On the Use of Stronger Synchronization to Boost Compression by Substring Enumeration

Danny Dubé (Danny.Dube@ift.ulaval.ca)
Université Laval, Canada

A new lossless data compression technique called compression by substring enumeration (CSE) has recently been introduced. CSE is competitive but it achieves lower performance on non-text-like data. More recent work confirmed that CSE incurs a penalty due to the fact that it is unaware of the position (or *phase*) of the bits relative to the byte boundaries [1]. That work demonstrated that CSE can be boosted by adding a preprocessing step in which synchronization bits are inserted in the data. Various synchronization schemes were used and, in general, it has been observed that the more we insert bits, the more we improve the compression, with the best results obtained using a *reliable* scheme that inserts 5 bits per byte ($n = 13$). In this work, we measure the boost when using even stronger schemes. The results are negative: the use of stronger schemes ($n < 13$) brings only minimal improvements.

Our synchronization schemes have a fixed shape. A k-bit scheme maps $b_1\, b_2\, \ldots\, b_8$ into $w_1\, b_1\, w_2\, b_2\, \ldots\, w_8\, b_8\, w_9$, where $k = |w_1\, \ldots\, w_9|$, $b_i \in \{0, 1\}$, and $w_i \in \{0, 1\}^*$.

A k-bit scheme is *n-reliable* if, by starting at an arbitrary position in mapped data, the phase can be identified by looking at the n next bits. A scheme is *reliable* if it is $(k + 8)$-reliable. A smaller n means stronger synchronization.

n	k	Synchronization Scheme
—	0	_ _ _ _ _ _ _ _
—	1	_ _ _ _ _ _ _ _ 0
—	2	_ _ _ _ _ _ _ _ 0 1
—	3	_ _ _ _ _ _ _ _ 0 1 1
—	4	_ _ _ _ _ _ _ _ 0 1 1 1
13	5	_ _ _ _ _ _ 0 _ _ 0 1 1 1

n	k	Synchronization Scheme
12	8	_ _ _ 0 _ _ 1 0 0 _ _ 1 _ 1 1 0
11	8	_ _ _ 0 _ 0 _ _ 1 1 0 _ _ 1 1 0
10	10	_ _ _ _ 0 _ 0 1 1 _ _ _ 0 1 0 0 1 1
9	10	_ _ _ _ 0 0 0 1 1 _ _ _ _ 0 1 0 1 1
8	15	_ _ _ 0 0 0 1 0 _ _ _ 1 1 1 0 1 _ _ 1 1 0 0 1
7	20	1 1 0 1 _ 1 _ 1 1 0 0 _ 0 _ 0 1 0 0 _ 0 _ 1 1 0 0 _ 1 _

Bits/Car.	BWT	PPM	Anti	k=0	k=1	k=2	k=3	k=4	n=13	n=12	n=11	n=10	n=9	n=8	n=7
bib	2.07	1.91	2.56	1.98	1.95	1.92	1.92	1.91	1.90	1.89	1.89	1.89	1.89	1.88	1.88
book1	2.49	2.40	3.08	2.27	2.26	2.25	2.25	2.25	2.25	2.25	2.25	2.25	2.25	2.29	2.33
book2	2.13	2.02	2.81	1.98	1.96	1.95	1.95	1.94	1.94	1.93	1.93	1.93	1.93	1.93	1.95
geo	4.45	4.83	6.22	5.35	5.21	4.98	4.81	4.70	4.63	4.58	4.59	4.58	4.58	4.58	4.57
news	2.59	2.42	3.42	2.52	2.49	2.46	2.45	2.44	2.43	2.43	2.43	2.43	2.43	2.42	2.42
obj1	3.98	4.00	4.87	4.46	4.53	4.43	4.32	4.24	4.17	4.03	4.05	4.02	4.01	4.00	3.99
obj2	2.64	2.43	3.61	2.71	2.69	2.59	2.53	2.49	2.47	2.45	2.46	2.45	2.45	2.45	2.44
paper1	2.55	2.37	3.17	2.54	2.51	2.48	2.47	2.46	2.44	2.41	2.41	2.41	2.41	2.41	2.41
paper2	2.51	2.36	3.14	2.41	2.39	2.38	2.38	2.37	2.36	2.35	2.35	2.34	2.35	2.34	2.34
paper3	—	—	—	2.73	2.70	2.69	2.68	2.67	2.65	2.63	2.63	2.63	2.63	2.63	2.63
paper4	—	—	—	3.20	3.16	3.13	3.13	3.10	3.07	3.02	3.02	3.02	3.02	3.01	3.01
paper5	—	—	—	3.33	3.29	3.27	3.24	3.22	3.19	3.12	3.13	3.12	3.12	3.11	3.10
paper6	—	—	—	2.65	2.61	2.58	2.56	2.55	2.52	2.50	2.50	2.49	2.50	2.49	2.49
pic	0.83	0.85	1.09	0.77	0.84	0.82	0.82	0.82	0.81	0.81	0.81	0.81	0.81	0.81	0.81
progc	2.58	2.40	3.18	2.60	2.58	2.54	2.52	2.50	2.48	2.44	2.44	2.44	2.44	2.44	2.43
progl	1.80	1.67	2.24	1.71	1.70	1.69	1.68	1.67	1.66	1.65	1.65	1.65	1.65	1.64	1.64
progp	1.79	1.62	2.27	1.78	1.76	1.73	1.71	1.70	1.68	1.66	1.66	1.66	1.66	1.66	1.65
trans	1.57	1.45	1.94	1.60	1.58	1.53	1.52	1.50	1.48	1.47	1.47	1.47	1.47	1.47	1.46

[1] Danny Dubé. Using synchronization bits to boost compression by substring enumeration. In *Proceedings of ISITA*, Taichung, Taiwan, October 2010.