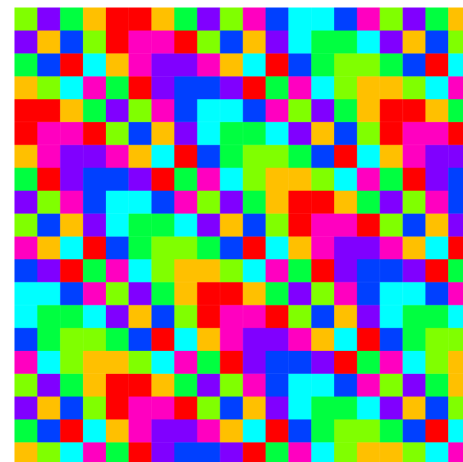


Codes correcteurs

Michel Rigo (Université de Liège)

<http://www.discmath.ulg.ac.be/mam/>



Welcome to a digital world !



0 et 1 suffisent pour *tout coder...*

$$237 = 2 \times 100 + 3 \times 10 + 7$$

Unités, dizaines, centaines, ...

Comment écrire les nombres entiers en *base 2*

0 et 1 suffisent pour *tout coder...*

$$237 = 2 \times 100 + 3 \times 10 + 7$$

Unités, dizaines, centaines, ...

Comment écrire les nombres entiers en *base 2*

$$100101 = 1 \times 2^5 + 1 \times 2^2 + 1$$

Unités, “deuxaines”, “quatraines”, ...

8	4	2	1	
			1	= 1
		1	0	= 2
		1	1	= 3
	1	0	0	= 4
	1	0	1	= 5
	1	1	0	= 6
	1	1	1	= 7
1	0	0	0	= 8
1	0	0	1	= 9
1	0	1	0	= 10

Quelles données ?

- textes (.doc, .rtf, SMS, html, RSS, ...)
- images, photos (.jpg, .png, .bmp, MMS, ...)
- sons, musique (CD, .wav, .mp3, ...)
- vidéos (DVD, .mov, .mpg, divx, .mp4, ...)
- séquence génétique (AGCTCCATGAGTC...)

CODAGE

Permettre l'échange et le stockage d'informations

Code ASCII *American Standard Code for Information Interchange*

1	33 !	65 A	97 a	129 Ì	161 ÿ	193 Á	225 á
2 ı	34 "	66 B	98 b	130 ,	162 ø	194 Â	226 â
3 ¸	35 #	67 C	99 c	131 f	163 £	195 Ã	227 ã
4 ¨	36 \$	68 D	100 d	132 „	164 ¨	196 Ä	228 ä
5	37 %	69 E	101 e	133 ...	165 ¥	197 Å	229 å
6 -	38 &	70 F	102 f	134 †	166 †	198 Æ	230 æ
7 •	39 '	71 G	103 g	135 ‡	167 §	199 Ç	231 ç
8 ■	40 (	72 H	104 h	136 ^	168 ¨	200 È	232 è
9	41)	73 I	105 i	137 ‰	169 ©	201 É	233 é
10	42 *	74 J	106 j	138 Š	170 ¨	202 Ê	234 ê
11 º	43 +	75 K	107 k	139 <	171 «	203 Ë	235 ë
12 □	44 ,	76 L	108 l	140 Œ	172 ¬	204 Ì	236 ì
13	45 -	77 M	109 m	141 Ì	173 -	205 Í	237 í
14 ¢	46 .	78 N	110 n	142 Ž	174 ®	206 Î	238 î
15 ¤	47 /	79 O	111 o	143 Ì	175 ¨	207 Ĩ	239 ĩ
16 †	48 0	80 P	112 p	144 Ì	176 °	208 Ð	240 ð
17 ◀	49 1	81 Q	113 q	145 '	177 ±	209 Ñ	241 ñ
18 ↓	50 2	82 R	114 r	146 '	178 ²	210 Ò	242 ò
19 !!	51 3	83 S	115 s	147 "	179 ³	211 Ó	243 ó
20 ¶	52 4	84 T	116 t	148 "	180 ´	212 Ô	244 ô
21 ⊥	53 5	85 U	117 u	149 ¨	181 µ	213 Õ	245 õ
22 τ	54 6	86 V	118 v	150 -	182 ¶	214 Ö	246 ö
23 †	55 7	87 W	119 w	151 —	183 ·	215 ×	247 ÷
24 ↑	56 8	88 X	120 x	152 ~	184 ¨	216 Ø	248 ø
25 †	57 9	89 Y	121 y	153 ™	185 ¨	217 Ù	249 ù
26 →	58 :	90 Z	122 z	154 š	186 °	218 Ú	250 ú
27 ←	59 ;	91 [	123 {	155 >	187 »	219 Û	251 û
28	60 <	92 \	124	156 œ	188 ¼	220 Ü	252 ü
29	61 =	93]	125 }	157 Ì	189 ½	221 Ý	253 ý
30	62 >	94 ^	126 ~	158 ž	190 ¾	222 Þ	254 þ
31	63 ?	95 _	127 □	159 Ÿ	191 ¨	223 ß	255 ÿ
32	64 @	96 `	128 €	160	192 Ä	224 à	

Code ASCII
Savoir compter
de 0 à 255
en base 2...

65 A	97 a
66 B	98 b
67 C	99 c
68 D	100 d
69 E	101 e
70 F	102 f
71 G	103 g
72 H	104 h
73 I	105 i
74 J	106 j
75 K	107 k
76 L	108 l
77 M	109 m
78 N	110 n
79 O	111 o
80 P	112 p
81 Q	113 q
82 R	114 r
83 S	115 s
84 T	116 t
85 U	117 u

128	-64	-32	-16	-8	-4	-2	-1	
0	0	0	0	0	0	0	0	= 0
0	0	0	0	0	0	0	1	= 1
...								
0	1	0	0	0	0	1	0	= 66
...								
0	1	1	0	1	0	1	0	= 106
...								
1	1	1	1	1	1	1	1	= 255

Bonjour

66, 111, 110, 106, 111, 117, 114

01000010 01101111 01101110 01101010
01101111 01110101 01110010

Coder des images avec des 0 et des 1...



