An introduction to error correcting codes

Mathematics sin fronteras

María Chara

Universidad Nacional del Litoral

May 12, 2021

FIQ UNL. FACULTAD DE INGENIERÍA QUÍMICA

Mariner 9

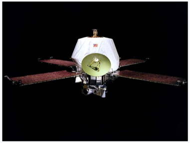
From Wikipedia, the free encyclopedia

Mariner 9 (**Mariner Mars '71** / **Mariner-I**) was a robotic space probe that contributed greatly to the exploration of Mars and was part of the NASA Mariner program. Mariner 9 was launched toward Mars on May 30, 1971[1][2] from LC-36B at Cape Canaveral Air Force Station, Florida, and reached the planet on November 14 of the same year,[1][2] becoming the first spacecraft to orbit another planet[3] – only narrowly beating the Soviet probes *Mars 2* (launched May 19) and *Mars 3* (launched May 28), which both arrived at Mars only weeks later.

After the occurrence of dust storms on the planet for several months following its arrival, the orbiter managed to send back clear pictures of the surface. Mariner 9 successfully returned 7,329 images over the course of its mission, which concluded in October 1972.[4]

Contents [hide]

Mariner 9



The Mariner 9 spacecraft

Mission type	Mars orbiter
Operator	NASA / JPL
COSPAR ID	1971-051A
SATCAT no.	5261
Mission duration	1 year, 4 months, 27 days
Spacecraft properties	
Manufacturer	Jet Propulsion Laboratory
Launch mass	997.9 kilograms (2,200 lb)

Objectives [edit]

Figure: https://en.wikipedia.org/wiki/Mariner_9

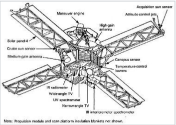
Mariner 9, 1971

Construction [edit]

The ultraviolet spectrometer aboard Mariner 9 was constructed by the Laboratory for Atmospheric and Space Physics at the University of Colorado, Boulder, Colorado. The ultraviolet spectrometer team was led by Professor Charles Barth.

The Infrared Interferometer Spectrometer (IRIS) team was led by Dr. Rudolf A. Hanel from NASA Goddard Spaceflight Center (GSFC). The IRIS instrument was built by Texas Instruments, Dallas, Texas.

The Infrared Radiometer (IRR) team was led by Professor Gerald Neugebauer from the California Institute of Technology (Caltech).



A schematic of Mariner 9, showing the major components and features

Error-Correction Codes achievements [edit]

To control for errors in the reception of the grayscale image data sent by Mariner 9 (caused by a low signal-to-noise ratio), the data had to be encoded before transmission using a so-called forward error-correcting code (FEC). Without FEC, noise would have made up roughly a quarter of a received image, while the FEC encoded the data in a redundant way which allowed for the reconstruction of most of the sent image data at reception.

Since the flown hardware was constrained with regards to weight, power consumption, storage, and computing power, some considerations had to be put into choosing an FEC, and it was decided to use a Hadamard code for Mariner 9. Each image pixel was represented as a six-bit binary value, which had 64 possible grayscale levels. Because of limitations of the transmitter, the maximum useful data length was about 30 bits. Instead of using a repetition code, a [32, 6, 16] Hadamard code was used, which is also a 1st-order Reed-Muller code. Errors of up to seven bits per each 32-bit word could be corrected using this scheme.[10][11] Compared to a five-repetition code, the error correcting properties of this Hadamard code were much better, yet its data rate was comparable. The efficient decoding algorithm was an important factor in the decision to use this code. The circuitry used was called the "Green Machine", which employed the fast Fourier transform, increasing

Figure: https://en.wikipedia.org/wiki/Mariner_9

Mariner 9 image of the central caldera of the Martian volcano, Olympus Mons.

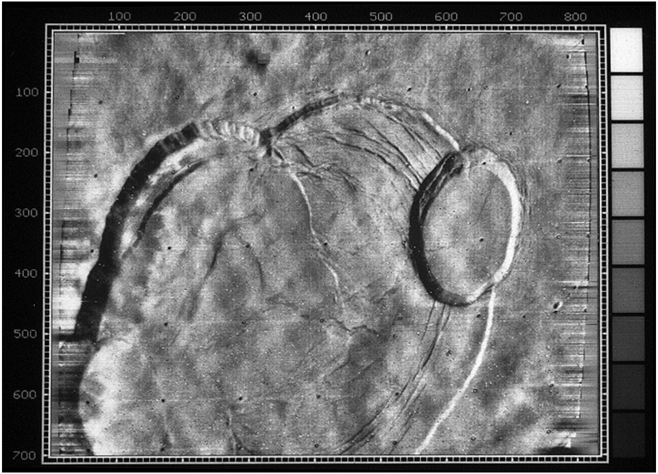


Figure: https://nssdc.gsfc.nasa.gov/imgcat/html/object_page/m09_mtvs4265_52.html

Patricia “Patsy” Conklin, an employee in the Bioscience and Planetology Section at NASA’s Jet Propulsion Laboratory assembles Mariner 9 photos into large mosaics.

Figure: https://www.upi.com/Top_News/2020/05/30/On-This-Day-Mariner-9-launched-toward-Mars/4991590342141/

Example: Transmission of pictures from NASA spaceships

Example: Transmission of pictures from NASA spaceships

Mariner 4 (1964$/$1965) performed the first successful flyby of the planet Mars, returning the first close-up pictures of the Martian surface. It captured the first images of another planet ever returned from deep space, taking 22 complete pictures of Mars.

Example: Transmission of pictures from NASA spaceships

Mariner 4 (1964/1965) performed the first successful flyby of the planet Mars, returning the first close-up pictures of the Martian surface. It captured the first images of another planet ever returned from deep space, taking 22 complete pictures of Mars.

Each picture was partitioned into 200×200 pixels and each image pixel was represented as a six-bit binary value, which had 64 possible grayscale levels from white (000000) to black (111111).

Example: Transmission of pictures from NASA spaceships

Mariner 4 (1964/1965) performed the first successful flyby of the planet Mars, returning the first close-up pictures of the Martian surface. It captured the first images of another planet ever returned from deep space, taking 22 complete pictures of Mars.

