

Tatouage numérique avec information
(2^{ème} génération) :

Rappel général
Introduction au tatouage informé

L'insertion informée :

Le principe de l'insertion informée
Insertion informée à corrélation linéaire
Insertion informée à corrélation normalisée
Schéma d'insertion « Broken Arrows »

Le codage informé :

Le principe du codage informé
Technique de tatouage QIM
Dirty-paper Trellis code
Travail à effectuer

CONTACT :

Email : kouider@lirmm.fr
<http://www2.lirmm.fr/~kouider>

Dissimulation de données

Tatouage numérique informé - 2^{ème} génération (Cours 2)

Tatouage numérique avec information (2^{ème} génération) :

Rappel général
Introduction au tatouage informé

L'insertion informée :

Le principe de l'insertion informée
Insertion informée à corrélation linéaire
Insertion informée à corrélation normalisée
Schéma d'insertion « Broken Arrows »

Le codage informé :

Le principe du codage informé
Technique de tatouage QIM
Dirty-paper Trellis code
Travail à effectuer

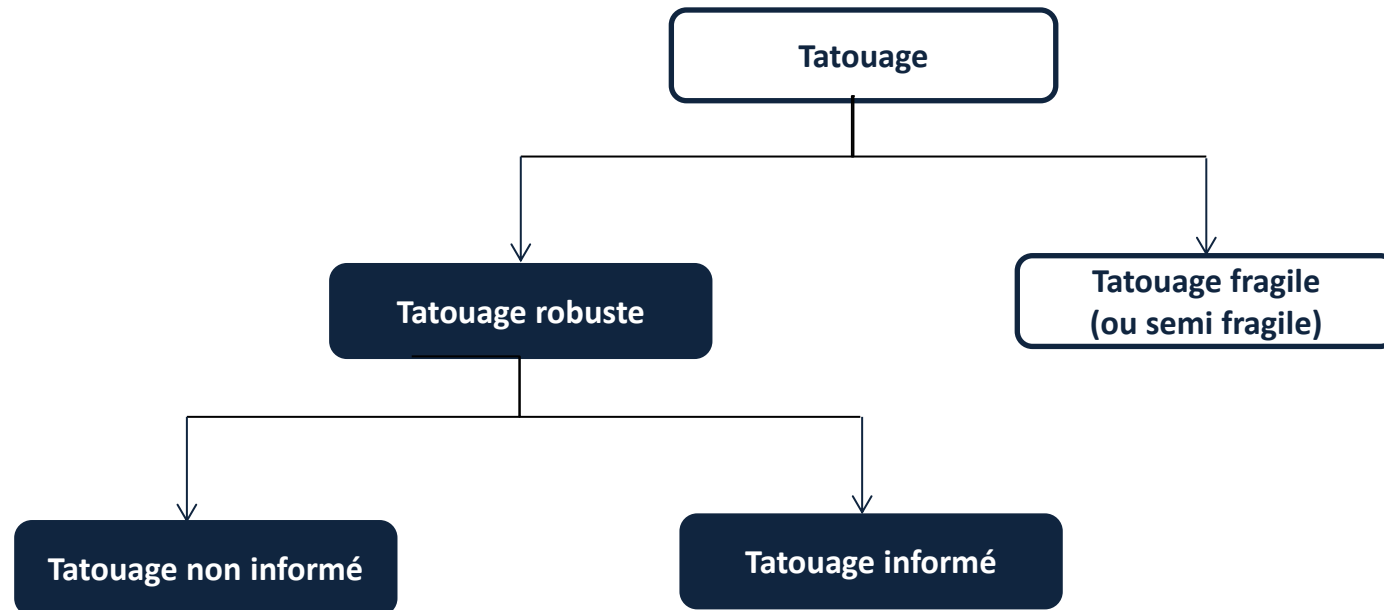
CONTACT :

