

Département d'informatique et de génie logiciel
Compression de données
IFT-4003/IFT-7023

Notes de cours
Compression d'image sans perte

Édition Hiver 2012

Mohamed Haj Taieb

Local: PLT 2113

Courriel: mohamed.haj-taieb.1@ulaval.ca

Faculté des sciences et de génie
Département de génie électrique et de
génie informatique

Plan de la présentation

□ Compression d'image sans perte:

- Introduction
- JPEG
- CALIC
- JPEG-LS
- Approche multi-résolution
- Transmission progressive d'image
- Encodage Facsimile
- JBIG2-T.88
- Contenu raster mixte

Introduction

❑ Techniques de compression

- Ces techniques dépendent de l'application et des données à transmettre.
- → On s'intéresse ici à la compression d'image.

❑ Débalancement des probabilités

- Probabilités débalancées → plus de compression.
- Cas extrême: $P(1)=1$ et $P(0)=0$ → $H=0$.
- Transformer les données pour avoir plus de débalancement.
- Transformation réversible.

Exemple de la transformation des données

❑ Séquence initiale:

1	2	5	7	2	-2	0	-5	-3	-1	1	-2	-7	-4	-2	1	3	4
---	---	---	---	---	----	---	----	----	----	---	----	----	----	----	---	---	---

- $H_1 = 3.5305$ bits/symbole
- Encodage arithmétique ou de Huffman: utilisation de presque 4 bits/symbole.

❑ Transformation réversible:

- Ajouter deux au précédent comme prédiction puis prendre la différence.

1	-1	0	0	-7	-4	0	-7	0	0	0	-5	-7	1	0	1	0	-1
---	----	---	---	----	----	---	----	---	---	---	----	----	---	---	---	---	----

- $H_2 = 2.2585$ bits/symbole

JPEG (1)

□ Joint Photographic Experts Group

- Composante sans perte de JPEG.
- Comporte 8 schémas de prédiction.
 1. $\hat{I}(i,j) = 0$: pas de prédiction.
 2. $\hat{I}(i,j) = I(i-1,j)$: prédiction unidimensionnelle.
 3. $\hat{I}(i,j) = I(i,j-1)$: unidimensionnelle.
 4. $\hat{I}(i,j) = I(i-1,j-1)$: unidimensionnelle.
 5. $\hat{I}(i,j) = I(i,j-1) + I(i-1,j) - I(i-1,j-1)$: bidimensionnelle.
 6. $\hat{I}(i,j) = I(i,j-1) + [I(i-1,j) - I(i-1,j-1)]/2$: bidimensionnelle.
 7. $\hat{I}(i,j) = I(i-1,j) + [I(i,j-1) - I(i-1,j-1)]/2$: bidimensionnelle.
 8. $\hat{I}(i,j) = [I(i,j-1) + I(i-1,j)]/2$: bidimensionnelle.

JPEG (2)

□ Joint Photographic Experts Group

- Pour des applications non temps réel. Les 8 modes de prédiction peuvent être testés → choix du mode le plus efficace.
- L'encodage des images test avec les différents modes JPEG, génère des images résiduelles qui sont ensuite encodées avec un code arithmétique adaptative.

Image	JPEG 0	JPEG 1	JPEG 2	JPEG 3	JPEG 4	JPEG 5	JPEG 6	JPEG 7
Sena	53,431	37,220	31,559	38,261	31,055	29,742	33,063	32,179
Sensin	58,306	41,298	37,126	43,445	32,429	33,463	35,965	36,428
Earth	38,248	32,295	32,137	34,089	33,570	33,057	33,072	32,672
Omaha	56,061	48,818	51,283	53,909	53,771	53,520	52,542	52,189

JPEG (3)

□ JPEG vs GIF vs PNG

- PNG: prédiction sur chaque ligne + dictionnaire.
- GIF: compression basé sur les dictionnaires.
- Compression d'images → prédictif surpasse dictionnaire.
- Earth: La compression par dictionnaire donne des séquences longues car la le fond a une valeur constante.

Image	Best JPEG	GIF	PNG
Sena	31,055	51,085	31,577
Sensin	32,429	60,649	34,488
Earth	32,137	34,276	26,995
Omaha	48,818	61,341	50,185

CALIC

□ CALIC

- Context Adaptive Lossless Image Compression.
- Un pixel dans une image à niveau de gris est généralement proche d'une certaine valeur de son entourage: laquelle?
- Ceci dépend de la structure locale de l'image.
- Choix de la prédiction: pixel au dessus, ou pixel à gauche ou combinaison pondérée des pixels du voisinage.
- Ce choix dépend de l'existence de frontière ou de contour horizontal ou vertical dans le voisinage.

□ Étude de la texture locale

- La détermination de la texture doit se baser sur des pixels du voisinage disponible à l'encodeur et le décodeur.

Prédiction (1)

□ Détection de frontière

- Différence horizontale
- → frontière verticale
- $d_h = |W - WW| + |N - NW| + |NE - N|$.

		NN	NNE
	NW	N	NE
WW	W	X	

- Différence verticale
- → frontière horizontale
- $d_v = |W - NW| + |N - NN| + |NE - NNE|$.

		NN	NNE
	NW	N	NE
WW	W	X	

Prédiction (2)

```
if  $d_h - d_v > 80$  // sharp horizontal edge
     $X^* = N$ 
else if  $d_v - d_h > 80$  // sharp vertical edge
     $X^* = W$ 
else { // assume smoothness first
     $X^* = (N+W)/2 + (NE-NW)/4$ 
    if  $d_h - d_v > 32$  // horizontal edge|
         $X^* = (X^*+N)/2$ 
    else if  $d_v - d_h > 32$  // vertical edge
         $X^* = (X^*+W)/2$ 
    else if  $d_h - d_v > 8$  // weak horizontal edge
         $X^* = (3X^*+N)/4$ 
    else if  $d_v - d_h > 8$  // weak vertical edge
         $X^* = (3X^*+W)/4$ 
}
```

		NN	NNE
	NW	N	NE
WW	W	X	

Calcul de d_h

		NN	NNE
	NW	N	NE
WW	W	X	

Calcul de d_v

Prédiction (3)

		NN	NNE
	NW	N	NE
WW	W	X	

d

		200	200
	200	200	200
100	100	X	

Frontière horizontale brusque

X^*

W

80

		160	160
	160	160	160
100	100	X	

Frontière horizontale

$(t+W)/2$

32

		130	130
	130	130	130
100	100	X	

Frontière horizontale faible

$(3t+W)/4$

8

		100	100
	100	100	100
100	100	X	

Homogène

t

-8

		100	100
	130	100	100
130	130	X	

Frontière verticale faible

$(3t+N)/4$

-32

		100	100
	160	100	100
160	160	X	

Frontière verticale

$(t+N)/2$

-80

		100	100
	200	100	100
200	200	X	

Frontière verticale brusque

N

$$d = d_v - d_h$$

$$t = \frac{W + N}{2} + \frac{NE - NW}{4}$$

Prédiction (4)

□ Exemple:

		200	100
	200	100	100
200	100	X	

$$d_h = |W - WW| + |N - NW| + |N - NE|$$

$$d_v = |W - NW| + |N - NN| + |NE - NNE|$$

$$d_h = |100 - 200| + |100 - 200| + |100 - 100| = 200$$

$$d_v = |100 - 200| + |100 - 200| + |100 - 100| = 200$$

$$d = d_v - d_h = 200 - 200 = 0$$

$$X^* = t = \frac{W + N}{2} + \frac{NE - NW}{4} = \frac{100 + 100}{2} + \frac{100 - 200}{4} = 100 - 25 = 75$$

Erreur de prédiction $e = X - X^*$

Pour $X=100$ t \rightarrow $e=100-75=25$

Raffinement de la prédiction

□ Éliminer la moyenne de l'erreur

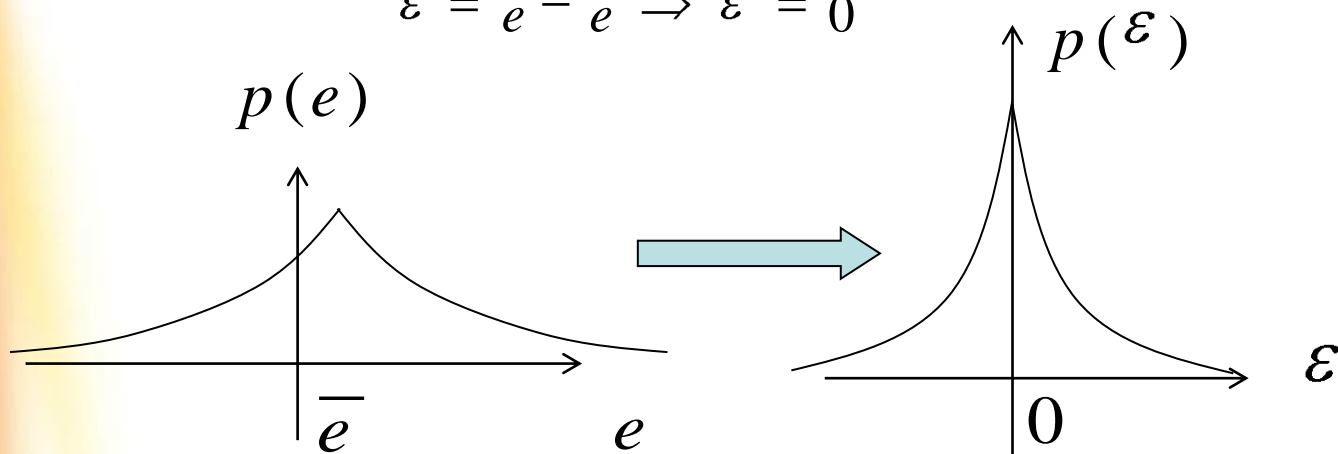
- Classification des pixels en groupe selon le contexte.
- Un contexte donné \rightarrow une moyenne de l'erreur: \bar{e} .

