

14. The media of claim 12, said sequence including determining whether Zmin is equal to Zmax, and if so, using a different depth offset compression scheme, or a different compression algorithm altogether with the remaining bits.

15. The media of claim 14, said sequence including reusing the bits otherwise allocated for an index mask to extend residuals with one bit per sample.

16. The media of claim 12, said sequence including determining whether Zmin is greater than Zmax and, if so, swapping Zmin and Zmax for an extra bit to improve precision of Zmin and Zmax.

17. The media of claim 12, said sequence including, if Zmin is greater than Zmax, then using the next WxH bits as a clear mask to indicate whether the pixel is cleared or not, where W is a tile's width and H is a tile's height.

18. The media of claim 17, said sequence including storing depths in uncompressed form depending on how many non-cleared depths exist.

19. The media of claim 12, said sequence including detecting whether an index mask is all ones or zeros, and, if so, using unused bits of Zmin or Zmax to improve precision of reference values that are actually used, or redistributing the extra bits among the residuals.

20. The media of claim 12, said sequence including determining more than two depth values per tile.

21. The media of claim 20, said sequence including breaking a tile into subtiles and assigning depth values to said subtiles, and selecting at least one subtile's depth value as a depth value of the tile.

22. The media of claim 12 including using the same technique for color data.

23. An apparatus comprising:

determine, using a hardware processor, Zmin and Zmax for depth offset compression, determine residual bits per depth with respect to at least one of Zmin or Zmax that are unused for depth offset compression, and use bits that are unused for depth offset compression to improve compression; and

a storage coupled to said processor.

24. The apparatus of claim 23, said processor to identify and use redundant bit combinations that arise from overlapping Zmin and Zmax ranges.

25. The apparatus of claim 23, said processor to determine whether Zmin is equal to Zmax, and if so, use a different depth offset compression scheme, or a different compression algorithm altogether with the remaining bits. .

26. The apparatus of claim 23, said processor to reuse the bits otherwise allocated for an index mask to extend residuals with one bit per sample

27. The apparatus of claim 23, said processor to determine whether Zmin is greater than Zmax and, if so, swapping Zmin and Zmax for an extra bit to improve precision of Zmin and Zmax..

28. The apparatus of claim 23, said processor to, if Zmin is greater than Zmax, then using the next WxH bits as a clear mask to indicate whether the pixel is cleared or not, where W is a tile's width and H is a tile's height.

29. The apparatus of claim 23, said processor to store depths in uncompressed form depending on how many non-cleared depths exist.

30. The apparatus of claim 23, said processor to detect whether an index mask is all ones or zeros, and, if so, use unused bits of Zmin or Zmax to improve precision of reference values that are actually used, or redistribute the extra bits among the residuals.