Email : kouider@lirmm.fr
<http://www2.lirmm.fr/~kouider>

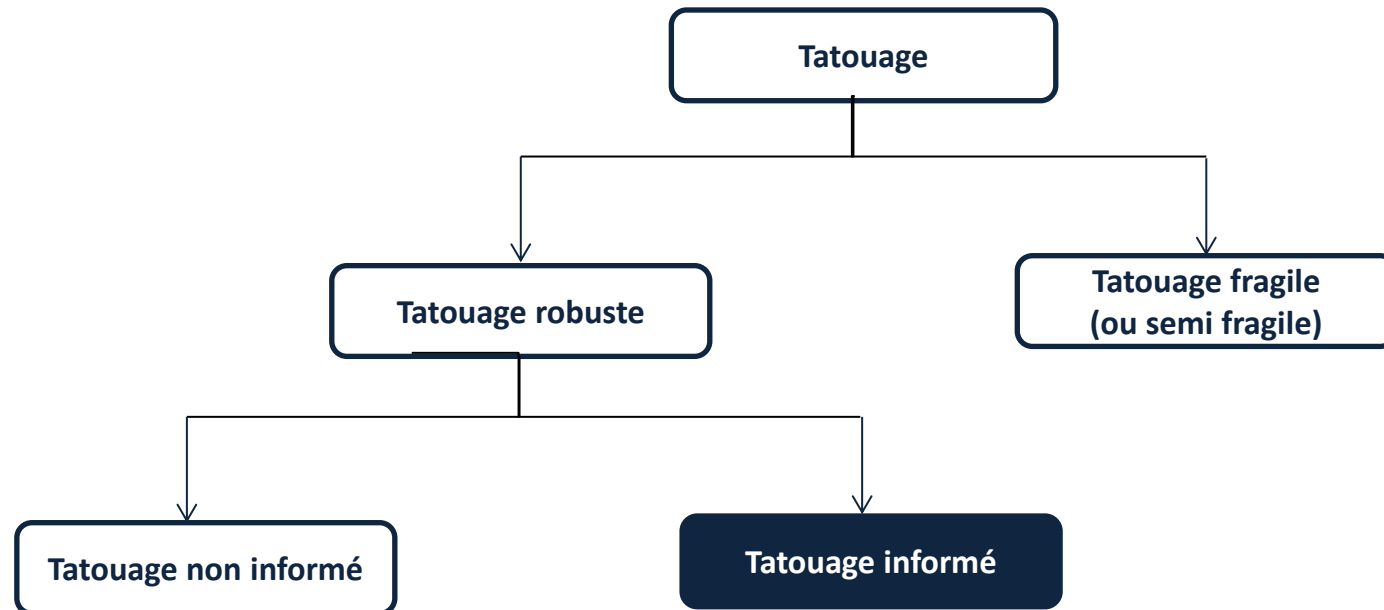
Dissimulation de données

I. Tatouage numérique avec information adjacente

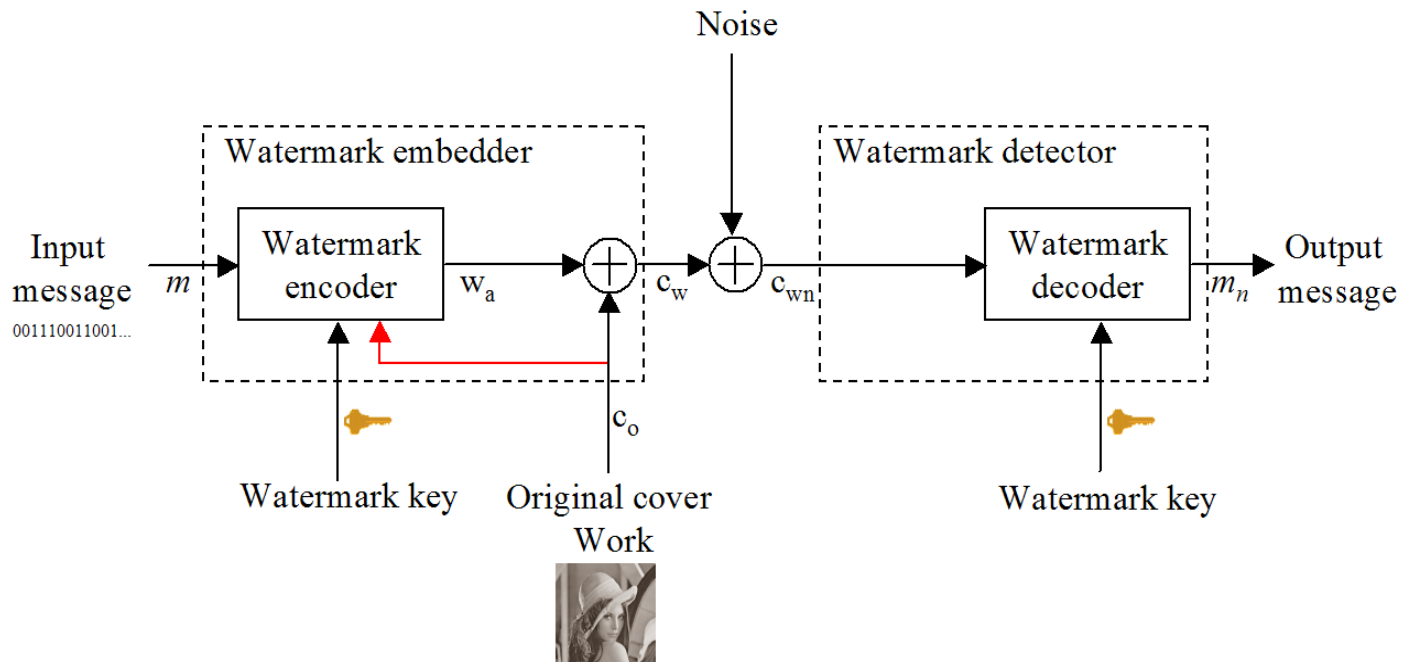
Les grandes classes du tatouage



Les grandes classes du tatouage

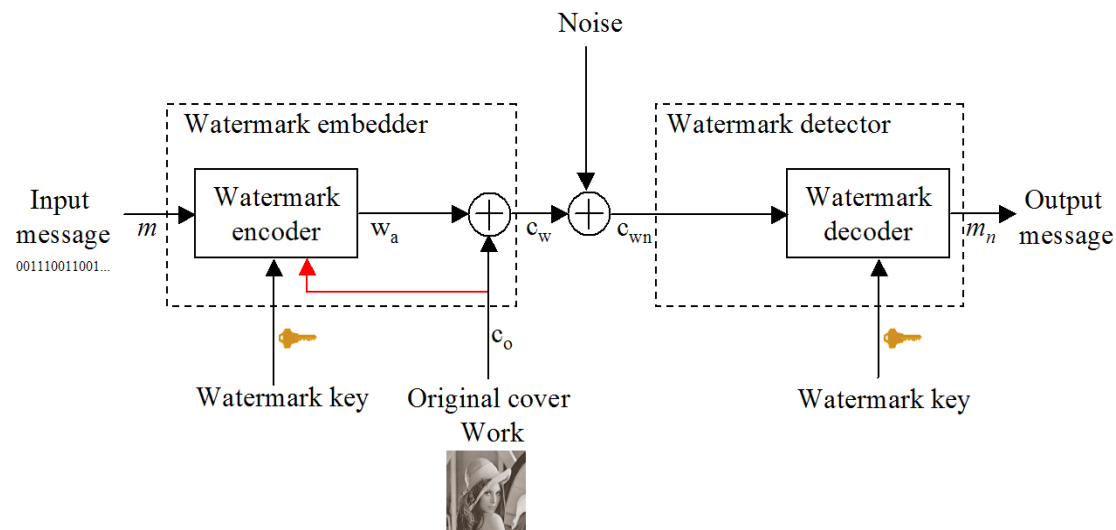


Le tatouage avec information adjacente (2^{ème} génération)

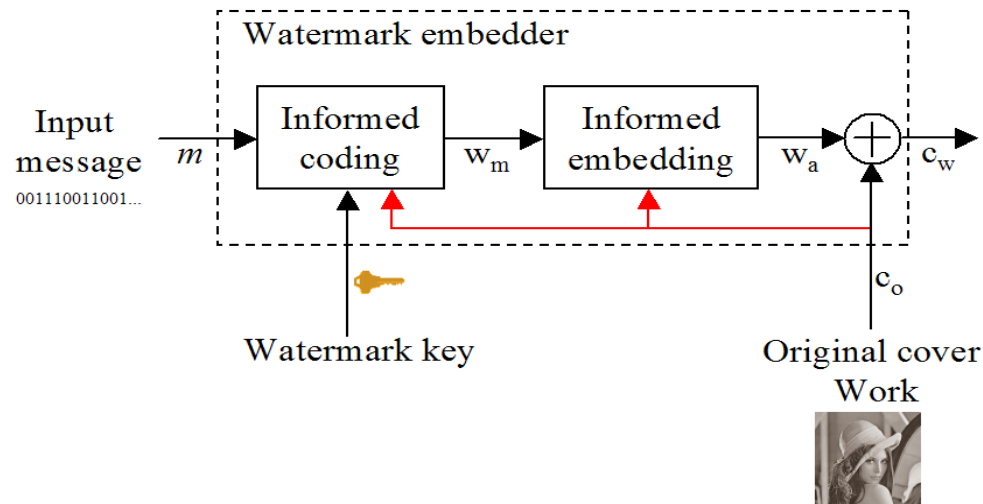


Le tatouage avec information adjacente (2^{ème} génération)

Puisqu'à l'insertion le signal hôte est connu, il est possible d'exploiter cette connaissance pour améliorer l'efficacité de l'algorithme. Le codeur examine donc c_o avant de générer la marque w_a . Plusieurs études des communications ont montré que pour certains types de canaux, l'utilisation de l'information de bord permettait de supprimer son interférence.



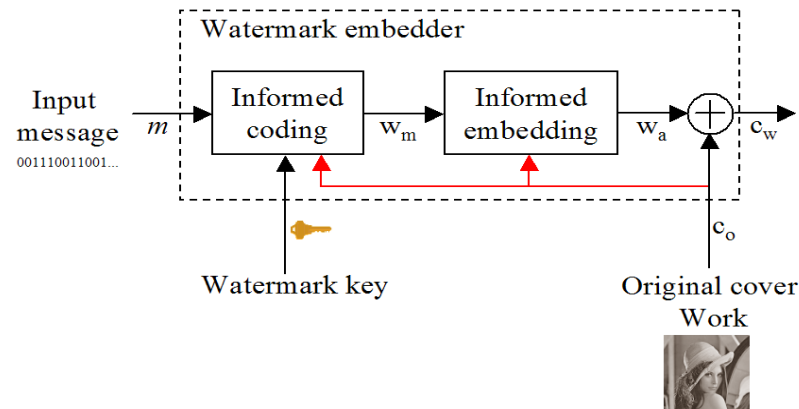
Le tatouage avec information adjacente (2^{ème} génération)



On peut distinguer (dans certains cas) :

- **étape 1 : le codage informé** (codage du message) (cf. résultat de Costa 83),
- **étape 2 : l'insertion informée** (l'insertion à proprement parler du message codé dans le signal hôte).

Le tatouage avec information adjacente (2^{ème} génération)



Les algorithmes de tatouage informé sont plus complexes à implanter que les algorithmes de tatouage aveugle. En effet, ils imposent :

- **Une bonne connaissance du détecteur,**
- **La prise en compte de l'impact d'éventuelles attaques sur les zones de détection,**
- **Trouver le moyen de faire passer un mot de code dans une zone de détection.**

Tatouage numérique avec information
(2ème génération) :

Rappel général
Introduction au tatouage informé

L'insertion informée :

Le principe de l'insertion informée
Insertion informée à corrélation linéaire
Insertion informée à corrélation normalisée
Schéma d'insertion « Broken Arrows »

Le codage informé :

Le principe du codage informée
Technique de tatouage QIM
Dirty-paper Trellis code
Travail à effectuer

CONTACT :

Email : kouider@lirmm.fr
<http://www2.lirmm.fr/~kouider>

Dissimulation de données

II. Insertion Informée

Enoncé du problème (optimisation)

Le problème de l'insertion peut être vu comme un problème d'optimisation :

- Soit trouver la marque qui maximise la robustesse (position dans l'hyper-plan ou l'hyper-cône) tout en conservant une distortion perceptuelle fixée (distance de Watson ...) (cas non détaillé dans ce cours).
- Soit trouver la marque qui minimise la distortion perceptuelle tout en maintenant un niveau de robustesse fixé (cas non détaillé dans ce cours)

Principe de fonctionnement de l'insertion informée

Le problème de l'insertion peut être vu comme un problème d'optimisation :

- Soit trouver la marque qui maximise la robustesse (position dans l'hyper-plan ou l'hyper-cône) tout en conservant une distortion perceptuelle fixée (distance de Watson ...) (cas non détaillé dans ce cours).
- Soit trouver la marque qui minimise la distortion perceptuelle tout en maintenant un niveau de robustesse fixé (cas non détaillé dans ce cours)

La démarche consiste à adapter le signal **w** au signal hôte **x** selon les contraintes d'imperceptibilité de détection et de **robustesse**.

Mesure – définition – de la robustesse

Une **mesure simpliste de robustesse** est de supposer qu'une marque insérée avec une valeur de corrélation forte est plus robuste qu'une marque insérée avec une faible valeur de corrélation.

- c'est **vrai** pour la corrélation linéaire,
- c'est **faux** pour la corrélation normalisée.

Les transparents suivants expliquent ce point.

Rappel - la corrélation linéaire

Détection aveugle :

Pour détecter la marque, il faut détecter $\pm w_r$ en présence du bruit causé par le signal hôte c_o et le bruit n . La manière optimale pour détecter ce signal en présence de bruit additif Gaussien est de calculer la corrélation linéaire entre l'image reçue c_{wn} et le pattern w_r :

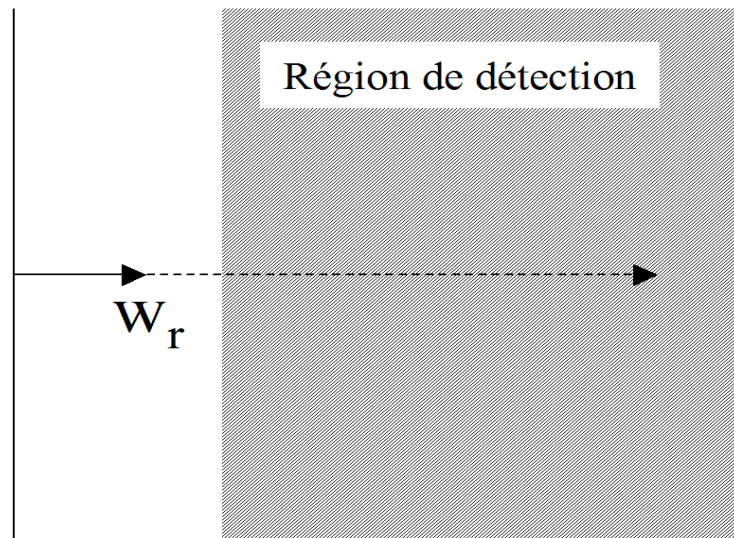
$$Z_{lc}(c_{wn}, w_r) = \frac{1}{N} c_{wn} \cdot w_r = \frac{1}{N} \sum_{i=1}^N c_{wn}[i] \cdot w_r[i]$$

Sortie du détecteur :