MOI
À
L'ÉCOLE

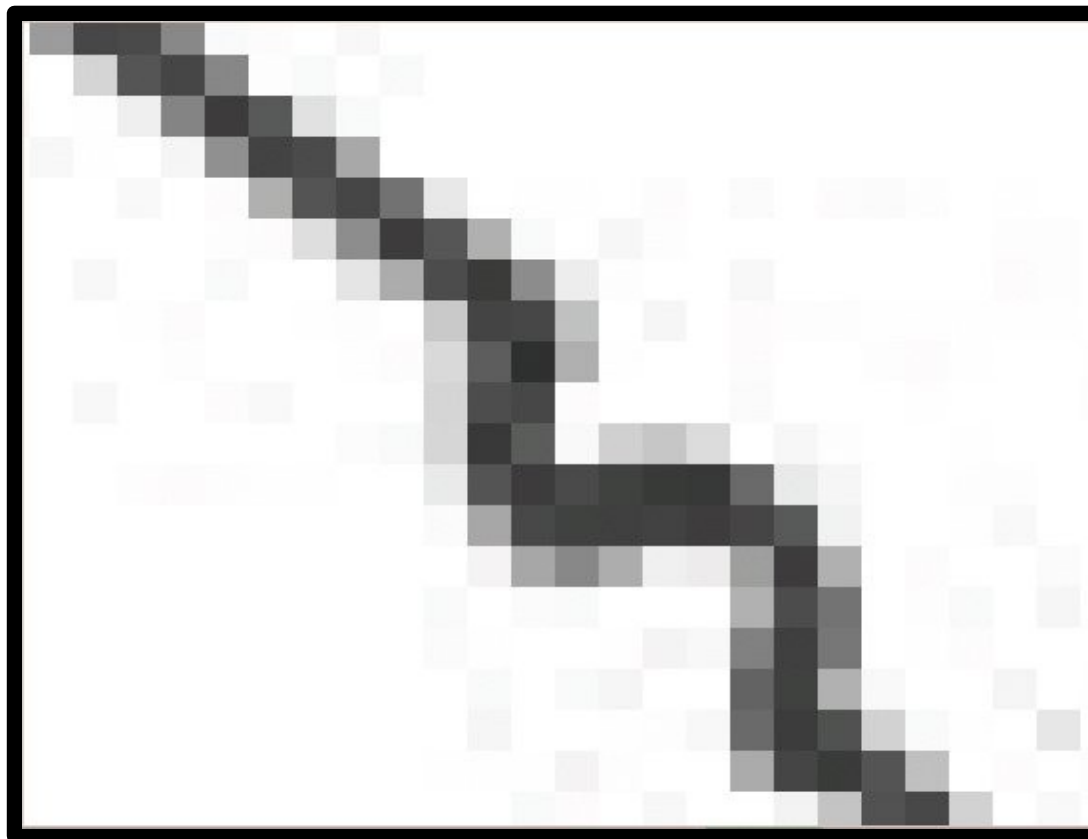
EN
ALGÈBRE

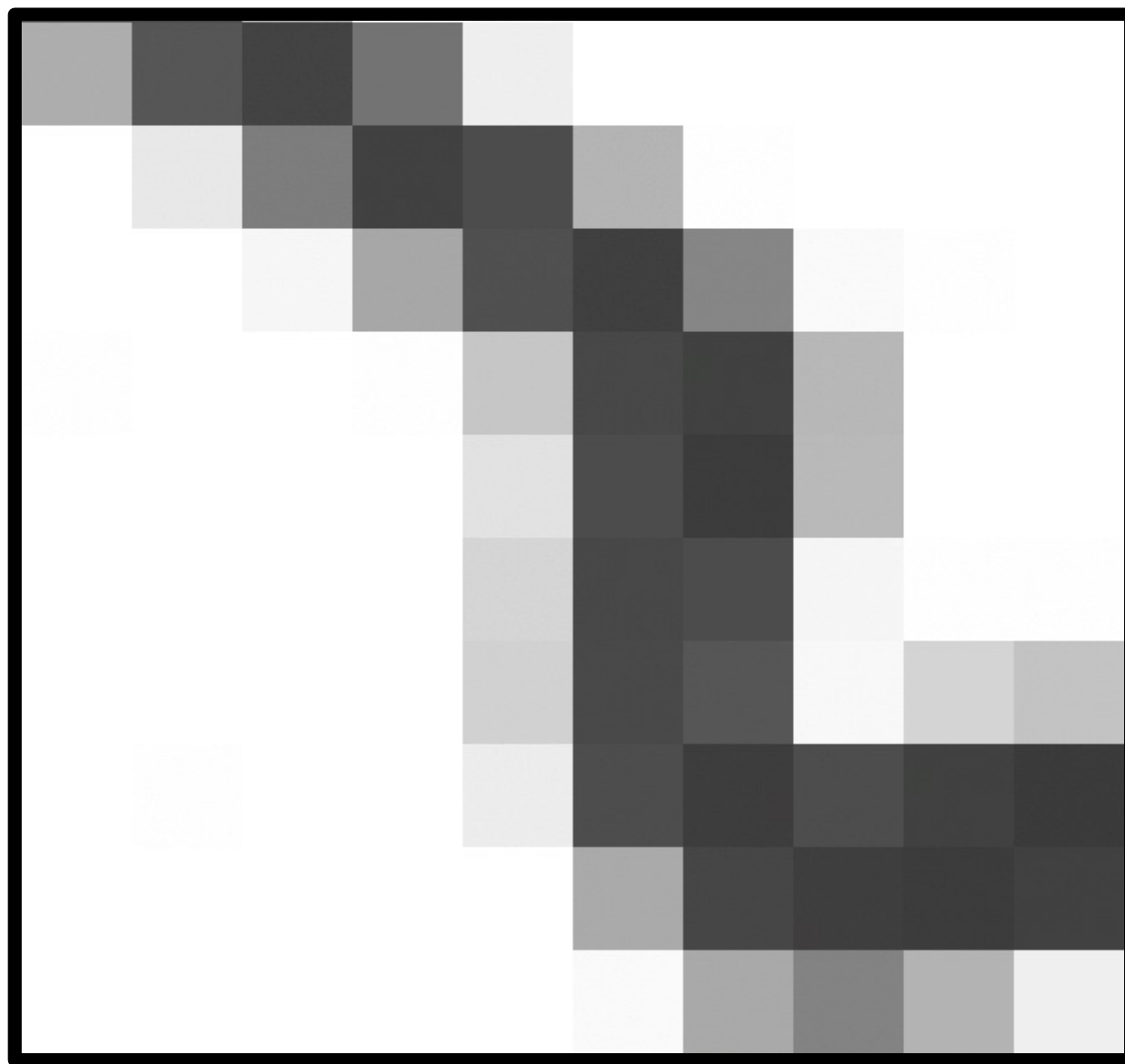




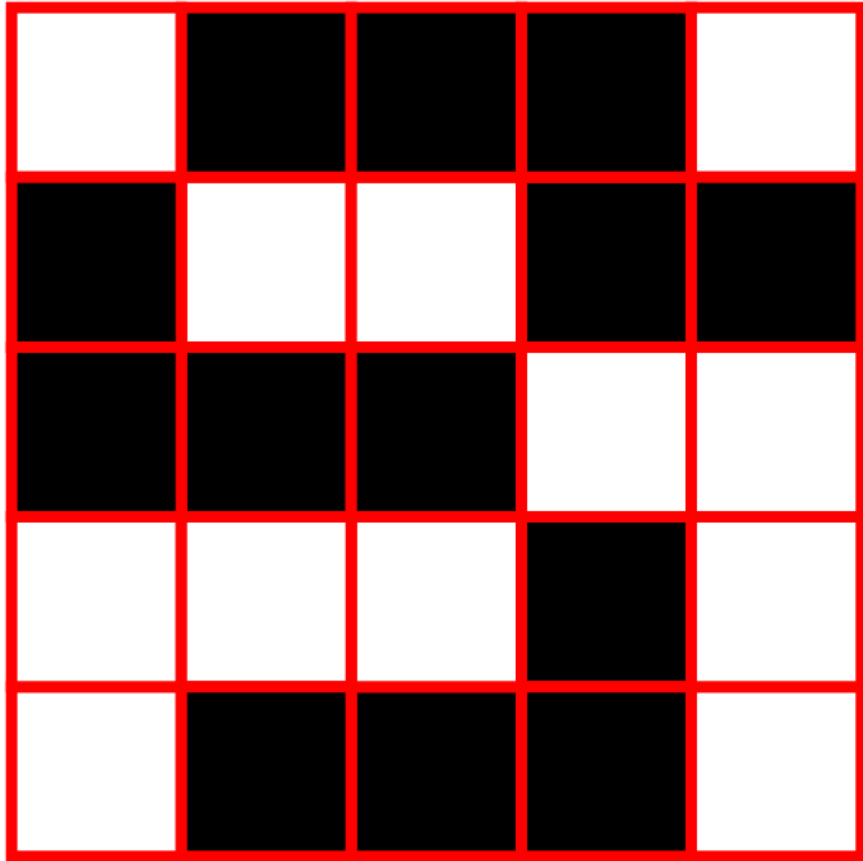




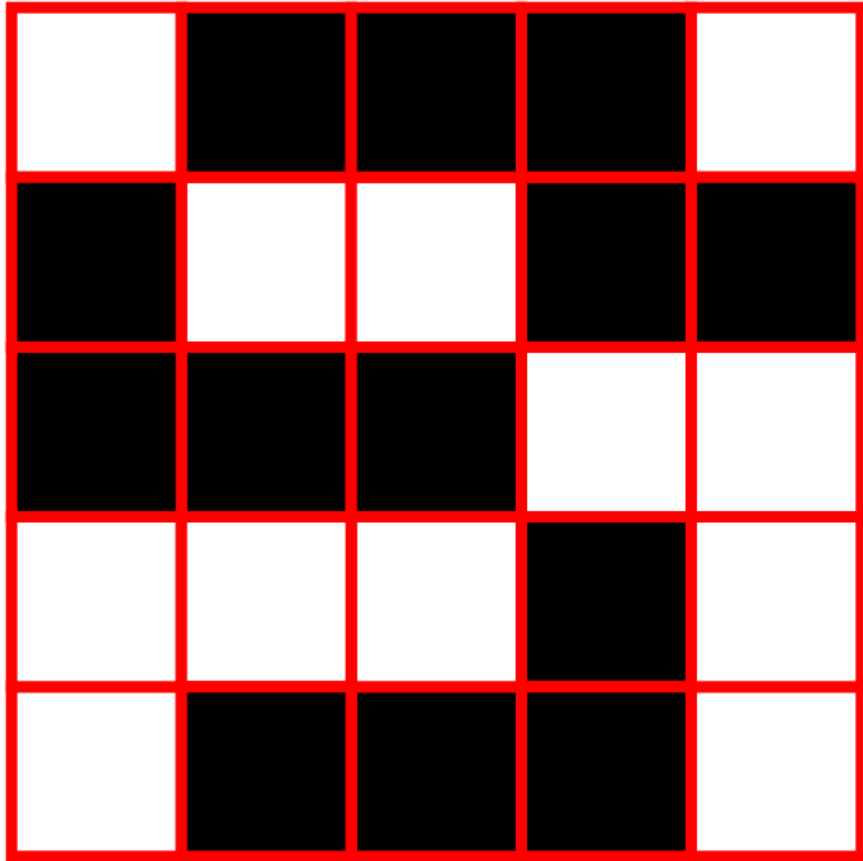




Images en noir et blanc

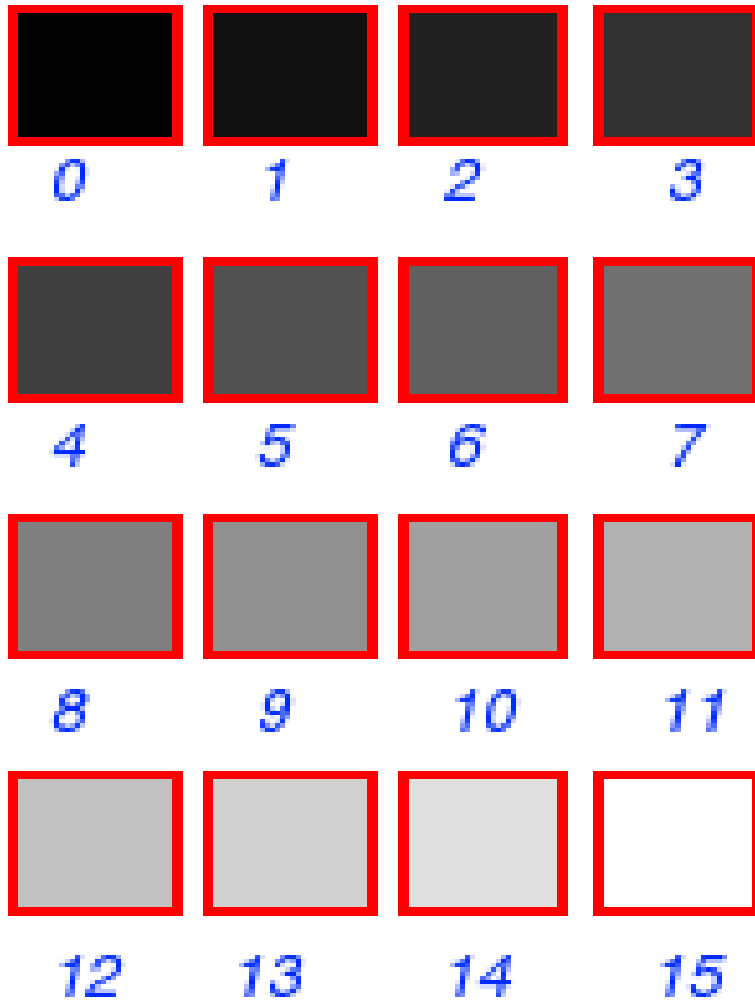


Images en noir et blanc



1	0	0	0	1
0	1	1	0	0
0	0	0	1	1
1	1	1	0	1
1	0	0	0	1

Images en 16 dégradés de gris



Images en 16 dégradés de gris



0

1

2

3

0000 0001 0010 0011



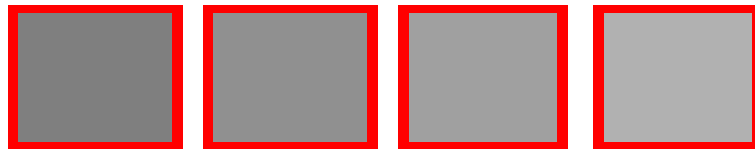
4

5

6

7

0100 0101 0110 0111



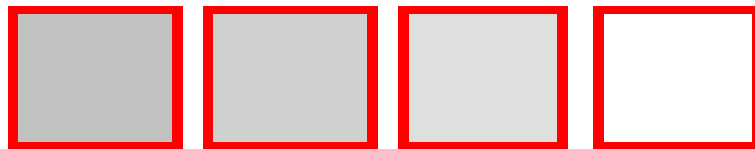
8

9

10

11

1000 1001 1010 1011



