中国科学技术大学
University of Science and Technology of China

Compressing the Tree of Canonical Huffman Coding

Qi Cheng[1,2] Wei Yan[1,2] Sian-Jheng Lin[3] Nenghai Yu[1,2]

[1]University of Science and Technology of China

[2]CAS Key Laboratory of Electromagnetic Space Information

[3]Theory Lab, Cenrtal Research Institute , 2012 Labs, Huawei Technology Co.Ltd

CONTENTS

1 Backgrounds

2 The proposed method

3 Experimental results

4 Conclusions

Backgrounds

The introduction of two conventional methods for compressing the tree of canonical Huffman coding

two conventional methods

encode the number of leaf nodes in each layer

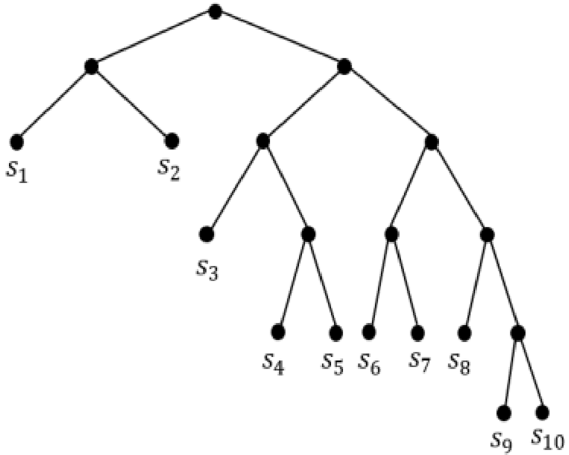
The first method need to encode the string of number $(l_{max}, n_1, \cdots, n_{l_{max}})$.
Use an integer with $\lceil log_{2(N-1)} \rceil$ bits to encode l_{max}.
Then use $\mathrm{L} = \lceil log_{2(N+1)} \rceil$ bits to encode each n_i for $1 \leq i \leq l_{max}$.
Thus, the total bits L_1: $L_1 = \lceil log_{2(N-1)} \rceil + l_{max} \lceil log_{2(N+1)} \rceil$.

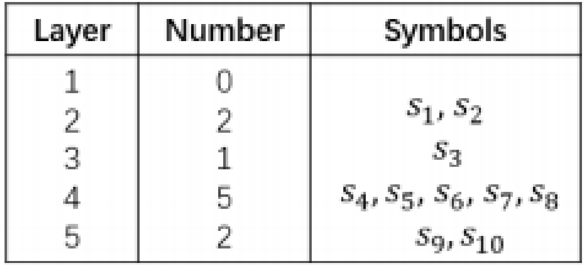
Encodes n_i from the l_{min}-th layer instead of the first layer.
Then use $\mathrm{M} = \lceil log_{2(n_{max}+1)} \rceil$ bits to encode each n_i for $l_{min} \leq i \leq l_{max}$.
The second method need to encode the string of number $(l_{min}, l_{max}, M, n_{l_{min}}, \cdots, n_{l_{max}})$.
Thus, the total bits L_2:

$$L_2 = 2\lceil log_{2(N-1)} \rceil + \lceil log_{2(\lceil log_{2(N+1)} \rceil - 1)} \rceil + (l_{max} - l_{min})\lceil log_{2(N+1)} \rceil.$$

Backgrounds

- an example





Layer	Number	Symbols
1	0	
2	2	s_1, s_2
3	1	s_3
4	5	s_4, s_5, s_6, s_7, s_8
5	2	s_9, s_{10}

- The first method need total bits L_1:

$$L_1 = \lceil log_{2(N-1)} \rceil + l_{max} \lceil log_{2(N+1)} \rceil = 24.$$

- The second method need total bits L_2:

$$L_2 = 2\lceil log_{2(N-1)} \rceil + \lceil log_{2(\lceil log_{2(N+1)} \rceil - 1)} \rceil + (l_{max} - l_{min}) \lceil log_{2(N+1)} \rceil = 22.$$

Motivation: how to compress the canonical Huffman coding tree more effective?