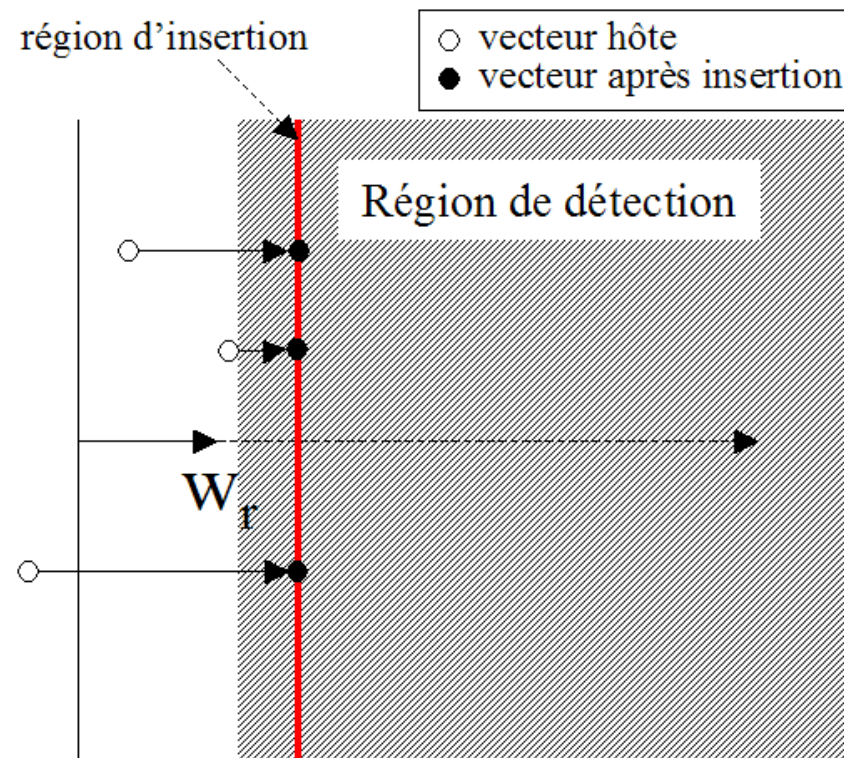
$$m_n = \begin{cases} 1 & \text{si } Z_{lc}(c_{wn}, w_r) > \tau_{lc} \\ \text{pas de marque} & \text{si } -\tau_{lc} \leq Z_{lc}(c_{wn}, w_r) \leq \tau_{lc} \\ 0 & \text{si } Z_{lc}(c_{wn}, w_r) < -\tau_{lc} \end{cases}$$

Interprétation géométrique de la corrélation linéaire

Lorsque la corrélation entre un vecteur c_{w_n} et une marque w_n dépasse un seuil τ , on dira que les deux vecteurs sont corrélés. On peut alors voir la *région de détection* comme un **hyper-plan** perpendiculaire au vecteur marque. Tout signal c étant dans la région de détection est considéré comme tatoué.



Insertion informée avec une stratégie d'insertion à corrélation linéaire fixée



Modélisation de la stratégie d'insertion à corrélation linéaire fixée

La corrélation linéaire entre un signal marqué $c_w = c_o + \alpha w_m$ et un pattern w_m de taille N est :

$$Z_{lc}(c_w, w_m) = \frac{1}{N}(c_o \cdot w_m + \alpha w_m \cdot w_m)$$

Si le seuil de détection est fixé à $\tau_{lc} + \beta$, on peut déduire la valeur de α :

$$\alpha = \frac{N(\tau_{lc} + \beta) - c_o \cdot w_m}{w_m \cdot w_m}$$

Les défauts de la corrélation linéaire

Un des problèmes de la corrélation linéaire est que le seuil de détection est fortement dépendant de l'amplitude des vecteurs. La mesure de corrélation linéaire est donc **non robuste aux attaques valuemetriques** (exemple : réduction de la luminosité).

Un autre problème réside dans la difficulté à modéliser la probabilité de *random-work* faux positif (même quand la marque suit une distribution blanche Gaussienne).

La corrélation normalisée - Définitions

La corrélation normalisée permet de résoudre les problèmes liés à la corrélation linéaire. Le vecteur c_{w_n} et le vecteur w_r sont normalisés à 1 avant d'effectuer le produit scalaire :

$$Z_{nc}(c_{w_n}, w_r) = \frac{c_{w_n}}{|c_{w_n}|} \cdot \frac{w_r}{|w_r|} = \frac{1}{|c_{w_n}| |w_r|} \sum_{i=1}^N c_{w_n}[i] \cdot w_r[i]$$

Le produit scalaire entre deux vecteurs c_{w_n} et w_r est égal au produit des longueurs des vecteurs par le cosinus de l'angle θ entre les deux vecteurs : $c_{w_n} \cdot w_r = |c_{w_n}| |w_r| \cos(\theta)$. Ainsi la corrélation normalisée est égale au cosinus de l'angle entre c_{w_n} et w_r :

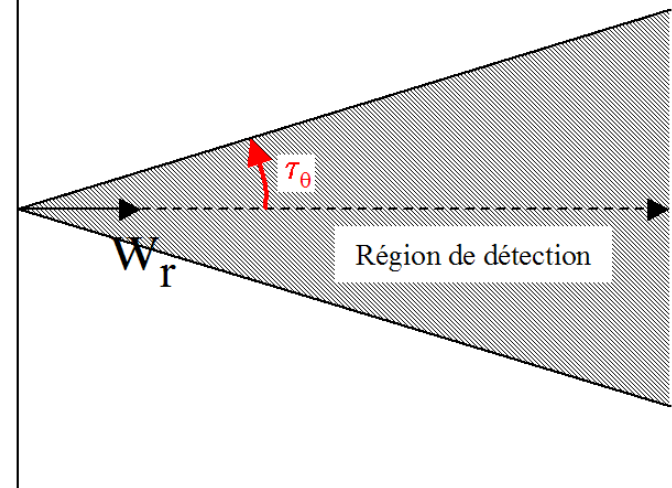
$$Z_{nc}(c_{w_n}, w_r) = \cos(\theta)$$

Interprétation géométrique de la corrélation normalisée

Lorsque la corrélation normalisée entre un vecteur c_{w_n} et une marque w_r dépasse un seuil τ , on dira que les deux vecteurs sont corrélés. On peut alors voir la *région de détection* comme un hyper-cone. Appliquer un seuil sur la corrélation est équivalent à appliquer un seuil sur l'angle entre les deux vecteurs c_{w_n} et w_n :

$$\frac{c_{w_n} \cdot w_r}{|c_{w_n}| \cdot |w_r|} > \tau_{nc} \Leftrightarrow \theta < \tau_\theta$$

avec $\tau_\theta = \cos^{-1}(\tau_{nc})$.



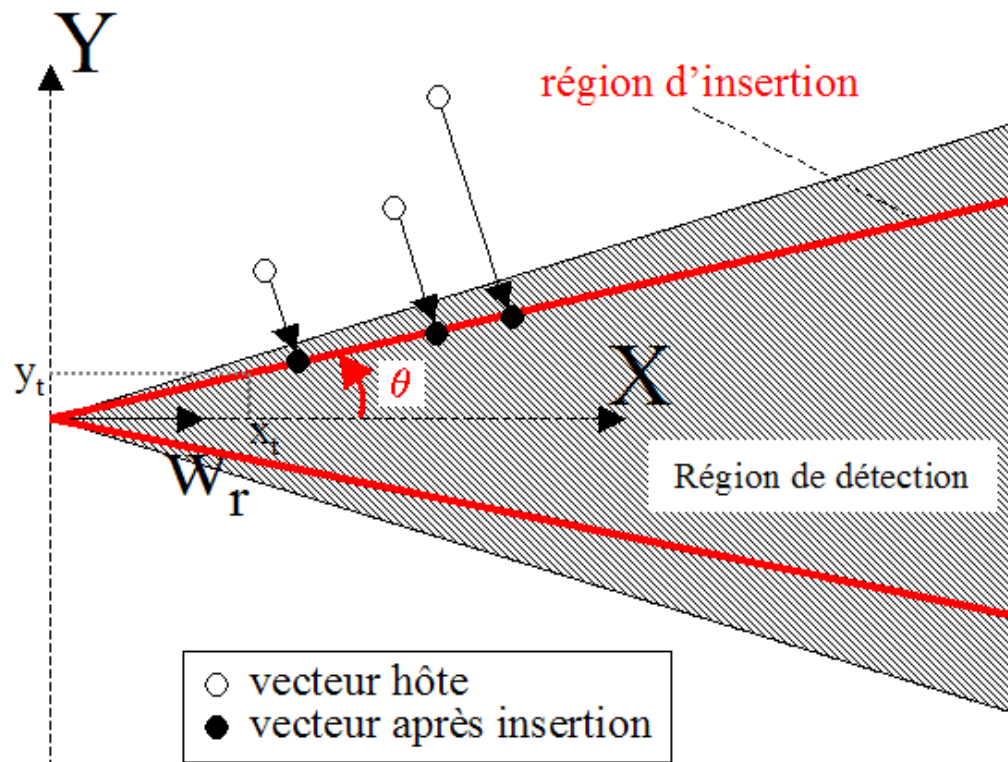
Mesure équivalente

De manière proche, certains auteurs utilisent la mesure suivante :

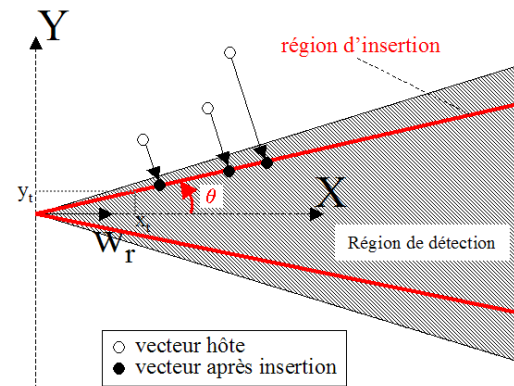
$$z_1(c_{wn}, w_r) = \frac{c_{wn} \cdot w_r}{|c_{wn}|}$$

qui est équivalente à la mesure de corrélation normalisée à un facteur près (la norme du vecteur w_r).

Insertion informée avec une stratégie d'insertion à corrélation normalisée fixée



Insertion informée avec une stratégie d'insertion à corrélation normalisée fixée



Solution :

- fixer la corrélation normalisée à τ_{nc}
- trouver le point c_w issu de la projection du signal hôte c_o sur la surface du cône. Le cône est centré sur le pattern de référence w_r .