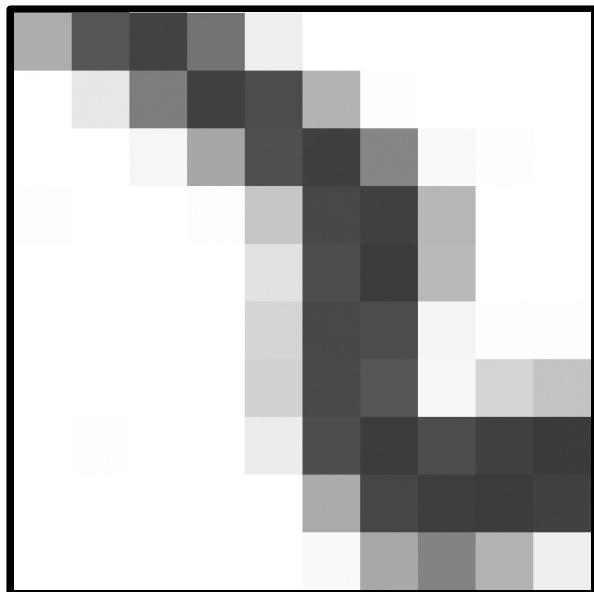
12

13

14

15

1100 1101 1110 1111



1011 0110 0100 1001 1110 1111 1111 1111 1111 1111
1111 1110 0111 0100 0101 1010 1111 1111 1111 1111
1111 1111 1110 1000 0101 0100 1001 1111 1111 1111
1111 1111 1111 1111 1010 0101 0100 1011 1111 1111
1111 1111 1111 1111 1011 0101 0011 1011 1111 1111
...



Et pour les images
en couleur ...



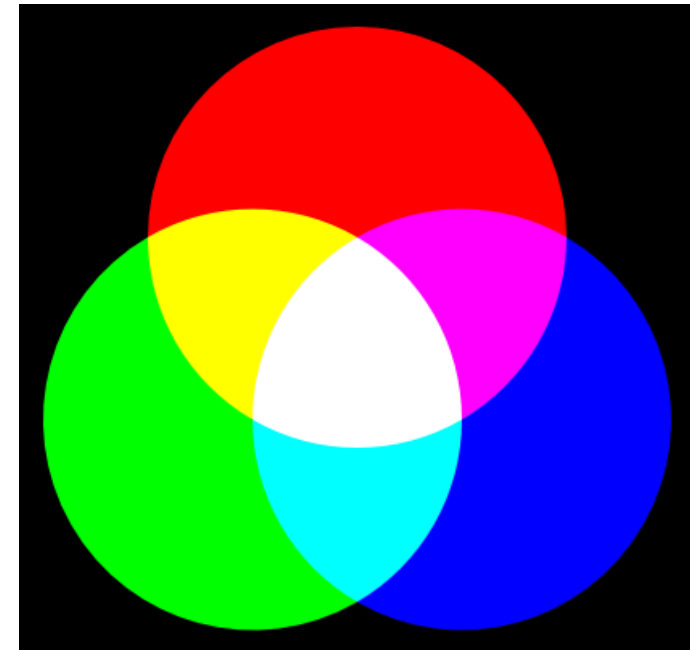
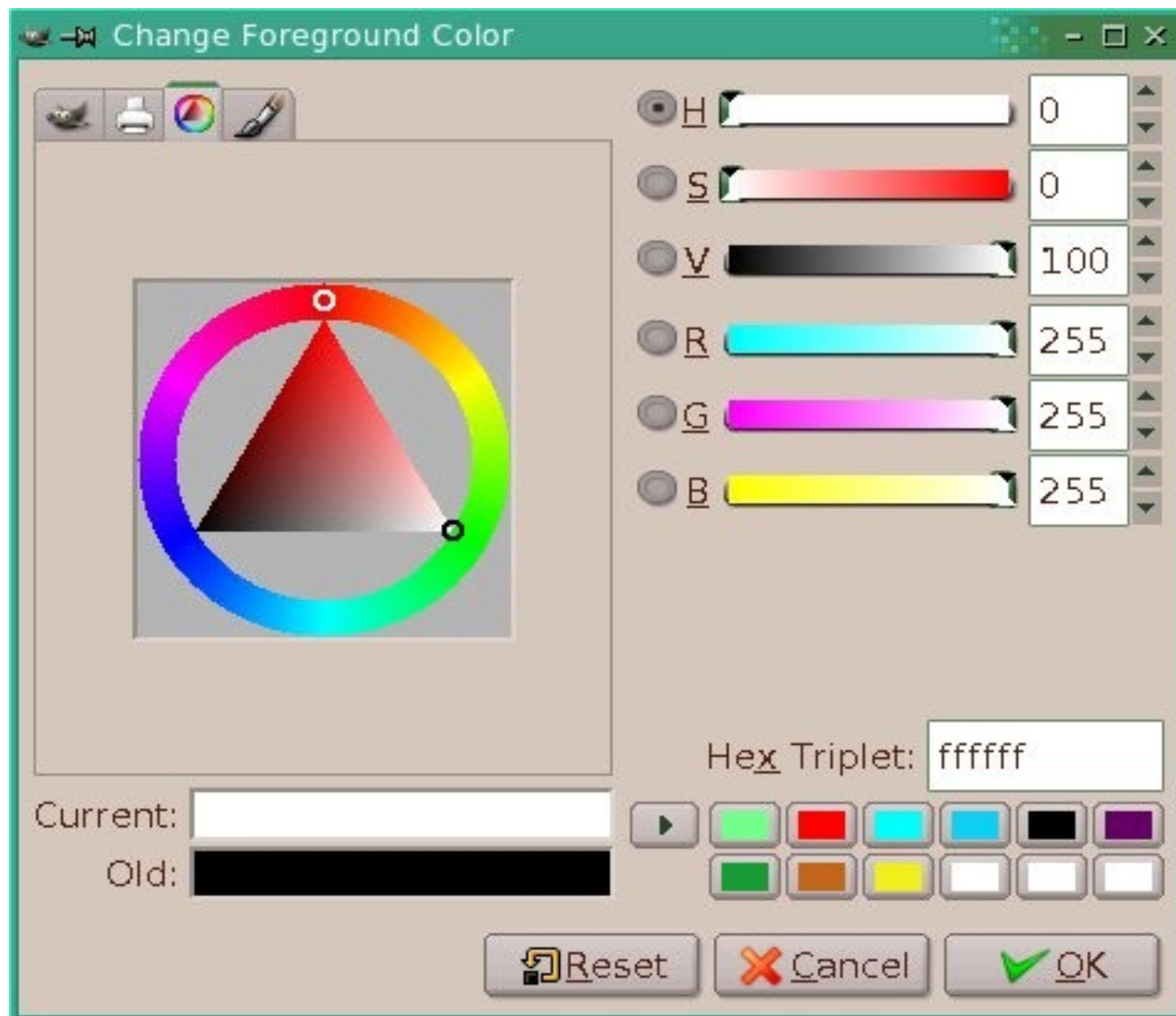
01101100 11001110
11000000 10101010
00010100 11001100 ...



00111100 01000111
01010101 00101111
11001110 10101100 ...



01111100 10001010
11010110 10101110
01110101 01011100 ...





