LOCO-ANS: An Optimization of JPEG-LS Using an Efficient and Low-Complexity Coder Based on ANS

Full Paper

Tobías Alonso (Ph.D. supervisors: Gustavo Sutter, and Jorge E. López de Vergara)
High Performance Computing and Networking Reasearch Group. Universidad Autónoma de Madrid, Spain

Motivation

- Lossless image codecs:
  Highly valuable images (e.g. hard to obtain).
  Legal reasons
  To ensure system robustness

- Near-lossless compression (generalization of lossless):
  Allow to set limits to the peak errors introduced
  Higher compression ratios

- Typical application restrictions:
  1. Limited resources
  2. Constrained energy consumption
  3. Low latency
  4. High throughput

- Applications (Main restrictions):
  Image capturing satellites (1,2,4)
  Medical imaging (1,2)
  Industry (3,4)
  Drones (1,2,3,4)

- Benefit from or require custom hardware. Particularly, implemented using FPGAs
  Production tends to be in low volumes
  Reconfigurability: update deployed systems
  Image sensors can be connected directly

JPEG-LS image CODEC

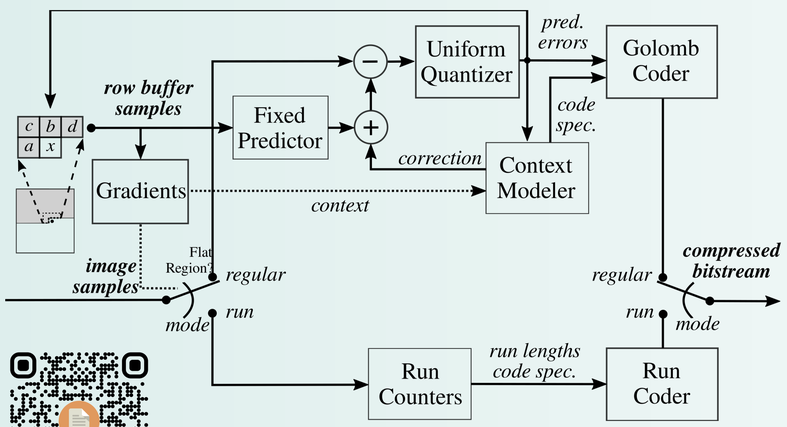
- Not the highest compression codec in the state of the art, but very well suited for these applications:
  Provides a competitive lossless compression
  Very good near-lossless performance
  High performance and low resource implementations

- As a result, several hardware designs have been published and it has even been used in NASA's Mars Exploration Rover mission

- Prediction error (ε) modeled as:

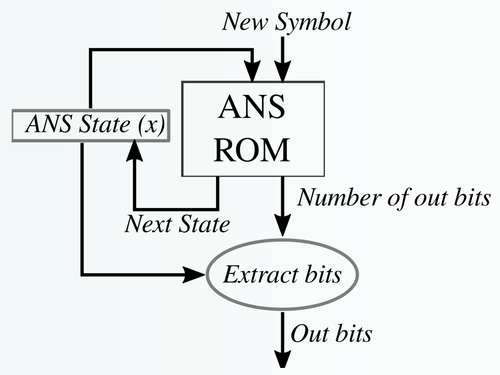
$$P(\theta, s)(\epsilon) = C(\theta, s)\theta^{|\epsilon - s|}, \epsilon = 0, \pm 1, \pm 2, \ldots,$$

- Problem: Coder does not perform well for medium and low entropies.



More about JPEG-LS

Entropy coding with tANS



From a black box perspective, tANS works as a Finite state machine (FSM) where the symbol to encode is the input and the current state is an integer, the ANS state, where ANS stores fractional bits of information. The output of the FSM ROM is the next state and the number of bits to take from the least significant part of the current state, which are then stored in the output bit file. From its design, tANS is meant to be implemented as a microcoded FSM (at least partially),and the FSM ROM is referred to as the tANS table. After a block of symbols is finished, the final state needs to be stored in the output bit file.