Each picture was partitioned into 200×200 pixels and each image pixel was represented as a six-bit binary value, which had 64 possible grayscale levels from white (000000) to black (111111).

The total number of binary digits per picture was 240000. Each individual photograph took approximately six hours to be transmitted back to Earth.

Example: Transmission of pictures from NASA spaceships

Mariner 4 (1964/1965) performed the first successful flyby of the planet Mars, returning the first close-up pictures of the Martian surface. It captured the first images of another planet ever returned from deep space, taking 22 complete pictures of Mars.

Each picture was partitioned into 200×200 pixels and each image pixel was represented as a six-bit binary value, which had 64 possible grayscale levels from white (000000) to black (111111).

The total number of binary digits per picture was 240000. Each individual photograph took approximately six hours to be transmitted back to Earth.

All images were stored onto a on-board magnetic tape recorder and then sent to our planet. All images were transmitted twice to ensure no data was missing or corrupt.

Mariner 4, 1964 1965

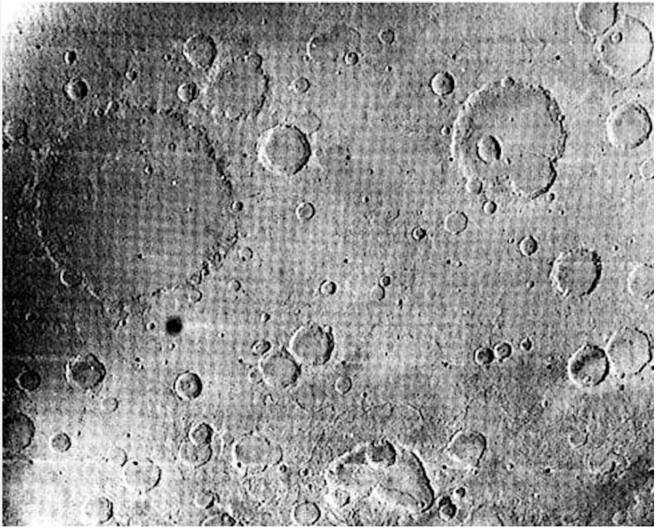
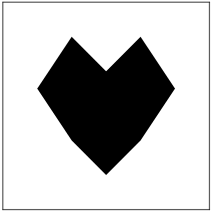


Figure: Photo:NASA

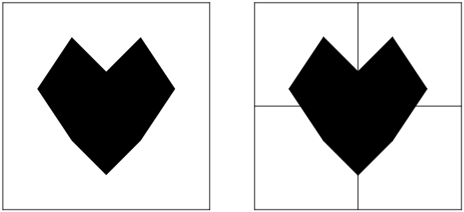
FIQ UNL. FACULTAD DE INGENIERÍA QUÍMICA

How can we do this?

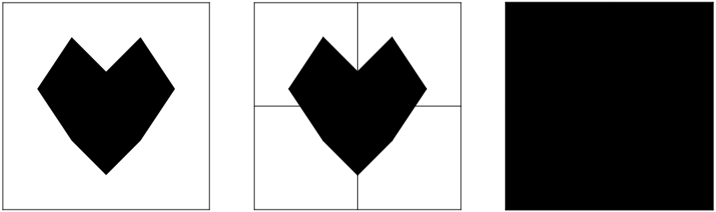
How can we do this?



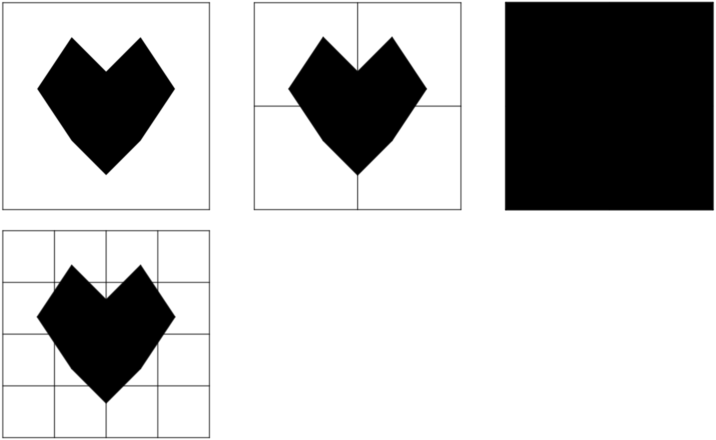
How can we do this?



How can we do this?

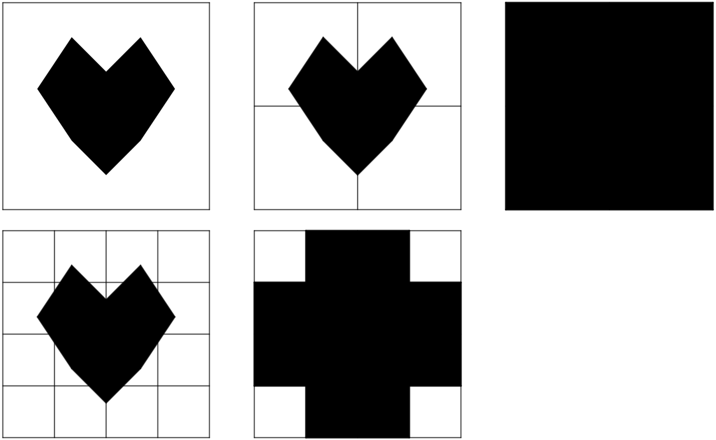


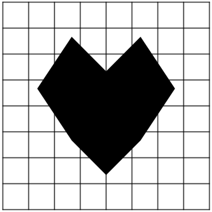
How can we do this?

