Unlike DFT, the DCT produces only real coefficients, while the DFT produces complex coefficients. Working with complex numbers makes both the computation and the quantization process more complex- making DCT more efficient for image compression.

Additionally, the DCT is even and symmetric along the y-axis, minimizing discontinuities compared to the DFT, which does not. The reduced discontinuities result in more of the information(energy) being stored in the low-frequency coefficients- meaning the higher-frequency coefficients have less impact. And so, we could remove those higher-frequency components without losing much of the information.

The amount of intensity levels remains the same, and the probability of each intensity level remains the same(after mapping). The intensity level itself does not matter, only its probability and the amount of intensity levels. Therefore, the histogram equalization would have no effect on the following Huffman encoding.

LZW’s advantage lies in reducing spatial redundancy, while Huffman encoding relies on encoding each symbol by itself, based on probability.  
Huffman requires two passages on the data- one for the probability tree and one for the actual data, while LZW requires only a single passage on the data. Moreover, LZW doesn’t need any previous data about the probabilities of symbols before encoding.  
Additionally, Huffman requires transmitting the tree as well(if it wasn’t agreed upon by the two parties), unlike LZW.

Golomb’s advantage becomes apparent when smaller values occur very often, with the probabilities decreasing exponentially as values increase.  
In Golomb coding, smaller values are assigned shorter codes.  
Therefore, when the data follows a geometric distribution(meaning smaller values are much more frequent than larger values, and the probability decreases exponentially as the value increases) Golomb coding would be better than Huffman coding.