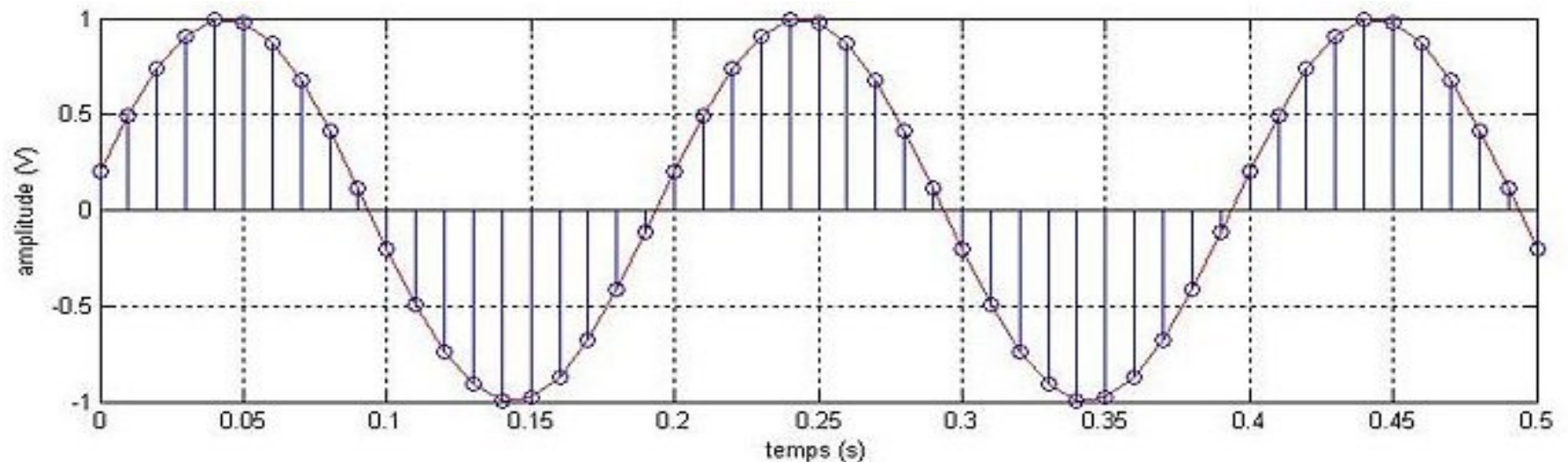
Pour les sons...

1) Son :

onde produite par la vibration mécanique d'un support et propagée grâce à l'élasticité du milieu sous forme d'onde longitudinale. L'air est un milieu compressible, le son se propage sous forme de variations de pression.

2) Membrane du **micro** vibre -> impulsions électriques
convertir les variations de pressions en signaux électriques

3) Echantillonneur analogique/digital : 0010010000111....





Pour les sons...

0010010000111....

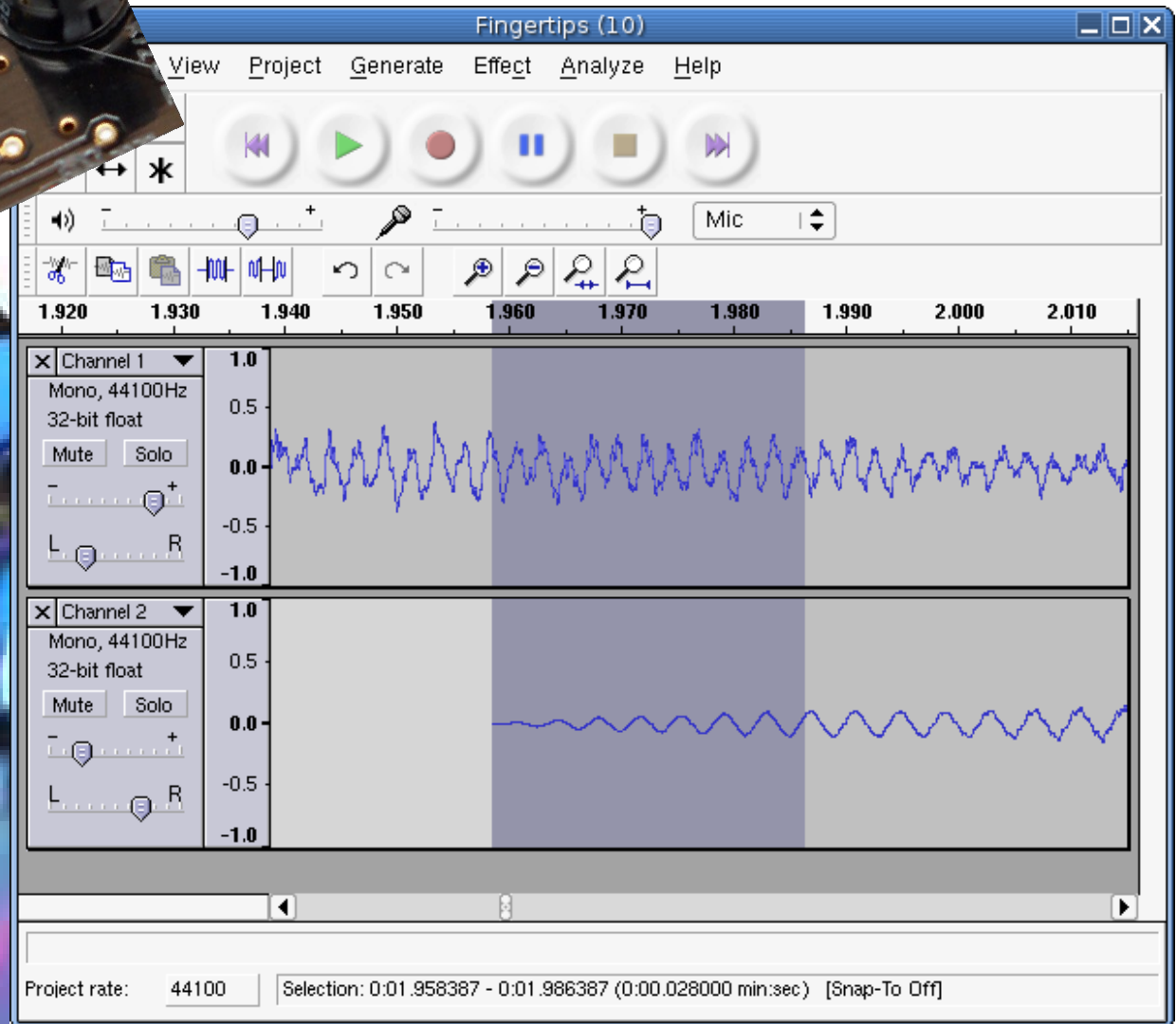
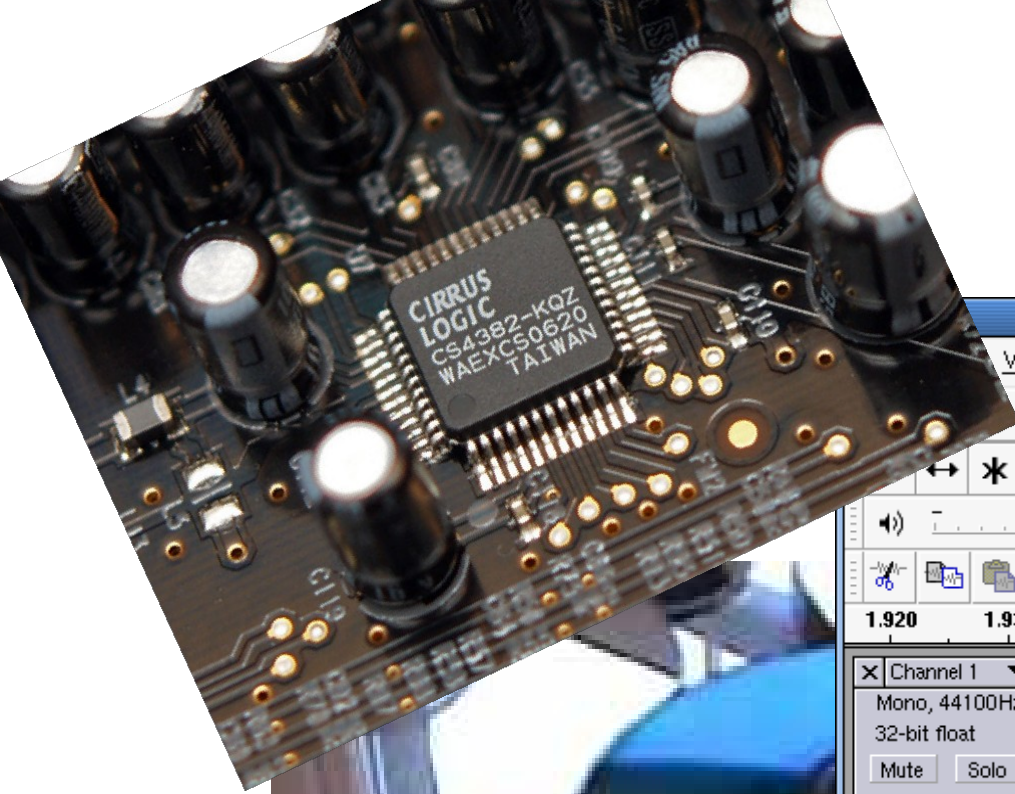
3) Convertisseur digital/analogique (DAC)

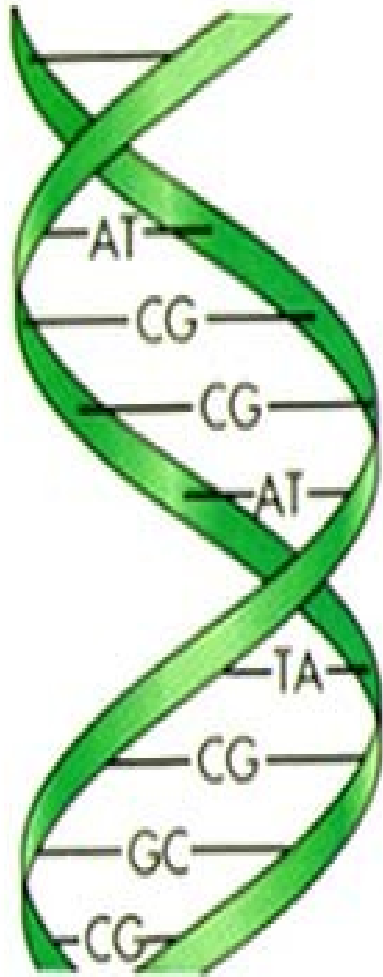
2) Haut-parleur

1) Son



Pour les sons...