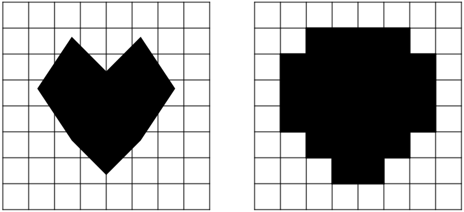


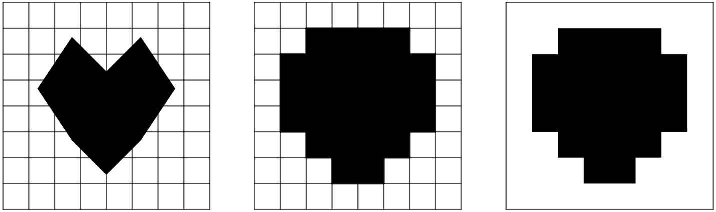
FIQ
UNL. FACULTAD DE INGENIERÍA QUÍMICA

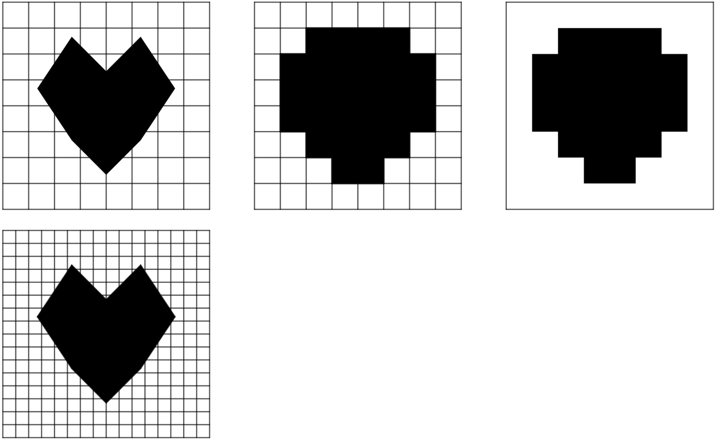
How can we do this?

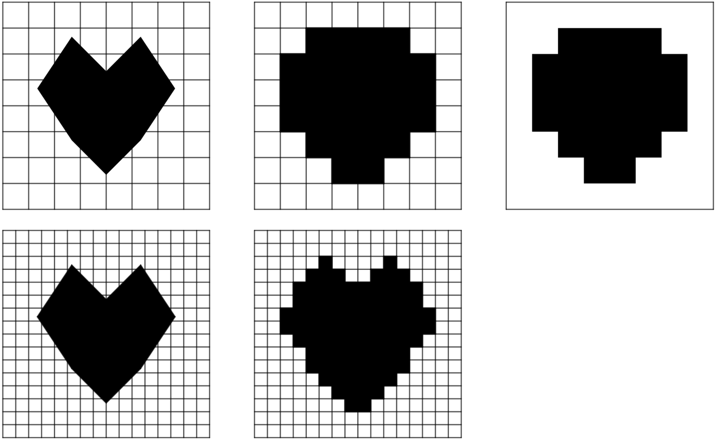


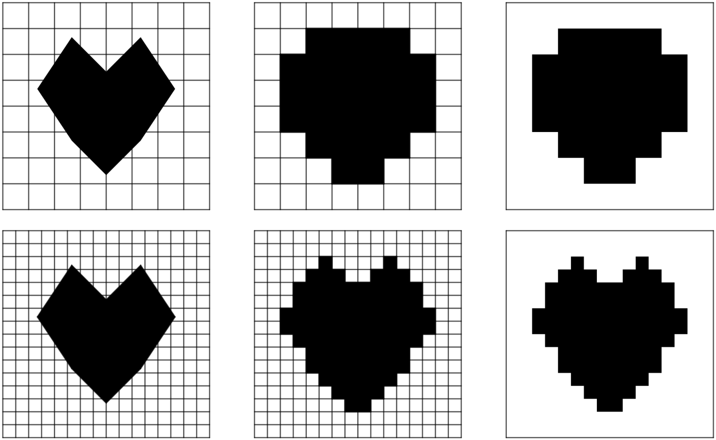


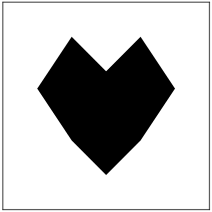


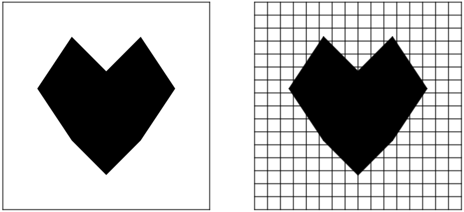


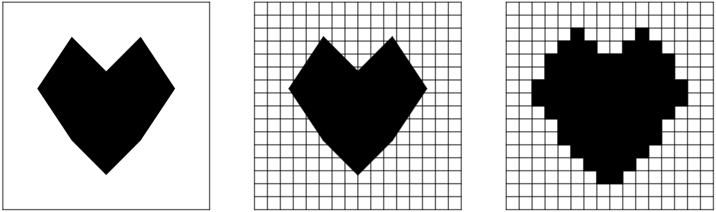




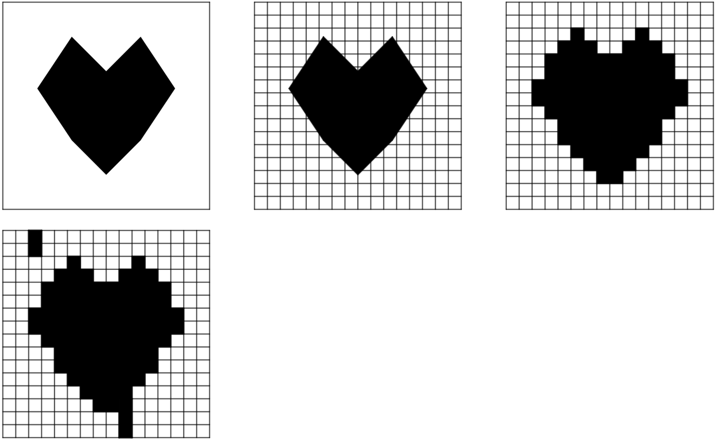


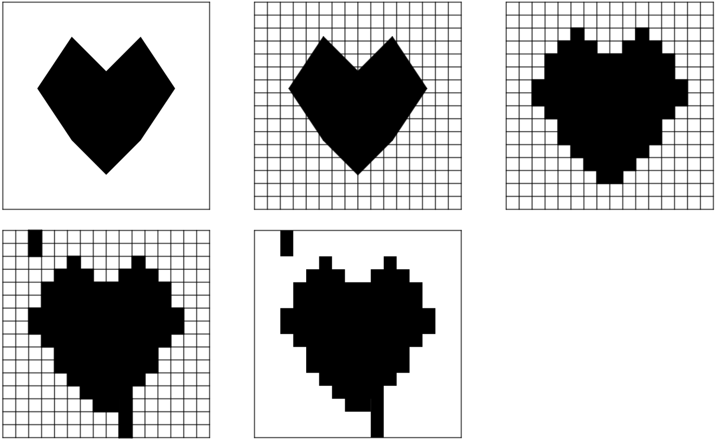






UNL . FACULTAD DE INGENIERÍA QUÍMICA





Example: $\mathcal{C} = \{0, 1\}$

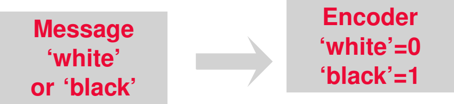
Example: $\mathcal{C} = \{0, 1\}$

Message 'white' or 'black'