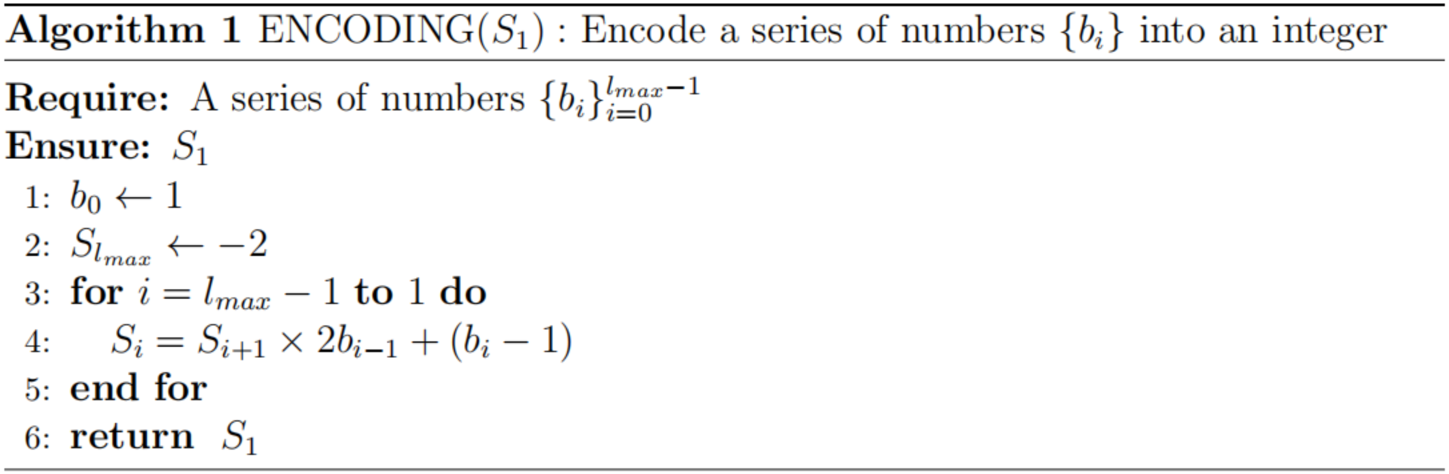
Proposed method

The new method for compressing the tree of canonical Huffman coding

idea: store the number of internal nodes $\{b_i\}$ instead of the leaves $\{n_i\}$

encode method: the mixed radix conversion

the algorithm :



Algorithm 1 ENCODING(S_1) : Encode a series of numbers $\{b_i\}$ into an integer

Require: A series of numbers $\{b_i\}_{i=0}^{l_{max}-1}$

Ensure: S_1

```
$b_0 \leftarrow 1$
$S_{l_{max}} \leftarrow -2$
for $i = l_{max} - 1$ to 1 do
    $S_i = S_{i+1} \times 2b_{i-1} + (b_i - 1)$
end for
return $S_1$
```

Proposed method

The proposed method need total bits L_3:

$$L_3 = \lceil log_{2^{|s_1|}} \rceil + \boxed{\lceil log_{2^{(N-1)}} \rceil}.$$

to encode the number of bits for encoding S_1

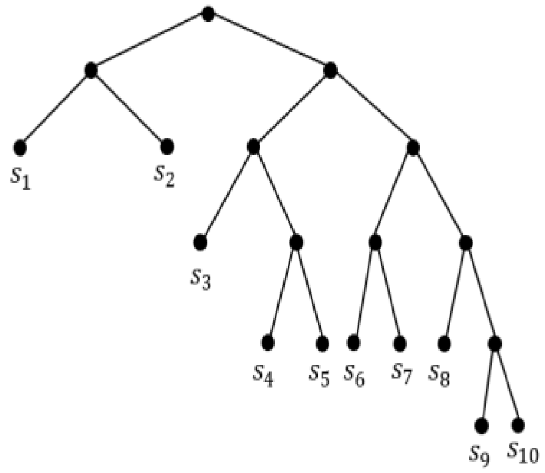
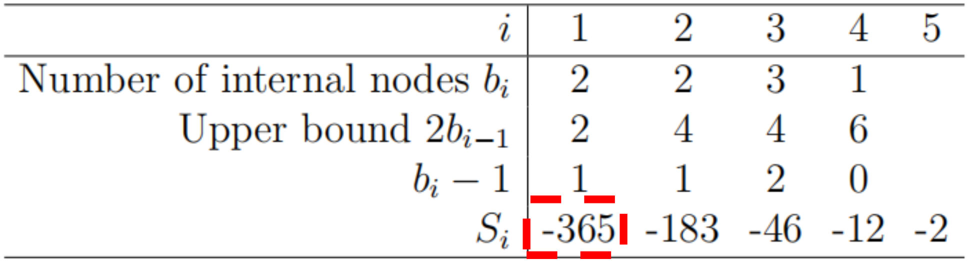