Wikipédia :

molécule, retrouvée dans toutes les cellules vivantes, qui renferme l'ensemble des **informations** nécessaires au développement et au fonctionnement d'un organisme. L'ADN est aussi le support de l'hérédité car il est transmis lors de la reproduction, de manière intégrale ou non. Il porte donc l'**information génétique**, il constitue le génome des êtres vivants.

Adenine
Thymine
Guanine
Cytosine

En résumé, on code “tout” avec des suites de 0 et de 1...

Dans **un monde parfait**, il n'y a pas d'erreur !

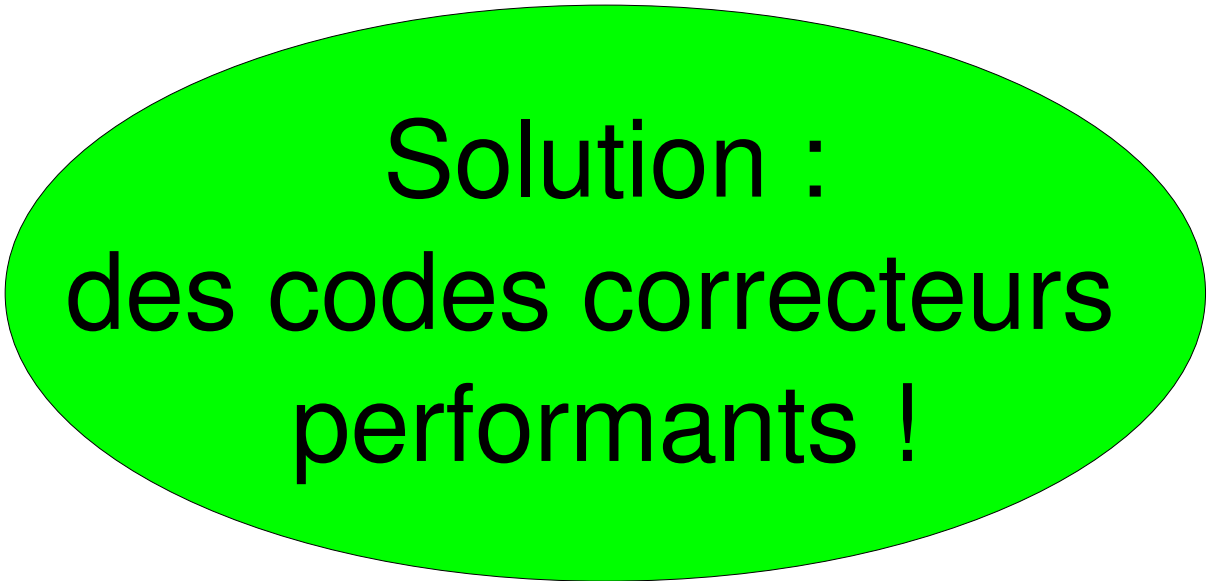
En résumé, on code “tout” avec des suites de 0 et de 1...

Dans **un monde parfait**, il n'y a pas d'erreur !



Dans **le monde réel** :

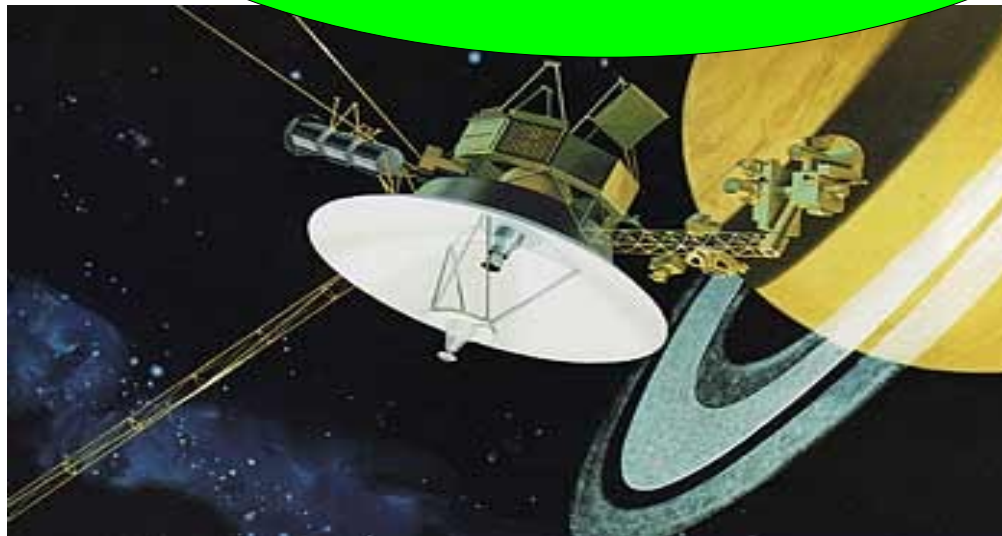
- interférences
ligne haute tension, orage, champ magnétique, ...
- grattes sur un CD, poussières, ...
- erreurs de lecture, d'écriture
- perte de puissance du signal sur longue distance
- mutation génétique

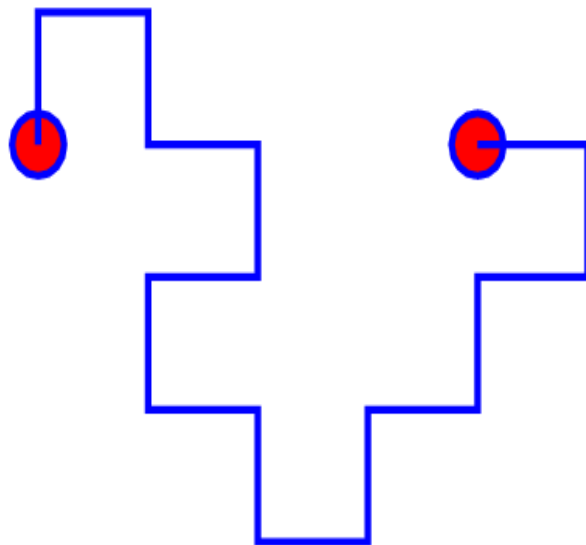


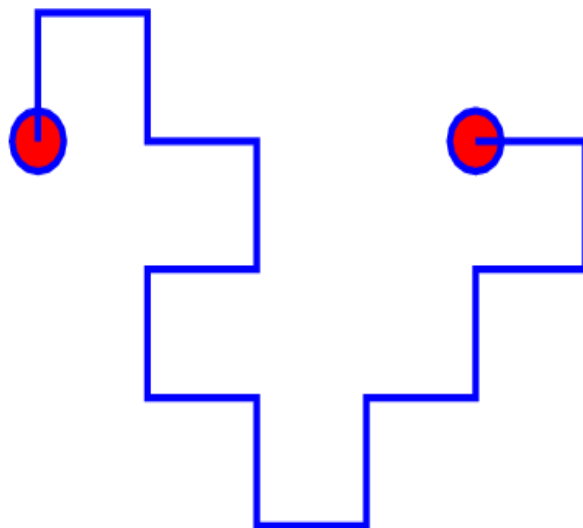
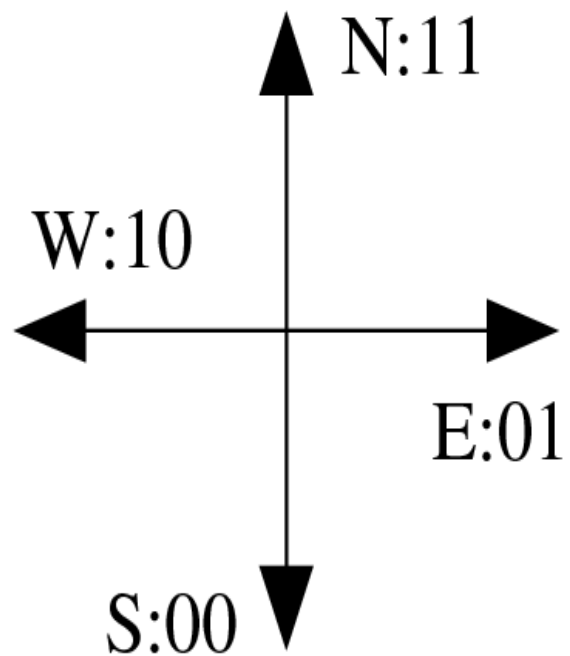
Solution :
des codes correcteurs
performants !