$$e = X - X^* \Rightarrow \bar{e}$$

\bar{e} : moyenne de l'erreur du contexte

$$\tilde{X} = X^* + \bar{e} \Rightarrow \varepsilon = X - \tilde{X} = X - X^* - \bar{e}$$

$$\varepsilon = e - \bar{e} \Rightarrow \bar{\varepsilon} = 0$$



Classification en groupe (1)

Contexte

- Motif de la texture:
- $C = [N, W, NW, NE, NN, WW, 2N-NN, 2W-WW]$
- Pour $k=1:8$
 - Si $X^* > C[k] \rightarrow \alpha[k] = 1$
 - Sinon $\rightarrow \alpha[k] = 0$
- Fin: $\alpha = b_7 b_6 \dots b_0$

		NN	NNE
	NW	N	NE
WW	W	X	

Exemple

		200	100
	200	100	100
200	100	X	

$$X^* = 75$$

$$C = \{100, 100, 200, 100, 200, 200, 0, 0\}$$

$$b_{0 \sim 7} = 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1$$

$$\alpha = 1100000_2$$

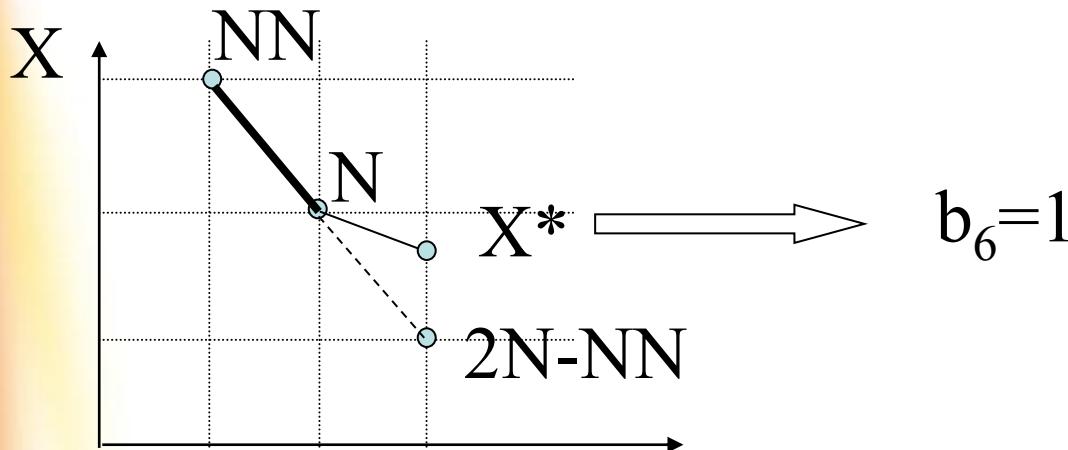
Classification en groupe (2)

Contexte

- $C=[N, W, NW, NE, NN, WW, 2N-NN, 2W-WW]$
- $\alpha=b_7b_6\dots b_0$

Signification de 2N-NN ou 2W-WW

		NN	NNE
	NW	N	NE
WW	W	X	



Classification en groupe (3)

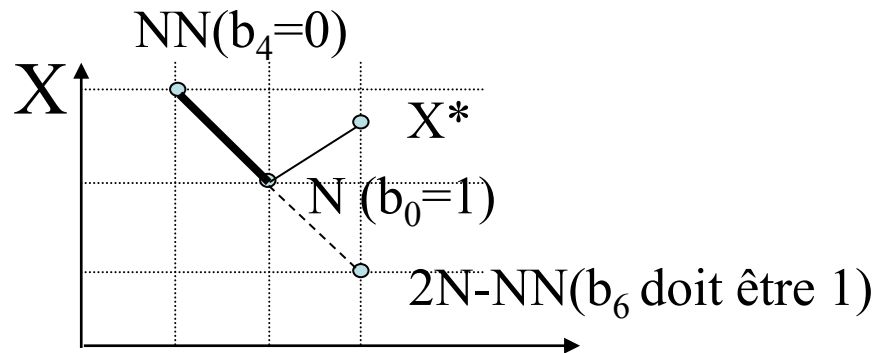
Contexte

- $C=[N, W, NW, NE, NN, WW, 2N-NN, 2W-WW]$
- $\alpha= b_7 b_6 \dots b_0$

		NN	NNE
	NW	N	NE
WW	W	X	

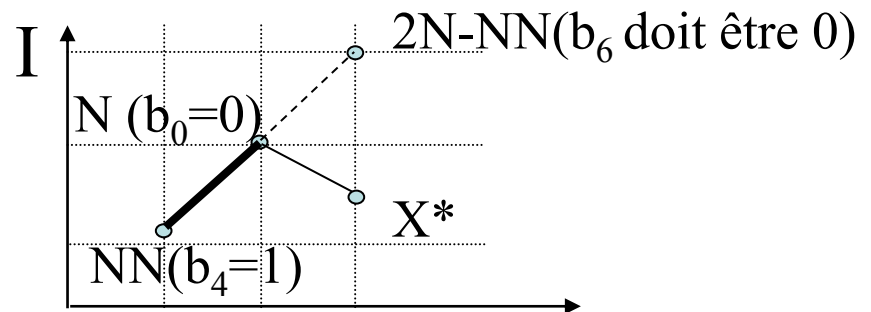
Cas non possible de α :

- $(b_0, b_4, b_6) = (1,0,0)$
- ~~$(1,0,0)$~~ $\rightarrow (1,0,1)$



Cas non possible de α :

- $(b_0, b_4, b_6) = (0,1,1)$
- ~~$(0,1,1)$~~ $\rightarrow (0,1,0)$



Classification en groupe (4)

□ Cas possibles de α : normalement $2^8=256$

- $b_0, b_4, b_6 = 1, 0, 0 \rightarrow (b_7 \ b_6 \neq 0 \ b_5 \ b_4 \neq 0 \ b_3 \ b_2 \ b_1 \ b_0 \neq 1)$: 32 cas non possibles.
- $b_0, b_4, b_6 = 0, 1, 1 \rightarrow (b_7 \ b_6 \neq 1 \ b_5 \ b_4 \neq 1 \ b_3 \ b_2 \ b_1 \ b_0 \neq 0)$: 32 cas non possibles.
- $b_1, b_5, b_7 = 1, 0, 0 \rightarrow (b_7 \neq 0 \ b_6 \ b_5 \neq 0 \ b_4 \ b_3 \ b_2 \ b_1 \neq 1 \ b_0)$: 32 cas non possibles.
- $(b_7 \neq 0 \ b_6 \neq 1 \ b_5 \neq 0 \ b_4 \neq 1 \ b_3 \ b_2 \ b_1 \neq 1 \ b_0 \neq 0)$: 4 cas déjà comptabilisés
- $(b_7 \neq 0 \ b_6 \neq 1 \ b_5 \neq 0 \ b_4 \neq 1 \ b_3 \ b_2 \ b_1 \neq 1 \ b_0 \neq 0)$: 4 cas déjà comptabilisés
- $\rightarrow 32 - (4 + 4) = 24$.
- $b_1, b_5, b_7 = 1, 0, 0 \rightarrow (b_7 \neq 1 \ b_6 \ b_5 \neq 1 \ b_4 \ b_3 \ b_2 \ b_1 \neq 0 \ b_0)$: 24 cas non possibles et non préalablement comptabilisés.
- Cas possibles de α : $256 - (32 + 32 + 24 + 24) = 144$

Classification en groupe (5)

Contexte

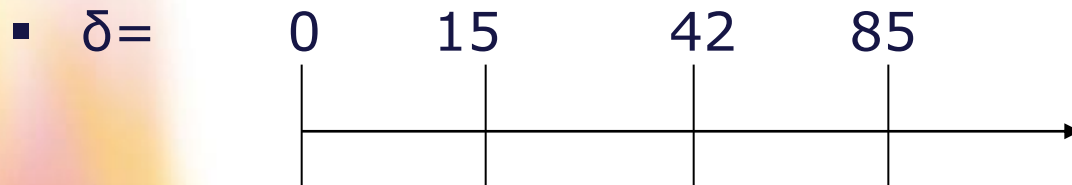
		NN	NNE
	NW	N	NE
WW	W	X	

- $C=[N, W, NW, NE, NN, WW, 2N-NN, 2W-WW]$

- $d_h = |W-WW| + |N-NW| + |NE-N|.$

- $d_v = |W-NW| + |N-NN| + |NE-NNE|$

- $\delta = d_h + d_v + 2|N-N^*|$



- $Q(\delta) =$

0	1	2	3
---	---	---	---

Quantification

- $\beta =$

00	01	10	11
----	----	----	----

Représentation binaire

- Exemple: $\delta = 70 \rightarrow Q(\delta) = 2, \beta = 10_2.$

Encodage (3)

□ Solution proposée par CALIC

- $x_n = 0, 7, 4, 3, 5, 2, 1, 7 \in \{0, 7\}$
- $r_n = 0, 7, -3, -1, 2, -3, -1, 6 \in \{-7, 7\}$.
- $r_n = x_n - x_{n-1}$ avec x_{n-1} étant connu au décodeur.
- $0 \leq x_n \leq 7 \rightarrow 0 - x_{n-1} \leq x_n - x_{n-1} \leq 7 - x_{n-1}$
- $r_n \in \{-x_{n-1}, 7 - x_{n-1}\} = \{-x_{n-1}, M-1-x_{n-1}\}$
- \rightarrow Translation vers $\{0, M-1\}$

x_{n-1}	0	0	7	4	3	5	2	1
r_n	0	7	-3	-1	2	-3	-1	6
translation	0	7	3	2	3	5	2	7

Classification en groupe (6)

□ Contexte

- $\alpha = b_7 b_6 \dots b_0$
- $\beta = \quad \quad 00 \quad \quad 01 \quad \quad 10 \quad \quad 11$
- Le contexte est déterminé à partir de la paire (α, β) .