Table 1: The encoding of the string of numbers $(2, 2, 3, 1)$.



i	1	2	3	4	5
Number of internal nodes b_i	2	2	3	1	
Upper bound $2b_{i-1}$	2	4	4	6	
$b_i - 1$	1	1	2	0	
S_i	$\boxed{-365}$	-183	-46	-12	-2

$$L_3 = \lceil log_{2^{|-365|}} \rceil + \lceil log_{2^{(10-1)}} \rceil = 13$$

Experimental results

The comparison of the compression ratio between the proposed method and two conventional methods.

Experimental data: Calgary corpus

We use a conventional way to construct the canonical Huffman tree for various data sets.

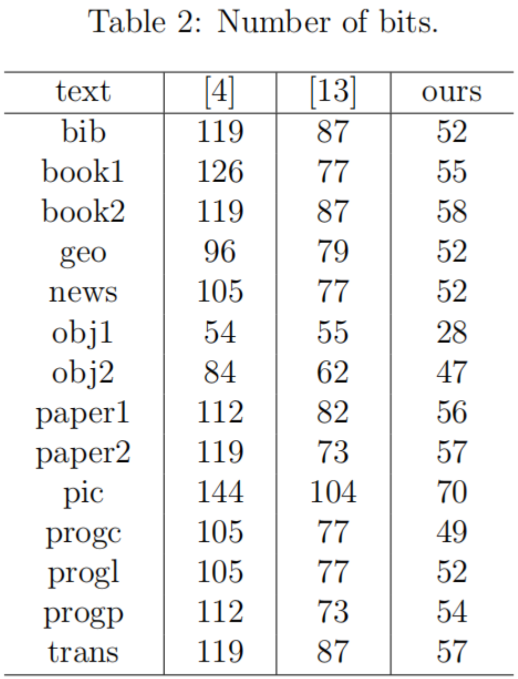


Table 2: Number of bits.

text	[4]	[13]	ours
bib	119	87	52
book1	126	77	55
book2	119	87	58
geo	96	79	52
news	105	77	52
obj1	54	55	28
obj2	84	62	47
paper1	112	82	56
paper2	119	73	57
pic	144	104	70
progc	105	77	49
progl	105	77	52
progp	112	73	54
trans	119	87	57