Nous essayons de trouver le point le plus proche sur le cône pour le signal hôte C_o



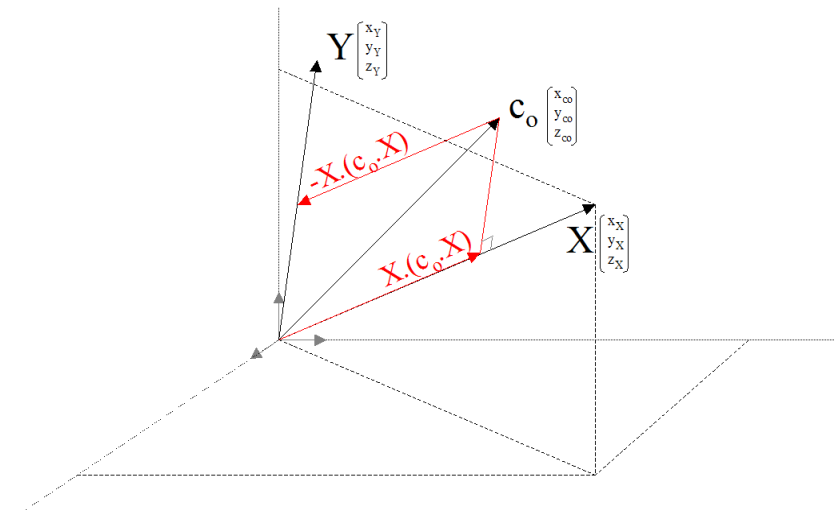
Pour ce faire, il faut réduire le problème initial en un problème à deux dimension

Réduction du problème à un problème à deux dimension - définition du plan de Cox

Soit (X,Y) deux axes orthogonaux définissant le plan contenant c_o et w_r . On utilise la technique d'orthonormalisation de **Gram-Schmidt** :

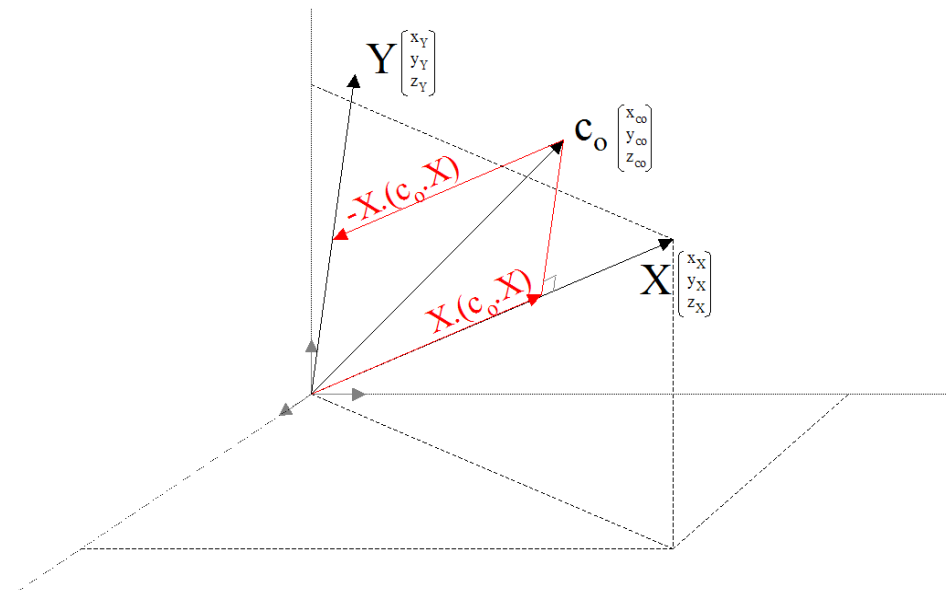
$$X = \frac{w_r}{|w_r|}$$

$$Y = \frac{c_o - X.(c_o.X)}{|c_o - X.(c_o.X)|}$$

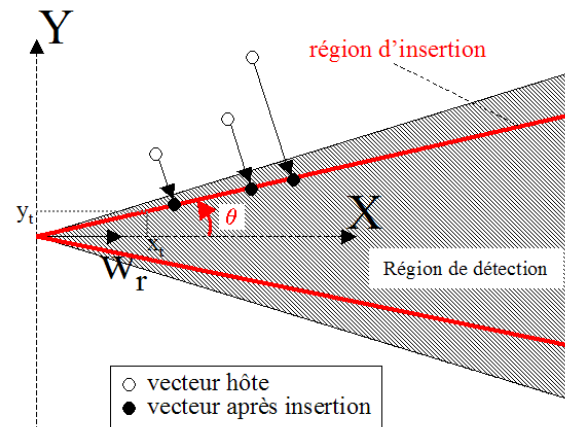


Réduction du problème à un problème à deux dimension - définition du plan de Cox

Chaque point du plan (\mathbf{X}, \mathbf{Y}) peut être exprimé par un point 2D (x, y) et dans le repère initiale par $x\mathbf{X} + y\mathbf{Y}$.



Equation de la région d'insertion



L'intersection du plan XY avec la région d'intersection nous donne deux droites formant un triangle.

Puisque le Y le point le plus proche du signal hôte C_0 sera forcément sur la ligne en haut notée ici \mathcal{D} .

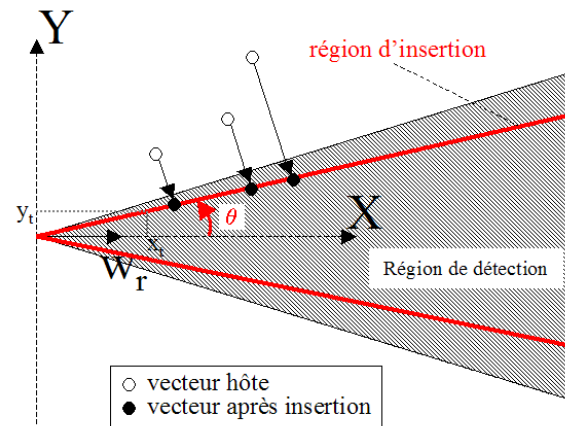
La droite \mathcal{D} peut être décrite par un vecteur (x_t, y_t) dans le plan (X, Y) :

$$x_t = \cos(\theta)$$

$$y_t = \sin(\theta)$$

avec θ l'angle entre w_r et la droite \mathcal{D} .

Equation de la région d'insertion

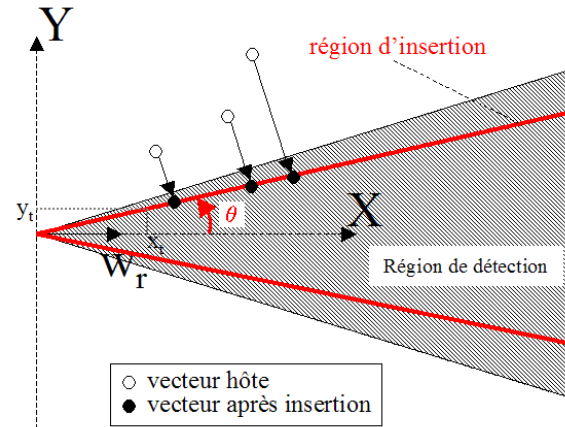


$\theta = \cos^{-1}(\tau_{nc})$ avec τ_{nc} le seuil de détection (corr. norm.) fixé par l'utilisateur. D'où :

$$x_t = \tau_{nc}$$

$$y_t = \sin(\cos^{-1}(\tau_{nc})) = \sqrt{1 - \tau_{nc}^2}$$

Equation de la région d'insertion



Les coordonnées de c_o dans le plan (X, Y) sont

$$x_{c_o} = c_o \cdot X$$

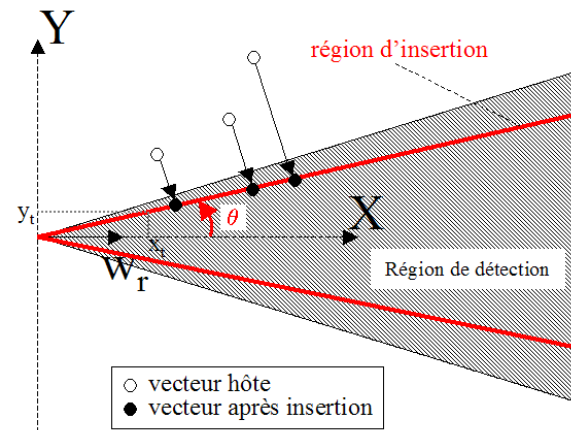
$$y_{c_o} = c_o \cdot Y$$

Le point $c_w = (x_{c_w}, y_{c_w})$ (exprimé dans le plan (X, Y)) le plus proche (distance euclidienne) de c_o appartenant à \mathcal{D} est le point :

$$x_{c_w} = x_t \cdot (x_t \cdot x_{c_o} + y_t \cdot y_{c_o})$$

$$y_{c_w} = y_t \cdot (x_t \cdot x_{c_o} + y_t \cdot y_{c_o})$$

Equation de la région d'insertion



L'équation du point c_w exprimé dans le repère initial est :

$$c_w = x_{c_w} \cdot X + y_{c_w} \cdot Y$$

Code pour le calcul de la région d'insertion

```
/*-----*
| Orthonormalize -- convert two vectors into two unit-length, orthogonal |
|                  vectors that lie in the same plane                     |
| Arguments:                                                |
|   X -- vector who's direction will not be changed (changed in place) |
|   Y -- vector who's direction will be changed (changed in place)     |
|   size -- number of component for a vector                 |
|-----*/
void WMTools::Orthonormalize( double *X, double *Y , int size) {
    double XDotY;                /* inner product of original Y and
                                unit-length X */

    double len;                 /* Euclidian length (magnitude) of a
                                vector */

    int i;                      /* index into marks */

    /* Normalize X to unit length. */
    len = 0;
    for( i = 0; i < size; i = i + 1 )
        len = len + X[ i ] * X[ i ];
    len = sqrt( len );
    for( i = 0; i < size; i = i + 1 )
        X[ i ] = X[ i ] / len;

    ...
}
```

