Estimation du mouvement

Marco Cagnazzo

14 mai 2012

Plan

- Introduction
- La représentation du mouvement
- Les différentes techniques
- Applications et examples

Le mouvement dans les images

Elément naturel constitutif des séquences d'images

- Perception par le HVS (système visuel humain)

  Surtout en vision périphérique : - faible résolution
  - faible sensibilité aux couleurs

- Formation et nature des images

  - projection d'une scène 3D sur un plan mobile, puis sous-échantillonnage spatial et temporel

    ➔ objets en mouvement ?

  - but : visualisation/interprétation par l'homme, ou un système automatique.

    ➔ forte similarité entre images

POURQUOI ESTIMER LE MOUVEMENT ?
LE CODAGE DES IMAGES (1)

Les 2 idées de base de la compression:

- Exploiter la redondance spatiale: techniques de codage pixel, codage par transformée, codage fractal, ...
- Exploiter la redondance temporelle: en vidéo, il y a une forte similarité entre des images successives.

⇒ D 'où l 'idée de construire un codage spatio-temporel

POURQUOI ESTIMER LE MOUVEMENT ?
LE CODAGE DES IMAGES (2)

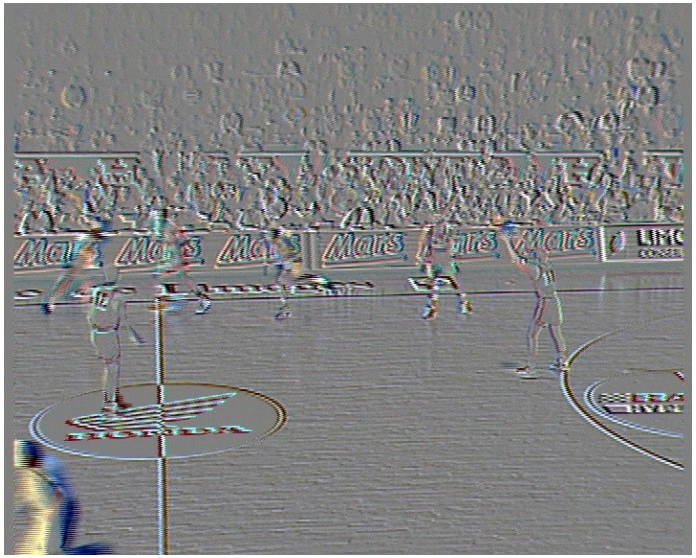
Le but est donc de diminuer la quantité d'information à transmettre

- ➡ Eliminer la redondance temporelle
- ➡ Mais aussi adapter le codage au contenu de l 'image :

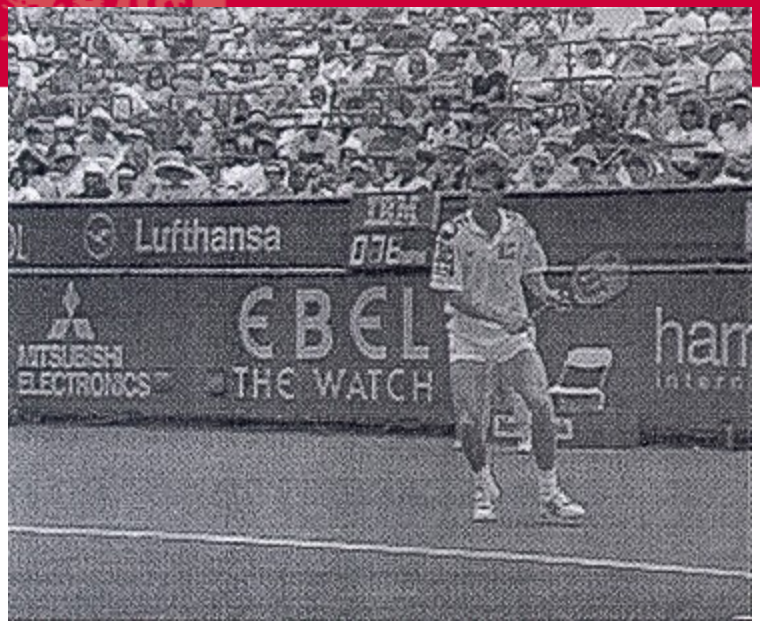
  - zone immobile : faible résolution temporelle. Inutile de tout coder, on fera du skipping.

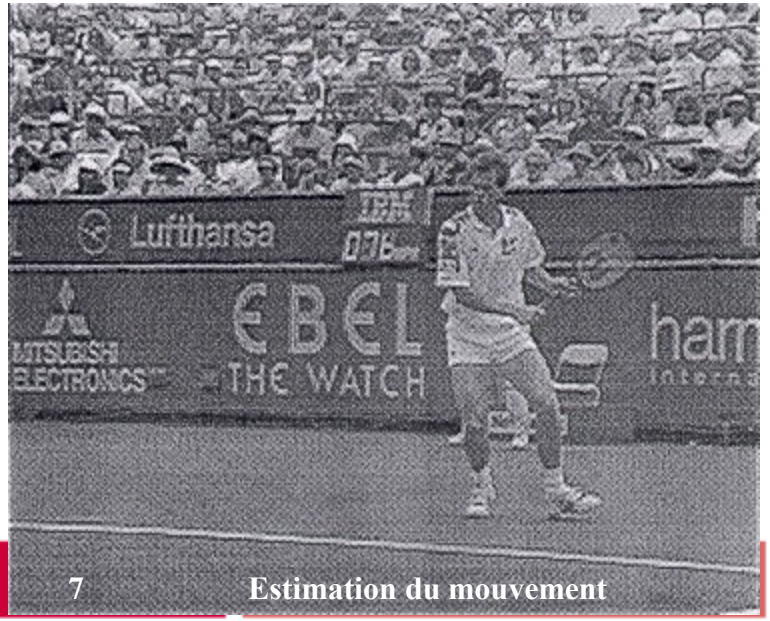
  - zone en mouvement : faible résolution spatiale (vue par les bâtonnets, en périphérie). Inutile de dépenser trop, on quantifiera plus, ou bien on sous-échantillonnera.

LA DIFFERENCE INTER-IMAGE



EXEMPLES D'IMAGES D'ERREUR





LA DIFFERENCE INTER IMAGE

- La différence inter-image n'est pas une bonne mesure de la redondance temporelle
  - ⇨ Projeter sur l 'axe des mouvements est meilleur
  - ⇨ Les mouvements sont souvent peu complexes

⇨ Estimation de mouvement

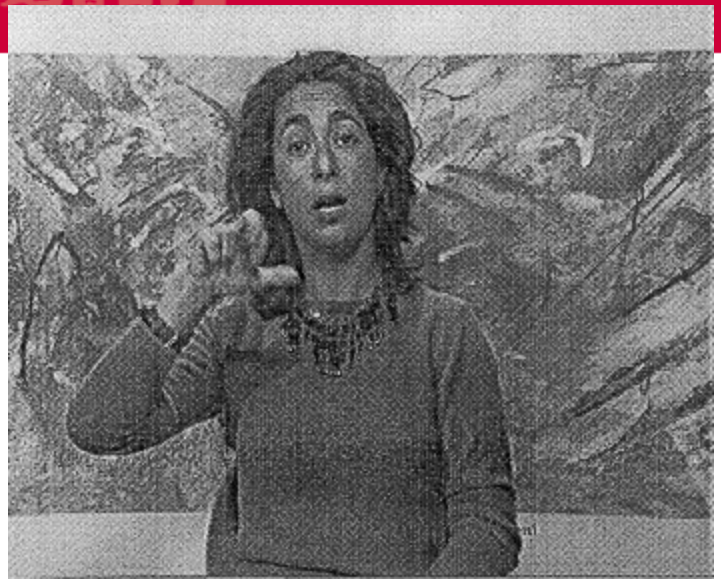
Prédiction par compensation de la référence

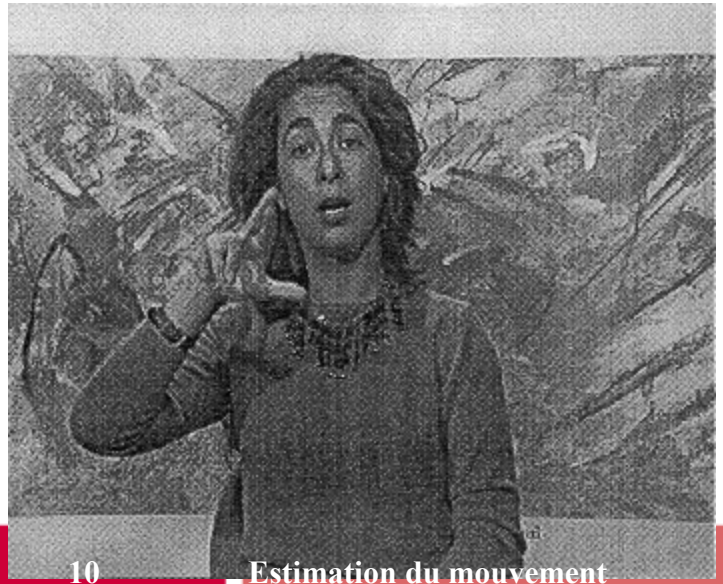
Codage du mouvement et de l'erreur

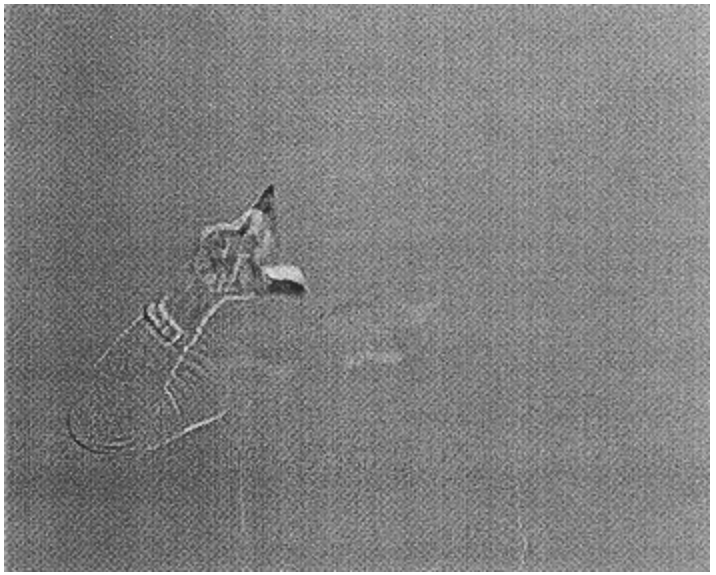
POURQUOI ESTIMER LE MOUVEMENT ?
L 'ANALYSE DES IMAGES

- Possibilités :
  - discrimination des objets (segmentation)
  - suivi des objets (tracking)
- Applications :
  - surveillance, reconnaissance…
  - militaire (guidage de missiles…)
  - codage (sprites, mosaïques)