Example: $\mathcal{C} = \{0, 1\}$

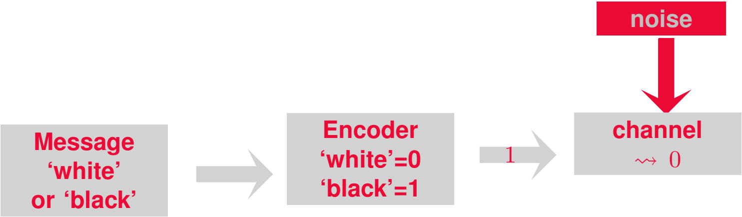
Message ‘white’ or ‘black’ →

Example: $\mathcal{C} = \{0, 1\}$

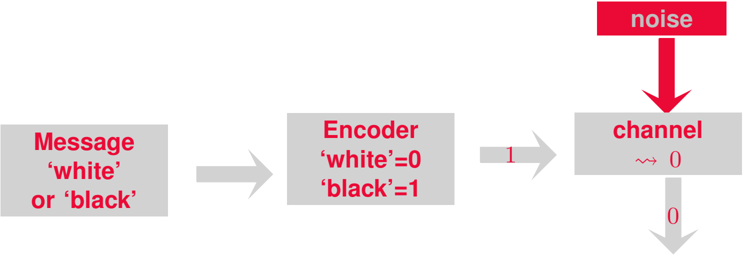


Example: $\mathcal{C} = \{0, 1\}$

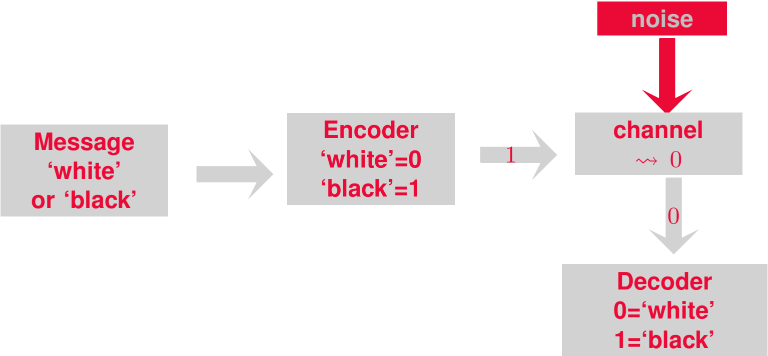
Example: $\mathcal{C} = \{0, 1\}$



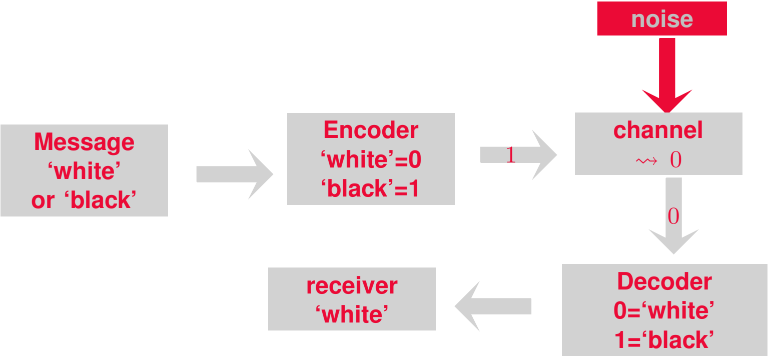
Example: $\mathcal{C} = \{0, 1\}$



Example: $\mathcal{C} = \{0, 1\}$



Example: $\mathcal{C} = \{0, 1\}$



Example: $\mathcal{C} = \{00, 11\}$

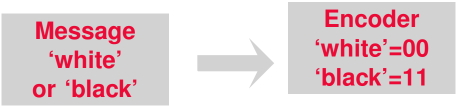
Example: $\mathcal{C} = \{00, 11\}$

Message
'white'
or 'black'