Code pour le calcul de la région d'insertion

```
...

/* Subtract X * (X dot Y) from Y to ensure that X and Y are orthogonal. */
XDotY = PatPatInnerProduct( X, Y , size);
for( i = 0; i < size; i = i + 1 )
    Y[ i ] = Y[ i ] - XDotY * X[ i ];

/* Normalize Y to unit length. */
len = 0;
for( i = 0; i < size; i = i + 1 )
    len = len + Y[ i ] * Y[ i ];
len = sqrt( len );
for( i = 0; i < size; i = i + 1 )
    Y[ i ] = Y[ i ] / len;
}
```

Code pour le calcul de la région d'insertion

```

/*-----*/
| MixFixedCC -- compute a vector that is close to a given extracted |
|                vector, and has a fixed correlation coefficient with a |
|                given message mark |
| |
| The correlation between the new vector and the reference vector is |
| specified as the sum of a detection threshold and a "strength" |
| parameter. The new vector is as close as possible to the given |
| extracted vector, measured by Euclidian distance. |
| |
| Arguments: |
|   co -- original image |
|   tcc -- detection threshold |
|   beta -- strength parameter |
|   wr -- message mark |
|   vw -- where to store resulting vector |
/*-----*/

void WME_FIXED_CC::MixFixedCC( unsigned char* co, int width, int height,
                               double tcc, double beta, double *wr, double *cw ) {

    double* X = new double [ width*height ]; /* unit vector aligned with wm */
    double* Y = new double [ width*height ]; /* unit vector orthogonal to X, such
                                                that X and Y describe the plane
                                                containing wm, co and the origin */

```

Code pour le calcul de la région d'insertion

```
...

double wrMean;           /* mean of wr */
double coMean;           /* mean of co */
double xco, yco;         /* coordinates of co in the XY plane */
double xt, yt;           /* unit vector in the XY plane that
                           has the desired correlation with
                           the watermark */

double xcw, ycw;         /* coordinates of new vector in the
                           XY plane */
int i;                   /* index into vectors */

/* An initial version of X. */
wrMean = WMTools::MarkMean( wr , width*height );
for( i = 0; i < width*height; i = i + 1 )
    X[ i ] = wr[ i ] - wrMean;

/* An initial version of Y. */
coMean = WMTools::MarkMean( co , width*height );
for( i = 0; i < width*height; i = i + 1 )
    Y[ i ] = (double) co[ i ] - coMean;

/* Apply Gram-Schmidt orthonormalization to obtain two orthogonal
   unit vectors. */
WMTools::Orthonormalize( X, Y , width*height);
```


Code pour le calcul de la région d'insertion

```
...

/* Find projection of co into the XY plane. */
xco = WMTools::ImgPatInnerProduct( co, X, width, height );
yco = WMTools::ImgPatInnerProduct( co, Y , width, height );

/* Find unit vector in the XY plane that has a normalized correlation
   with the watermark of tcc + beta */
xt = tcc + beta;
yt = sqrt( 1 - xt * xt );

/* Find the point on the line described by xt,yt that is closest to
   xco,yco */
xcw = xt * (xt * xco + yt * yco);
ycw = yt * (xt * xco + yt * yco);

/* Project xcw,ycw back into mark space */
for( i = 0; i < width*height; i = i + 1 )
    cw[ i ] = xcw * X[ i ] + ycw * Y[ i ] + coMean;

/* Delete X and Y */
delete [] X;
delete [] Y;
}
```

Réflexion sur la robustesse des deux approches

- Pour la corrélation linéaire, augmenter le seuil de détection revient à augmenter la robustesse (on déplace l'hyper-plan),
- Pour la corrélation normalisée, augmenter le seuil de détection revient à diminuer l'angle du cône et donc obtenir un point c_w plus proche de l'origine du cône. Un bruit (attaque) sur l'image c_w peut alors plus facilement faire sortir le point du cône de détection.

Critère de robustesse pour la corrélation linéaire

Problématique

La corrélation normalisée ne mesure pas directement la robustesse.

Il faut trouver une mesure fonction d'un bruit que peut subir le vecteur c_w avant de sortir de la région de détection. Il est intéressant de choisir un bruit blanc Gaussien car la corrélation normalisée n'y est pas spécialement robuste et de plus c'est un modèle simple.

Equation de la robustesse pour un niveau de bruit fixé

Supposons un bruit blanc Gaussien \mathbf{n} ajouté à l'image tatouée c_w modélisant une attaque. La corrélation normalisée au détecteur est alors :

$$z_{nc}(c_w + n) = \frac{(c_w + n) \cdot w_r}{|c_w + n| |w_r|}$$

En supposant que le bruit \mathbf{n} est probablement orthogonal à c_w et à w_r , nous obtenons :

$$z_{nc}(c_w + n) \approx \frac{c_w \cdot w_r}{\sqrt{c_w \cdot c_w + n \cdot n} |w_r|}$$

Equation de la robustesse pour un niveau de bruit fixé

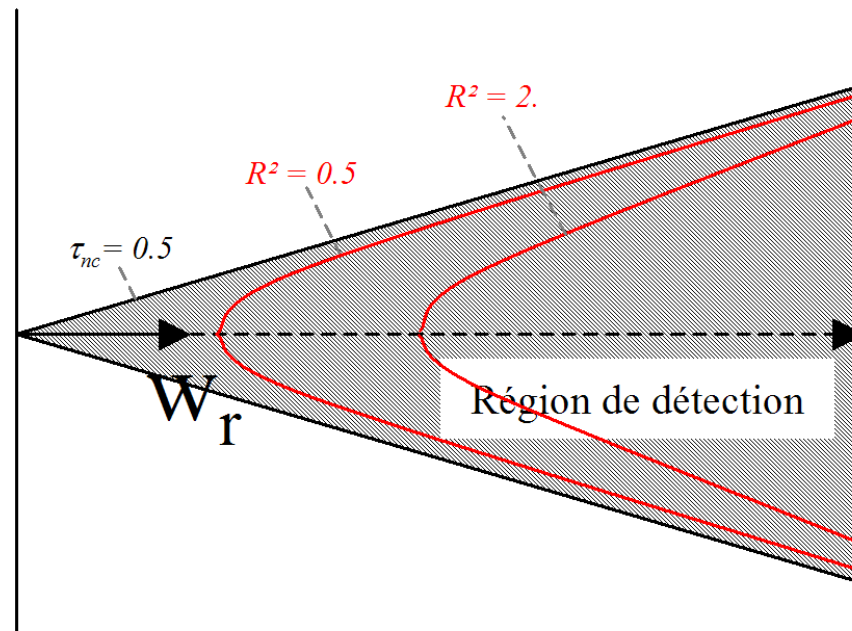
Nous cherchons à trouver l'amplitude du bruit $R = \sqrt{n \cdot n}$ qui fait passer $z_{nc}(c_w + n)$ sous le seuil τ_{nc} . En remplaçant $z_{nc}(c_w + n)$ par τ_{nc} , nous obtenons :

$$R^2 = \left(\frac{c_w \cdot w_r}{\tau_{nc} |w_r|} \right)^2 - c_w \cdot c_w \quad (1)$$

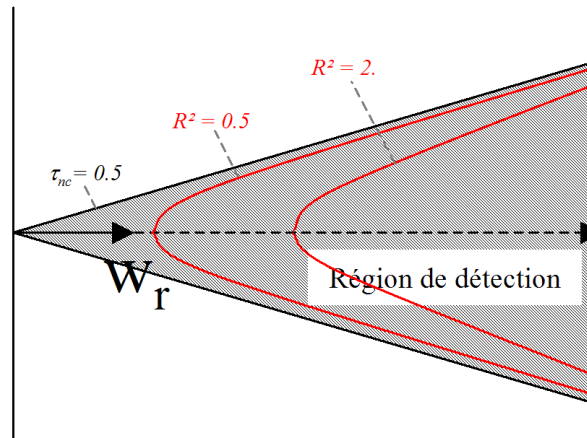
... si l'on fixe un niveau de bruit R^2 , on peut déduire c_w . On vient donc de définir une **mesure grossière** de robustesse.

Représentation graphique de la région d'insertion pour un niveau de bruit R^2 constant

Pour être robuste à un bruit \mathbf{n} blanc Gaussien $R^2 = \mathbf{n}.\mathbf{n}$ constant, fixé par l'utilisateur, la zone d'insertion (zone de c_w) est une moitié d'hyperbole de dimension N .



Solution approchée de la projection sur la demi hyperbole

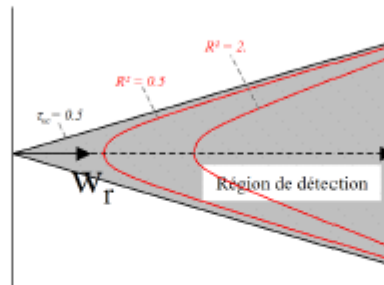


$$R^2 = \left(\frac{c_w \cdot w_r}{\tau_{nc} |w_r|} \right)^2 - c_w \cdot c_w$$

À l'insertion, on doit résoudre une équation quadratique pour déterminer c_w sachant c_o , w_r et R^2 . Par simplicité, on peut faire une recherche exhaustive en testant la distance entre c_o (exprimée dans (X, Y)) et des points d'abscisses dans l'intervalle $[0, y_{c_o}]$ appartenant à l'hyperbole.