EXEMPLE: RECONNAISSANCE DE FORMES







LA PROBLEMATIQUE DU MOUVEMENT : LE 'VRAI' MOUVEMENT

Différence entre mouvement et changement de luminance

- **Tout mouvement n'implique pas un changement de luminance:**

  - objets uniformes ou de texture invariante selon la direction du déplacement

- **La réciproque est 'encore moins' vraie:**

  - variations d'illumination (ombres, réflexions, flashs)

  - bruit

LA PROBLEMATIQUE DU MOUVEMENT : LE 'VRAI' MOUVEMENT

- **'Vrais' mouvements ou changements de luminance?**

  but : analyse / codage

  => 2 hypothèses dans les estimateurs de mouvement:

- **L'hypothèse de la luminance constante le long des trajectoires:**

  $L(x, y, t) = L(x-dx, y-dy, t-1)$ pour tout point de l'image

- **Prise en compte des variations d'illumination:**

  $L(x, y, t) = a(x,y,t)\, L(x-dx, y-dy, t-1)$

DUALITE ESTIMATION / SEGMENTATION

- **Problème de l'œuf et de la poule...**

  - pour estimer le mouvement correctement, il faut connaître les zones de mouvement homogène.
  - pour délimiter ces zones, il faut connaître le mouvement dans l'image.

- **Les solutions**

  - estimation / segmentation menées de façon conjointe (robustesse de l'estimation)

DUALITE ESTIMATION / SEGMENTATION (2)

… mais l'ambiguïté forme / mouvement demeure

- Pour représenter un mouvement, il faut

  - définir un modèle de mouvement: transformation définie par ***n*** paramètres

  - définir la/les zone(s) de l'image où l'appliquer

- **Le modèle contient à la fois les informations de mouvement et de surface de la zone considérée.**

LA COMPENSATION DE MOUVEMENT

- **On a vu que la différence inter-image n'était pas une bonne solution (trop de paramètres).**

- **Principe de la compensation :**

A partir :

- d 'une image de référence I_R
- du champ des déplacements estimés de l'image I_R à l 'image I

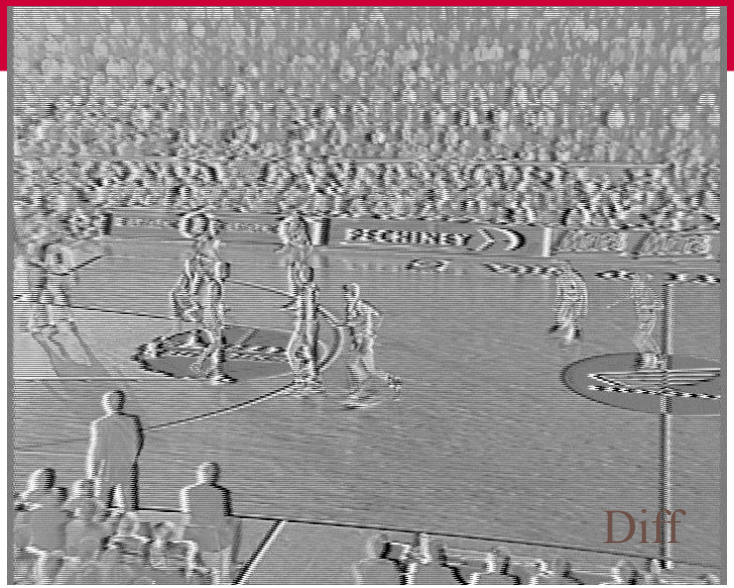
on reconstruit une image compensée I_C, prédiction de I

