Image Processing Assignment

(Hamming Code: Error detection and Correction)

NAME: KAPIL KUMAR CHHIPA

ROLL-NUMBER: M180265CA

Type your text

In real time data transmission sometimes what data we send not received by receiver they received something else. So, with the help of hamming code we can get what are the wrong data got by receiver and after knowing the error receiver can remove it with the help of Hamming Code.

Hamming code is well known for its error detection and correction capability.

CONCEPT OF HAMMING CODE

When we send some data, it will go in the form of **bits** so if the data received by the receiver is wrong means there will be some **bits are changed** either **1 to 0** or **0 to 1**.

In hamming code, we add some **extra bits** along with the actual data these extra bits are known as **Redundant bits**.

HOW MANY REDUNDANT BITS ARE REQUIRED?

With the help of $\mathbf{2^r >= m + r + 1}$ formula we can get number of redundant bits required.

Where **r = redundant bit** and **m = message bit.**

Suppose number of message bits are 7 then total redundant bits are required equal to 4. ($\mathbf{2^4 >= 7 + 4 + 1}$).

Now if we want to understand what is the actual algorithm for hamming code we have to have knowledge about the parity bit so

WHAT IS PARITY BIT?

Parity bit is a bit appended to the data bits which ensures that the total number of 1's are even (even parity) or odd (odd parity).

While checking the parity, if the total number of 1's are odd then write the value of parity bit **P1**(or **P2** etc.) as **1** (which means the error is there) and if it is even then the value of parity bit is **0** (which means no error).

ALGORITHM OF HAMMING CODE

1. Write the bit positions starting from 1 in binary form (1, 10, 11, 100, etc).
2. All the bit positions that are a power of 2 are marked as parity bits (1, 2, 4, 8, etc).
3. All the other bit positions are marked as data bits.
4. Each data bit is included in a unique set of parity bits, as determined its bit position in binary form.
   a. Parity bit 1 covers all the bits positions whose binary representation includes a 1 in the least significant
   position (1, 3, 5, 7, 9, 11, etc).
   b. Parity bit 2 covers all the bits positions whose binary representation includes a 1 in

the second position from
the least significant bit (2, 3, 6, 7, 10, 11, etc).
c. Parity bit 4 covers all the bits positions whose binary representation includes a 1 in the third position from
the least significant bit (4–7, 12–15, 20–23, etc).
d. Parity bit 8 covers all the bits positions whose binary representation includes a 1 in the fourth position from
the least significant bit bits (8–15, 24–31, 40–47, etc).
e. In general, each parity bit covers all bits where the bitwise AND of the parity position and the bit position is
non-zero.

5. Since we check for even parity set a parity bit to 1 if the total number of ones in the positions it checks is
odd.

6. Set a parity bit to 0 if the total number of ones in the positions it checks is even.

DETERMINING THE POSITION OF REDUNDANT BITS

These redundancy bits are placed at the positions which correspond to the power of 2.

As in the above example:

1. The number of data bits = 7
2. The number of redundant bits = 4
3. The total number of bits = 11
4. The redundant bits are placed at positions corresponding to power of 2 -: 1, 2, 4, and 8.

Representation of Hamming Code with Parity bits and Data

11	10	9	8	7	6	5	4	3	2	1
D11	D10	D9	P8	D7	D6	D4	P4	D3	P2	P1

Determining the Parity bits

1. P1 bit is calculated using parity check at all the bits positions whose binary representation includes a 1 in the least significant position. (1, 3, 5, 7, 9, 11).
2. P2 bit is calculated using parity check at all the bits positions whose binary representation includes a 1 in the second position from the least significant bit. (2,3,6,7,10,11)
3. P4 bit is calculated using parity check at all the bits positions whose binary representation includes a 1 in the third position from the least significant bit. (4, 5, 6, 7).
4. R8 bit is calculated using parity check at all the bits positions whose binary representation includes a 1 in the fourth position from the least significant bit. (8,9,10,11)

EXAMPLE: Suppose the data to be transmitted is **1011001**.

AFTER CALCULATIONS THE BITS WILL BE PLACED AS FOLLOWS:

D11	D10	D9	P8	D7	D6	D5	P4	D3	P2	P1
1	0	1	0	1	0	0	1	1	1	0

Error detection and correction –
Suppose in the above example the 6th bit is changed from 0 to 1 during data transmission.

So the message with error is: **10101101110**

HAMMING CODE: ERROR DETECTION

Again, Check the Parity bits:

CHECK FOR P1:

As we can observe the total number of bits are even so we will write the value of parity bit as **P1 = 0**. This means **there is no error**.

CHECK FOR P2:

As we can observe that the number of 1's are odd, then we will write the value of **P2 = 1**. This means **error is there.**

CHECK FOR P4:

As we can observe that the number of 1's are odd, then we will write the value of **P4 = 1**. This means the **error is there.**

CHECK FOR P8:

As we can observe that the number of 1's are odd, then we will write the value of **P8 = 0**. This means **there is no error**.

HAMMING CODE: ERROR CORRECTION

With the help of error checking method, we know that our code has error. So, the error is present at bit number 6 (Decimal vale of 0110 is 6). So, change the bit of D6.

P8	P4	P2	P1
0	1	1	0

2^{ND} EXAMPLE

Let's we received a hamming code of 7-bits which is **1011011.**

Redundant bits required = 3

HAMMING CODE: ERROR DETECTION

D7	D6	D5	P4	D3	P1	P2
1	0	1	1	0	1	1

First thing what we will do? We check is there any error in received message.

CHECK FOR P1:

As we can observe the total number of bits are odd so we will write the value of parity bit as **P1 = 1**. This means **error is there**.

CHECK FOR P2:

As we can observe that the number of 1's are even, then we will write the value of **P2 = 0**. This means **there is no error**.

CHECK FOR P4:

As we can observe that the number of 1's are odd, then we will write the value of **P4 = 1**. This means the error is there.

So, from the above parity analysis, P1 & P4 are not equal to 0, so we can clearly say that the received hamming code has errors.

HAMMING CODE: ERROR CORRECTION

With the help of error checking method, we know that our code has error. So the error is present at bit number 5 (Decimal vale of 101 is 5). So change the bit of D5.

P4	P2	P1
1	0	1

Correct message is: 1001011

Received Message is: 1011011