Example: $\mathcal{C} = \{00, 11\}$

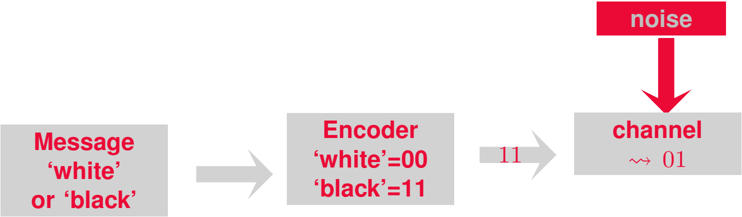
Message 'white' or 'black' →

Example: $\mathcal{C} = \{00, 11\}$

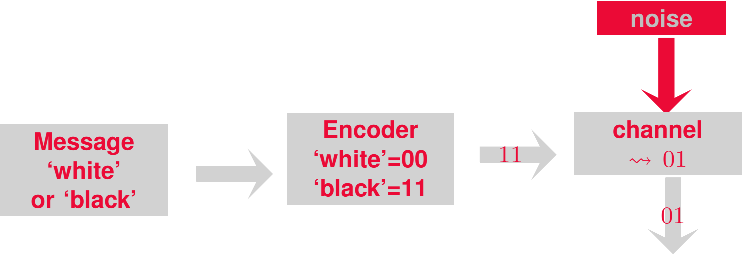


Example: $\mathcal{C} = \{00, 11\}$

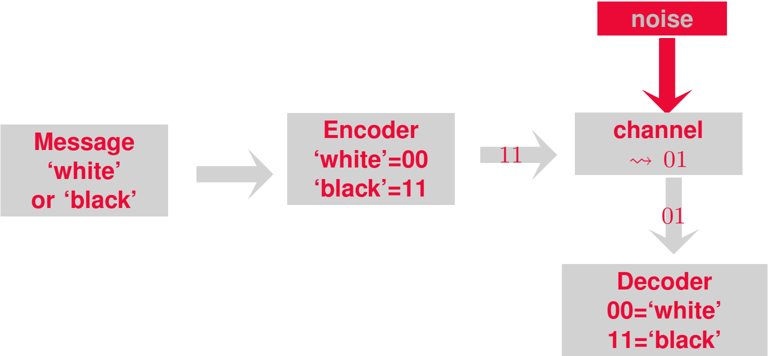
Example: $\mathcal{C} = \{00, 11\}$



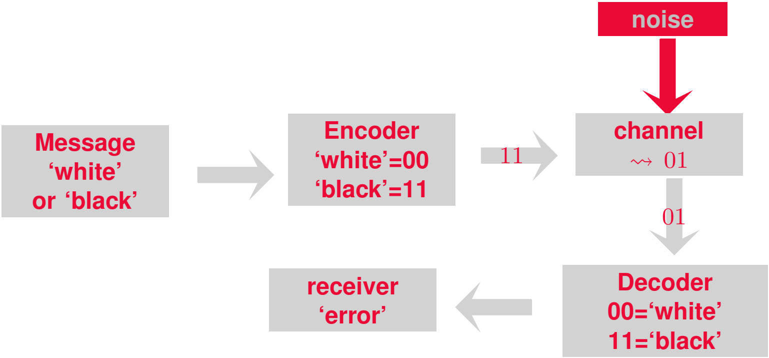
Example: $\mathcal{C} = \{00, 11\}$



Example: $\mathcal{C} = \{00, 11\}$



Example: $\mathcal{C} = \{00, 11\}$



Example: $\mathcal{C} = \{000, 111\}$

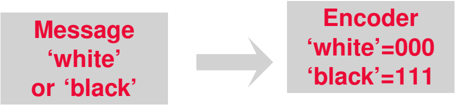
Example: $\mathcal{C} = \{000, 111\}$

Message 'white' or 'black'

Example: $\mathcal{C} = \{000, 111\}$

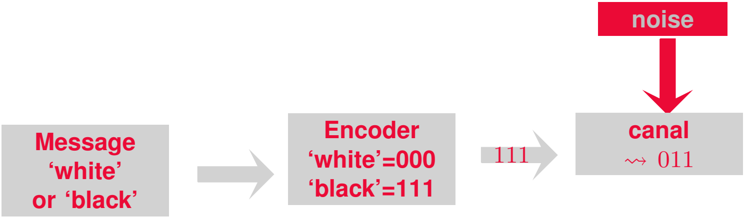
Message
'white'
or 'black'

Example: $\mathcal{C} = \{000, 111\}$

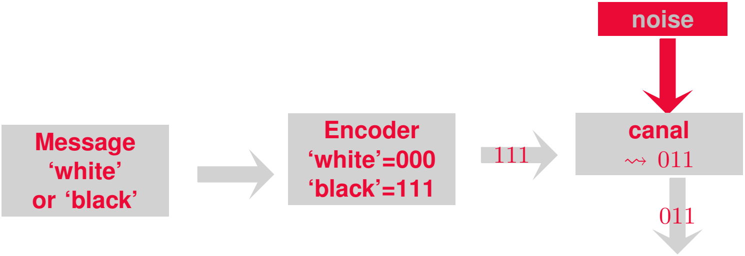


Example: $\mathcal{C} = \{000, 111\}$

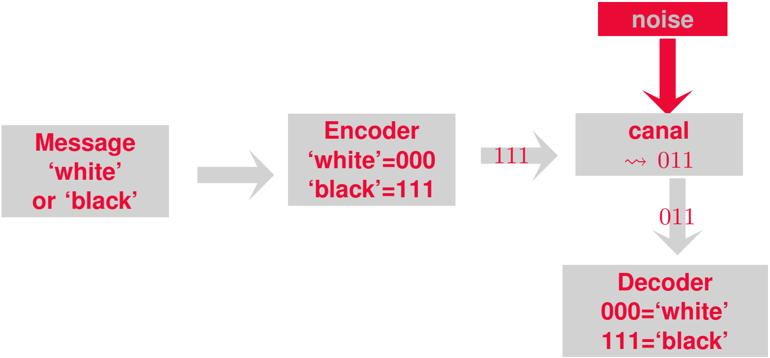
Example: $\mathcal{C} = \{000, 111\}$



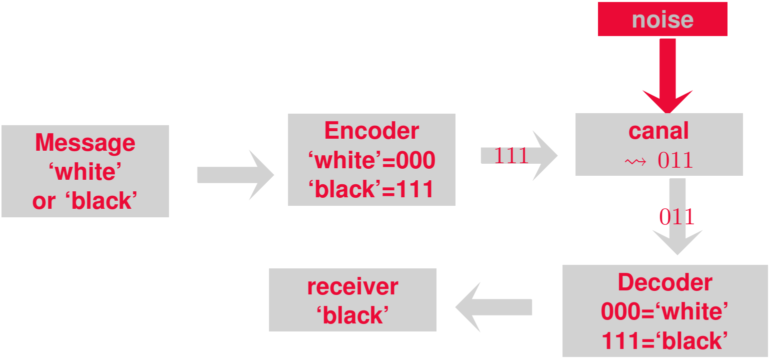
Example: $\mathcal{C} = \{000, 111\}$



Example: $\mathcal{C} = \{000, 111\}$



Example: $\mathcal{C} = \{000, 111\}$



- A (n, M)-**code** $\mathcal{C}$ over a finite set $\mathcal{A}$ is a subset of $\mathcal{A}^n$ with M elements. $\mathcal{A}$ is called the **alfabet**.

- A (n, M)-**code** $\mathcal{C}$ over a finite field $\mathbb{F}_q$ is a subset of $\mathbb{F}_q^n$ with M elements. $\mathbb{F}_q$ is called the **alfabet**.

