Quantization is the mapping of analog signals into digital signals, usually having a high bit rate. Data compression in the general sense is the mapping of analog or high rate digital data into relatively low rate digital data. Compression systems often consist of two steps, quantization (which can expand bandwidth) of an analog signal followed by purely digital compression to reduce the number of bits per second. The encoding of analog or high rate digital information into a low rate digital sequence is useful because:

- Analog information must be converted into digital form in order to use digital communications, storage, and signal processing. Compressed data can yield better quality reproductions for a given digital communications or digital storage capacity. Equivalently, a given quality reproduction can be achieved with less capacity if the signals have been compressed.
- If the original data can be compressed without losing essential information or features, subsequent digital signal processing can be significantly less complex. For example, enhancement, inverse transforms, and simple classifiers can be incorporated into the compression algorithms or performed separately by table lookup.

The cost of compression is the complexity required by the compression algorithm. As one might expect, greater compression often (but not always) comes at the expense of increased implementation complexity. Whether the performance improvement merits the increased cost usually depends on the application.

The goal of this talk is to survey the menagerie of quantization and compression algorithms in the specific context of image compression and to provide some general comparisons based on performance, complexity, and side benefits of particular coding techniques. The author’s bias for vector quantization and classification tree design techniques is admitted at the outset.

The following is a list of topics that will be touched upon. Lack of time will not permit much detail on any topic, but I hope to at least raise the issues, list a variety of approaches, make some general comparisons, and point out some useful attributes.

- Why compress?
- Lossless vs. lossy compression,
- Scalar quantization.
Compression based on scalar quantization: Predictive codes and transform codes. The problem of bit allocation.
Quantization and entropy coding.

- Vector quantization
  - Design techniques, including the Lloyd clustering algorithm.
  - Complexity and other problems of simple vector quantization.

- Constrained vector quantization
  - A list of variations.
  - Three favorites: Classified VQ, Predictive VQ, Tree structured VQ.
  - Tree structured codes and progressive transmission ("Quick Look").

- Variable-rate vector quantizers

- Decision trees and data compression: Growing and pruning trees.

- Closing thoughts and suggestions of where to find out more.