□ Dans notre exemple

- $\alpha = \mathbf{1100000}$
- $\beta = 10$
- $C(\alpha, \beta) = \underline{1100000}10$

□ Nombre de contextes

- α : 144 valeurs possibles.
- β : 4 valeurs possibles.
- $C(\alpha, \beta)$: $144 \times 4 = 576$ contextes possibles.

Modification de la prédiction

❑ Mise à jour de la moyenne de chaque groupe (contexte)

- Lors de l'encodage, il faut garder une trace des moyennes de l'erreur de chaque groupe défini par $C(\alpha, \beta)$: $\bar{e}(\alpha, \beta) = \frac{\sum e(\alpha, \beta)}{N(\alpha, \beta)}$

❑ Prédiction modifiée: $\tilde{X} = X^* + \bar{e}(\alpha, \beta)$

❑ Erreur modifiée: $\varepsilon = X - \tilde{X}$

❑ Exemple $X = 10, 11, 13, 15, 18$

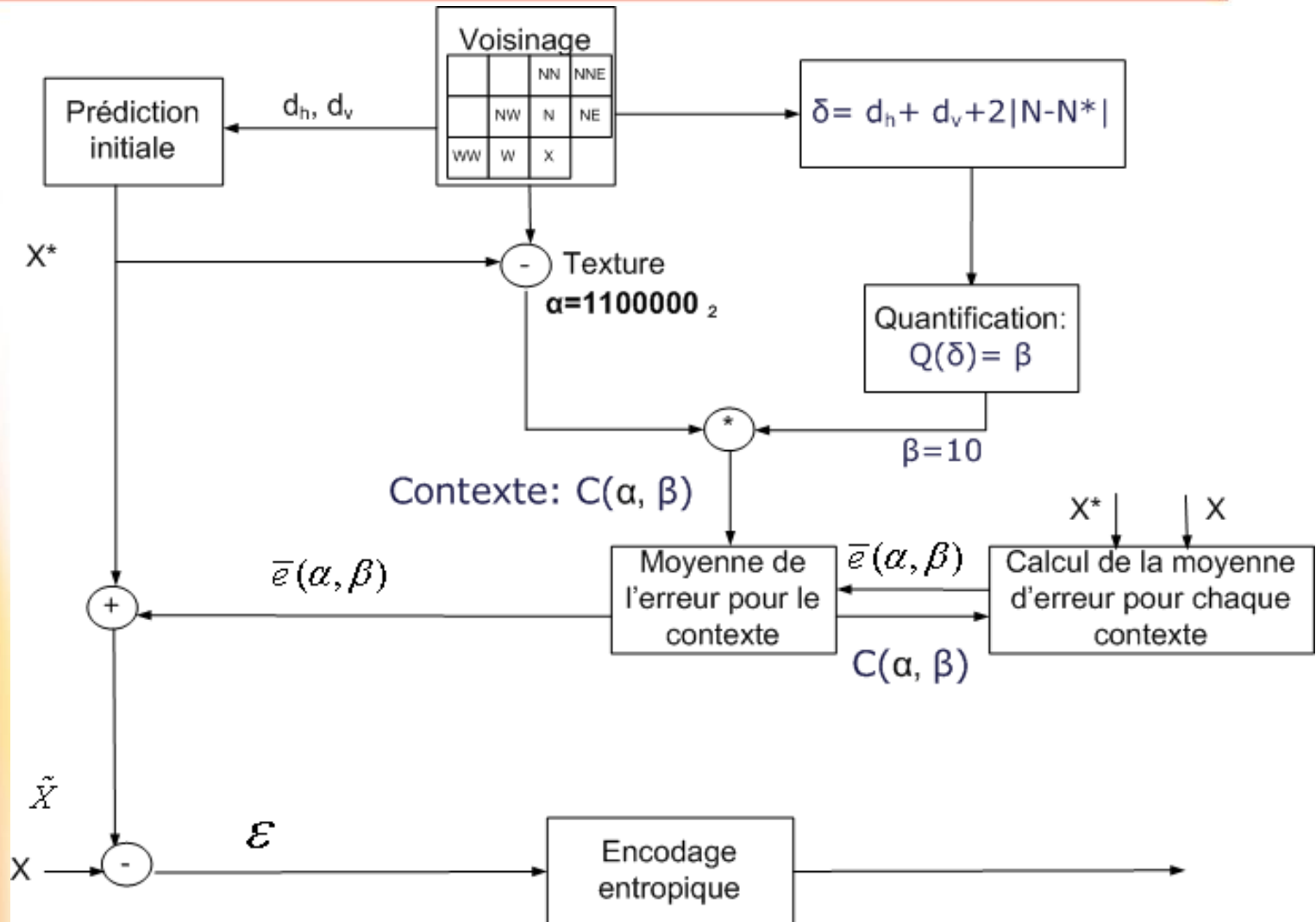
$$X^* = 8, 10, 13, 16, 14$$

$$e = 2, 1, 0, -1, 4 \rightarrow \frac{2 + 1 + 0 - 1 + 4}{5} = 1.2 \rightarrow \bar{e} = 1$$

$$\tilde{X} = X^* + \bar{e} = 9, 11, 14, 17, 15$$

$$\varepsilon = X - \tilde{X} = 1, 0, -1, -2, 3 \rightarrow \bar{\varepsilon} = 0.2$$

Récapitulation: Algorithme CALIC



Encodage (1)

□ Encodage dans CALIC

- Après avoir déterminé la prédiction finale \tilde{x} , la différence entre \tilde{x} et X est encodée. → Encodage résiduelle.
- La séquence résiduelle comporte toujours une certaine structure
- Réduire cette structure: encodage en fonction du contexte.
- Le contexte du résidu est donné par $\delta = d_h + d_v + 2|N - N^*|$.
- Réduire la complexité: on considère 8 plages:

$$0 \leq \delta < q_1 \Rightarrow \text{Context 1}$$

$$q_1 \leq \delta < q_2 \Rightarrow \text{Context 2}$$

$$q_2 \leq \delta < q_3 \Rightarrow \text{Context 3}$$

$$q_3 \leq \delta < q_4 \Rightarrow \text{Context 4}$$

$$q_4 \leq \delta < q_5 \Rightarrow \text{Context 5}$$

$$q_5 \leq \delta < q_6 \Rightarrow \text{Context 6}$$

$$q_6 \leq \delta < q_7 \Rightarrow \text{Context 7}$$

$$q_7 \leq \delta < q_8 \Rightarrow \text{Context 8}$$

Encodage (2)

□ Intervalle des résidus de prédiction

- $X \in \{0, M-1\}, X^* \in \{0, M-1\} \rightarrow X-X^* \in \{-(M-1), M-1\}$
- $X-X^*$ ont une valeur généralement proche de zéro.
- Pour le codage arithmétique il faut allouer une valeur de comptage à toutes les valeurs possibles.
- Il faut un nombre de bits élevés pour distinguer entre tous les compteurs.

□ Solution proposée par CALIC

- Soit la séquence suivante: $x_n = 0, 7, 4, 3, 5, 2, 1, 7$.
- $x_n \in \{0, 7\} \rightarrow 3$ bits.
- Prédiction utilisant l'élément précédent \rightarrow Résidu $r_n = x_n - x_{n-1}$
- $r_n = 0, 7, -3, -1, 2, -3, -1, 6 \in \{-7, 7\}$.