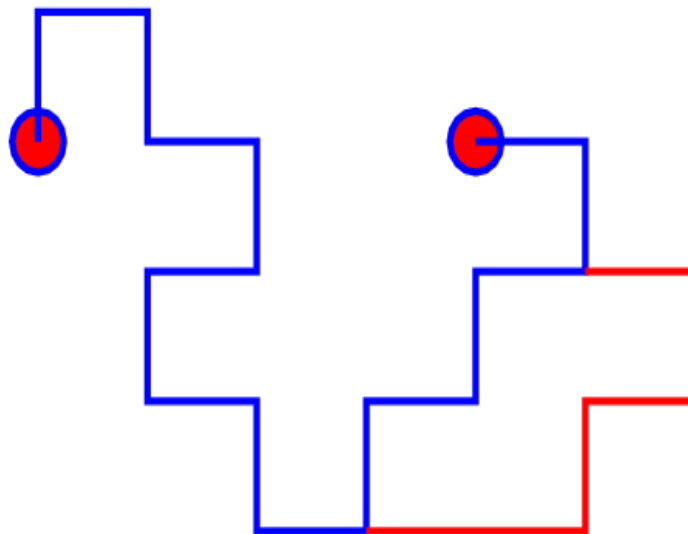
Solution :
des codes correcteurs
performants !







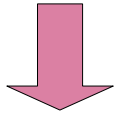
N E S E S W S E S E N E N E N W
 11010001001000010001110111011110
 11010001001000010001010111011110



101

Essai 1 : Doubler chaque chiffre

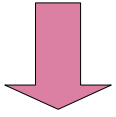
0 1 0 0 1 0 1 1 1 0



00 11 00 00 11 00 11 11 11 00

Essai 1 : Doubler chaque chiffre

0 1 0 0 1 0 1 1 1 0



00 11 00 00 11 00 11 11 11 00

Si une erreur se produit ?

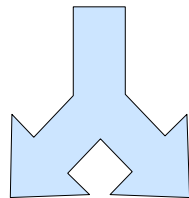
00 11 00 00 11 10 11 11 11 00

Essai 1 : Doubler chaque chiffre

0 1 0 0 1 0 1 1 1 0
↓
00 11 00 00 11 00 11 11 11 00

Si une erreur se produit ?

00 11 00 00 11 10 11 11 11 00



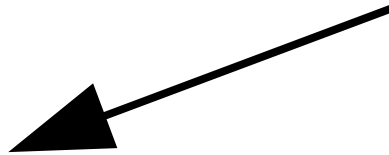
0 ou 1 ?

On sait détecter mais
PAS corriger !

Essai 2 : Bit de parité

0 1 0 0 1 0 1 1 1 0

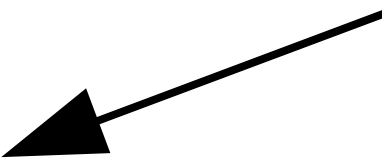
0 1 0 0 1 0 1 1 1 0 1



Essai 2 : Bit de parité

0 1 0 0 1 0 1 1 1 0

0 1 0 0 1 0 1 1 1 0 1

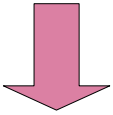


Si une erreur se produit ?

0 1 0 0 1 1 1 1 1 0 1

Essai 3 : Tripler chaque chiffre

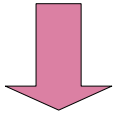
0 1 0 0 1 0 1 1 1 0



000 111 000 000 111 000 111 111 111 000

Essai 3 : Tripler chaque chiffre

0 1 0 0 1 0 1 1 1 0



000 111 000 000 111 000 111 111 111 000

Si une erreur se produit ?

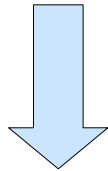
000 111 100 000 111 000 111 111 111 000

Essai 3 : Tripler chaque chiffre

0 1 0 0 1 0 1 1 1 0
↓
000 111 000 000 111 000 111 111 111 000

Si une erreur se produit ?

000 111 100 000 111 000 111 111 111 000

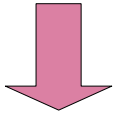


0

On sait détecter
ET corriger !

Essai 3 : Tripler chaque chiffre

0 1 0 0 1 0 1 1 1 0



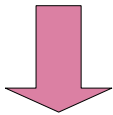
000 111 000 000 111 000 111 111 111 000

Si PLUS d'une erreur
se produit ?

000 111 100 000 111 001 111 111 111 000

Essai 3 : Tripler chaque chiffre

0 1 0 0 1 0 1 1 1 0



000 111 000 000 111 000 111 111 111 000

Si PLUS d'une erreur
se produit ?

000 111 100 000 111 001 111 111 111 000

000 111 000 000 111 000 111 100 111 000



0

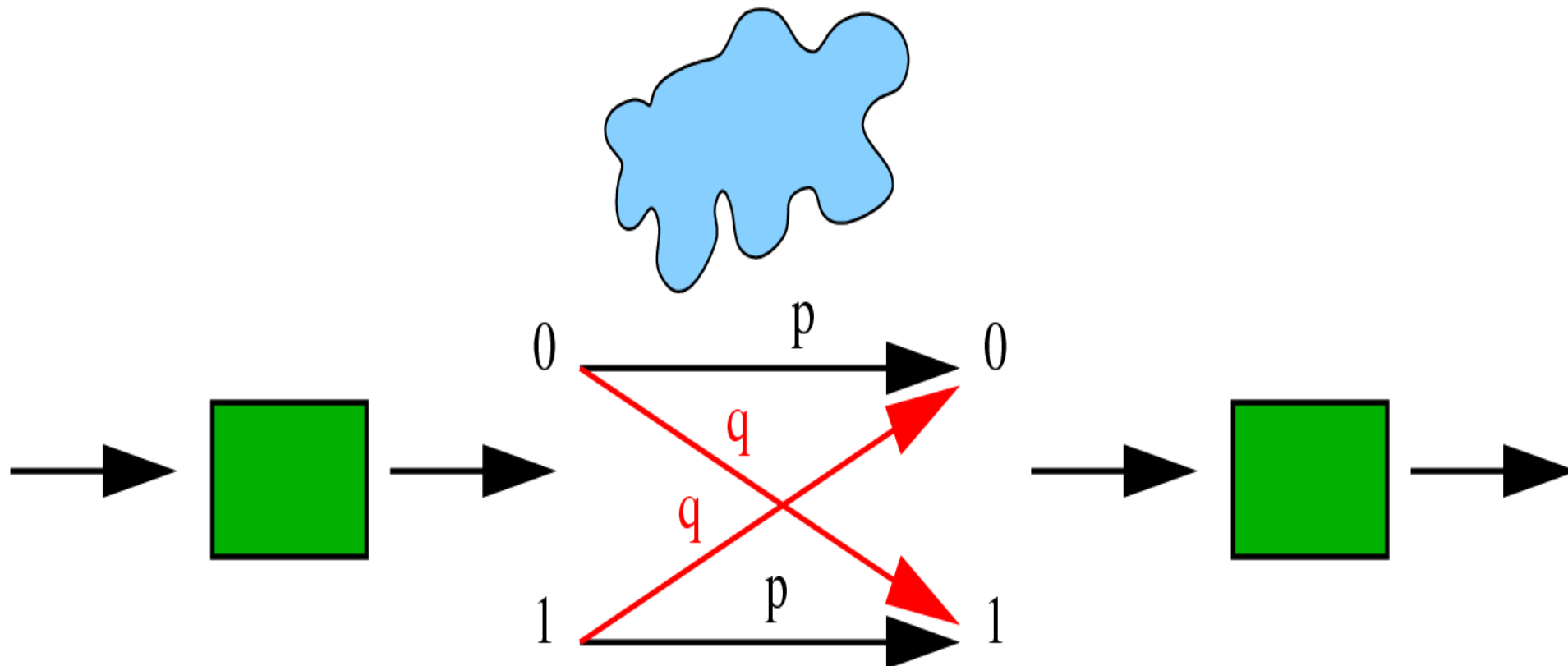
On peut corriger 1 erreur
et même,
un nombre arbitraire d'erreurs
si elles sont dans des *triplets distincts*

seul $1/3$ des chiffres contient
de l'information...