- **Exemple :**

$D(x,y) = (dx,dy)$

$I_C(x,y) = I_R(x - dx, y - dy)$

EXEMPLES

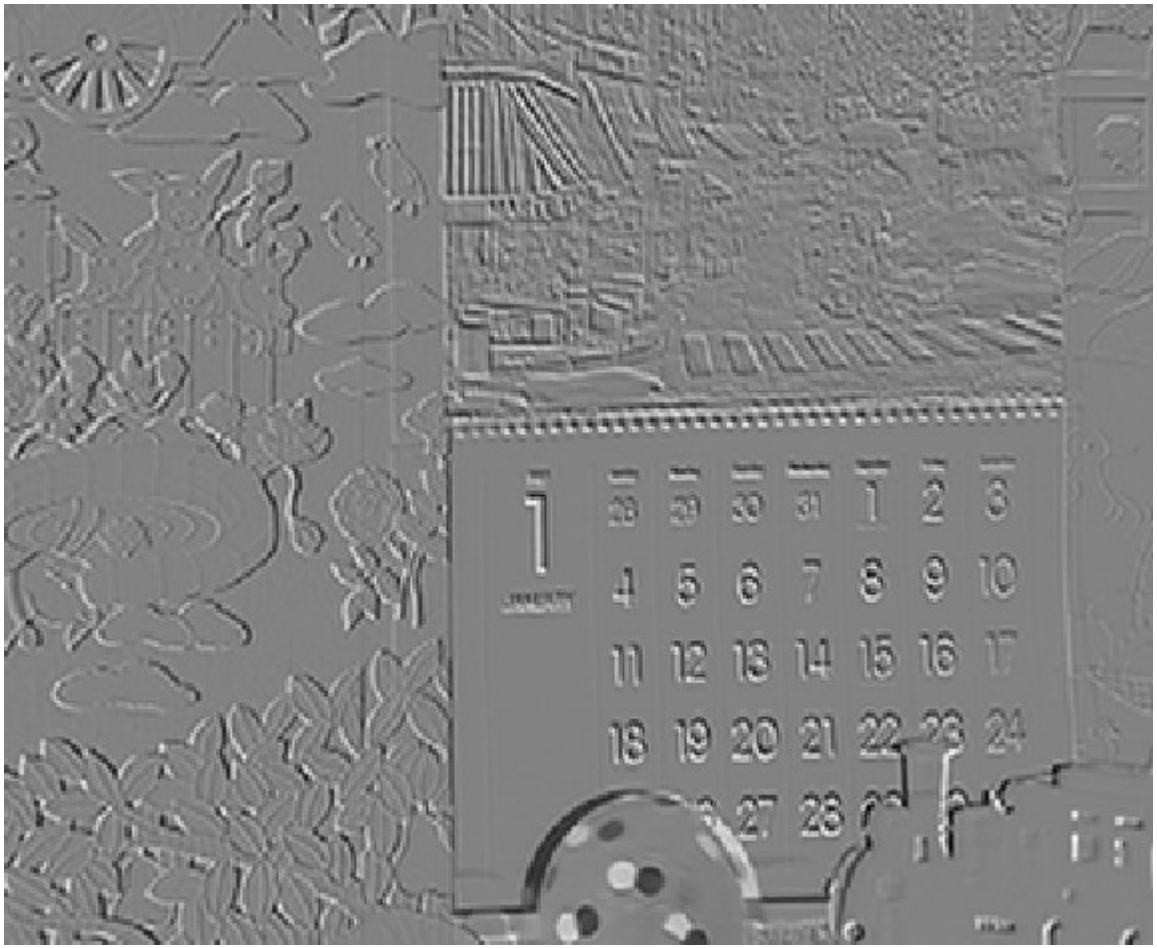


TELECOM ParisTech

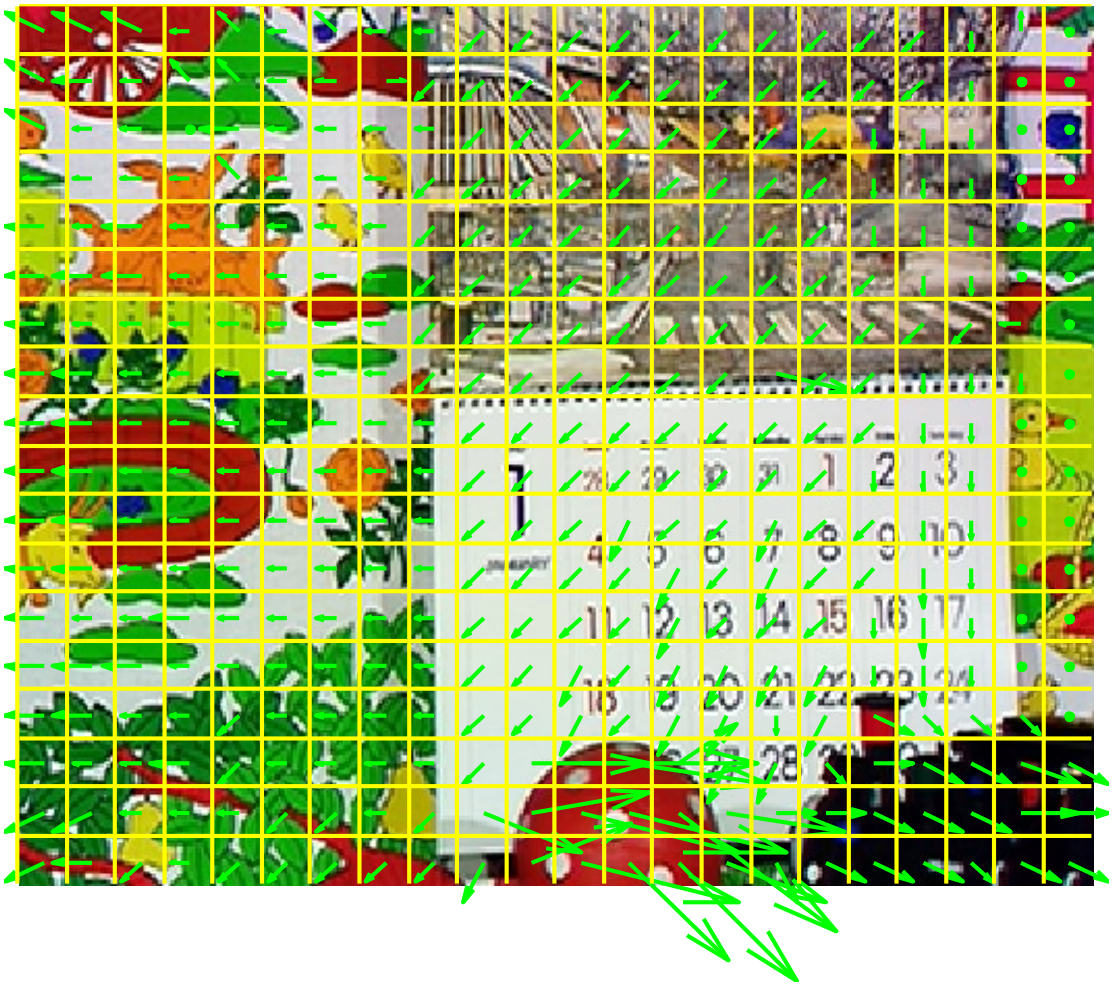
Reference image

Current image

Difference image



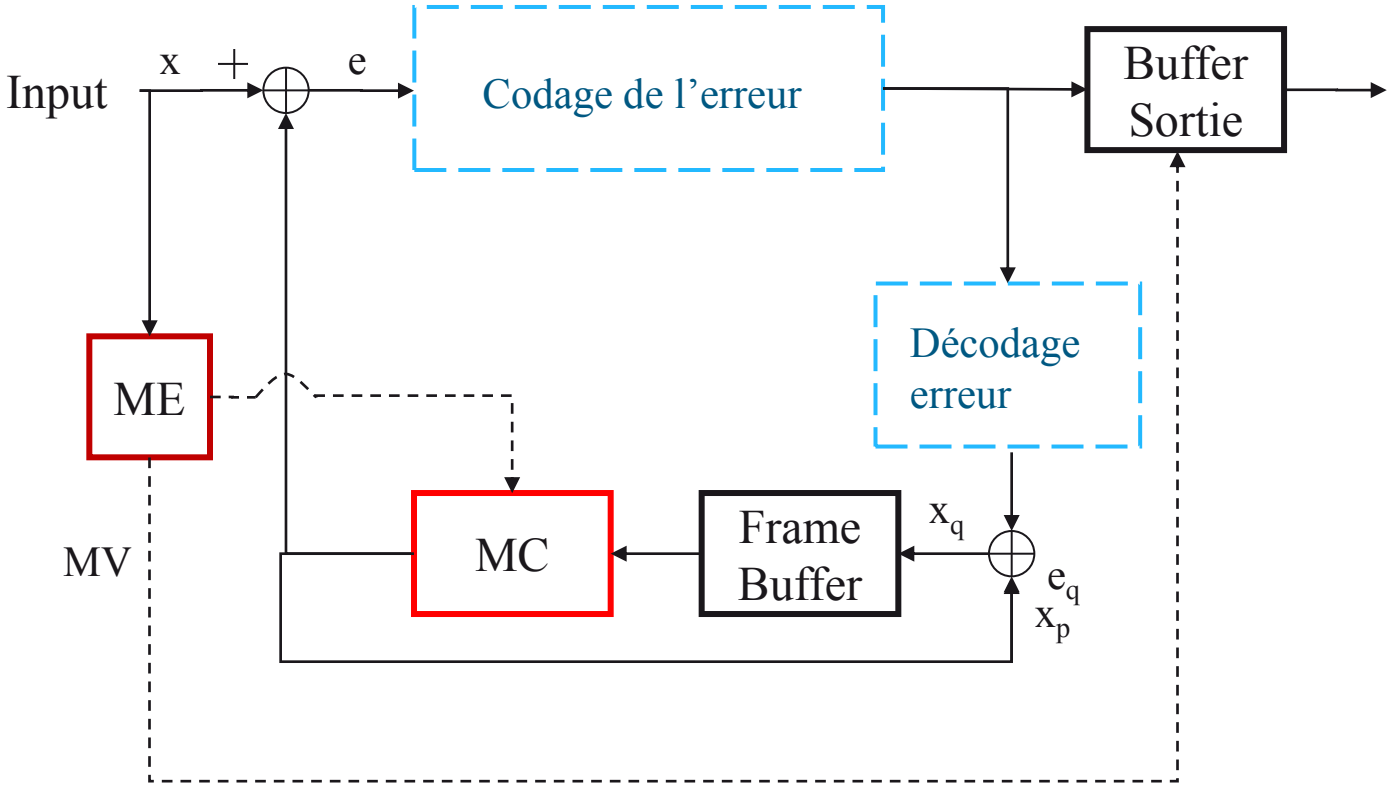
Motion vector field



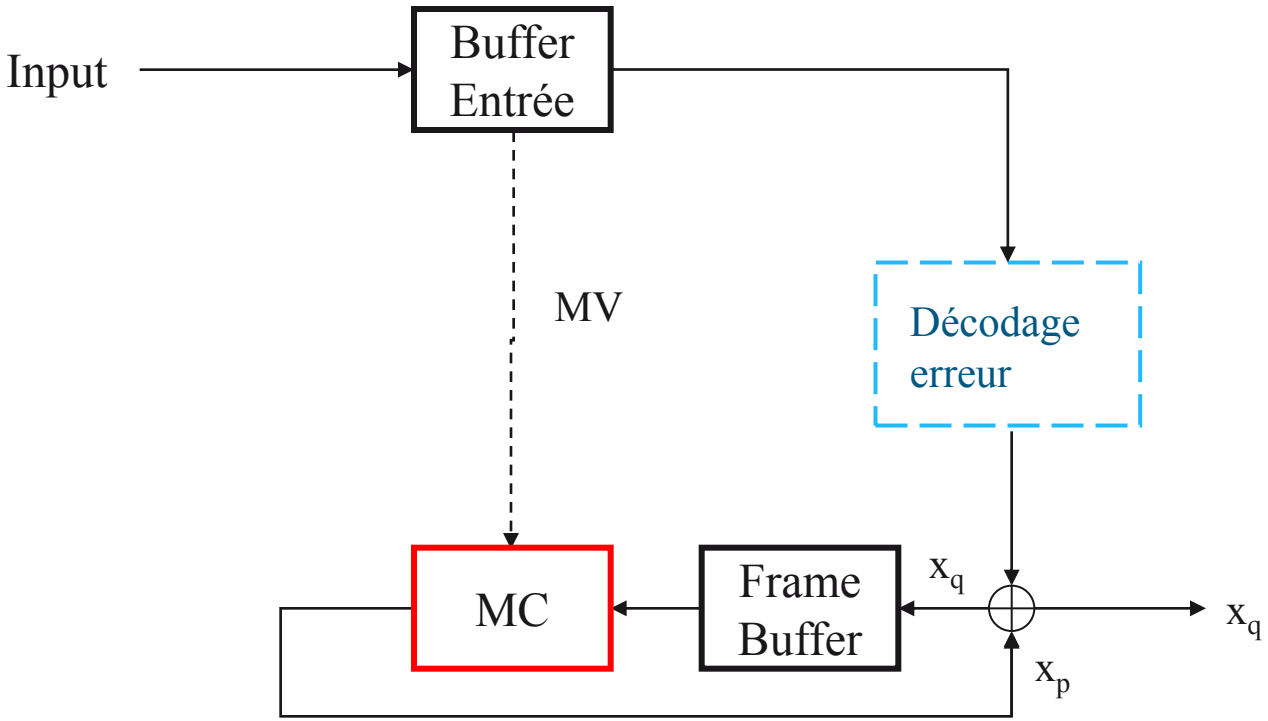
Motion-compesated image

MC error

D'OU LE SCHEMA CLASSIQUE DE CODAGE



DECODEUR


Input
Buffer
Entrée