Encodage (3)

□ Solution proposée par CALIC

- $x_n = 0, 7, 4, 3, 5, 2, 1, 7 \in \{0, 7\}$
- $r_n = 0, 7, -3, -1, 2, -3, -1, 6 \in \{-7, 7\}$.
- $r_n = x_n - x_{n-1}$ avec x_{n-1} étant connu au décodeur.
- $0 \leq x_n \leq 7 \rightarrow 0 - x_{n-1} \leq x_n - x_{n-1} \leq 7 - x_{n-1}$
- $r_n \in \{-x_{n-1}, 7 - x_{n-1}\} = \{-x_{n-1}, M-1-x_{n-1}\}$
- \rightarrow Translation vers $\{0, M-1\}$

x_{n-1}	0	0	7	4	3	5	2	1
r_n	0	7	-3	-1	2	-3	-1	6
translation	0	7	3	2	3	5	2	7

Encodage (4)

Translation vers $\{0, M-1\}$, $X^* \leq (M-1)/2$

- $0 \rightarrow 0$
- $1 \rightarrow 1$
- $-1 \rightarrow 2$
- $2 \rightarrow 3$
- ...
- $X^* \rightarrow 2X^* - 1$
- $-X^* \rightarrow 2X^*$
- $X^* + 1 \rightarrow 2X^* + 1$
- $X^* + 2 \rightarrow 2X^* + 2$
- ...
- $M-1-X^* \rightarrow M-1$

<u>M=8</u>	<u>M=8</u>	<u>M=8</u>	<u>M=8</u>
<u>X*=0</u>	<u>X*=1</u>	<u>X*=2</u>	<u>X*=3</u>
0→0	0→0	0→0	0→0
1→1	1→1	1→1	1→1
2→2	-1→2	-1→2	-1→2
3→3	2→3	2→3	2→3
4→4	3→4	-2→4	-2→4
5→5	4→5	3→5	3→5
6→6	5→6	4→6	-3→6
7→7	6→7	5→7	4→7

Encodage (5)

□ Translation vers $\{0, M-1\}$, $X^* \geq (M-1)/2$

- $0 \rightarrow 0$
- $1 \rightarrow 1$
- $-1 \rightarrow 2$
- $2 \rightarrow 3$
- ...
- $(M-1)-X^* \rightarrow 2(X^*-(M-1))-1$
- $X^*-(M-1) \rightarrow 2(X^*-(M-1))$
- $X^*-(M-1)-1 \rightarrow 2(X^*-(M-1))+1$
- ...
- $-(X^*-1) \rightarrow M - 2$
- $-X^* \rightarrow M - 1$

<u>M=8</u>	<u>M=8</u>	<u>M=8</u>	<u>M=8</u>
<u>X*=4</u>	<u>X*=5</u>	<u>X*=6</u>	<u>X*=7</u>
0→0	0→0	0→0	0→0
1→1	1→1	1→1	-1→1
-1→2	-1→2	-1→2	-2→2
2→3	2→3	-2→3	-3→3
-2→4	-2→4	-3→4	-4→4
3→5	-3→5	-4→5	-5→5
-3→6	-4→6	-5→6	-6→6
-4→7	-5→7	-6→7	-7→7

JPEG-LS (1)

□ Prédiction initiale

- Encodeur prédictive plus simple que CALIC: LOCO-I.
- Prédiction adaptative médiane: choix de la valeur médiane entre W , N et NW si cette valeur est N ou W .

```
if  $NW \geq \max(W, N)$ 
 $\hat{X} = \max(W, N)$ 
else
{
    if  $NW \leq \min(W, N)$ 
 $\hat{X} = \min(W, N)$ 
    else
 $\hat{X} = W + N - NW$ 
}
```

		NN	NNE
	NW	N	NE
WW	W	X	

JPEG-LS (2)

□ Raffinement de la prédiction initiale

- Soustraction de la moyenne de l'erreur de prédiction du contexte en question.

□ Contextes

- Un contexte est conditionné par les variations locales.
- Calcul de 3 différences:
 - $D_1 = \text{NE} - \text{N}$
 - $D_2 = \text{N} - \text{NW}$
 - $D_3 = \text{NW} - \text{W}$.

		NN	NNE
	NW	N	NE
WW	W	X	

JPEG-LS (3)

Quantification des différences

- D_1, D_2 et $D_3 \rightarrow \mathbf{Q}=(Q_1, Q_2, Q_3)$.
- $T_{i=1,2,3}$: définis par l'utilisateur.
- 9 possibilités pour chaque Q_i .
- $9 \times 9 \times 9 = 729$ contextes.

Réduction du nombre de contextes

- Remplacer le vecteur contexte \mathbf{Q} par $-\mathbf{Q}$ lorsque le premier élément non nulle est négatif.
- $\mathbf{Q}=(2, -1, -1) \rightarrow \mathbf{Q}=(2, -1, -1)$
- $\mathbf{Q}=(-3, 2, -1) \rightarrow \mathbf{Q}=(3, -2, 1)$
- $\mathbf{Q}=(0, -2, 2) \rightarrow \mathbf{Q}=(0, 2, -2)$

$$D_i \leq -T_3 \Rightarrow Q_i = -4$$

$$-T_3 < D_i \leq -T_2 \Rightarrow Q_i = -3$$

$$-T_2 < D_i \leq -T_1 \Rightarrow Q_i = -2$$

$$-T_1 < D_i \leq 0 \Rightarrow Q_i = -1$$

$$D_i = 0 \Rightarrow Q_i = 0$$

$$0 < D_i \leq T_1 \Rightarrow Q_i = 1$$

$$T_1 < D_i \leq T_2 \Rightarrow Q_i = 2$$

$$T_2 < D_i \leq T_3 \Rightarrow Q_i = 3$$

$$T_3 < D_i \Rightarrow Q_i = 4$$

JPEG-LS (4)

□ Combien devient le nombre de contextes?

- Remplacer le vecteur contexte \mathbf{Q} par $-\mathbf{Q}$ lorsque le premier élément non nulle est négatif.
- **Si** $Q_1 < 0$: $\mathbf{Q}(Q_1, Q_2, Q_3) \rightarrow -\mathbf{Q}$: $4 \times 9 \times 9 = 324$ cas.
- **Si** $Q_1 = 0$ et $Q_2 < 0$: $\mathbf{Q}(Q_1, Q_2, Q_3) \rightarrow -\mathbf{Q}$: $4 \times 9 = 36$ cas.
- **Si** $Q_1 = 0$ et $Q_2 = 0$ et $Q_3 < 0$: $\mathbf{Q}(Q_1, Q_2, Q_3) \rightarrow -\mathbf{Q}$: 4 cas.
- $729 - (324 + 36 + 4) = 365$ contextes.

□ La variable SIGN

- $\mathbf{Q} \rightarrow \mathbf{Q}$: SIGN = +1
- $\mathbf{Q} \rightarrow -\mathbf{Q}$: SIGN = -1
- SIGN est utilisé lors du raffinement de la prédiction:
- $\tilde{X} = X^* + \text{correction} \times \text{SIGN}$.

JPEG-LS (5)

□ Translation de l'erreur de prédiction

- Erreur de prédiction: $r = X - \tilde{X} \in [-M + 1, M - 1]$
- Translation de l'erreur de prédiction vers l'intervalle de même taille que celui des pixels (pixel $\in [0, M-1] \rightarrow$ taille M):
 - $r < -M/2 \rightarrow r = r + M < M/2 \rightarrow r \in [-M/2 - 1, M/2 - 1]$
 - $r > M/2 \rightarrow r = r - M > -M/2 \rightarrow r \in [-M/2 - 1, M/2 - 1]$

□ Encodage de l'erreur de prédiction r :

- Choix adaptatif d'une variété des codes de Golomb.
- [Rappel]: les codes de Golomb sont optimal pour une séquence qui suit une distribution géométrique.

Performance

- Old Jpeg: choix de la meilleure performance entre 8 schéma de prédiction → algorithme à 8 passages.
- New JPEG et CALIC → Algorithme à passage unique.

Tailles des images après compression

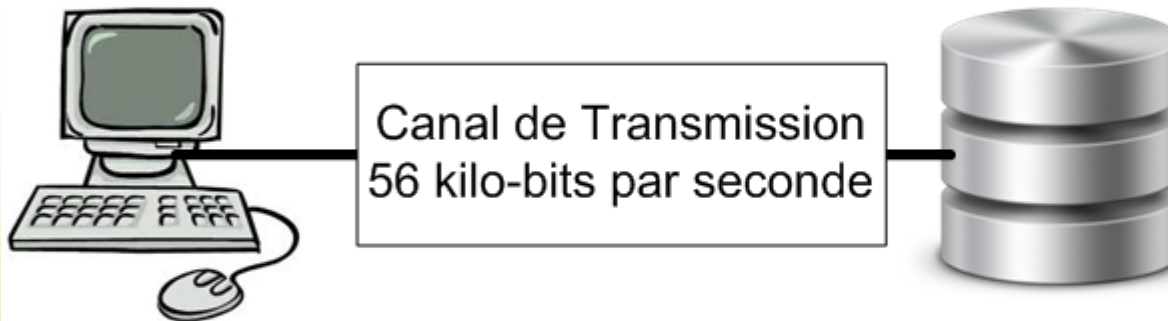
image	Old JPEG	New JPEG	CALIC
Sena	31,055	27,399	26,433
Sensin	32,429	30,344	29,213
Earth	32,137	26,088	25,280
Omaha	48,818	50,765	48,249

Le nouveau standard JPEG a des performances proches de CALIC et surpasse l'ancien standard par 6% à 8%.

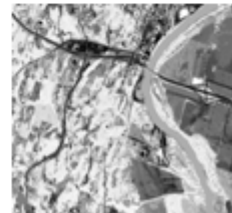
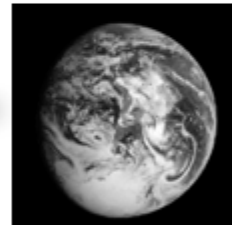
Approche multi-résolution

- ❑ Application : Recherche dans une base de données distante.