Solution approchée de la projection sur la demi hyperbole

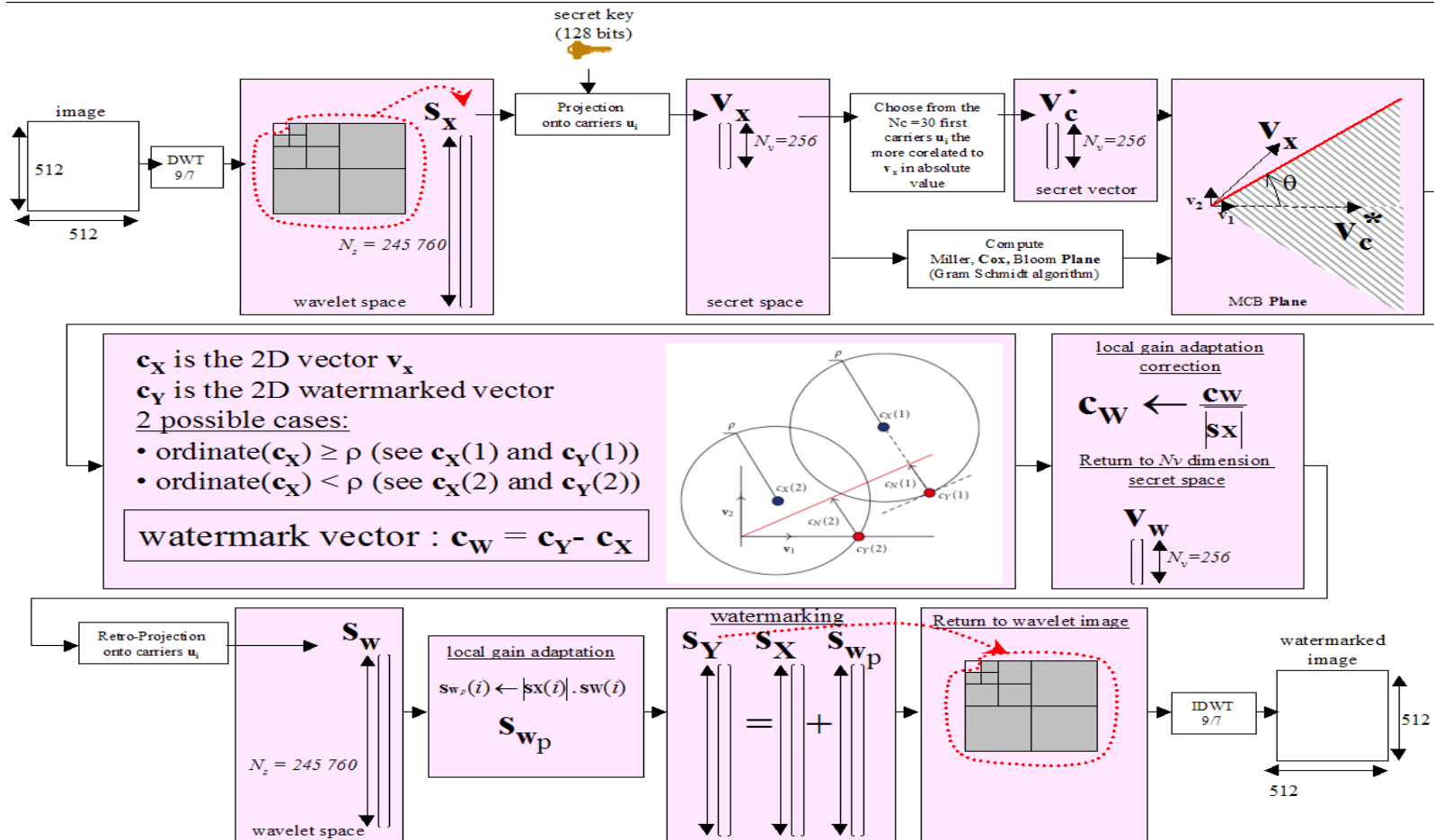
$$R^2 = \left(\frac{c_w \cdot w_r}{\tau_{nc} |w_r|} \right)^2 - c_w \cdot c_w$$



Dans le repère (\mathbf{X}, \mathbf{Y}) , soit $c_w = (x_{c_w}, y_{c_w})$ et $w_r = (|w_r|, 0)$. On a alors :

$$x_{c_w}^2 = \frac{\tau_{nc}^2 (R^2 + y_{c_w}^2)}{1 - \tau_{nc}^2}$$

Insertion par « flèche brisée »



Tatouage numérique avec information
(2ème génération) :

Rappel général
Introduction au tatouage informé

L'insertion informée :

Le principe de l'insertion informée
Insertion informée à corrélation linéaire
Insertion informée à corrélation normalisée
Schéma d'insertion « Broken Arrows »

Le codage informé :

Le principe du codage informé
Technique de tatouage QIM
Dirty-paper Trellis code
Travail à effectuer

CONTACT :

Email : kouider@lirmm.fr
<http://www2.lirmm.fr/~kouider>

Dissimulation de données

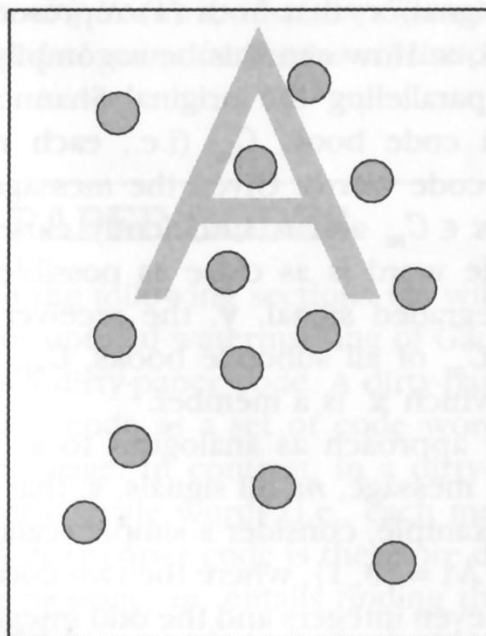
III. Codage Informé

Résultats de Cox 1983

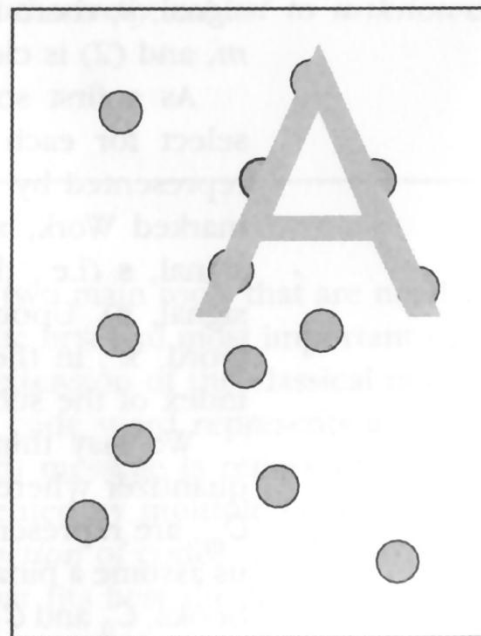
Problème

Soit un papier couvert **de taches de saletés** d'intensités distribuées selon une loi normale. On écrit alors un message sur ce papier avec une quantité limitée d'encre. Le papier sale est alors envoyé avec le message et est également taché selon une loi normale. Si le récepteur ne peut distinguer l'encre de la saleté, combien d'information fiable peut-on envoyer ?

Résultats de Cox 1983



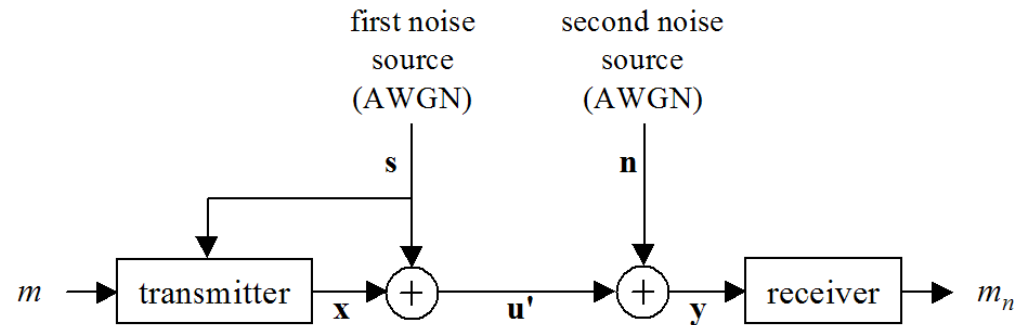
Blind writing



Informed writing

Ecriture sur un papier sale

Résultats de Cox 1983



Problème équivalent :

Un canal de transmission possède 2 sources de bruit indépendantes Gaussiennes. Avant de transmettre x , l'émetteur connaît s le premier bruit. Le signal x est transmis avec une puissance limitée : $\frac{1}{N} \sum_i x[i] \leq p$. Le deuxième bruit n est inconnu.

Résultat prouvé par Max Costa en 1983

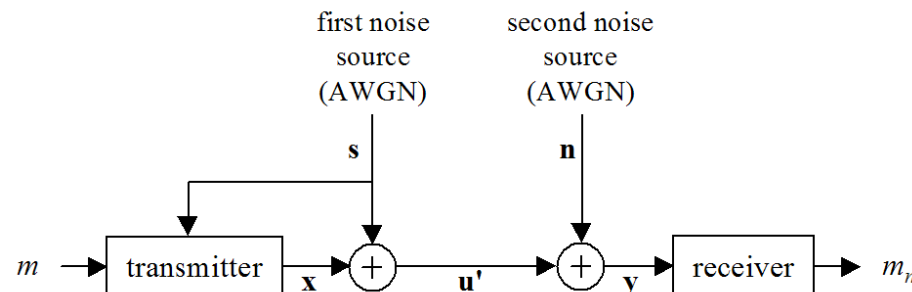
Le premier bruit s n'a pas d'influence sur la capacité du canal

Résultats de Cox 1983

Ce résultat peut être appliqué pour le tatouage :

- s (le papier) \equiv signal hôte,
- x (l'écriture du message) \equiv pattern ajouté,
- p (quantité d'encre) \equiv contrainte de fidélité,
- n (saleté supplémentaire) \equiv attaque.