MV
Décodage
erreur
MC
Frame
Buffer
x_q
x_q
x_p

LA PROBLEMATIQUE DU MOUVEMENT: LES OCCLUSIONS

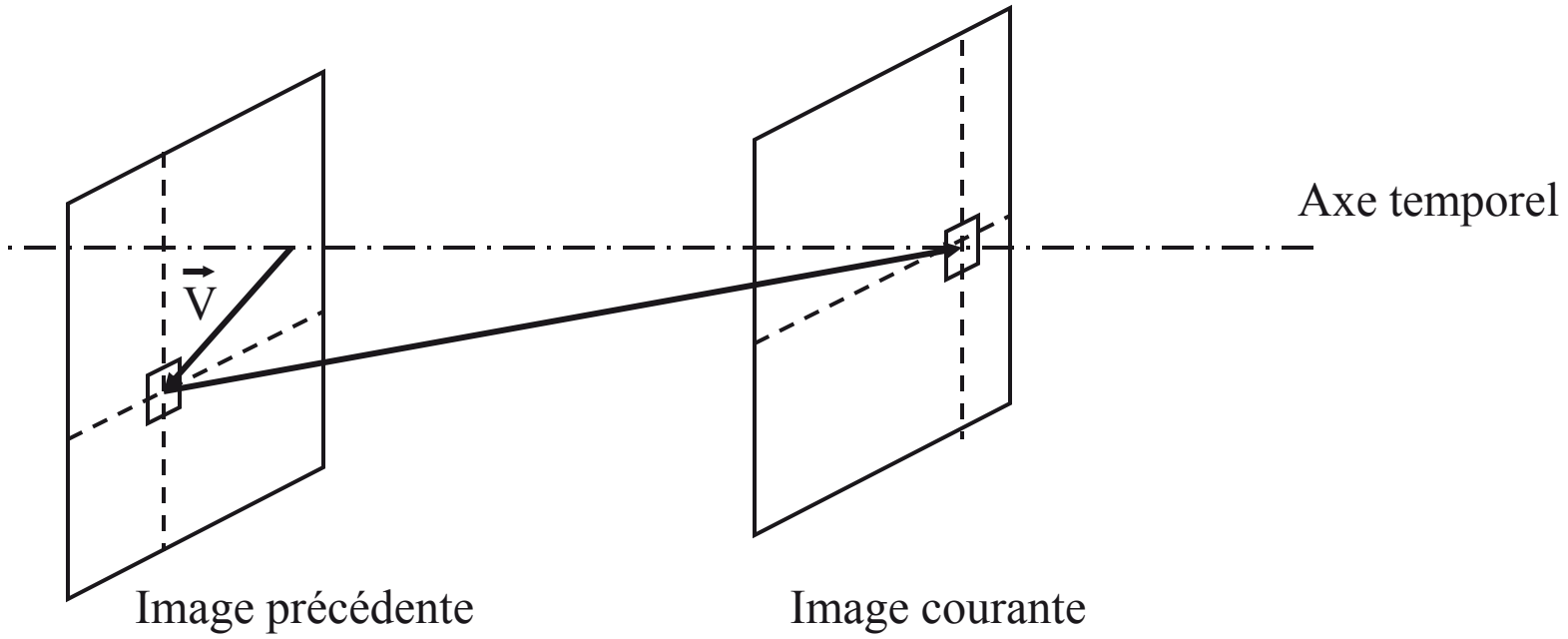
LES MODES DE COMPENSATION

Occlusions => le codage n'est pas forcément causal!

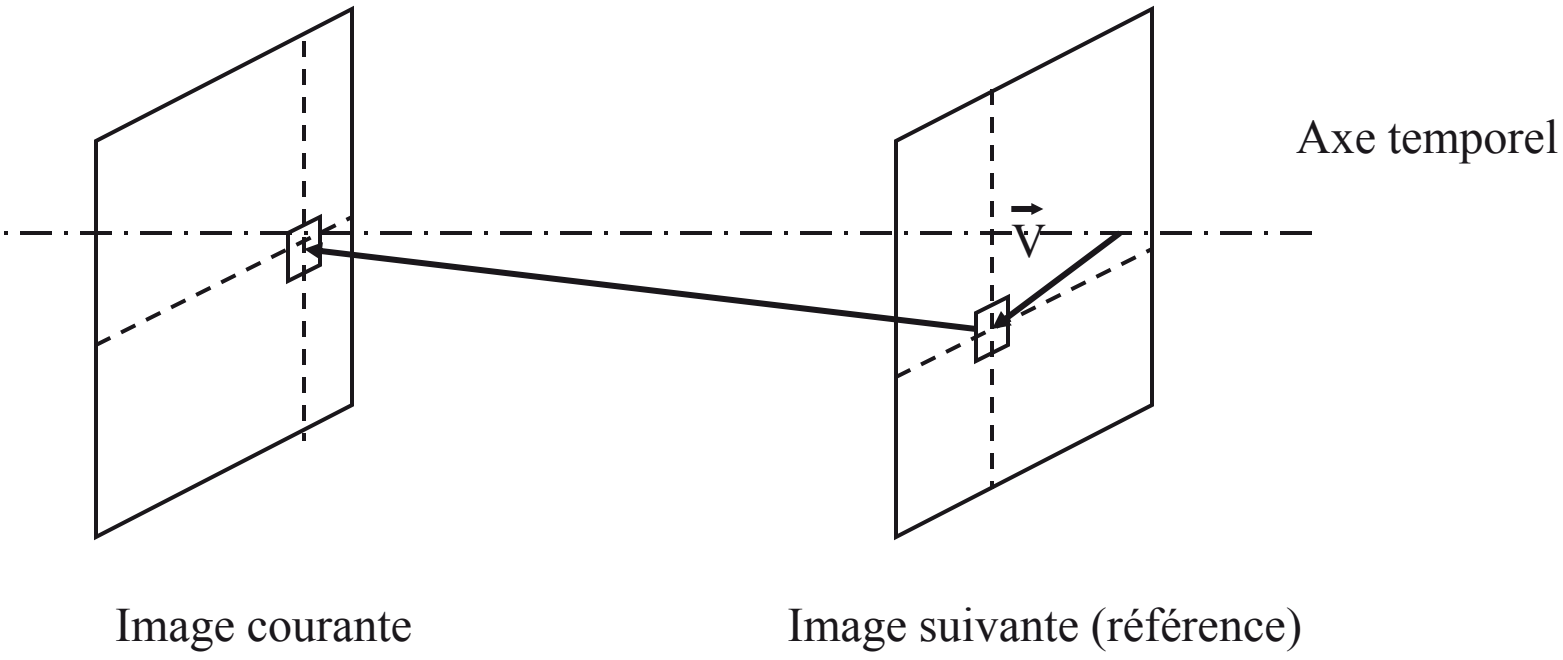
Il vaut mieux parfois faire des prédictions par rapport au futur plutôt que par rapport au passé.

- **Compensation backward**
- **Compensation forward**
- **Compensation bidirectionnelle**