- A (n, M)-**code** $\mathcal{C}$ over a finite field $\mathbb{F}_q$ is a subset of $\mathbb{F}_q^n$ with M elements. $\mathbb{F}_q$ is called the **alfabet**.
- The elements of $\mathbb{F}_q^n$ are called **words** and the elements of $\mathcal{C}$ are called **code words**.

- A (n, M)-**code** $\mathcal{C}$ over a finite field $\mathbb{F}_q$ is a subset of $\mathbb{F}_q^n$ with M elements. $\mathbb{F}_q$ is called the **alfabet**.
- The elements of $\mathbb{F}_q^n$ are called **words** and the elements of $\mathcal{C}$ are called **code words**.
- For $a = (a_1, a_2, \ldots, a_n)$ and $b = (b_1, b_2, \ldots, b_n) \in \mathbb{F}_q^n$ let

$$d(a, b) = |\{i : 1 \leq i \leq n,\ a_i \neq b_i\}|.$$

this function d is called the **Hamming distance** on $\mathbb{F}_q^n$.

- A (n, M)-**code** $\mathcal{C}$ over a finite field $\mathbb{F}_q$ is a subset of $\mathbb{F}_q^n$ with M elements. $\mathbb{F}_q$ is called the **alfabet**.
- The elements of $\mathbb{F}_q^n$ are called **words** and the elements of $\mathcal{C}$ are called **code words**.
- For $a = (a_1, a_2, \ldots, a_n)$ and $b = (b_1, b_2, \ldots, b_n) \in \mathbb{F}_q^n$ let

  $$d(a, b) = |\{i : 1 \leq i \leq n,\ a_i \neq b_i\}|.$$

  this function d is called the **Hamming distance** on $\mathbb{F}_q^n$.
- The **weight** of an element $a \in \mathbb{F}_q^n$ is defined by

  $$w(a) := d(a, 0) = |\{i : 1 \leq i \leq n,\ a_i \neq 0\}|.$$

- A (n, M)-**code** $\mathcal{C}$ over a finite field $\mathbb{F}_q$ is a subset of $\mathbb{F}_q^n$ with M elements. $\mathbb{F}_q$ is called the **alfabet**.
- The elements of $\mathbb{F}_q^n$ are called **words** and the elements of $\mathcal{C}$ are called **code words**.
- For $a = (a_1, a_2, \ldots, a_n)$ and $b = (b_1, b_2, \ldots, b_n) \in \mathbb{F}_q^n$ let

$$d(a, b) = |\{i : 1 \leq i \leq n,\ a_i \neq b_i\}|.$$

  this function d is called the **Hamming distance** on $\mathbb{F}_q^n$.
- The **weight** of an element $a \in \mathbb{F}_q^n$ is defined by

$$w(a) := d(a, 0) = |\{i : 1 \leq i \leq n,\ a_i \neq 0\}|.$$

- For example

$$d((00000), (01010)) = 2$$

  and

$$w((00000)) = 0 \qquad \text{y} \qquad w((01010)) = 2.$$

- A (n, M)-**code** $\mathcal{C}$ over a finite field $\mathbb{F}_q$ is a subset of $\mathbb{F}_q^n$ with M elements. $\mathbb{F}_q$ is called the **alfabet**.
- The elements of $\mathbb{F}_q^n$ are called **words** and the elements of $\mathcal{C}$ are called **code words**.
- For $a = (a_1, a_2, \ldots, a_n)$ and $b = (b_1, b_2, \ldots, b_n) \in \mathbb{F}_q^n$ let

  $$d(a, b) = |\{i : 1 \leq i \leq n,\ a_i \neq b_i\}|.$$

  this function d is called the **Hamming distance** on $\mathbb{F}_q^n$.
- The **weight** of an element $a \in \mathbb{F}_q^n$ is defined by

  $$w(a) := d(a, 0) = |\{i : 1 \leq i \leq n,\ a_i \neq 0\}|.$$

Obs.: The Hamming distance is a metric on $\mathbb{F}_q^n$.

- The **minimum distance** $d(\mathcal{C})$ of a code $\mathcal{C}$ is the minimum distance between distinct codewords, e. i.,

$$d(\mathcal{C}) = \min\{d(x, y) : x \in \mathcal{C},\ y \in \mathcal{C},\ x \neq y\}.$$

- The **minimum distance** $d(\mathcal{C})$ of a code $\mathcal{C}$ is the minimum distance between distinct codewords, e. i.,

$$d(\mathcal{C}) = \min\{d(x, y) : x \in \mathcal{C},\ y \in \mathcal{C},\ x \neq y\}.$$

- An (n, M, d)-code is a code $\mathcal{C}$ with M words of length n and minimum distance d.

- The **minimum distance** $d(\mathcal{C})$ of a code $\mathcal{C}$ is the minimum distance between distinct codewords, e. i.,

$$d(\mathcal{C}) = \min\{d(x, y) : x \in \mathcal{C},\ y \in \mathcal{C},\ x \neq y\}.$$

- An (n, M, d)-code is a code $\mathcal{C}$ with M words of length n and minimum distance d.

- For example, the spacecraft Mariner 9 used an $(32, 64, 16)$ binary code.

- The **minimum distance** $d(\mathcal{C})$ of a code $\mathcal{C}$ is the minimum distance between distinct codewords, e. i.,

$$d(\mathcal{C}) = \min\{d(x, y) : x \in \mathcal{C},\ y \in \mathcal{C},\ x \neq y\}.$$

- An (n, M, d)-code is a code $\mathcal{C}$ with M words of length n and minimum distance d.

- For example, the spacecraft Mariner 9 used an $(32, 64, 16)$ binary code.

- $C = \{(0000000), (0001111), (0010101), (0011010), (0100110), (0101001), (0110011), (0111100), (1000011), (1001100), (1010110), (1011001), (1100101), (1101010), (1110000), (1111111)\}.$

  is an $(7, 16, 3)$ binary code.