Le résultat de Costa, appliqué au tatouage indique que la quantité d'information que l'on peut embarquer via le tatouage ne dépend pas du signal couverture. La restriction quant à la quantité d'information que l'on peut insérer est due au deuxième bruit ; une redondance de codage est donc nécessaire pour que la transmission soit correcte.



Codage informé - Principe

Principe du codage informé

Le codage informé repose sur la construction d'un code **c** à partir du document hôte **x**.

Un dictionnaire de mots de codes est construit. Pour chaque message, le codage informé met à disposition plusieurs mots de code représentant le message. Le mot de code **c** est **choisi** parmi ces mots à **l'aide du signal hôte**. Le codage informé s'appuie sur les principes du schéma de Costa.

Codage informé - Principe

Par exemple, on souhaite transmettre un des 4 messages A,B,C,D. On choisit un ensemble \mathcal{U} de vecteurs, on le divise en 4 sous-ensemble (coset) $\mathcal{U}_A, \mathcal{U}_B, \mathcal{U}_C, \mathcal{U}_D$. **Chaque mot de code d'un coset représente un message.**

Le codeur choisit lorsqu'il veut coder un message m le mot de code u du (coset \mathcal{U}_m) qui est le plus proche du vecteur hôte s .

Le codeur transmet alors $x = u - s$ (cf. schéma).

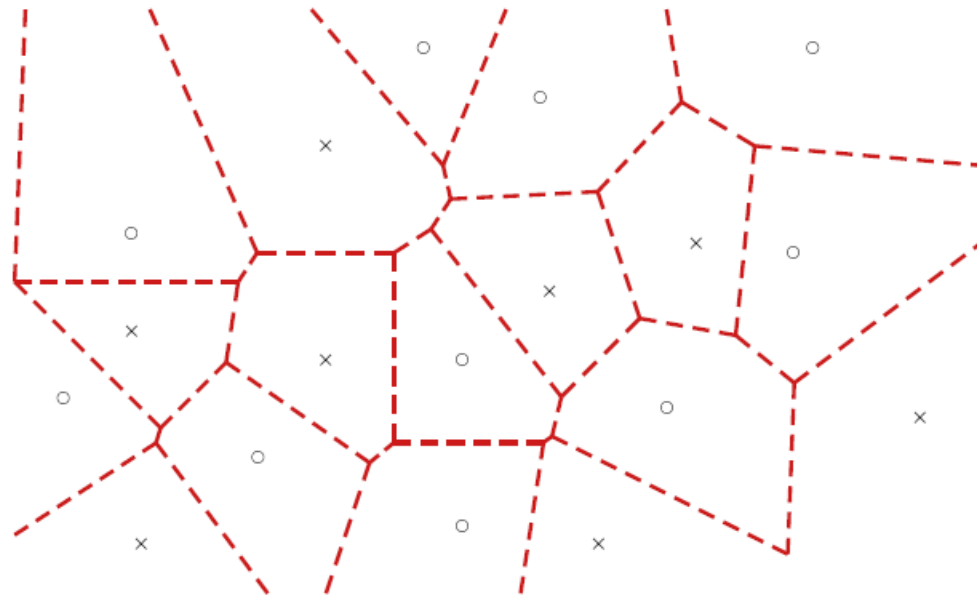
Le récepteur reçoit le vecteur y et détermine le mot de code u'' le plus proche de y , identifie le coset $\mathcal{U}_{m'}$ et donc le message m' .

Codage informé - Principe

Problème de l'approche : il est possible que certains mots de code u représentant un message m soient trop éloignés de s (contrainte de puissance non respectée). Plutôt que de transmettre $x = u - s$ on préfère transmettre $x = \alpha(u - s)$ avec α un scalaire de $[0, 1]$.
Remarque : après l'ajout du signal hôte, on a $u' = \alpha u + (1 - \alpha)s$.
Le choix de α dépend du canal.

Technique de tatouage QIM

A Voronoï view on QIM :

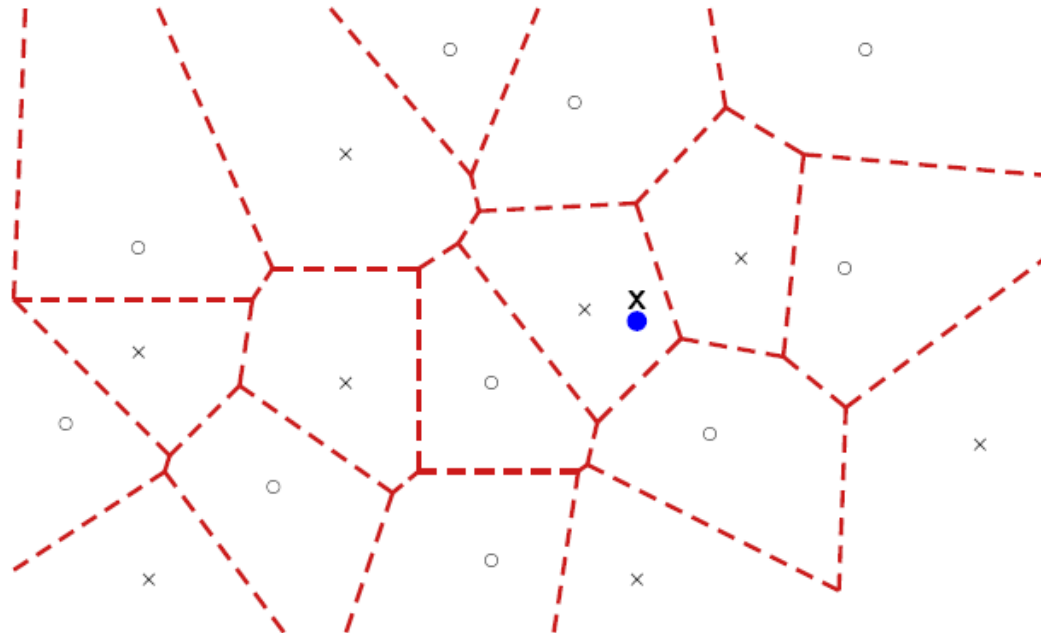


Binary (DC-)QIM. Quantizer for 0's (resp. 1's) are the \circ 's (resp. \times 's).

Dictionnaire de mots de codes

Technique de tatouage QIM

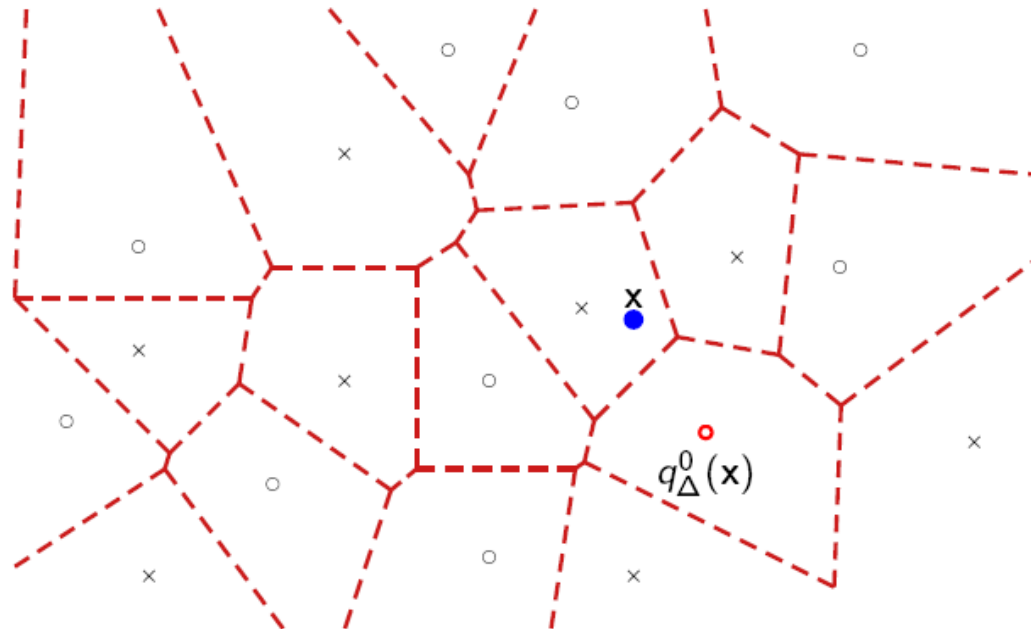
A Voronoï view on QIM and SCS



Binary (DC-)QIM. Quantizer for 0's (resp. 1's) are the \circ 's (resp. \times 's).

Technique de tatouage QIM

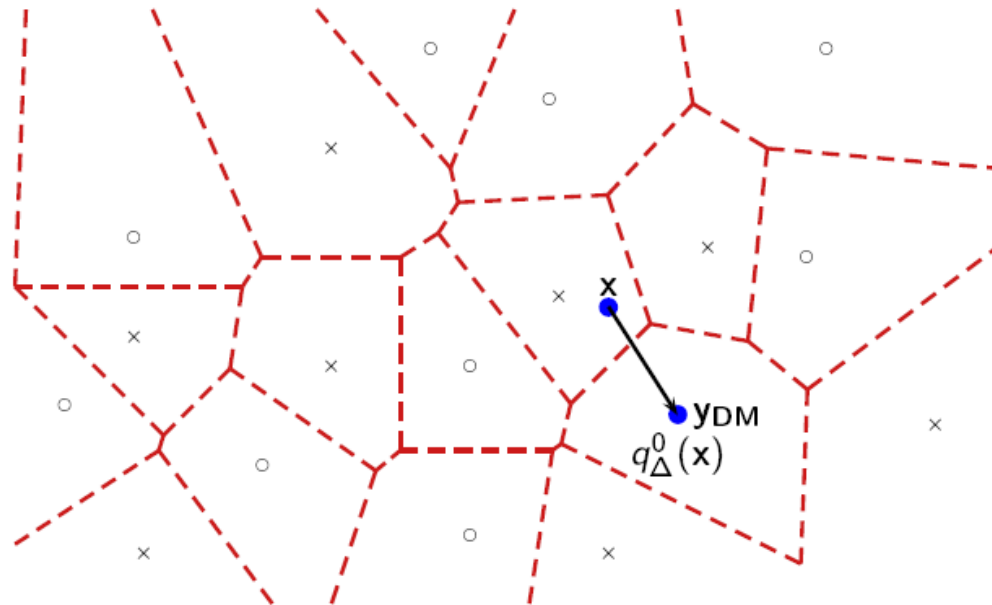
A Voronoï view on QIM and SCS



Binary (DC-)QIM. Quantizer for 0's (resp. 1's) are the o's (resp. x's).