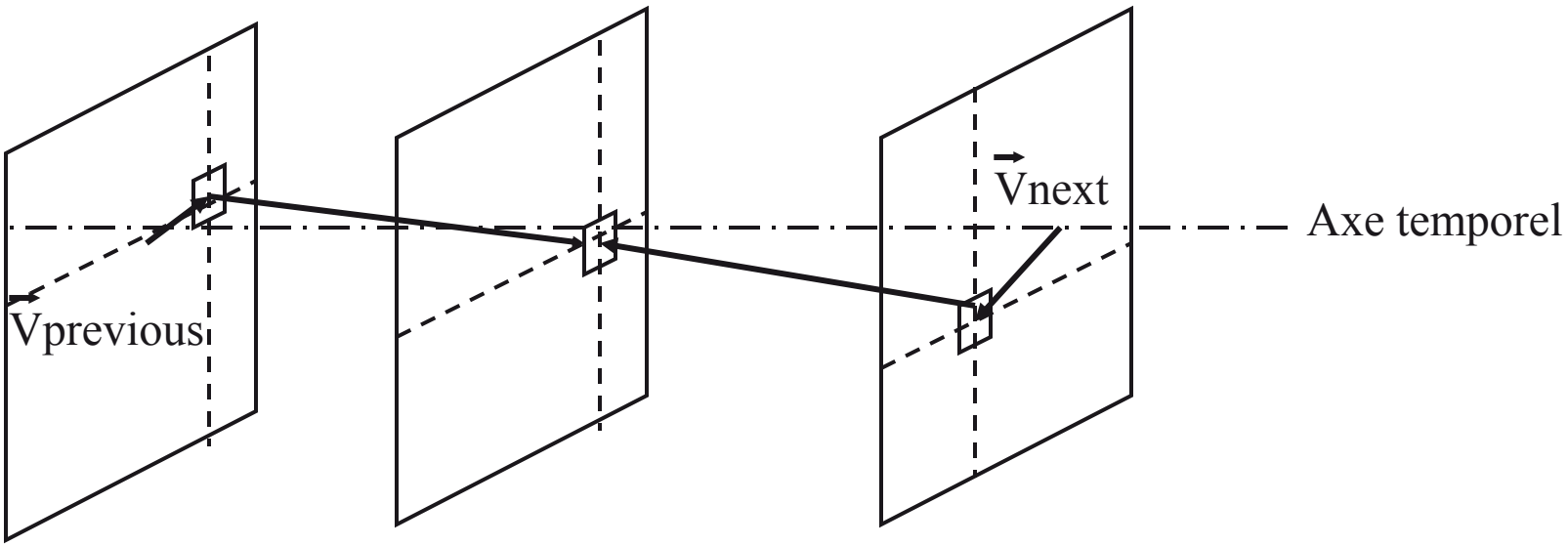
COMPENSATION FORWARD



COMPENSATION BACKWARD

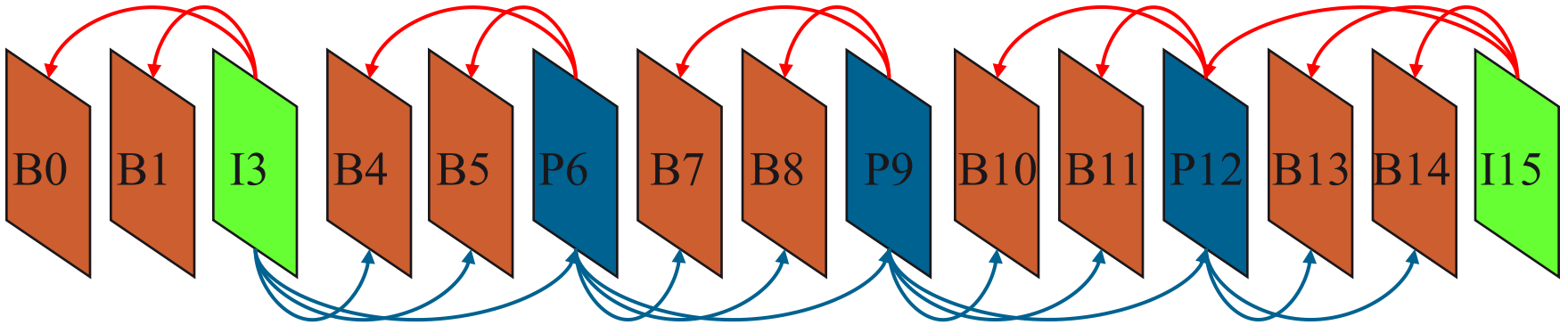


COMPENSATION BIDIRECTIONNELLE

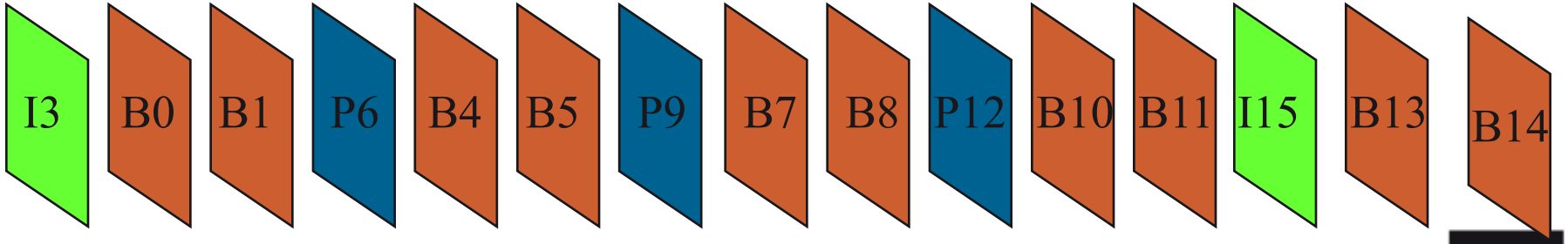


APPLICATION: CAS DE MPEG

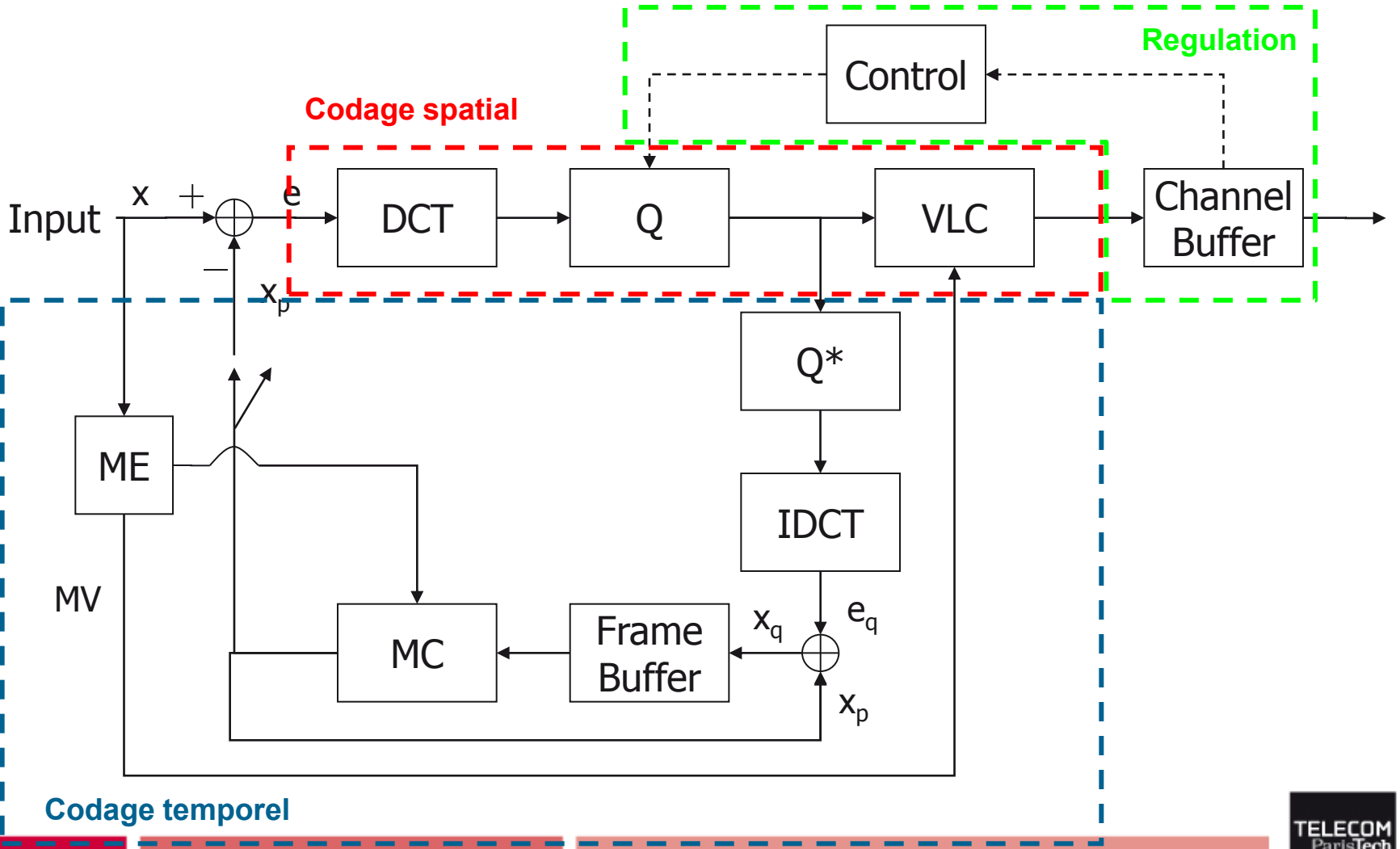
Ordre d'affichage:



Ordre de transmission:



SCHEMA CLASSIQUE DE CODAGE SPATIO-TEMPOREL HYBRID



LA PROBLEMATIQUE DU MOUVEMENT : L'EVALUATION DES RESULTATS

- **Ne pas perdre de vue que l'estimation de mouvement est un problème très complexe. Beaucoup de cas critiques:**

  - drapeau flottant au vent
  - l'eau, la mer
  - le bruit
  - flash, reflets

  } Définition d 'un test set important

- **L'évaluation est à relier au but recherché**

- En analyse : mouvement 'réel ', selon critère à définir
  => efficacité in fine du système d'analyse
- En codage : PSNR, quantité d 'information transmise
  => efficacité in fine du système de codage

L'EVALUATION DES RESULTATS EN CODAGE

2 méthodes

- qualité de reconstruction (PSNR, EQM), mais limité car ne permet pas de discriminer l'efficacité de l'estimation du mouvement de l'efficacité du codage spatial, de la régulation de débit etc…
- quantité d'information à transmettre (entropie). On effectue un codage à pas de quantification constant pour by-passer la régulation et on regarde le nombre de bits produit.

Généralement on combine les deux: relevé de la quantité de bits produite + inspection visuelle / PSNR.

L'EVALUATION DES RESULTATS EN CODAGE (2)

- **Prendre en considération le contexte de l'estimation**

 - ori->ori, cod->ori, connaissances à priori (segmentation, mouvement…)

- **Prendre en considération la complexité de l'estimation**

 - la rapidité de l'estimation, de la compensation : tailles d'images, complexités relatives
 - la complexité de l 'algorithme (implantation, HW/SW partitionning)

A titre d 'exemple, il vaut mieux faire une estimation de mouvement à partir d'une image codée / décodée. Mais les premiers codeurs MPEG utilisaient tous une estimation de mouvement sur les images originales pour des raisons de coût / temps calcul / mémoire.

SYNTHESE

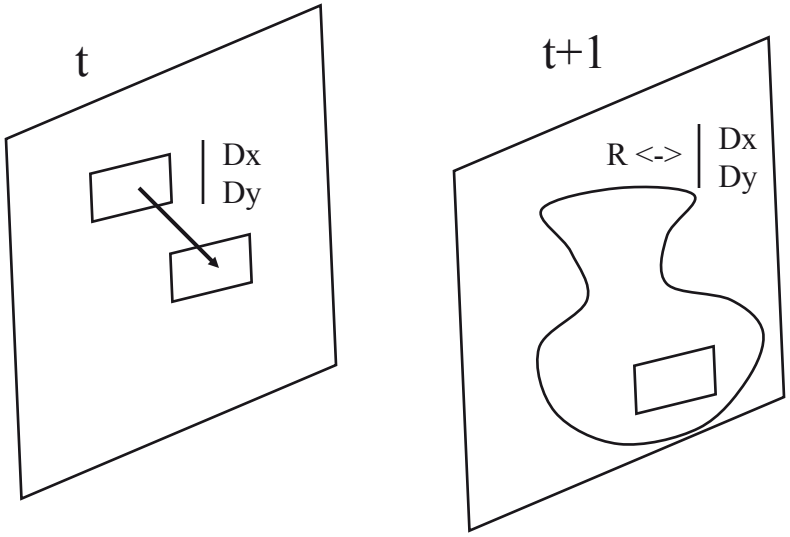
- But de l ’estimation de mouvement = non seulement suppression redondance temporelle (codage), mais ∃ beaucoup d’autres applications (tracking, reconnaissance, surveillance …)
- Notion de compensation de mouvement, meilleure que la différence inter-image
- Occlusion ➔ différents modes de compensation
- Evaluation des performances ➔ prendre en compte le contexte et la complexité de l’algorithme

Plan

- Introduction
- **La représentation du mouvement**
- Les différentes techniques
- Applications et examples

LA REPRESENTATION DU MOUVEMENT

- **Avant toute estimation de mouvement, il faut décider de sa description, i.e. du modèle mathématique par lequel on va représenter le mouvement 2D.**



- Le mouvement est représenté par :
  - l'équation formelle du modèle
  - les paramètres le définissant : paramètres à estimer
  - la zone sur laquelle ils s'appliquent

LE FLUX OPTIQUE

- **En théorie le problème est mal posé :**

  - image = projection 2D de la scène 3D

  - Pour trouver les mouvements 3D il faudrait analyser un flux d'images stéréos.