A measure for the error-correcting capability of a linear code is the minimum distance

A measure for the error-correcting capability of a linear code is the minimum distance

Theorem

A code $\mathcal{C}$ with minimum distance d can:

(i) *detect up to $d-1$ errors;*

(ii) *correct up to $\lfloor \frac{d-1}{2} \rfloor$ errors.*

A measure for the error-correcting capability of a linear code is the minimum distance

Theorem

A code $\mathcal{C}$ with minimum distance d can:

(i) *detect up to $d-1$ errors;*

(ii) *correct up to $\lfloor \frac{d-1}{2} \rfloor$ errors.*

Proof

A measure for the error-correcting capability of a linear code is the minimum distance

Theorem

A code $\mathcal{C}$ with minimum distance d can:

(i) *detect up to $d-1$ errors;*

(ii) *correct up to $\lfloor \frac{d-1}{2} \rfloor$ errors.*

Proof

(i) Assume that a codeword x is sent and a vector y is received with up to $d-1$ errors. Then y can not be a codeword because the minimum distance of $\mathcal{C}$ is d and

$$d(x, y) \leq d - 1 < d(\mathcal{C}).$$

Thus, the transmission errors has been detected.

A measure for the error-correcting capability of a linear code is the minimum distance

Theorem

A code $\mathcal{C}$ with minimum distance d can:

(i) *detect up to $d-1$ errors;*

(ii) *correct up to $\lfloor \frac{d-1}{2} \rfloor$ errors.*

Proof

(ii) Let $t = \lfloor \frac{d-1}{2} \rfloor$ and assume that a codeword x is sent and a vector y is received with up to t errors. Then $d(x, y) \leq t$.

A measure for the error-correcting capability of a linear code is the minimum distance

Theorem

A code $\mathcal{C}$ with minimum distance d can:

(i) *detect up to $d-1$ errors;*

(ii) *correct up to $\lfloor \frac{d-1}{2} \rfloor$ errors.*

Proof

(ii) Let $t = \lfloor \frac{d-1}{2} \rfloor$ and assume that a codeword x is sent and a vector y is received with up to t errors. Then $d(x,y) \leq t$. If z is another codeword, since

$$d(x,z) \leq d(x,y) + d(y,z)$$

then

$$d(y,z) \geq d(x,z) - d(x,y) \geq d - t > t$$

and therefore x is the closest codeword to y.

Nearest Neighbor Decoding

To decode y as the codeword x such that $d(y, x)$ is the minimum possible we have to assure that:

- each symbol has the same probability p of been transmitted with an error;
- if a symbol is received with an error, all of the remaining symbols have the same probability to appear as the error.

Linear codes

- A **linear code** $\mathcal{C}$ (over the aphabet $\mathbb{F}_q$) is a linear subspace of $\mathbb{F}_q^n$.

Linear codes

- A **linear code** $\mathcal{C}$ (over the aphabet $\mathbb{F}_q$) is a linear subspace of $\mathbb{F}_q^n$.
- We say tha n is the **length** of the code and $\dim \mathcal{C}$ is the **dimension** of the code (as vectorial subspace over $\mathbb{F}_q$).

Linear codes

- A **linear code** $\mathcal{C}$ (over the aphabet $\mathbb{F}_q$) is a linear subspace of $\mathbb{F}_q^n$.
- We say tha n is the **length** of the code and $\dim \mathcal{C}$ is the **dimension** of the code (as vectorial subspace over $\mathbb{F}_q$).
- An $[n, k]-$code $\mathcal{C}$ is a linear code of length n and dimension k.

Linear codes

- A **linear code** $\mathcal{C}$ (over the aphabet $\mathbb{F}_q$) is a linear subspace of $\mathbb{F}_q^n$.
- We say tha n is the **length** of the code and $\dim \mathcal{C}$ is the **dimension** of the code (as vectorial subspace over $\mathbb{F}_q$).
- An $[n,k]-$code $\mathcal{C}$ is a linear code of length n and dimension k. If d is the minimum distance of $\mathcal{C}$ we say that $\mathcal{C}$ is am $[n,k,d]-$code.
- Each codeword of $\mathcal{C}$ has k information symbols and $n-k$ redundant symbols: k/n is called the information rate of the code $\mathcal{C}$.

Linear codes

- A **linear code** $\mathcal{C}$ (over the aphabet $\mathbb{F}_q$) is a linear subspace of $\mathbb{F}_q^n$.
- We say tha n is the **length** of the code and $\dim \mathcal{C}$ is the **dimension** of the code (as vectorial subspace over $\mathbb{F}_q$).
- An $[n, k]-$code $\mathcal{C}$ is a linear code of length n and dimension k. If d is the minimum distance of $\mathcal{C}$ we say that $\mathcal{C}$ is am $[n, k, d]-$code.
- Each codeword of $\mathcal{C}$ has k information symbols and $n - k$ redundant symbols: k/n is called the information rate of the code $\mathcal{C}$.
- One of the main goals of the theory of error correcting codes is to construct *good codes*, i.e., codes with good parameters, maximizing k/n and d/n.

Proposition (Singleton Bound)

For an $[n, k, d]$ *code* $\mathcal{C}$ *holds*

$$k + d \leq n + 1.$$

Proposition (Singleton Bound)

For an $[n, k, d]$ *code* $\mathcal{C}$ *holds*

$$k + d \leq n + 1.$$

- Codes with $k + d = n + 1$ are in some sense optimal; such codes are called MDS codes (maximum distance separables).

Proposition (Singleton Bound)

For an $[n, k, d]$ *code* $\mathcal{C}$ *holds*

$$k + d \leq n + 1.$$

- Codes with $k + d = n + 1$ are in some sense optimal; such codes are called MDS codes (maximum distance separables).
- In general is hard to obtain non trivial lower bounds for a minimum distance of a given code or a given class of codes.