Parcours de 30 images en niveau de gris de tailles 1024x1024.
 $30 \times 8 \times 1024 \times 1024 = 251.66$ Méga-bits



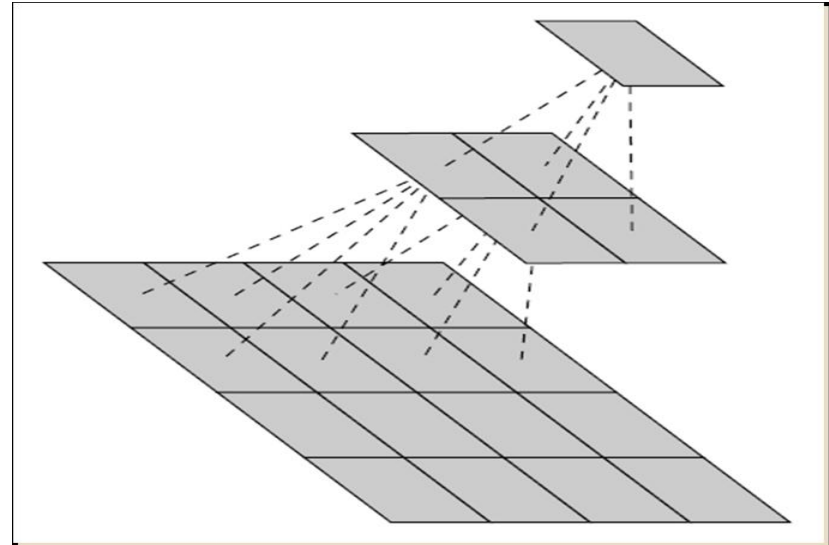
Durée de la recherche = $251 \text{ Mbs} / 56 \text{ kbps} =$
4493.9 secondes =
1 heure 15 minutes



Approche multi-résolution

□ Structure pyramidale de la transmission progressive

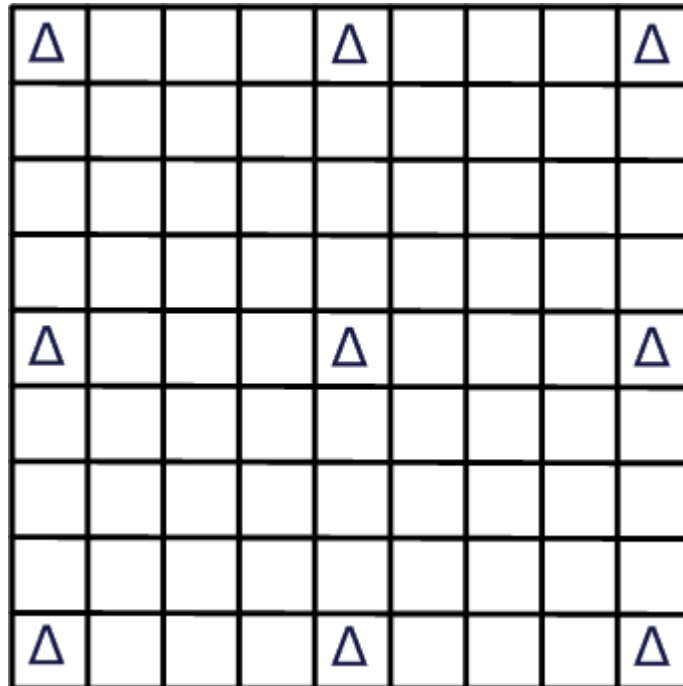
- Image initiale: 1024x1024.
- Groupement en bloc de 8x8.
- Envoi d'un pixel par bloc.
- $1024 \times 1024 \rightarrow 128 \times 128$.
- Parcours des 30 images:
- 1h 15min \rightarrow 1min 10sec.
- Il faut être capable de reconnaître l'image à partir de la trame échantillonnée.
- Sinon demande de plus de pixel de façon progressive.



Approche multi-résolution

□ HINT (Hierarchical INTerpolation) [1]

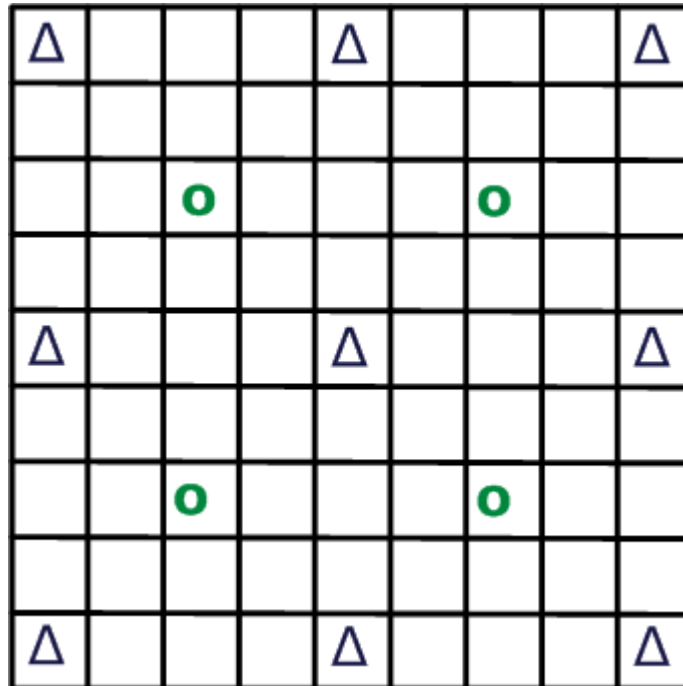
- $\Delta \rightarrow \circ \rightarrow X \rightarrow \diamond \rightarrow \square$



Approche multi-résolution

□ HINT (Hierarchical INTerpolation) [2]

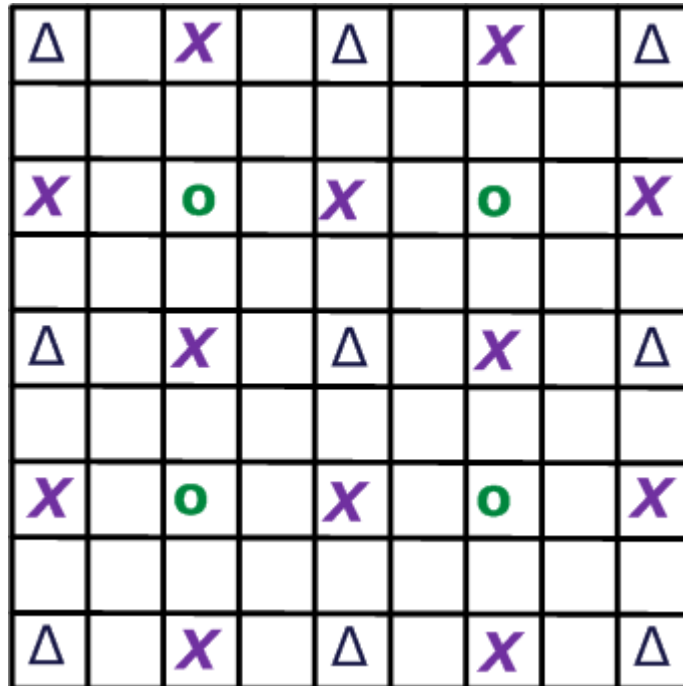
- $\Delta \rightarrow \circ \rightarrow X \rightarrow \diamond \rightarrow \square$



Approche multi-résolution

□ HINT (Hierarchical INTerpolation) [3]

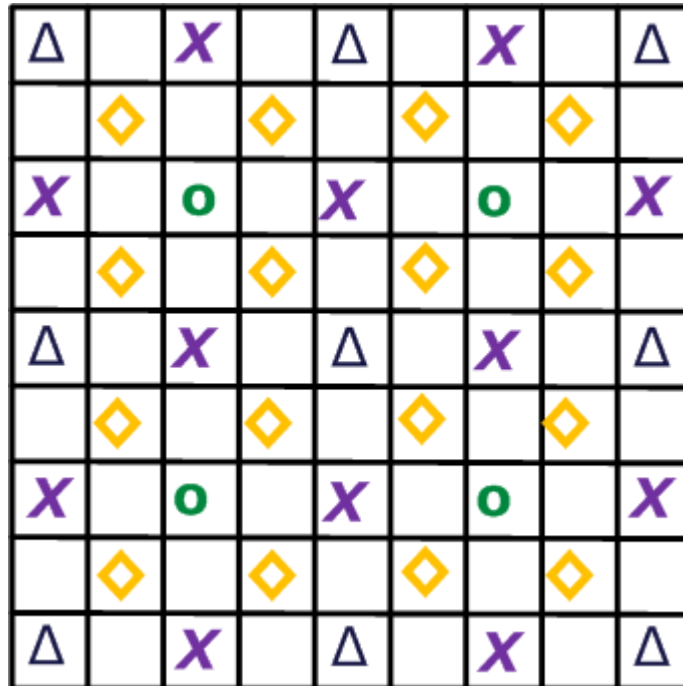
- $\Delta \rightarrow \circ \rightarrow X \rightarrow \diamond \rightarrow \square$



Approche multi-résolution

□ HINT (Hierarchical INTerpolation) [4]

