LOCO and JPEG-LS

A new method for **lossless** and **near lossless** image compression

Nimrod Peleg
Update: May 2009
Credit...

• **Suggested by HP Labs, 1996**

• Developed by: M.J. Weinberger, G. Seroussi and G. Sapiro,

“LOCO-I: A Low Complexity, Context-Based, Lossless Image Compression Algorithm”,

General Block Diagram

Digital Source
Image Data

Context Modeling

Prediction

Run Mode

Error Encoding

Compressed Data
Context Based Algorithm

- An efficient coding needs a statistical model, for a good prediction of pixel value.
- The statistical distribution of a pixel is predicted according to the previous pixels.
- The best distribution for coding is minimum entropy distribution.
- To achieve it, each pixel is assigned to a “context” (728 context types in LOCO).
Prediction

• After context definition, a simple prediction is activated.
• The predictor is NOT context dependent, but same for all contexts.
• It’s a simple and effective edge detector, based on 3 gradients.
• Prediction error is calculated between predicted and “real” pixel value.
Error Coding

• Basic assumption: for each context, a statistical information is available.

• A Geometric distribution is assumed for the prediction error, and LOCO uses 2 parameters for each context: Average and Decay factor.

• LOCO uses a Golomb-Rice entropy coding, which is optimal for two-sided geometric distribution (it’s a 1 parameter Huffman like method).
The Golomb-Rice technique is simple, low-complexity and efficient:

- Huffman-like complexity
- Almost Arithmetic coding efficiency
Run Mode

• When all local gradient are zero, we can assume that it’s a “smooth” area in the image.
• In this situation we skip the prediction and the prediction error coding stages.
• We go on with the “run-mode” until the condition $x=b$ is no more TRUE.
• This process saves lots of bits for long runs.
A more detailed Description
General Scheme

Single frames

Prediction

Context Modeling

Prediction Error

Entropy Coding

Statistical Parameters

Compressed BitStream

To Network
Context Modeling

Every pixel is assigned to a context based on locally calculated parameters.

Why?

Prediction errors for pixels in same context, have the same statistical distribution.
The assumption is -

The prediction error is distributed as a two sided geometric distribution.

- The extinction factor is context dependent.
- There is systematic bias for every context.
Calculating the context number

There is a **tradeoff** between the **number** of context and the **quality** of the estimated statistics for each context.

**In the LOCO algorithm:**
There are 728 contexts created by using the **loca** Gradients:

\[
\begin{align*}
D1 &= d-a \\
D2 &= a-c \\
D3 &= c-b
\end{align*}
\]
Prediction

The Loco predictor is "casual”, it uses nearby pixels which already have been scanned

\[
Px = \begin{cases} 
\min(a,b) & c \geq \max(a,b) \\
\max(a,b) & c \leq \min(a,b) \\
a + b - c & \text{otherwise}
\end{cases}
\]
The principle:

The assumption is:

‘x’ should be on the same plane created by a, b and c.
For example:

Here, the lowest value $x$ can get is the value of $b$ (from what we “have”).

$$P_x = \begin{cases} 
\min(a,b) & c \geq \max(a,b) \\
\max(a,b) & c \leq \min(a,b) \\
 a + b - c & \text{otherwise}
\end{cases}$$
Why not take the Average?

\[
x = \frac{a + b + c}{3}
\]
Let’s take a look on this case -

When $c$ is very close to $a$ . . .

But,

$$x_{avg} \approx \frac{2a + b}{3}$$

$$P_x = b$$
Prediction Example

Original Image

Predicted Image (SNR=25.3dB)
and more details...
The Quantizer

The gradients are quantized by a 9 level quantizer

\[(Q1, Q2, Q3)\]

The context number - Q
Entropy Coding

Golomb-Rice is an **optimal** entropy coder for geometric distribution

**Golomb - Rice Codes**

- Does not require tables (vs. Huffman)
- One parameter only: $k$

- By binary presentation: 8-k bits
- By unary presentation: k bits
Golomb-Rice coding example

Let’s take the number: \( x=14 \)

Unary presentation of \( 11_2 \) \((= 3_{10})\) = 0 0 0 1

Binary presentation - 1 0

The BitStream
Run mode

A special mode which allows efficient compression of a sequence of pixels with same value.

For example:
**Lossy Mode**

Maximal allowed restoration error per pixel = 0

The method:

The prediction error is quantized by a $2^{\text{Near}} + 1$ step quantizer.
And the profit is . . .

- Narrower geometric distribution
  
  
  Shorter Golomb-Rice Codes

- Spending more time in Run-Mode
Explanation
(last graph: Lena for different Near values)

Near לראות, שבפרכי נמוכים העלייה ביחס הדחיסה
Near לניירות עם, אולס בעריכים גבוהים של
Near אינטראקציה בין שיפור ביחס הדחיסה. השקטות לכל, היא
Near ועם אינטראקציה בין שיפור ביחס הדחיסה, התמונה
Nearยา עיניים, היא LOCO, מאבד את ה- Near
Near.Contexts של כל, מאבד את העיילות, בנוסף להרחבת הדחיסה
Near וביעילות ההגנה, מגוון, כאופן להדמאת לראות
 Near, SNR ב.
Different Near values compression

Near = 3

Near = 10
Benchmark Comparison

Compresson vs. Complexity trade-off

- CALIC-A
- JPEG-A
- JSLUG
- ALCM
- CALIC-H
- LOCO-I
- JPEG-H

- JPEG-H, JPEG-A: Current standard
- CALIC-A: Best available compression
- JSLUG, ALCM, CALIC-H: Other proposals

Download Free from UBC