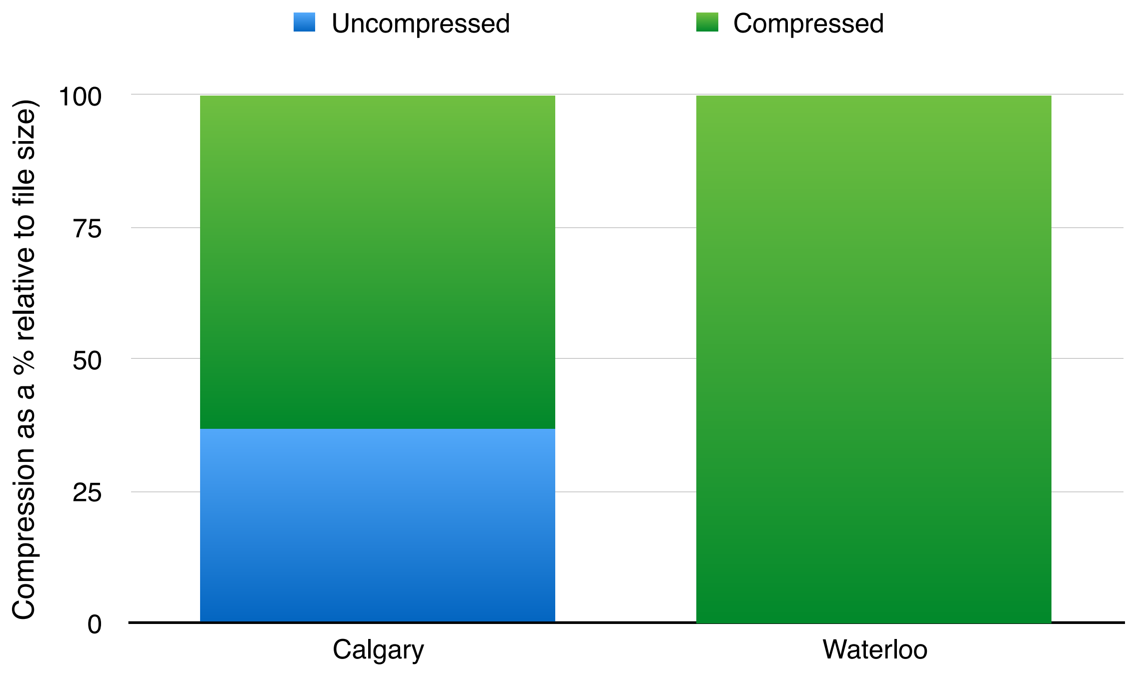
Statistical and Empirical Analysis of Huffman

Which compresses more, binary files or text files?

HuffMark with Calgary Corpus and Waterloo TIFs

	Calgary (Text)	Waterloo (Images)
Data read (bytes)	3,251,493	12,466,304
Data compressed (bytes)	1,196,008	8,286
Compression (in %)	63.21	99.934
Compression time (s)	2.545	0.757



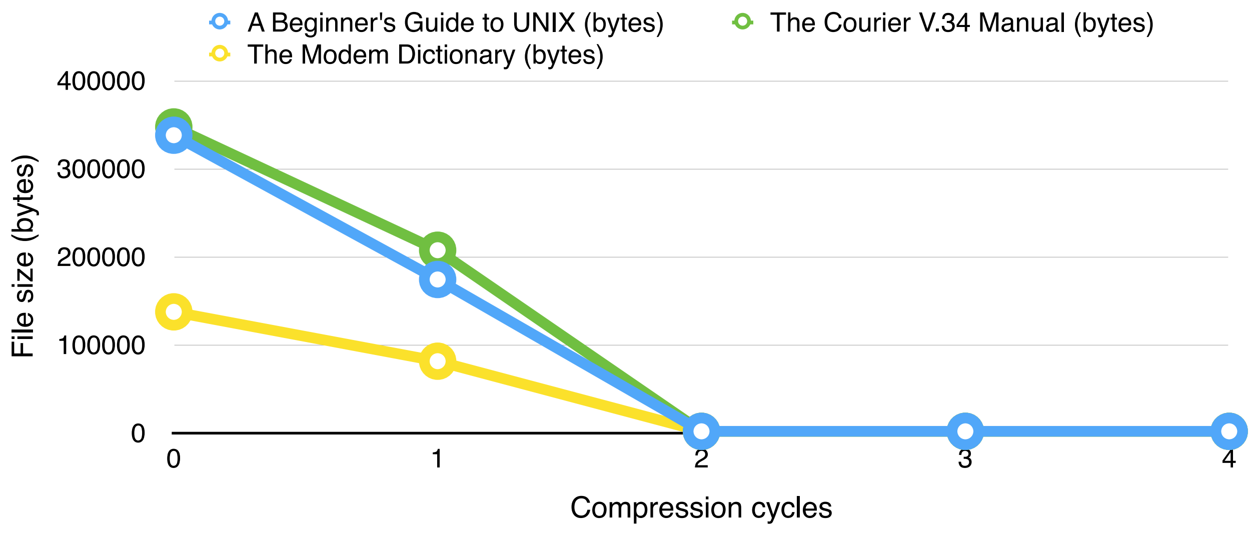
This test was surprising as I had guessed — without performing any tests — that the Calgary corpus would be more compressible. I thought that the structure of the text files would dovetail nicely with the compression algorithm used.

As it turns out, the images were very compressible, by a wide margin. If you examined the files, you would see that they are often a composition of several patterns repeated. There are also a few dominant colors in each image. Such characteristics allow for great savings in files size.

Can you gain additional compression by double-compressing an already compressed file? If so, is there eventually a limit to when this no longer saves space on ordinary files?

Compressed file sizes after varying compression cycles

Number of compression cycles	A Beginner's Guide to UNIX (bytes)	The Courier V.34 Manual (bytes)	The Modem Dictionary (bytes)
0	337,256	346,665	136,696
1	173,461	206,691	80,670
2	1,036	1,036	1,036
3	1,036	1,036	1,036
4	1,036	1,036	1,036



This experiment was interesting as I suspect we are encountering factors beyond the scope of the application. We witness a general halving of file sizes, in the first run, whereby subsequent compression cycles result in a file size of 1,036 bytes.

A possibility is that file sizes, are theoretically less than 1,036 bytes and each compression cycle sees some reduction in file size up to a certain point. Why we see a file size floor of 1,036 bytes is likely a product of the file system. Running on a HFS+ Journaled file system, the block size of a HFS catalog file is 512 bytes. If the file were slightly greater than 512 bytes, it would be allocated another 512 byte chunk, which corresponds roughly to the 1KB file size that all three documents approach. Therefore, compressing the file after such a file size is hit would not be useful.