More about ANS

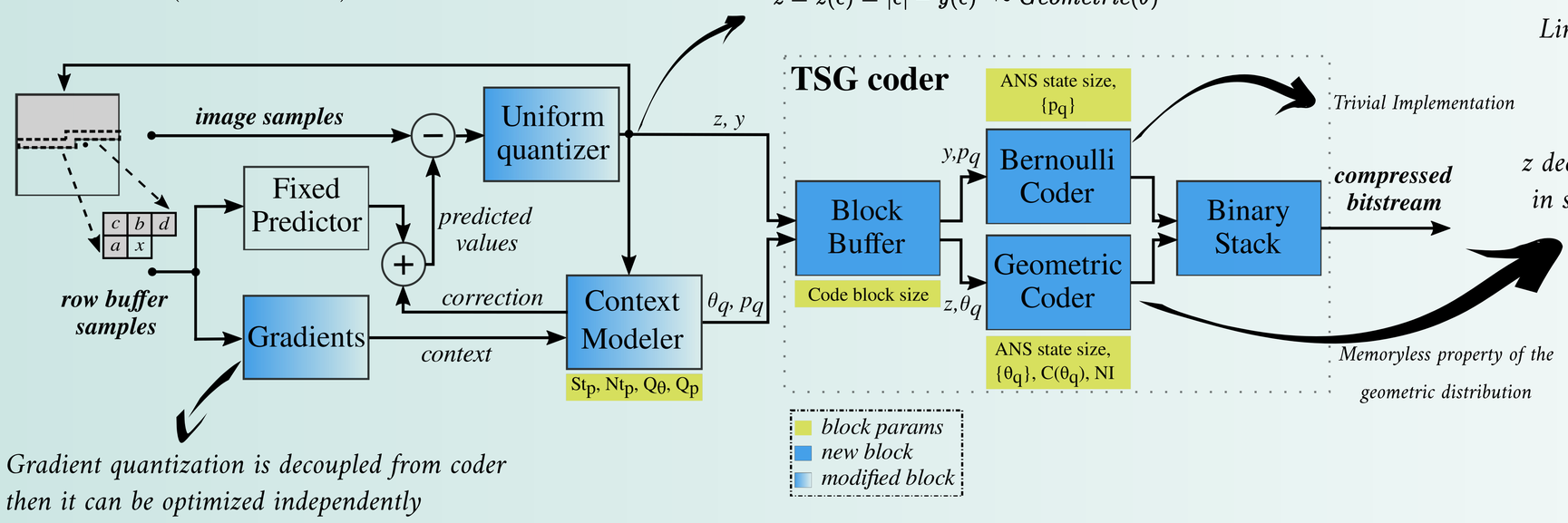
LOCO-ANS

Quantized prediction error modeled as Two-sided geometric (TSG) distribution (as in JPEG-LS) but using an equivalent model:

$$y = y(\epsilon) \triangleq \begin{cases} 0, & \epsilon \geq 0 \\ 1, & \epsilon < 0 \end{cases} \sim Bernulli(p)$$

$$z = z(\epsilon) \triangleq |\epsilon| - y(\epsilon) \sim Geometric(\theta)$$

Decouples distribution parameters, simplifying estimation and coder



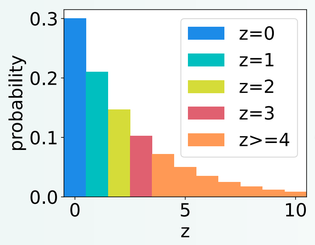
Gradient quantization is decoupled from coder then it can be optimized independently

```
Require: z_bits, z, param
c ← get_cardinality(param)
remaining_sym ← z
subsym ← mod(z, c)
if z ≥ NI * c then                          } Limit iteratons to NI
    store_in_binary(z, z_bits)
    remaining_sym ← NI * c
    subsym ← c
end if
repeat                                      } z decomposition in subsymbols
    remaining_sym ← remaining_sym − subsym
    obits ← ANS_table[param][state][subsym].bits
    store_in_binary_stack(state, obits)
    state ← ANS_table[param][state][subsym].nx_st
    subsym ← c
until remaining_sym = 0
```

Codification of single z limiting iterations.