Proposition (Singleton Bound)

For an $[n, k, d]$ *code* $\mathcal{C}$ *holds*

$$k + d \leq n + 1.$$

Proposition (Singleton Bound)

For an $[n, k, d]$ *code* $\mathcal{C}$ *holds*

$$k + d \leq n + 1.$$

Proof

Proposition (Singleton Bound)

For an $[n, k, d]$ *code* $\mathcal{C}$ *holds*

$$k + d \leq n + 1.$$

Proof

- Consider the linear subspace $E \subset \mathbb{F}_q^n$ given by

$$E = \{(a_1, \ldots, a_n) \in \mathbb{F}_q^n \; : \; a_i = 0 \text{ for all } i \geq d\}$$

Proposition (Singleton Bound)

For an $[n, k, d]$ *code* $\mathcal{C}$ *holds*

$$k + d \leq n + 1.$$

Proof

- Consider the linear subspace $E \subset \mathbb{F}_q^n$ given by

$$E = \{(a_1, \ldots, a_n) \in \mathbb{F}_q^n \ : \ a_i = 0 \text{ for all } i \geq d\}$$

  Every $a \in E$ has weight $w(a) \leq d-1$, hence $E \cap \mathcal{C} = \emptyset$.

Proposition (Singleton Bound)

For an $[n, k, d]$ *code* $\mathcal{C}$ *holds*

$$k + d \leq n + 1.$$

Proof

- Consider the linear subspace $E \subset \mathbb{F}_q^n$ given by

$$E = \{(a_1, \ldots, a_n) \in \mathbb{F}_q^n \ : \ a_i = 0 \text{ for all } i \geq d\}$$

Every $a \in E$ has weight $w(a) \leq d-1$, hence $E \cap \mathcal{C} = \emptyset$. As $\dim E = d-1$ we obtain

$$k+(d-1) = \dim \mathcal{C} + \dim E = \dim(\mathcal{C}+E) + \dim(\mathcal{C} \cap E) = \dim(E+\mathcal{C}) \leq n.$$

Generator matrix of a code $\mathcal{C}$

Generator matrix of a code $\mathcal{C}$

- If $\mathcal{B} = \{v_1, \ldots, v_k\}$ is a basis of an $[n, k]$ code $\mathcal{C}$, we define the generator matrix G of the code as the matrix $G_{k\times n}$ for which the rows are the vectors v_i of the base.

Generator matrix of a code $\mathcal{C}$

- If $\mathcal{B} = \{v_1, \ldots, v_k\}$ is a basis of an $[n, k]$ code $\mathcal{C}$, we define the generator matrix G of the code as the matrix $G_{k \times n}$ for which the rows are the vectors v_i of the base.

- G depend on the basis.

Generator matrix of a code $\mathcal{C}$

- If $\mathcal{B} = \{v_1, \ldots, v_k\}$ is a basis of an $[n, k]$ code $\mathcal{C}$, we define the generator matrix G of the code as the matrix $G_{k\times n}$ for which the rows are the vectors v_i of the base.
- G depend on the basis.
- Two equivalent matrixes define the same code.

Generator matrix of a code $\mathcal{C}$

- If $\mathcal{B} = \{v_1, \dots, v_k\}$ is a basis of an $[n, k]$ code $\mathcal{C}$, we define the generator matrix G of the code as the matrix $G_{k \times n}$ for which the rows are the vectors v_i of the base.
- G depend on the basis.
- Two equivalent matrixes define the same code.
- We shall say that G is in standard form (often called reduced echelon form) if

$$G = (I_k | A)$$

  where I_k is the $k \times k$ identity matrix and A is $k \times n - k$.

Coding and decoding

Coding and decoding

For an $[n, k]$ code $\mathcal{C}$ over $\mathbb{F}_q$ we can encode using the generator matrix G:

$$\begin{array}{ccc} \mathbb{F}_q^k & \longrightarrow & \mathbb{F}_q^n \\ u & \rightarrow & c = uG \end{array}$$

Coding and decoding

For an $[n, k]$ code $\mathcal{C}$ over $\mathbb{F}_q$ we can encode using the generator matrix G:

$$\begin{array}{ccc} \mathbb{F}_q^k & \longrightarrow & \mathbb{F}_q^n \\ u & \rightarrow & c = uG \end{array}$$

- Example: Mariner 9

$$u = (a_1, a_2, \ldots, a_6) \longrightarrow c = (c_1, c_2, \ldots, c_{32})$$

Coding and decoding

For an $[n, k]$ code $\mathcal{C}$ over $\mathbb{F}_q$ we can encode using the generator matrix G:

$$\begin{array}{ccc} \mathbb{F}_q^k & \longrightarrow & \mathbb{F}_q^n \\ u & \to & c = uG \end{array}$$

- Example: Mariner 9

$$u = (a_1, a_2, \ldots, a_6) \longrightarrow c = (c_1, c_2, \ldots, c_{32})$$

- The $[4, 2]$ binary code $\mathcal{C}$ generating by the matrix

$$G = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix}.$$

Coding and decoding

For an $[n, k]$ code $\mathcal{C}$ over $\mathbb{F}_q$ we can encode using the generator matrix G:

$$\begin{array}{ccc} \mathbb{F}_q^k & \longrightarrow & \mathbb{F}_q^n \\ u & \to & c = uG \end{array}$$

- Example: Mariner 9

$$u = (a_1, a_2, \ldots, a_6) \longrightarrow c = (c_1, c_2, \ldots, c_{32})$$

- The $[4, 2]$ binary code $\mathcal{C}$ generating by the matrix

$$G = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix}.$$

Then $\mathcal{C} = \{(0000), (1011), (0101), (1110)\}$.

Coding and decoding

For an $[n,k]$ code $\mathcal{C}$ over $\mathbb{F}_q$ we can encode using the generator matrix G:

$$\begin{array}{ccc} \mathbb{F}_q^k & \longrightarrow & \mathbb{F}_q^n \\ u & \rightarrow & c = uG \end{array}$$

- Example: Mariner 9

$$u = (a_1, a_2, \ldots, a_6) \longrightarrow c = (c_1, c_2, \ldots, c_{32})$$

- The $[4,2]$ binary code $\mathcal{C}$ generating by the matrix

$$G = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix}.$$

Then $\mathcal{C} = \{(0000), (1011), (0101), (1110)\}$.

If G is on standard form then decoding is trivial since

$$u \in \mathbb{F}_q^k \quad \longrightarrow \quad c = uG = (u|uA) \in \mathbb{F}_q^n \quad \longrightarrow \quad u = c_{|_{\mathbb{F}_q^k}} \in \mathbb{F}_q^k.$$