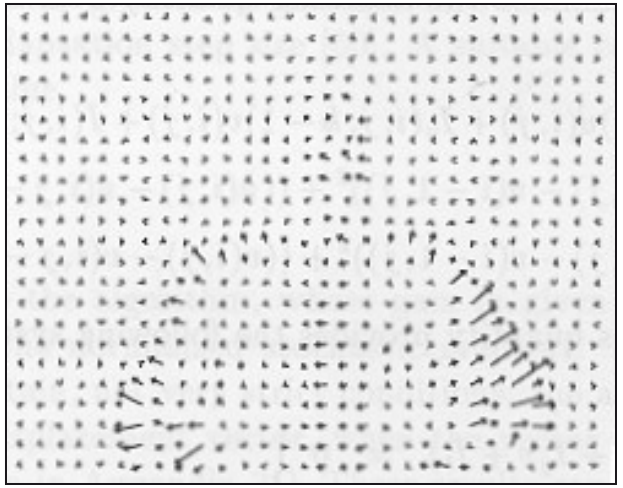
⇨ On ne cherchera pas les vrais vecteurs mais à évaluer les pseudo-vecteurs décrivant les mouvements des objets dans le plan image.

Ces pseudo-vecteurs constituent ce que l'on nomme le <u>flux optique.</u>

LE FLUX OPTIQUE

- Représentation précise, simple dans le principe mais coûteuse

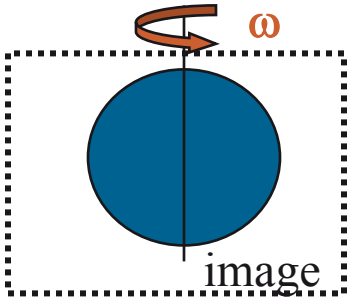
<u>1 pixel = 1 vecteur</u>



DIFFERENCE FLUX OPTIQUE / CHAMP DE VECTEURS REEL

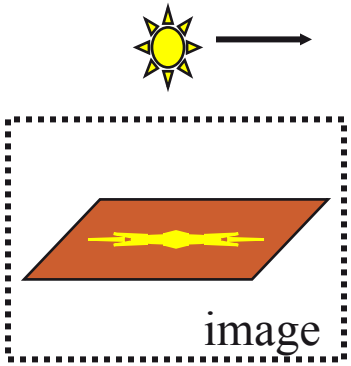
- **Boule en rotation pure**

  - flux optique nul, champ de vitesse réel non nul



- **Surface réfléchissante éclairée par une source en mouvement :**

  - flux optique non nul, champ de vitesse réel nul



LA REPRESENTATION DU MOUVEMENT: LES MODELES DE MOUVEMENT

- **Caractéristiques des modèles : (selon application)**
  - aptitude a bien représenter les mouvements, même complexes
  - compacité de la représentation (transmission, codage)
  - facilite d'interprétation et manipulation
  - flexibilité (adaptation à la complexité du mouvement)

- **Deux approches :**
  - calcul du champs dense et approximation
  - projection de la scène 3D sur le plan image

Plan

- Introduction
- La représentation du mouvement
- **Les différentes techniques**
- Applications et examples

LES PRINCIPALES TECHNIQUES

- **Nous nous limiterons à l'estimation de mouvement de translation**

  Les modèles plus complexes (affine, quadratique) rendent effectivement compte de mouvements plus complexes (rotation, extension).

  Cependant, en pratique, pour le codage, ils sont peu utilisés car trop coûteux.

  Ces méthodes sont de toute façon héritées des méthodes classiques présentées ci-après.

LES PRINCIPALES TECHNIQUES (2)

- **Deux grandes familles de méthodes d'estimation**
  - les techniques de mise en correspondance (matching)
  - les méthodes différentielles

LES TECHNIQUES DE MISE EN CORRESPONDANCE

- **Principe :**

  Mise en correspondance de zones / primitives entre deux images successives et recherche de la plus grande similarité (corrélation maximale)

  => block-matching, région-matching, contour-matching

- **Utilisation :**

  Massive : le block-matching est la technique la plus classique d'estimation de mouvement (MPEG1,2,4, H.26x ...)

LES TECHNIQUES DE MISE EN CORRESPONDANCE (2)

- **Avantages :**
  - minimisation directe de l'erreur de reconstruction
  - fiabilité
  - simplicité du principe, et parallèlisation possible
  - méthode connue, maîtrisée, optimisée...

- **Inconvénients :**
  - complexité pour un espace de recherche important
  - pauvreté de la modélisation pour une complexité raisonnable

LES TECHNIQUES DE MISE EN CORRESPONDANCE (3)

- **Caractéristiques :**

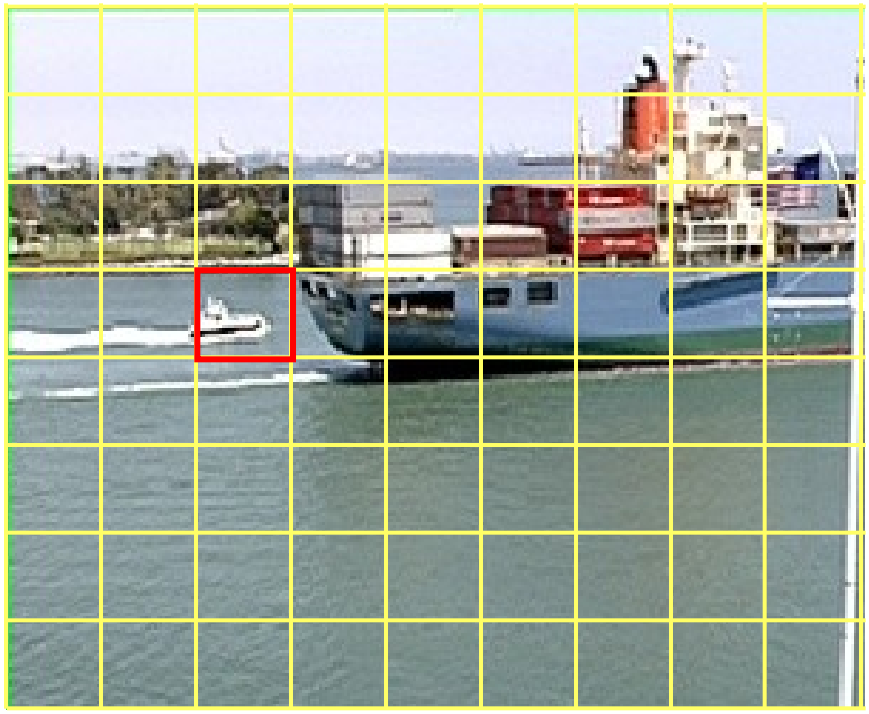
  -Support du matching

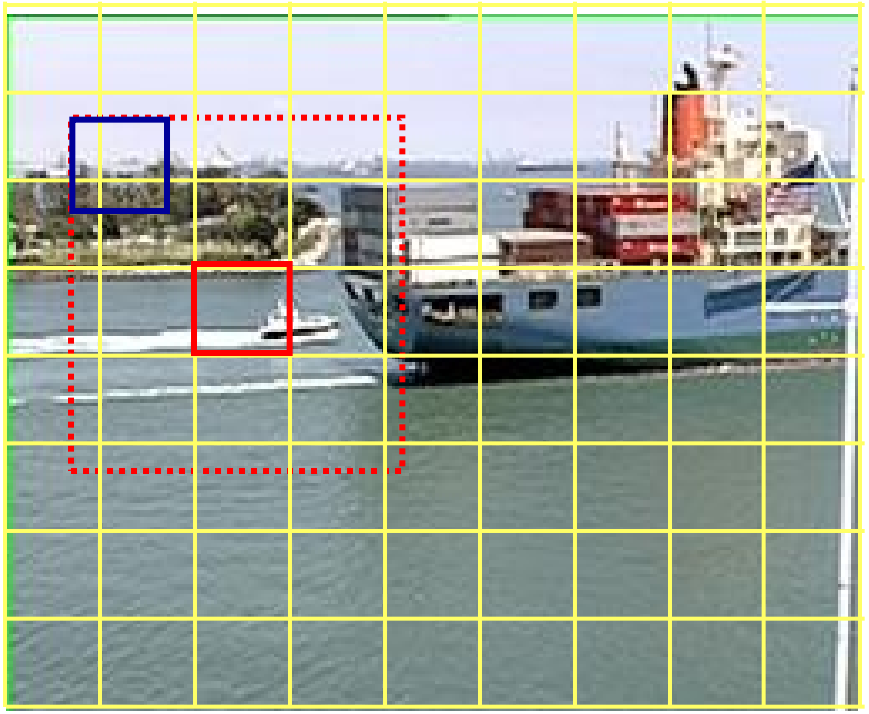
  -Modèle du mouvement

  - Critère à optimiser

  - Stratégie de recherche

Block-matching





$B_k^{(\mathbf{p})}$ $B_h^{(\mathbf{p}+\mathbf{v})}$ $d(\mathbf{v})=d(B_k^{(\mathbf{p})}, B_h^{(\mathbf{p}+\mathbf{v})})$

$$\mathbf{v}^* = \arg\min_{\mathbf{v}} d(\mathbf{v})$$

Block-matching

- **Test ME**

$$d(\mathbf{v}) = d\left(B_k^{(\mathbf{p})}, B_h^{(\mathbf{p}+\mathbf{v})}\right)$$