Technique de tatouage QIM

A Voronoï view on QIM and SCS



Binary (DC-)QIM. Quantizer for 0's (resp. 1's) are the \circ 's (resp. \times 's).

Codage dirty-paper trellis code

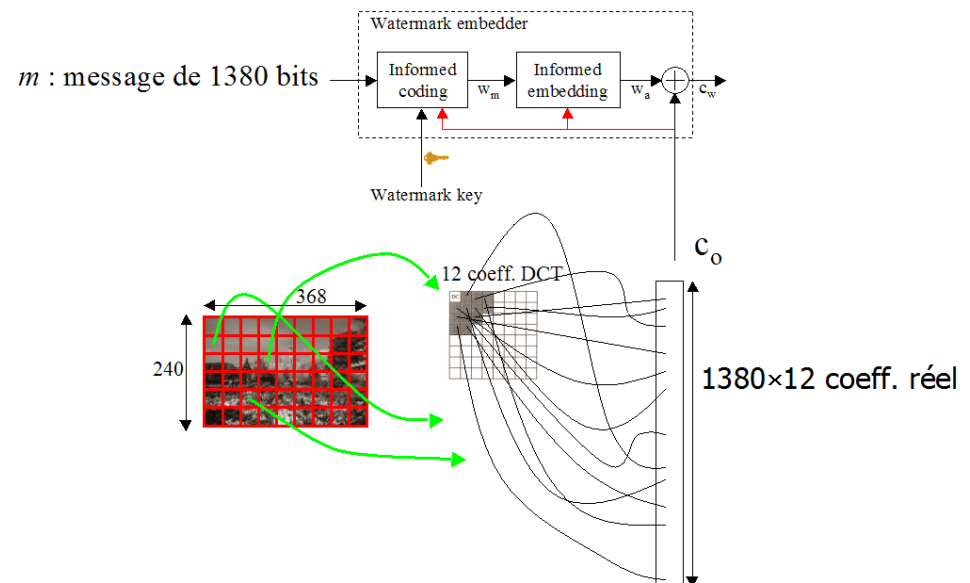
"Applying Informed Coding and Informed Embedding to Design a Robust, High Capacity Watermark". M. L. Miller, G. J. Doërr and I. J. Cox, In IEEE Transactions on Image Processing, 13(6) :792-807, 2004.

On souhaite embarquer un message m de taille L (ex : $L = 1380$ bits) dans une image de taille N (ex : $N = 240 \times 368$).

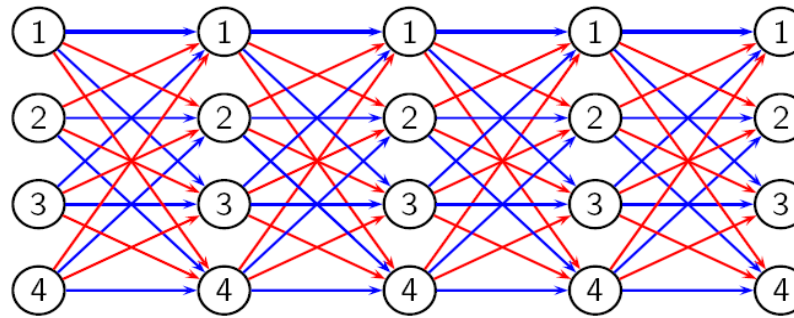
Codage dirty-paper trellis code

On souhaite embarquer un message m de taille L (ex : $L = 1380$ bits) dans une image de taille N (ex : $N = 240 \times 368$).

Soit c_o le vecteur hôte (espace d'insertion) contenant les 12 premiers coefficients ACs des blocs DCT 8×8 mélangés; le vecteur c_o est donc composé de $12 \times N/64$ coefficients réels (On a $12 \times N/64 = 12 \times L = 12 \times 1380$)



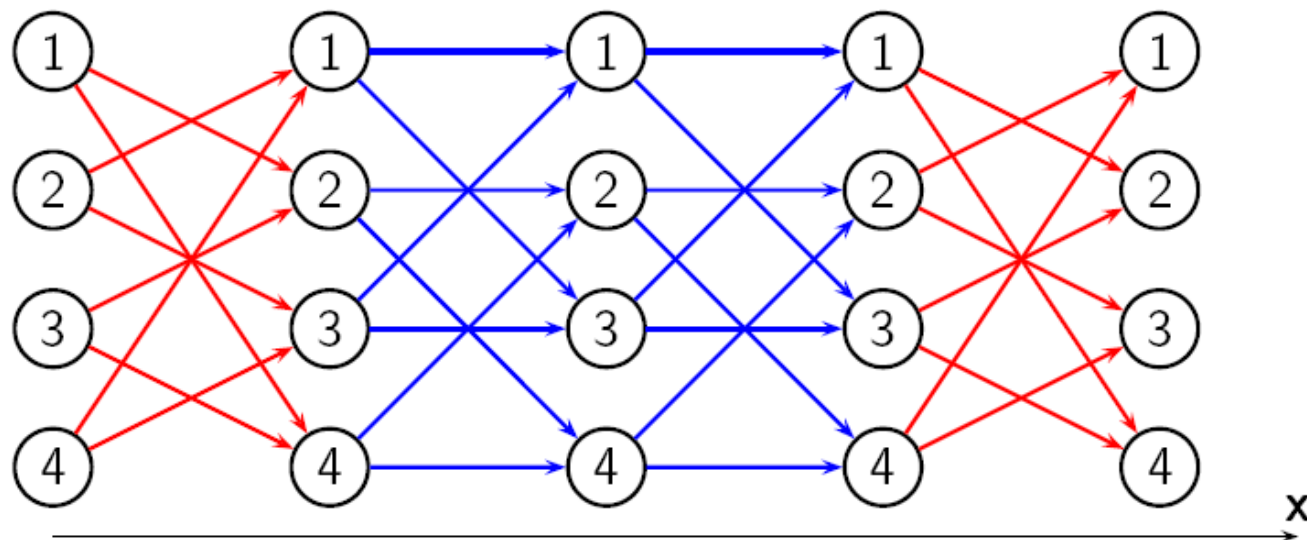
Codage dirty-paper trellis code



Définition du treillis :

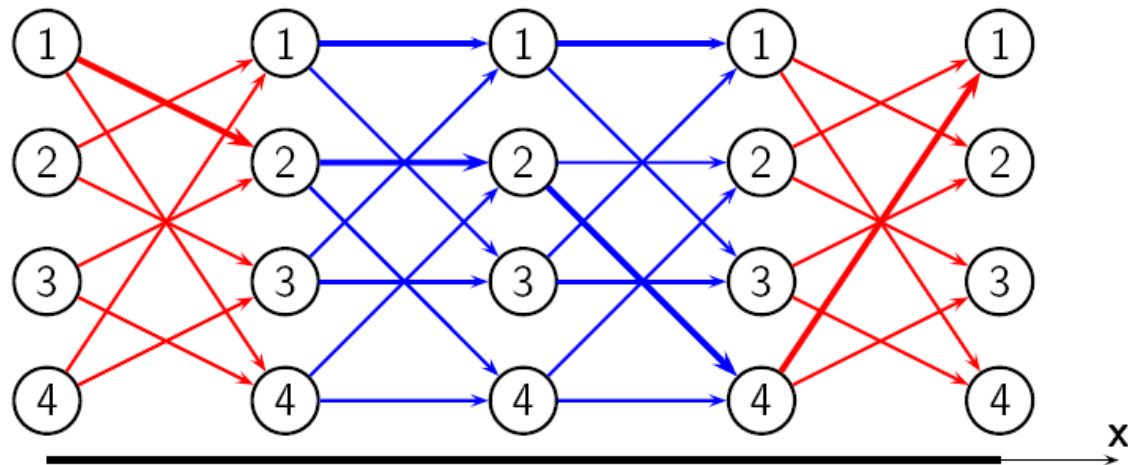
- 64 états,
- 64 arcs sortant d'un état,
- L étapes (exemple $L = 1380$),
- arc bleu = entrée 0, arc rouge = entrée 1,
- chaque arc est valué (sortie du codage) par une séquence pseudo-aléatoire de 12 coefficients réels.

Codage dirty-paper trellis code



Encoding of $\mathbf{m} = (1001)$

Codage dirty-paper trellis code



Encoding of $\mathbf{m} = (1001)$

On détermine le chemin le plus corrélé (produit scalaire) entre les arcs de sortie et le vecteur c_o (noté x ici) : Algorithme de Viterbi modifié.

B. Chen et G. W. Wornell : Quantization index modulation : a class of provably good methods for digital watermarking and information embedding. IEEE Transactions on Information Theory, 47(4):1423 –1443, May 2001. (QIM)

M. Miller, G. Doerr et I. Cox : Applying Informed Coding and Embedding to Design a Robust High Capacity Watermark. IEEE Transactions on Image Processing, 13(2):792–807, June 2004. (DPTC)

Travail à rendre avant la séance prochaine :

- Lire les deux articles et bien comprendre les deux méthodes QIM et DPTC,
- Rédiger un rapport de 4 à 5 pages décrivant et comparant les deux méthodes de tatouage.
- La traduction de mot à mot des articles n'est pas acceptée.

Les rapports en PDF devront être envoyés à sarra.kouider@lirmm.fr avant la séance prochaine, tout retard entraînera une diminution de la note.