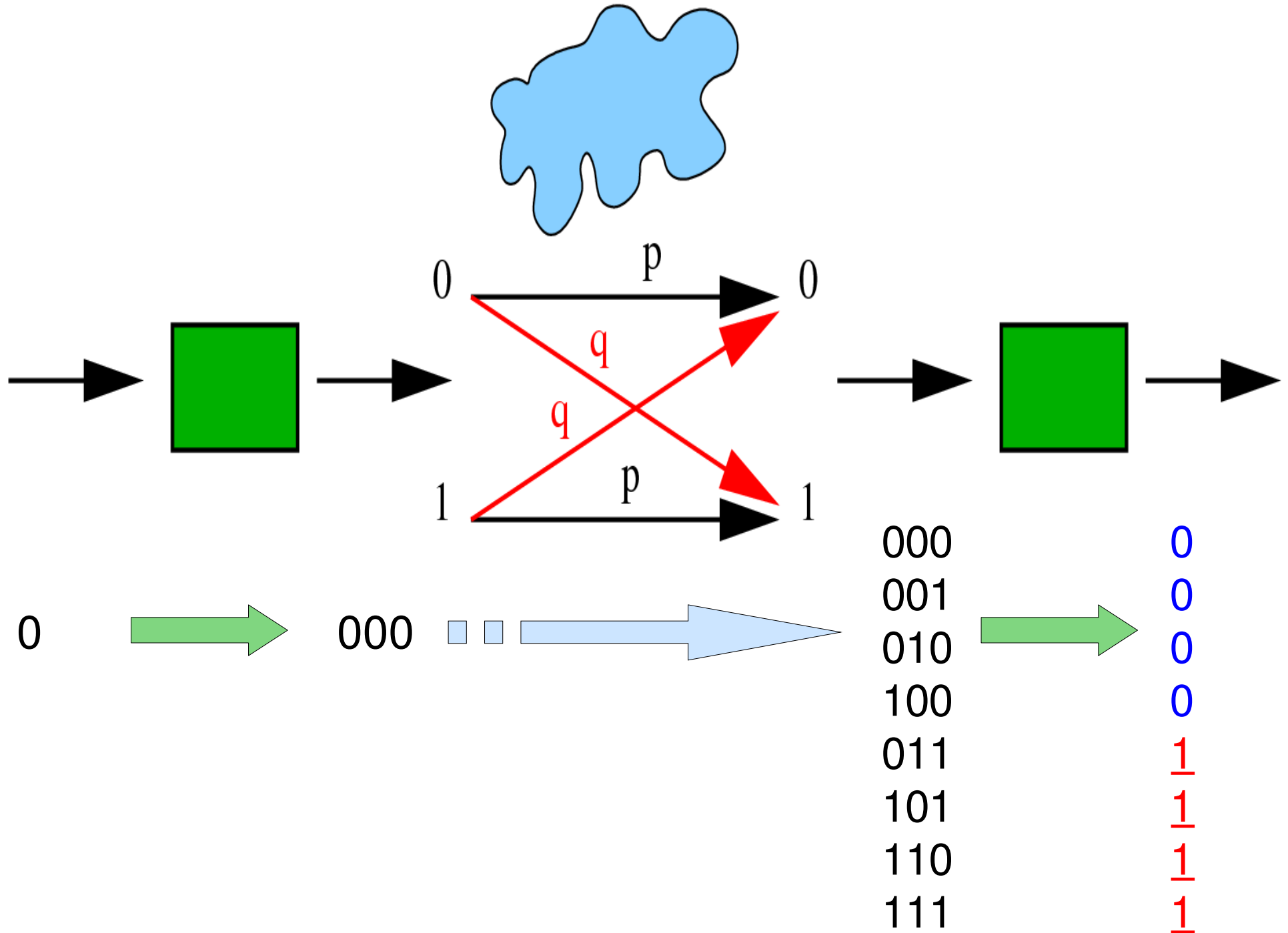
Moins de chansons
sur son iPod ;-(



Canal de communication



Canal de communication



Petit calcul de probabilités...

000	<u>0</u>	p^3
001	<u>0</u>	p^2q
010	<u>0</u>	p^2q
100	<u>0</u>	p^2q
011	<u>1</u>	pq^2
101	<u>1</u>	pq^2
110	<u>1</u>	pq^2
111	<u>1</u>	q^3

Petit calcul de probabilités...

000	0	p^3	$>$	$p^2(p+3q)=p^2(3-2p)$
001	0	p^2q		
010	0	p^2q		
100	0	p^2q		
011	<u>1</u>	pq^2		
101	<u>1</u>	pq^2		
110	<u>1</u>	pq^2		
111	<u>1</u>	q^3		

$$=0,972 \text{ si } p=0,9$$

décoder correctement 1 chiffre en en transmettant 3 sur un canal bruité



Image 4 dégradés

00 01 10 11

$p=0,9$



```

/home/michel/PrintempsSciences09/code-mathematica.nb
File Edit Cell Format Input Kernel Find Window Help

In[22]:= g = Import["pic-snoopy.bmp", "BMP"];

In[23]:= d = Flatten[Map[PadLeft[IntegerDigits[#, 2], 2] &, Flatten[g[[1, 1]] - 1]];

In[25]:= Shallow[d]
Out[25]//Shallow=
{1, 1, 1, 1, 1, 1, 1, 1, 1, 1, <<24990>>}

In[26]:= c = Mod[d + Table[If[Random[Integer, {1, 10}] < 10, 0, 1], {i, 1, Length[d]}], 2];

In[27]:= Shallow[c]
Out[27]//Shallow=
{1, 1, 1, 0, 1, 1, 1, 1, 1, 1, <<24990>>}

In[28]:= g[[1, 1]] =
  Partition[Partition[Map[FromDigits[#, 2] + 1 &, Partition[c, 2]], Length[g[[1, 1, 1]]],
    Length[g[[1, 1]]]][[1]];

Export["pic-snoopy-bruit.bmp", g];

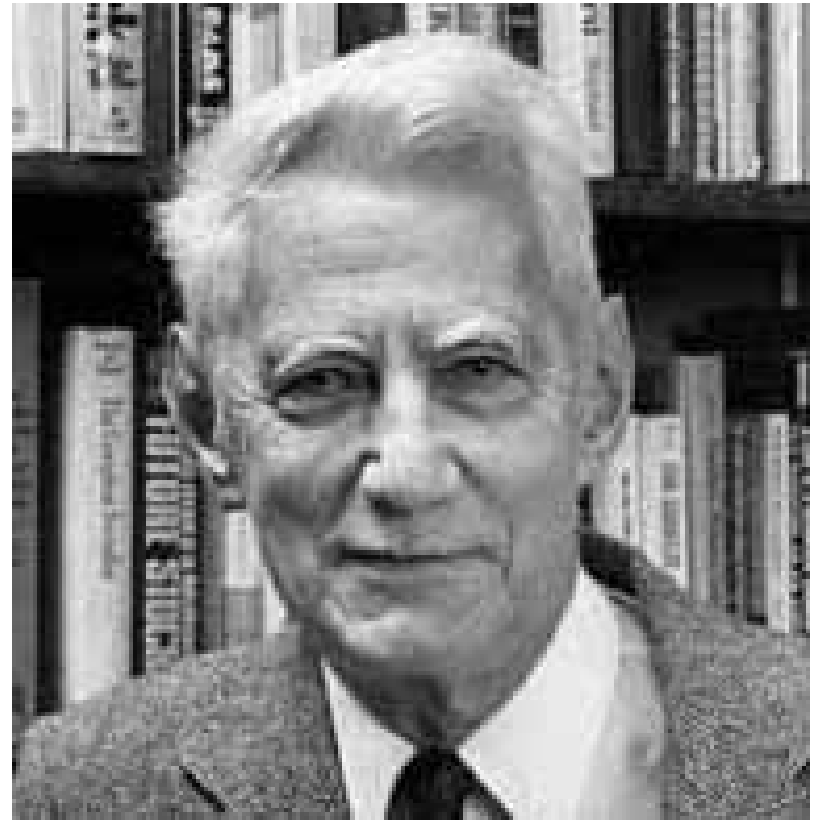
```



$p=0,9$



Avec codage
3 répétitions



Claude E. Shannon (1916-2001)

Théorie de l'information, théorèmes de Shannon (1948)