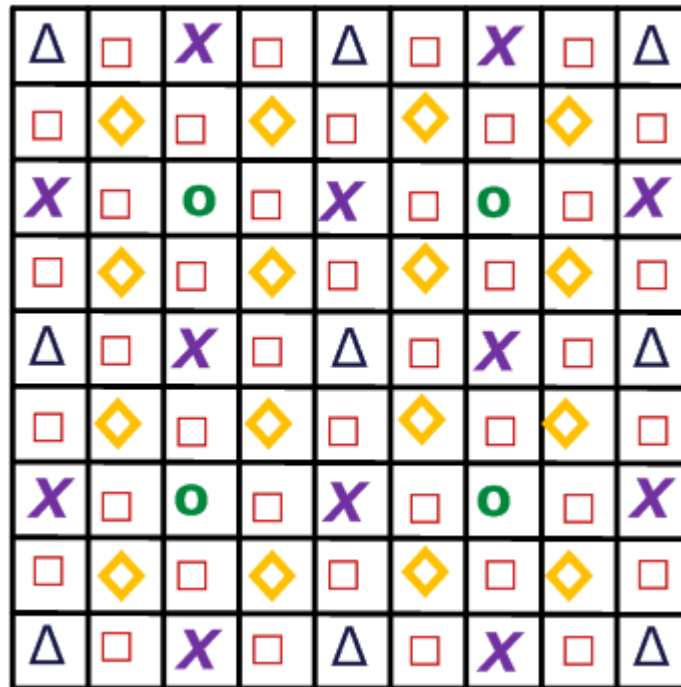
- $\Delta \rightarrow \circ \rightarrow X \rightarrow \diamond \rightarrow \square$



Approche multi-résolution

□ HINT (Hierarchical INTerpolation) [5]

- $\Delta \rightarrow \circ \rightarrow X \rightarrow \diamond \rightarrow \square$



Approche multi-résolution

- ❑ Exemple: Image Sena de taille 256x256 (1)
 - Résolution: 32x32



Approche multi-résolution

- Exemple: Image Sena de taille 256x256 (2)
 - Résolution: 32x32 → 64x64



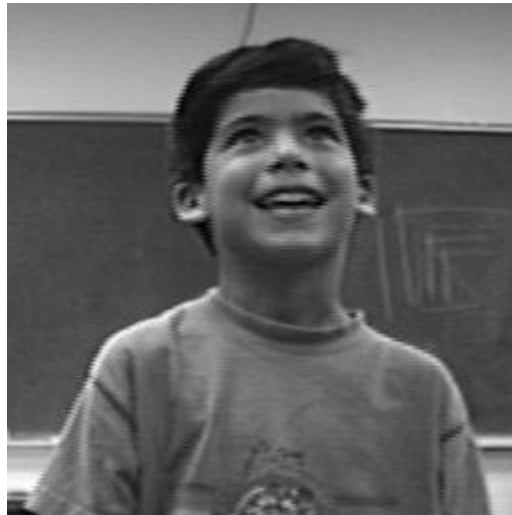
Approche multi-résolution

- Exemple: Image Sena de taille 256x256 (3)
 - Résolution: 32x32 → 64x64 → 128x128



Approche multi-résolution

- Exemple: Image Sena de taille 256x256 (4)
 - Résolution: $32 \times 32 \rightarrow 64 \times 64 \rightarrow 128 \times 128 \rightarrow 256 \times 256$



Comparaison entre la transmission progressive et le standard Raster Scan



Encodage Facsimile

❑ Équipement Fax

- **Groupe 1** (recommandation T.2): envoie un document A4 (21x29.7 cm)/6min sur une voix téléphonique analogique.
- **Groupe 2** (recommandation T.3): un document A4/3min sur une voix téléphonique analogique (pas de compression).
- **Groupe 3** (recommandation T.4): un document A4/1min, utilise une représentation binaire (compression numérique).
- **Groupe 4** (recommandation T.6, T.503, T.521 et T.563): un document A4/1min, compression numérique.

❑ Schémas de compression, protocoles, Recommandation ITU

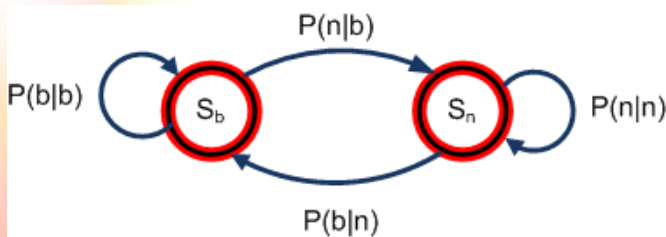
- On s'intéresse au standards plutôt qu'aux équipements: T.4, T.6, T.82(JBIG), T88(JBIG2) et T.42(MRC).

Rappel: Modèle de probabilité vs Modèle de Markov (1)

- Énoncé: Soit une image binaire tel que: $p(b|b) = 0.99$, $p(n|n) = 0.7$
- On en déduit:

$$p(b|b) = 0.99 \rightarrow p(n|b) = 0.01$$

$$p(n|n) = 0.7 \rightarrow p(b|n) = 0.3$$



$$P(S_b) = P(S_n)P(b|n) + P(S_b)P(b|b)$$

$$P(S_b) = 1 - P(S_b)P(b|n) + P(S_b)P(b|b)$$

$$(1 + P(b|n) - P(b|b))P(S_b) = P(b|n)$$

$$P(S_b) = \frac{P(b|n)}{1 + P(b|n) - P(b|b)} = \frac{0.3}{1 + 0.3 - 0.99} = \frac{30}{31}$$

$$P(S_n) = \frac{1}{31}$$



Rappel: Modèle de probabilité vs Modèle de Markov (2)

- Calcul de l'entropie en utilisant le modèle de probabilité:

$$H_{prob} = -P(S_b) \log_2 P(S_b) - P(S_n) \log_2 P(S_n)$$

$$H_{prob} = -\frac{30}{31} \log_2 \frac{30}{31} - \frac{1}{31} \log_2 \frac{1}{31} = 0.206 \text{ bits}$$

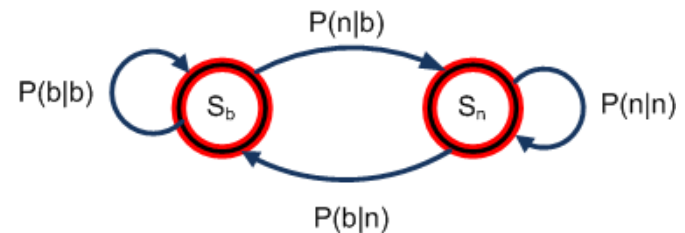
- Calcul de l'entropie en utilisant le modèle de Markov

$$H(S_b) = -0.01 \log_2 0.01 - 0.99 \log_2 0.99 = 0.081 \text{ bits}$$

$$H(S_n) = -0.3 \log_2 0.3 - 0.7 \log_2 0.7 = 0.881 \text{ bits}$$

$$H_{Markov} = P(S_b) H(S_b) + P(S_n) H(S_n)$$

$$H_{Markov} = \frac{30}{31} 0.081 + \frac{1}{30} 0.881 = 0.107 \text{ bits}$$

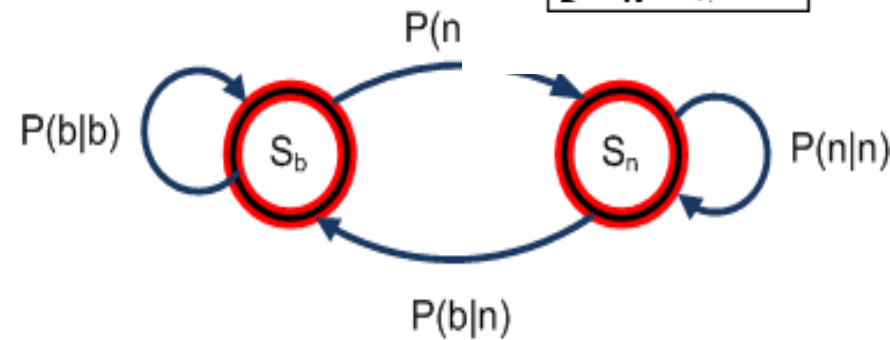


Run-Length Encoding (2)



□ Interprétation

- Probabilité très débalancée:
 - $P(b|b) \gg P(n|b)$.
- Probabilité débalancée:
 - $P(n|n) > P(b|n)$.
- Il est plus vraisemblable que le prochain pixel garde la même valeur que celui qui le précède.
- Au lieu d'encoder chaque couleur séparément on encode la longueur des *runs* de chaque couleur.
- Exemple: 190b-30n-210b.
- Encodage individuel de chaque pixel: 430 bits.
- Run length Encoding: si on considère 8 bit pour chaque run → 24 bits.



CCIT Groupe 3: Codage unidimensionnel

❑ Codage unidimensionnel

- Chaque ligne de la page est traitée séparément.
- Schéma RLE où chaque ligne est représentée par une série d'alternances entre les runs blancs et des runs noirs.
- Le premier run est supposé par convention blanc.
- Si le premier pixel est noir alors on a un premier run blanc de longueur 0.

❑ Encodage des runs

- La longueur des runs ont une probabilité différente → on utilise un code à longueur variable.
- Standards CCITT T.4 et T.6 utilise le code de Huffman.

Code de Huffman modifié MH

❑ Taille d'un document A4

- La longueur d'une ligne d'un document A4 est de 1728 pixels.
- Donc la longueur d'un run r peut atteindre 1728.
- Ce n'est pas pratique de concevoir un code de Huffman comportant 1729 mots-code.