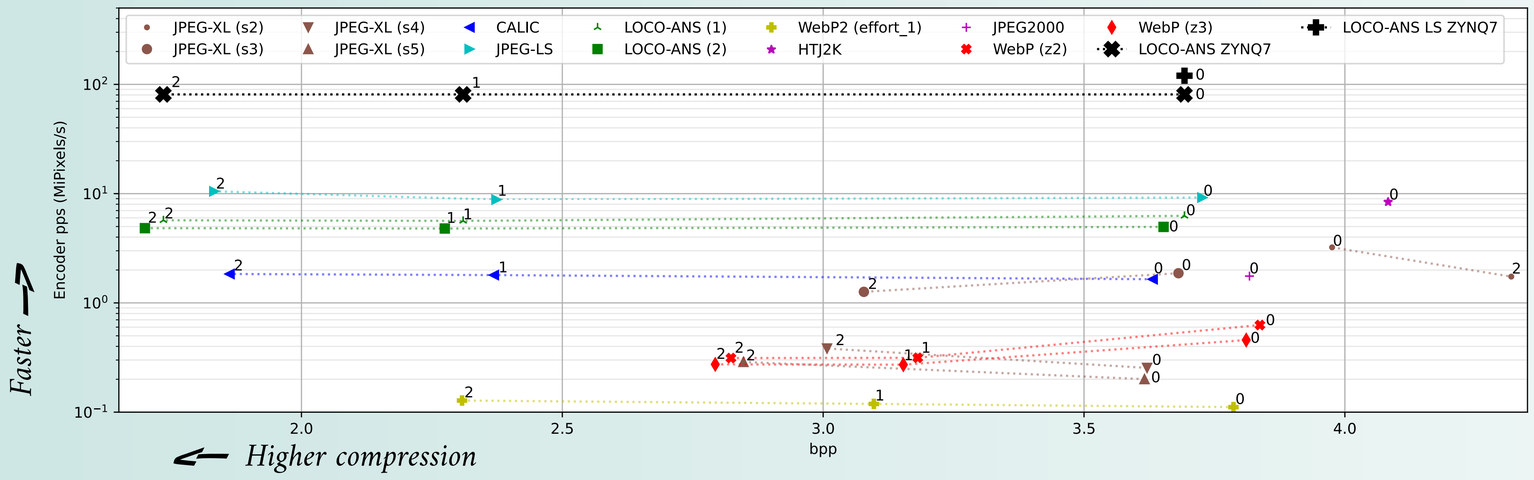
ANS coding of subsymbol

using for all iterations the same



ANS coder subsymbol source
(solves high cardinality and small probs problems)

Performance comparison



Peak error shown next to performance point. Software implementations were run in a Raspberry Pi 3 Model B. Preliminary hardware codec was implemented in a ZYNQ 7010/20.

Current Development (Hardware implementation)

- Hardware implementation is currently under development (prototype optimization phase).
- Preliminary results show that the TSG coder achieves in mean between 1.2 and 2.4 times higher throughput than pixel decorrelation stage (for lossless decorrelation, near-lossless is even slower)

Conclusions

- Compered to JPEG-LS using the same context size, LOCO-ANS achieves a bpp improvement of up to 1.6%, 6% and 37.6% for a peak error set to 0,1 and 10, respectively.
- The ANS based coder for two-sided geometric sources provides a highly efficient and low complexity coding. This coder enables further optimizations
- LOCO-ANS approaches lossless compression rates of more complex encoders, even surpassing them in near-lossless compression, while obtaining a much faster encoder speed and amenable hardware implementation.
- Recent developments support that the proposed coder does not bottleneck performance in pipelined hardware.
- Even a decade old, low end FPGA we achieve 3-4 times higher throughput than single thread software implementation in high end i7-6700K µP

LOCO-ANS open source Software implementation (Hw implementation coming soon)

This work was supported in part by the Spanish Research Agency under the project AgileMon (AEI PID2019-104451RB-C21).