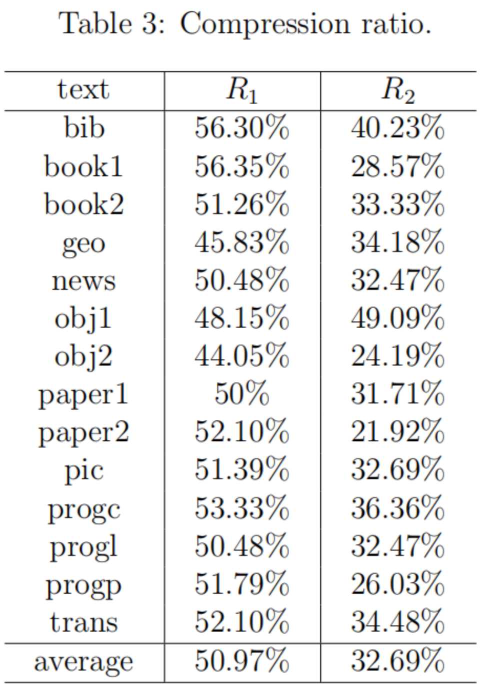
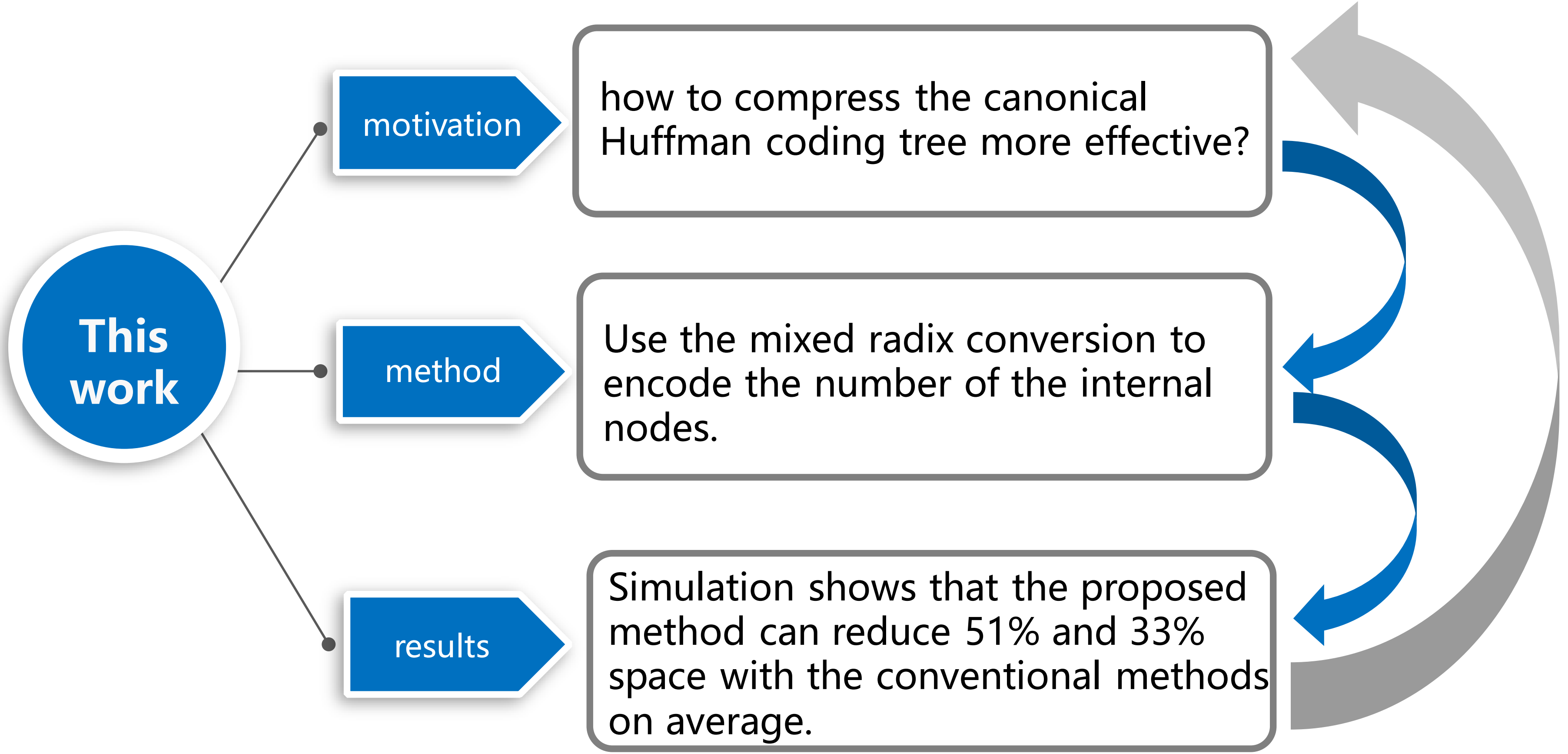


Table 3: Compression ratio.

text	R_1	R_2
bib	56.30%	40.23%
book1	56.35%	28.57%
book2	51.26%	33.33%
geo	45.83%	34.18%
news	50.48%	32.47%
obj1	48.15%	49.09%
obj2	44.05%	24.19%
paper1	50%	31.71%
paper2	52.10%	21.92%
pic	51.39%	32.69%
progc	53.33%	36.36%
progl	50.48%	32.47%
progp	51.79%	26.03%
trans	52.10%	34.48%
average	50.97%	32.69%



This work

motivation

how to compress the canonical Huffman coding tree more effective?

method

Use the mixed radix conversion to encode the number of the internal nodes.

results

Simulation shows that the proposed method can reduce 51% and 33% space with the conventional methods on average.

Data Compression Conference

中国科学技术大学
University of Science and Technology of China

Thank you !

For more details, please refer to our paper:
Compressing the Tree of Canonical Huffman Coding

That is all my presentation

Reporter：Qi Cheng