❑ Solution

- On représente le run r sous la forme suivante:
- $r = 64 \times m + t$ pour $t = 0, 1, \dots, 63$ et $m = 1, 2, \dots, 27$.
- $t \rightarrow$ terminating codes.
- $m \rightarrow$ make-up codes

MH terminating code

Run length	White code-word	Black code-word	Run length	White code-word	Black code-word
0	00110101	0000110111	32	00011011	000001101010
1	000111	010	33	00010010	000001101011
2	0111	11	34	00010011	000011010010
3	1000	10	35	00010100	000011010011
4	1011	011	36	00010101	000011010100
5	1100	0011	37	00010110	000011010101
6	1110	0010	38	00010111	000011010110
7	1111	00011	39	00101000	000011010111
8	10011	000101	40	00101001	000001101100
9	10100	000100	41	00101010	000001101101
10	00111	0000100	42	00101011	000011011010
11	01000	0000101	43	00101100	000011011011
12	001000	0000111	44	00101101	000001010100
13	000011	00000100	45	00000100	000001010101
14	110100	00000111	46	00000101	000001010110
15	110101	000011000	47	00001010	000001010111
16	101010	0000010111	48	00001011	000001100100
17	101011	0000011000	49	01010010	000001100101
18	0100111	0000001000	50	01010011	000001010010
19	0001100	00001100111	51	01010100	000001010011
20	0001000	00001101000	52	01010101	000000100100
21	0010111	00001101100	53	00100100	000000110111
22	0000011	00000110111	54	00100101	000000111000
23	0000100	00000101000	55	01011000	000000100111
24	0101000	00000010111	56	01011001	000000101000
25	0101011	00000011000	57	01011010	000001011000
26	0010011	000011001010	58	01011011	000001011001
27	0100100	000011001011	59	01001010	000000101011
28	0011000	000011001100	60	01001011	000000101100
29	00000010	000011001101	61	00110010	000001011010
30	00000011	000001101000	62	00110011	000001100110
31	00011010	000001101001	63	00110100	000001100111

MH make-up code

Run length	White code-word	Black code-word	Run length	White code-word	Black code-word
64	11011	0000001111	1344	011011010	0000001010011
128	10010	000011001000	1408	011011011	0000001010100
192	010111	000011001001	1472	010011000	0000001010101
256	0110111	000001011011	1536	010011001	0000001011010
320	00110110	000000110011	1600	010011010	0000001011011
384	00110111	000000110100	1664	011000	0000001100100
448	01100100	000000110101	1728	010011011	0000001100101
512	01100101	0000001101100	1792	00000001000	same as
576	01101000	0000001101101	1856	00000001100	white
640	01100111	0000001001010	1920	00000001101	from this
704	011001100	0000001001011	1984	000000010010	point
768	011001101	0000001001100	2048	000000010011	
832	011010010	0000001001101	2112	000000010100	
896	011010011	0000001110010	2176	000000010101	
960	011010100	0000001110011	2240	000000010110	
1024	011010101	0000001110100	2304	000000010111	
1088	011010110	0000001110101	2368	000000011100	
1152	011010111	0000001110110	2432	000000011101	
1216	011011000	0000001110111	2496	000000011110	
1280	011011001	0000001010010	2560	000000011111	

CCIT Groupe 3: Codage bidimensionnel (1)

□ MR: Modified Read (Relative Element Address Designate)

- On représente l'instant de transition au lieu de représenter la longueur du run.



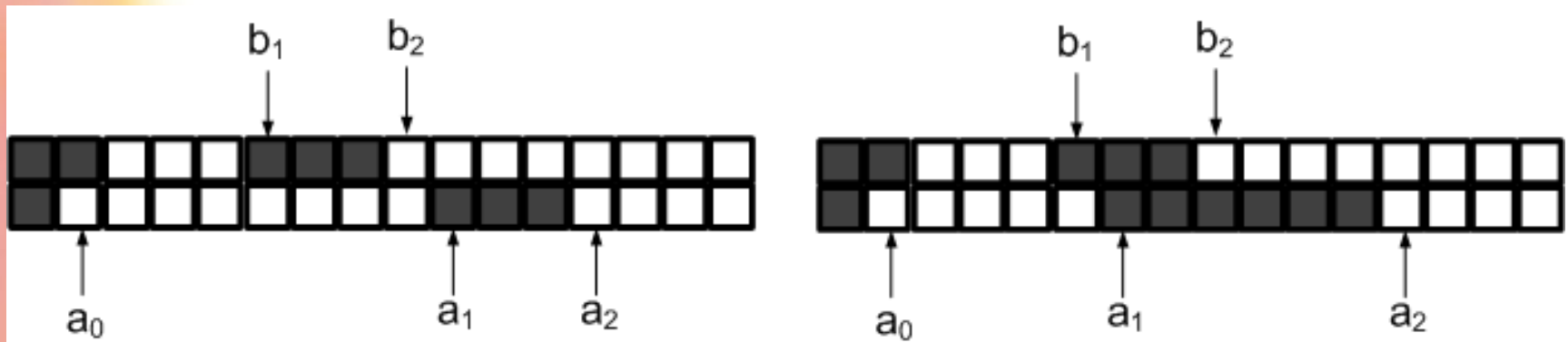
□ Notations:

- \mathbf{a}_0 : dernier pixel connu à l'encodeur et au décodeur. Lorsqu'on passe à une nouvelle ligne \mathbf{a}_0 désigne un pixel blanc imaginaire se trouvant à gauche.
- \mathbf{a}_1 : premier pixel de transition se trouvant à la droite de \mathbf{a}_0 . La couleur de \mathbf{a}_1 est l'opposé de celle de \mathbf{a}_0 . L'emplacement de \mathbf{a}_1 est connu uniquement au niveau de l'encodeur.

CCIT Groupe 3: Codage bidimensionnel (2)

□ Notations:

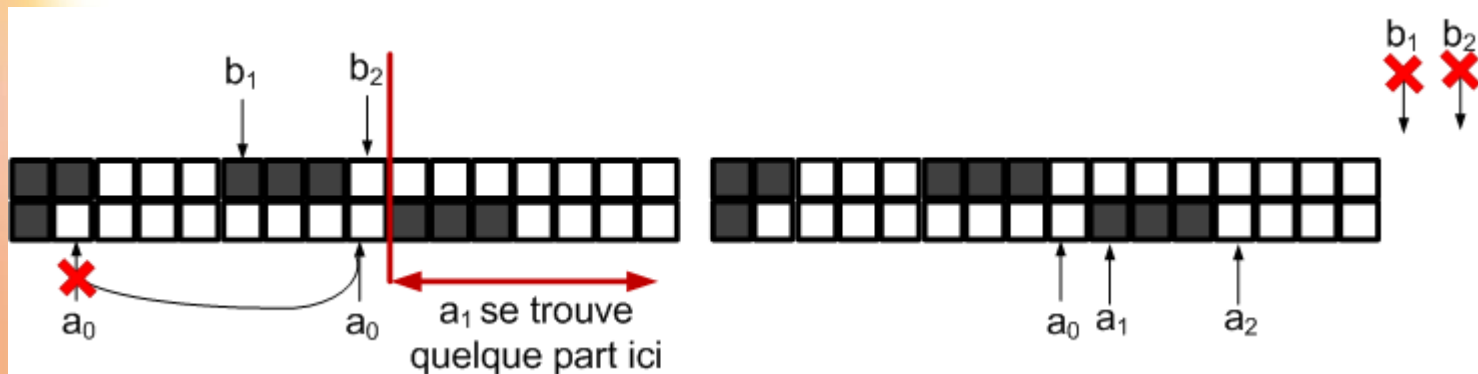
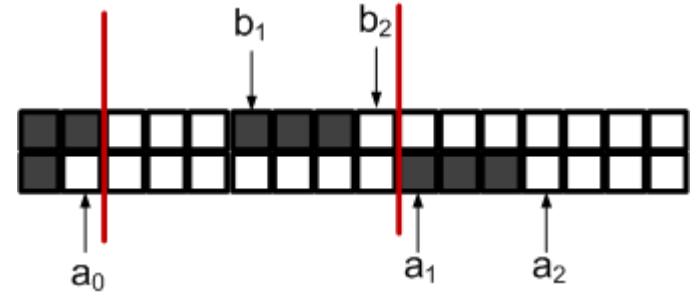
- \mathbf{a}_2 : second pixel de transition se trouvant à la droite de a_0 .
- \mathbf{b}_1 : premier pixel de transition se trouvant à la droite de a_0 de la ligne de dessus. La couleur de b_1 est l'opposé de celle de a_0 et son emplacement est connu à l'encodeur et le décodeur.
- \mathbf{b}_2 : premier pixel de transition se trouvant à la droite de b_1 de la même ligne.



CCIT Groupe 3: Codage bidimensionnel (3)

□ Pass mode:

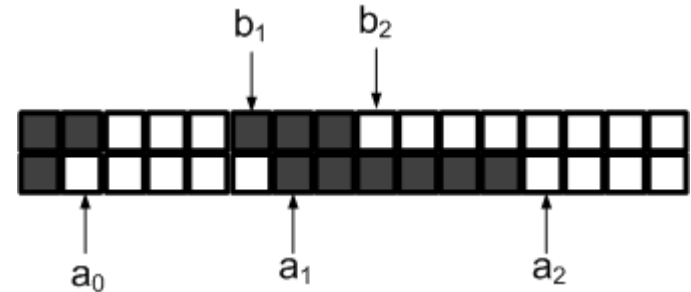
- b_1 et b_2 se trouve entre a_0 et a_1 .
- Envoi du code 0001.
- A la réception de ce code le récepteur, qui ne connaît pas a_1 , pourra savoir que a_1 se trouve au-delà de b_2 .
- Donc on peut déplacer a_0 au même niveau que b_2 et après mettre à jour b_1 et b_2 .