- **Vecteur choisi**

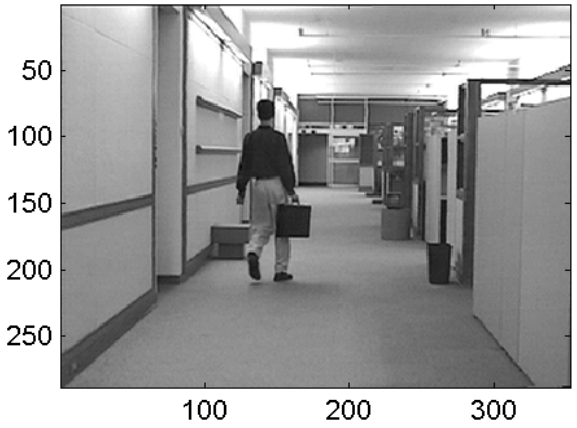
$$\mathbf{v}^* = \arg\min_{\mathbf{v}} d(\mathbf{v})$$

- **Bloc codé**

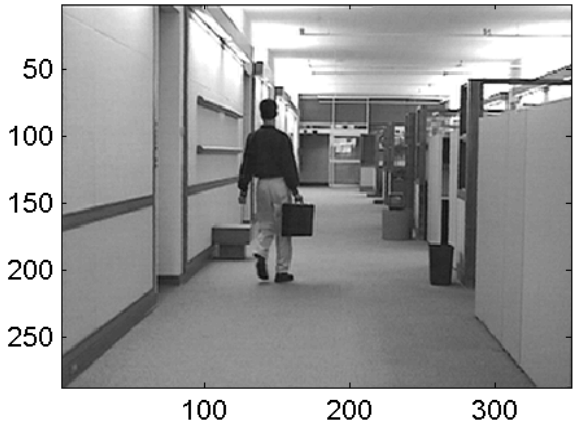
$$B_k^{(\mathbf{p})} - B_h^{(\mathbf{p}+\mathbf{v}^*)}$$