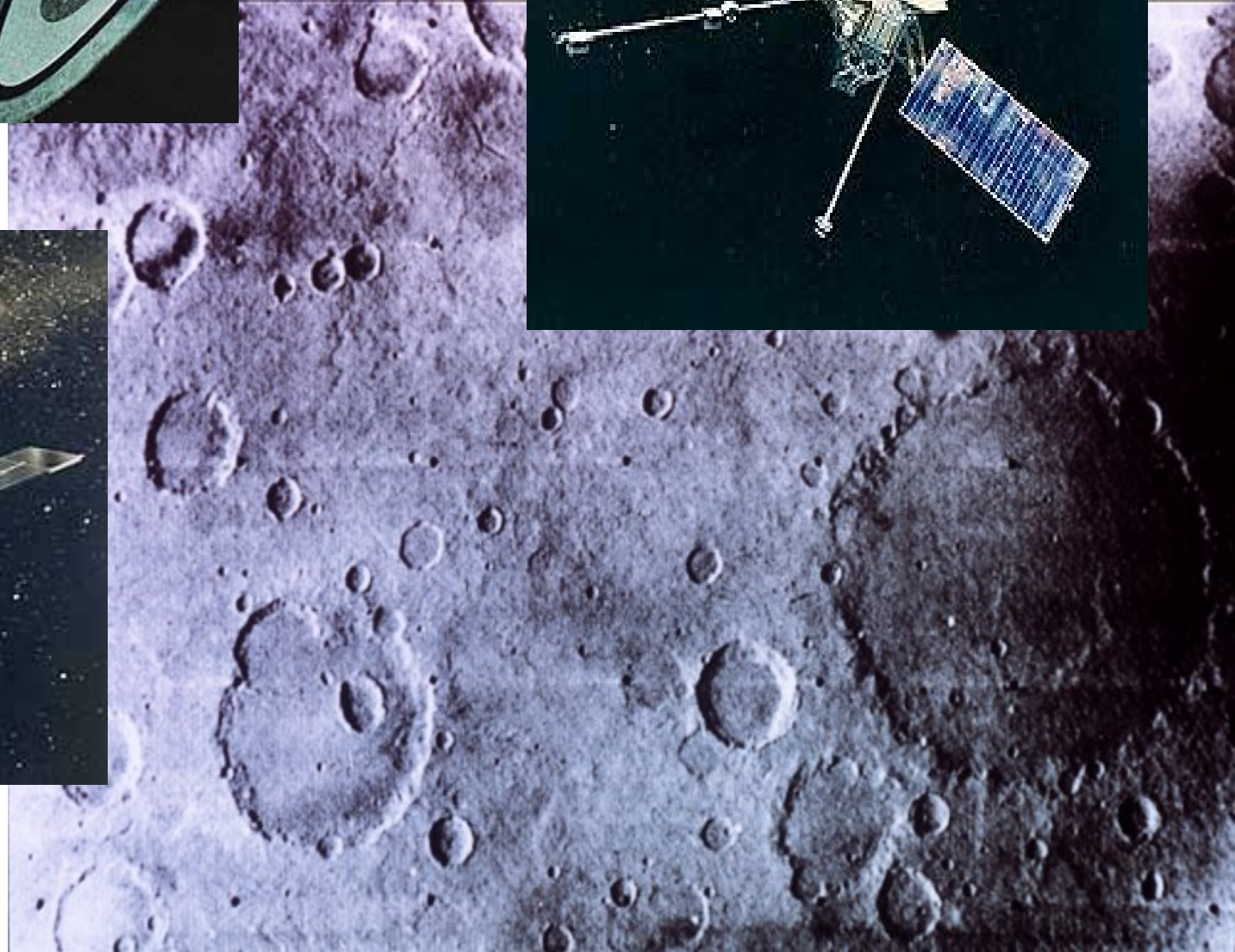
Voyager (1977) - Jupiter et Saturne



Mariner 10 (1973-75) –
Venus et Mercure



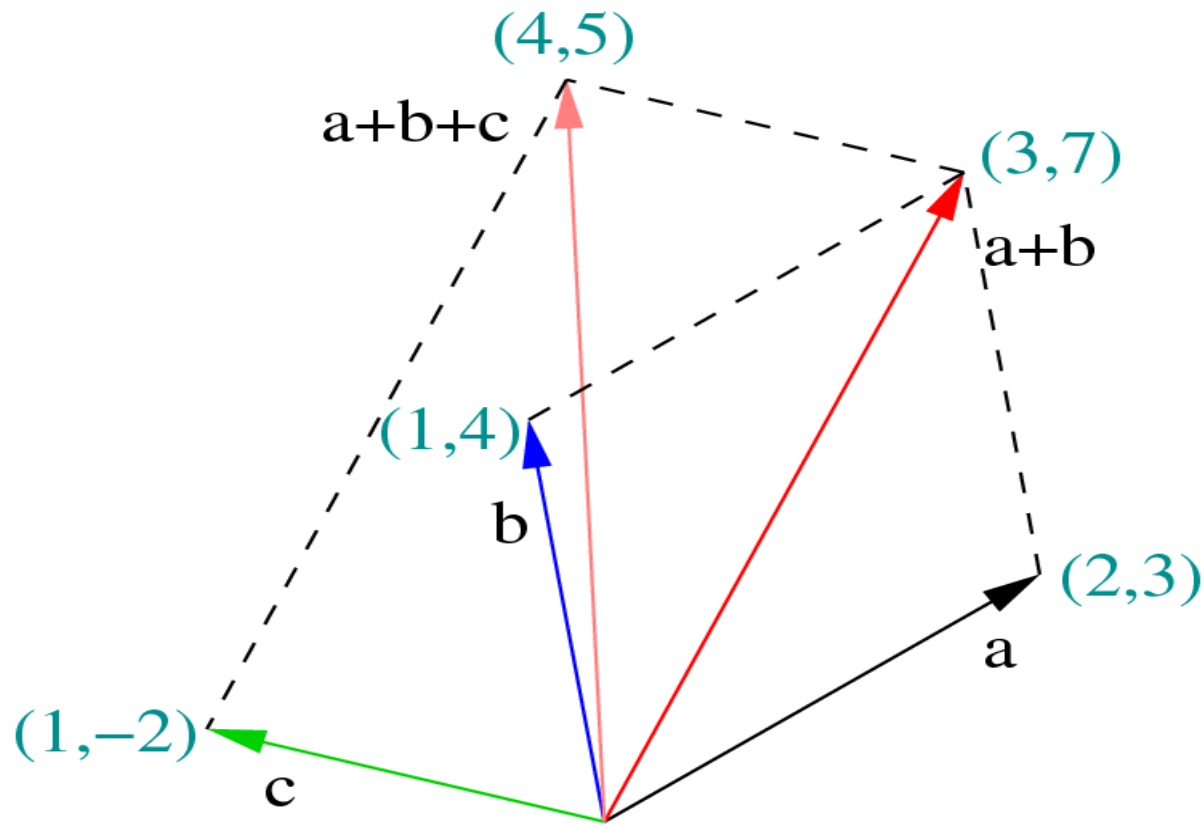
Mariner 2 (1962) - Venus



Un autre exemple de code correcteur : un code de Hamming



Richard Hamming



sous-espace vectoriel

$$a=(1, 0, 0, 0, 0, 1, 1)$$

$$b=(0, 1, 0, 0, 1, 0, 1)$$

$$c=(0, 0, 1, 0, 1, 1, 0)$$

$$d=(0, 0, 0, 1, 1, 1, 1)$$

$$a+b = ? \text{ si } 1+1=0$$

$a=(1, 0, 0, 0, 0, 1, 1)$

$b=(0, 1, 0, 0, 1, 0, 1)$

$c=(0, 0, 1, 0, 1, 1, 0)$

$d=(0, 0, 0, 1, 1, 1, 1)$

$(1, 1, 0, 0, 1, 1, 0)$ $a+b$

$(1, 0, 1, 0, 1, 0, 1)$ $a+c$

$(1, 0, 0, 1, 1, 0, 0)$ $a+d$

$(0, 1, 1, 0, 0, 1, 1)$ $b+c$

$(0, 1, 0, 1, 0, 1, 0)$ $b+d$

$(0, 0, 1, 1, 0, 0, 1)$ $c+d$

$(1, 1, 1, 0, 0, 0, 0)$ $a+b+c$

$(1, 1, 0, 1, 0, 0, 1)$ $a+b+d$

$(1, 0, 1, 1, 0, 1, 0)$ $a+c+d$

$(0, 1, 1, 1, 1, 0, 0)$ $b+c+d$

$(1, 1, 1, 1, 1, 1, 1)$ $a+b+c+d$

$(0, 0, 0, 0, 0, 0, 0)$

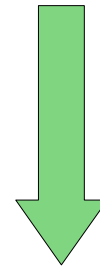
$a = (1, 0, 0, 0, 0, 1, 1)$
 $b = (0, 1, 0, 0, 1, 0, 1)$
 $c = (0, 0, 1, 0, 1, 1, 0)$
 $d = (0, 0, 0, 1, 1, 1, 1)$

(1, 1, 0, 0, 1, 1, 0)
(1, 0, 1, 0, 1, 0, 1)
(1, 0, 0, 1, 1, 0, 0)
(0, 1, 1, 0, 0, 1, 1)
(0, 1, 0, 1, 0, 1, 0)
(0, 0, 1, 1, 0, 0, 1)
(1, 1, 1, 0, 0, 0, 0)
(1, 1, 0, 1, 0, 0, 1)
(1, 0, 1, 1, 0, 1, 0)
(0, 1, 1, 1, 1, 0, 0)
(1, 1, 1, 1, 1, 1, 1)
(0, 0, 0, 0, 0, 0, 0)

Codage :

7 chiffres transmis pour
4 chiffres d'information,
taux $4/7$

0110



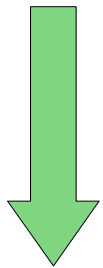
0110011

sous-espace vectoriel de
dimension 4 dans un espace
vectoriel de dimension 7

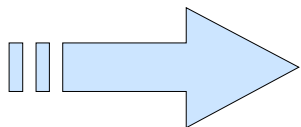
Décodage du code Hamming[7,4]

$$x=(0, 0, 0, 1, 1, 1, 1) \quad y=(0, 1, 1, 0, 0, 1, 1) \quad z=(1, 0, 1, 0, 1, 0, 1)$$

0110



0110011



0111011 = s

On calcule
3 produits scalaires

$$\begin{aligned} \langle s, x \rangle &= 1 & \langle s, y \rangle &= 0 & \langle s, z \rangle &= 0 \\ & & & & 100 \text{ en base } 2 : 4 \end{aligned}$$

THM : On peut détecter et corriger 1 erreur avec H[7,4]

Petit calcul de probabilités...

si $p=0,9$

décoder correctement l'info. 4 chiffres en en transmettant 7 sur un canal bruité

$$p^7 + 7p^6q = 0,850306$$

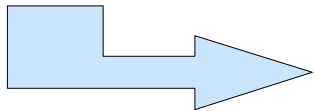
Hamming[7,4]

4/7

corrige 1 (100%) erreur

rappel : decoder correctement 1 chiffre en en transmettant 3 sur un canal bruité

$$p^2(3-2p)$$



$$p^3(3-2p)^4 = 0,892617$$

3 Répétitions

1/3

4 chiffres

en en transmettant 12

corrige 1 (100%), 2 (81%), 3 (49%), 4 (16%) erreurs

$$p^4 = 0,6561$$

Sans codage

1

ne corrige aucune erreur

Les erreurs peuvent se produire de manière isolée / par paquets

On peut ou non réenvoyer le message

S'adapter aux contraintes physiques (e.g. flux de données, etc.)

Construire des codes plus longs,
permettant de corriger plus d'erreurs :

Golay[23,12] peut corriger jusqu'à 3 erreurs, mieux que trois blocs
Hamming[7,4] qui ne corrigent pas tous les types d'au plus 3 erreurs,
code basé sur résidus quadratiques [47,24] corrige jusqu'à 5 erreurs

Codage et décodage efficaces

Encore bien d'autres choses...

Chiffrement des données, cryptographie
(sécurité, confidentialité, authentification)

Compression de données
(jpeg, mp3, zip, ...)