CCIT Groupe 3: Codage bidimensionnel (4)

□ Vertical mode & Horizontal mode:

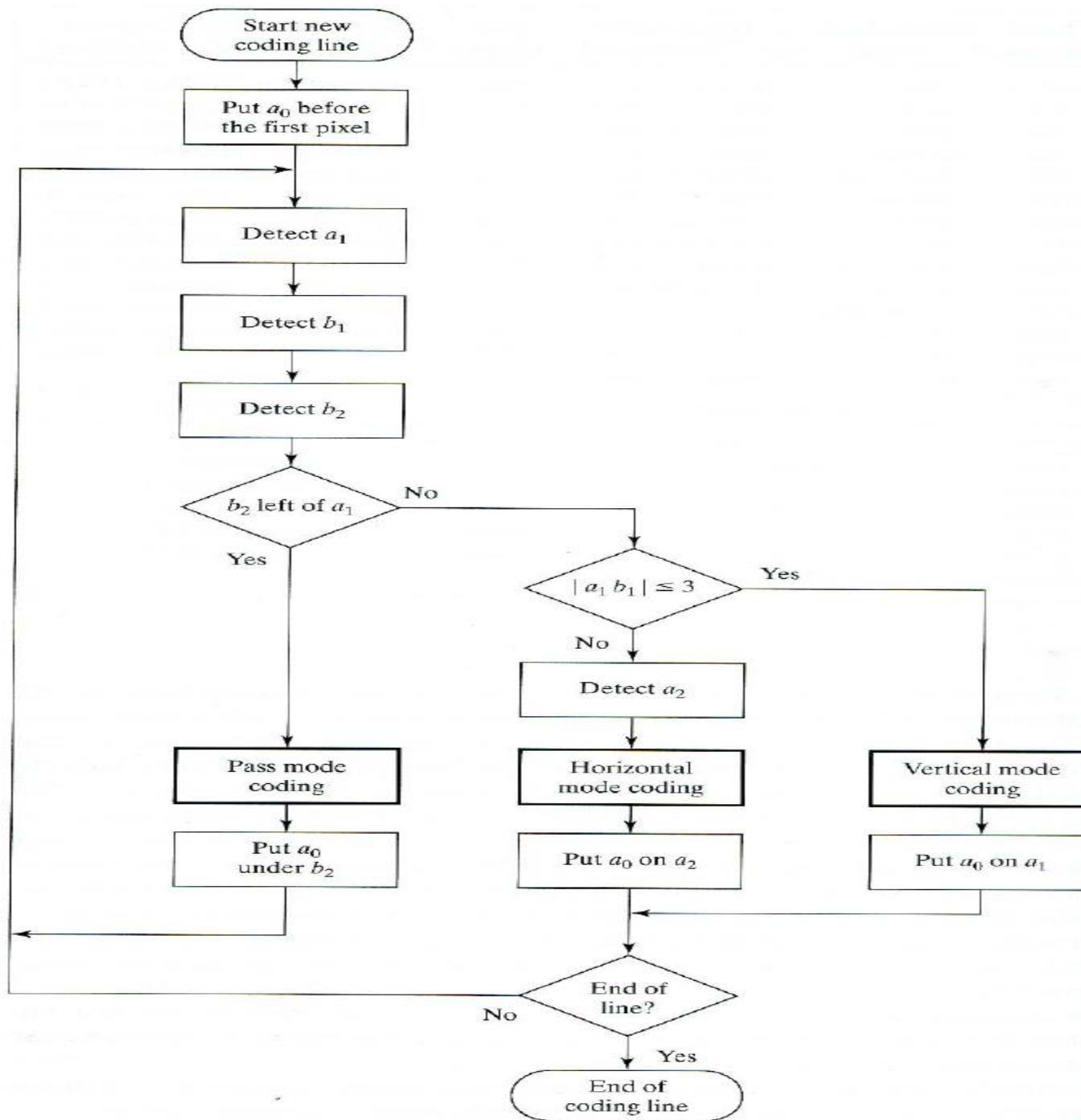
- a_1 se trouve avant b_2 .
- 2 cas se présentent:



- **Mode vertical:** $|a_1 - b_1| \leq 3$
- On envoie la position de a_1 par rapport à $b_1 \rightarrow a_1 - b_1$.
- $a_0 \leftarrow a_1$
- **Mode horizontal:** $|a_1 - b_1| > 3$
- On retourne à la technique unidimensionnelle et on envoie les distances $(a_1 - a_0)$ et $(a_2 - a_1)$ en utilisant le codage modifié de Huffman (MH).
- $a_0 \leftarrow a_2$
- Pour éviter la propagation d'erreur 1 ligne est encodé 1D et $K-1$ lignes sont encodés avec 2D ($K=2, 4$).

CCIT Groupe 3: Codage bidimensionnel (5)

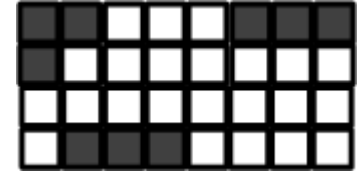
Mode	Code Word
Pass	0001
Horizontal	$001 + M(a_0a_1) + M(a_1a_2)$
Vertical	
a_1 below b_1	1
a_1 one to the right of b_1	011
a_1 two to the right of b_1	000011
a_1 three to the right of b_1	0000011
a_1 one to the left of b_1	010
a_1 two to the left of b_1	000010
a_1 three to the left of b_1	0000010
Extension	0000001 $\times\times\times$



Exemple de codage MR K=2 (1)

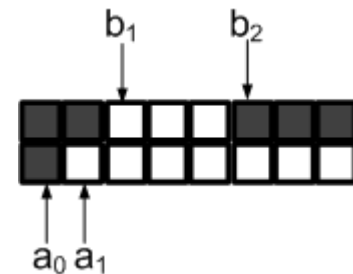
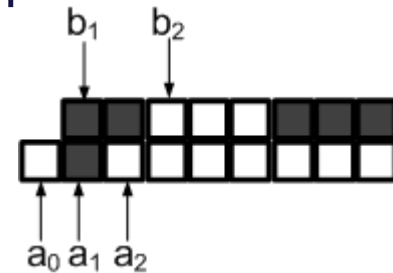
□ Codage de la ligne 1

- Utilisation du codage de Huffman modifié:
- Ligne 1 = $b(0), n(2), b(3), n(3)$
- HM → envoi: **00110101,11,1000,10**



□ Codage de la ligne 2

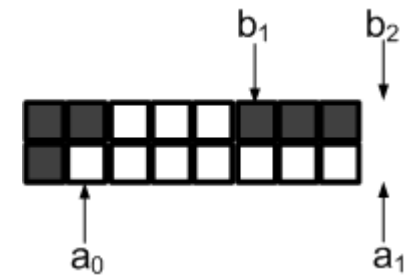
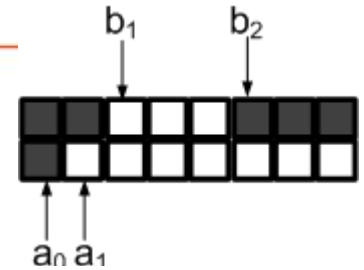
- On suppose un pixel blanc fictif au début de chaque ligne
- On détecte a_1, b_1 et b_2
- b_2 est à gauche de $a_1 \rightarrow$ non
- $|a_1 - b_1| = 0 \leq 3 \rightarrow$ mode vertical
- On envoie $a_1 - b_1 = 0$ (a_1 below b_1) \rightarrow envoi: **1**
- $a_0 \leftarrow a_1$
- mode vertical: On envoie $a_1 - b_1 = -1 \rightarrow$ envoi: **010**



Exemple de codage MR K=2 (2)

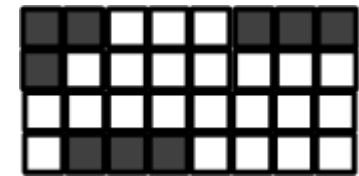
□ Codage de la ligne 2 (suite)

- $a_0 \leftarrow a_1$
- b_2 est à gauche de $a_1 \rightarrow$ non
- $|a_1 - b_1| = 4 > 3 \rightarrow$ mode horizontal
- On envoie $(a_1 - a_0) = 7$ blancs et $(a_2 - a_1) = \text{NA}$
- On utilise MH \rightarrow envoi: **1111**
- Fin de la ligne 2



□ Codage de la ligne 3

- Utilisation du codage de Huffman modifié:
- Ligne 1 = $b(8)$
- HM \rightarrow envoi: **10011**



Exemple de codage MR K=2 (3)

□ Codage de la ligne 4

- b_2 est à gauche de $a_1 \rightarrow$ non
- $|a_1 - b_1| = 7 > 3 \rightarrow$ mode horizontal
- On envoie $(a_1 - a_0) = 2$ blancs et $(a_2 - a_1) = 3$ noirs
- On utilise MH \rightarrow envoi: **0111 et 01**
- $a_0 \leftarrow a_2$
- b_2 est à gauche de $a_1 \rightarrow$ non
- $|a_1 - b_1| = 0 \leq 3 \rightarrow$ mode vertical
- On envoie $a_1 - b_1 = 0$ (a_1 below b_1) \rightarrow envoi: **1**
- Fin de la ligne 4

