0.1 PNG (Encoding and Decoding)

- *Algorithm:* PNG (Encoding and Decoding) (algo. 1)

- *Input:* PNG image

- *Complexity:* $\mathcal{O}(n \log n)$

- *Data structure compatibility:* Huffman tree

- *Common applications:* Face recognization

Problem. PNG (Encoding and Decoding)

PNG[1], "Portable Network Graphics", is a computer file format for storing, transmitting, and displaying images. It has features of lossless compression, transparency information, interlacing and a range of color depths. We analyze PNG's file structure, lossless filtering and compression to encode and decode PNG.

Description

PNG format is widely used now. Comparing with other image format like JPG, BMP, the advantage of PNG is smaller file size and that's why it's called "portable". And the advantage benefits from its lossless filtering and compression. We will introduce the file structure of PNG, lossless filtering and compression algorithm.

File Structure

The fundamental building block of PNG images is the *chunk*[2]. A PNG file consists of PNG signture and chunks. PNG format defines two kinds of chunk, *critical chunk* and *ancillary chunks*. It is essential for PNG encoding and decoding softerwares to support critical chunks encoding and decoding. Hence we will focus on critical chunks. A general PNG file consists of PNG signture and three critical chunks, IHDR (header chunk), IDAT (image data chunk), IEND (image trailer chunk) (Tbl. 1). The components are in this order.

Table 1: General PNG file structure

PNG File			
PNG Signature	IHDR	IDAT	IEND

1. **PNG Signature**
   PNG Signture is the first 8-byte of a PNG file, which is just a simple identifier code. It is designed to detect the most common types of file-transfer corruption. The contents of PNG signature are 0x89, 0x50, 0x4E, 0x47, 0x0D, 0x0A, 0x1A(Tbl. 2).

Table 2: PNG Signature Bytes[2]

Decimal Value	ASCII Interpretation
137	A byte with its most significant bit set ("8-bit character")
80	P
78	N
71	G
13	Carriage-return (CR) character, a.k.a. CTRL-M or ^M
10	Line-feed (LF) character, a.k.a. CTRL-J or ^J
26	CTRL-Z or ^Z
10	Line-feed (LF) character, a.k.a. CTRL-J or ^J

2. **Chunks**

Although there are chunks identified as different types, they are bulit in common structure. Each chunk can be divided into 4 parts (in sequence) (Tbl. 3).

Table 3: Chunk structure

Name	Bytes	Description
Length	4	Length of Chunk data, not longer than $(2^{31} - 1)$bytes
Chunk Type Code	4	Consists of ASCII character
Chunk data	Determined by Length	Main data of chunk
CRC	4	Cyclic redundancy check

- IHDR (header chunk)
  IHDR stores the basic information of image data. It has to be the first chunk and the only one. The chunk data of IHDR is 13-byte long. It determines the width and height of the image, bit depth, color type, compression method, filter method, interlace method.

- IDAT (image data chunk)
  IDAT stores the main image data of PNG file. Then compression and filtering method are applied to generate this chunk.

- IEND (image trailer chunk) IEND marks the end of PNG file. It has to be the end of the file. The contents of it are 0x00, 0x00, 0x00, 0x00, 0x49, 0x45, 0x4E, 0x44, 0xAE, 0x42, 0x60, 0x82, which corresponds to the chunk structure.

Filtering method[3]

The filtering method of PNG is differential coding, which is reversible. The data of a pixel is determined by the pixels on the upper, left, upper-left of the pixel. PNG provides 5 ways to calculate the substraction, which is determined by the filter method byte (Tbl. 4).

Table 4: Filtering method

Name	Description
None	Unchanged
Sub	Difference between the left pixel and itself
Up	Difference between the upper pixel and itself
Average	Difference between the averge of upper and left pixels and itself
Paeth	Difference between the left, upper, upper-left pixels and itself

Compression method

The compression method of PNG is deflate algorithm, which is a mixture of Huffman coding (Alg. 1) and LZ77.

Algorithm 1: Huffman coding

Input : Symbols
Output: Encoded symbols

1 Calculate the probablities of each different symbols
2 Assign the probability to each node
3 Sort the probabilities in non-decreasing order
4 **while** *all probablities of nodes in the tree are less than 1* **do**
5   Generate a parent for the least and the second least probablity nodes
6   Assign the sum of the probablities of these two childs to the parent
7   Mark the child with less probability as 0
8   Mark the child with less probability as 1
9 **end while**
10 The path to the symbol (consisting of zeros and ones) is Huffman code of the symbol.
11 **return** Pathes

References.

[1] libpng.org. *Chapter 1. An Introduction to PNG*. http://www.libpng.org/pub/png/book/chapter01.html Accessed October 7, 2020 (cit. on p. 1).

[2] libpng.org. *Chapter 8. PNG Basics*. http://www.libpng.org/pub/png/book/chapter08.html Accessed October 7, 2020 (cit. on pp. 1, 2).

[3] libpng.org. *Chapter 9. Compression and Filtering*. http://www.libpng.org/pub/png/book/chapter09.html Accessed October 7, 2020 (cit. on p. 2).