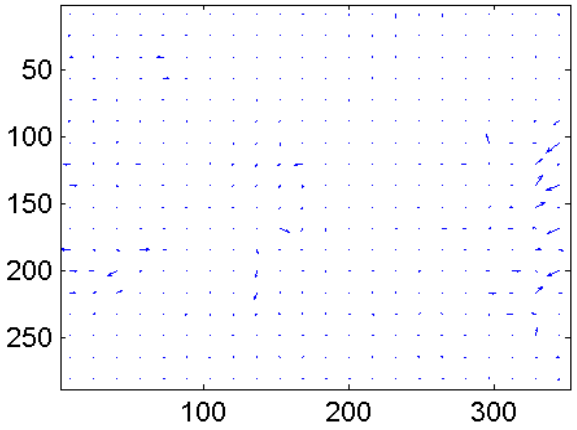
Block-matching : exemple

reference frame

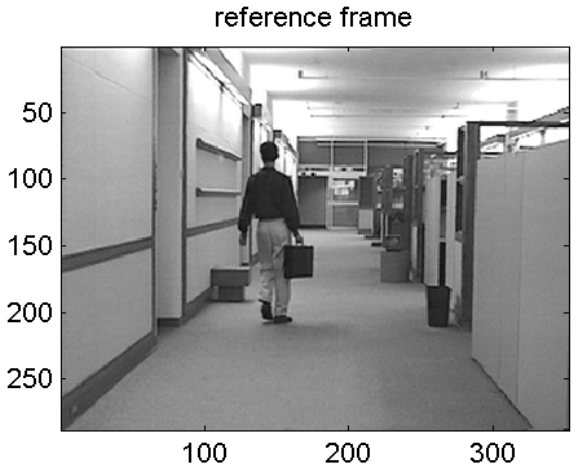


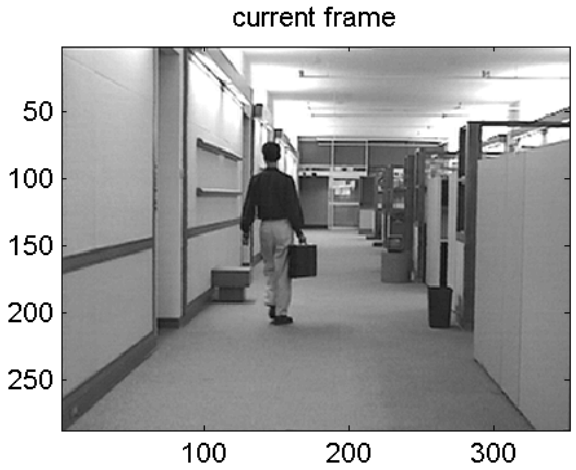
current frame

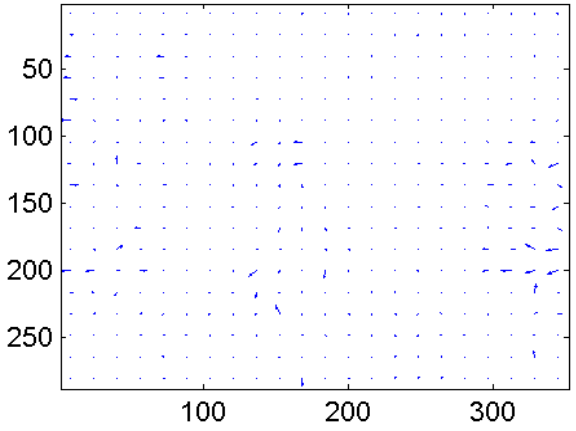




Block-matching : exemple

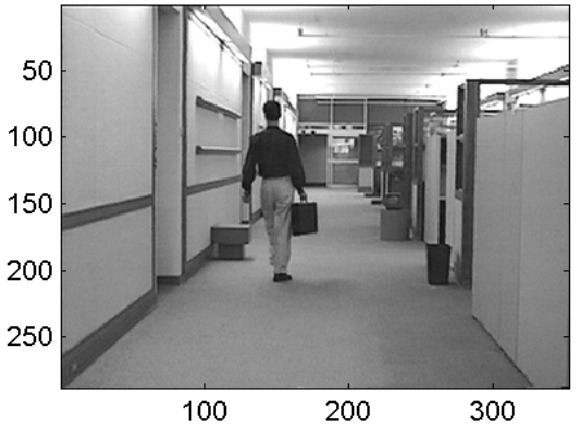




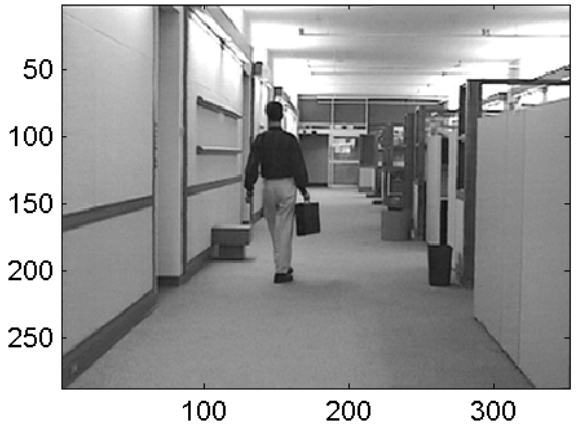


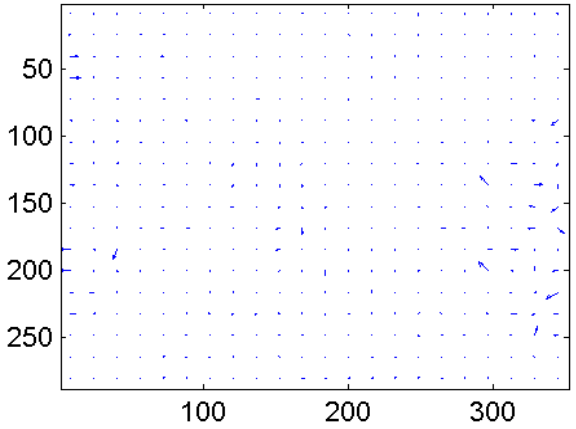
Block-matching : exemple

reference frame



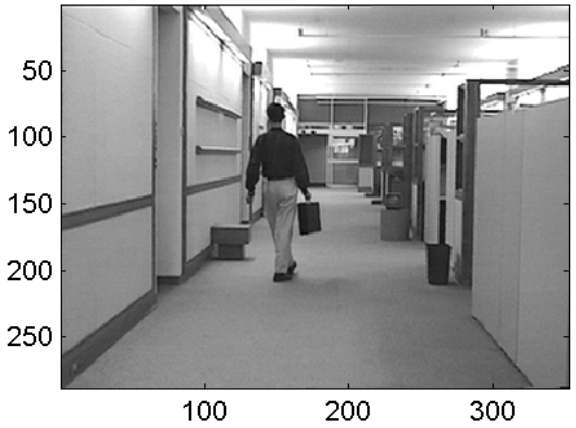
current frame



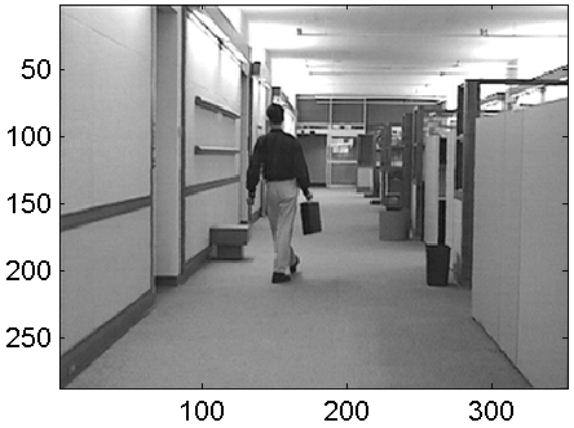


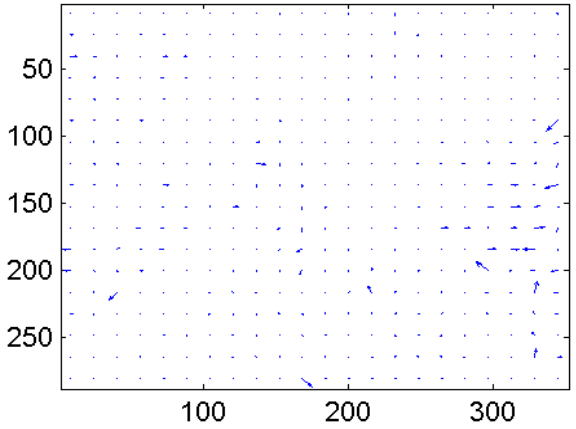
Block-matching : exemple

reference frame



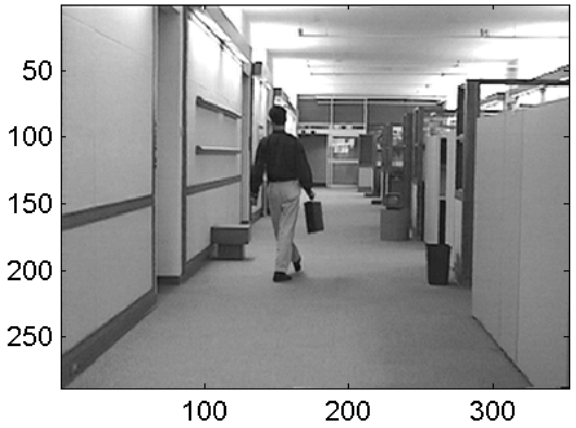
current frame



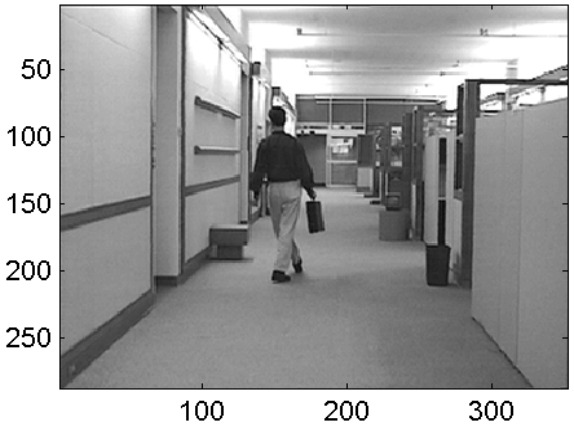


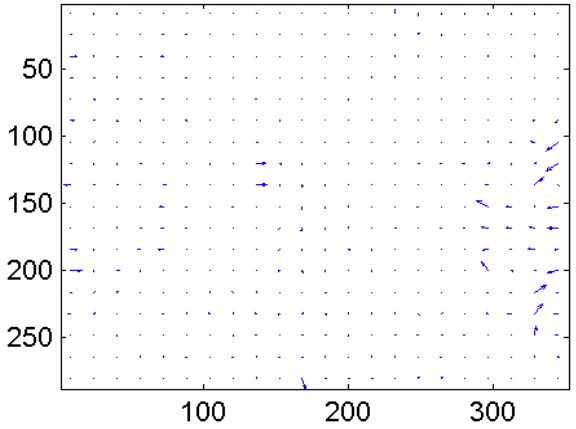
Block-matching : exemple

reference frame

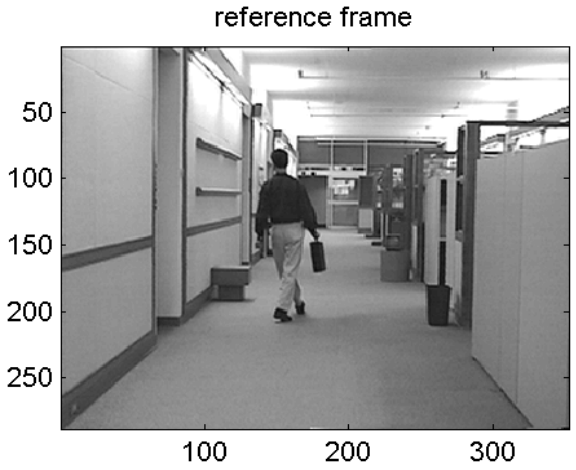


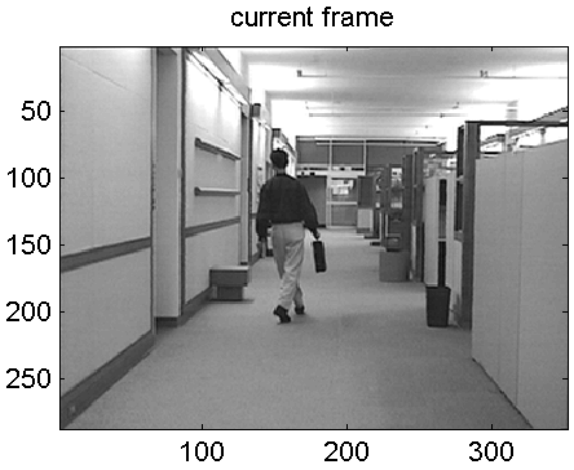
current frame

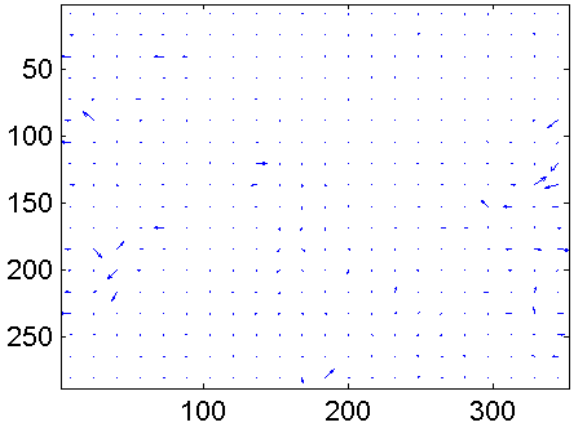




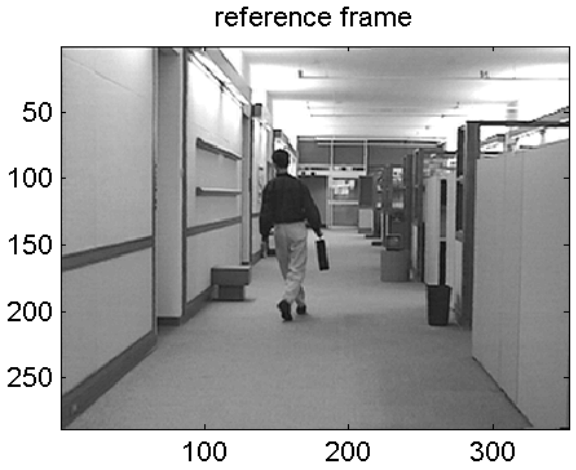
Block-matching : exemple

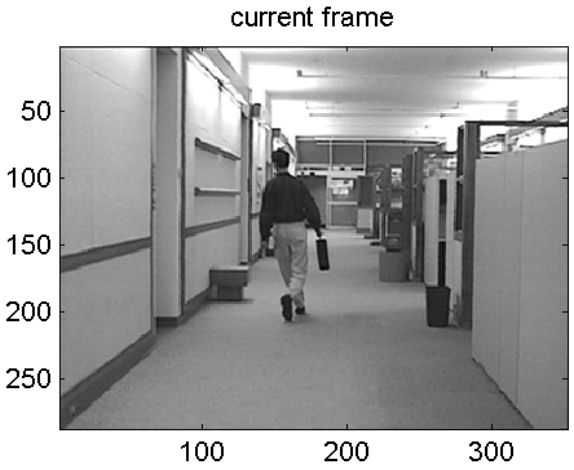


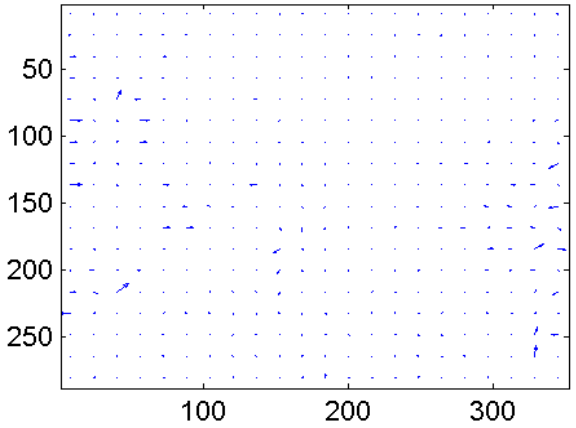




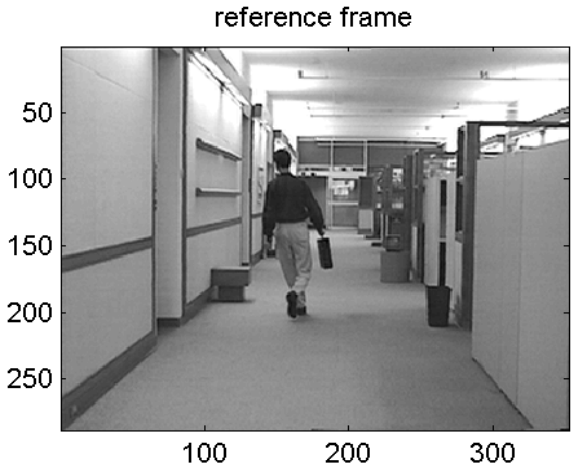
Block-matching : exemple

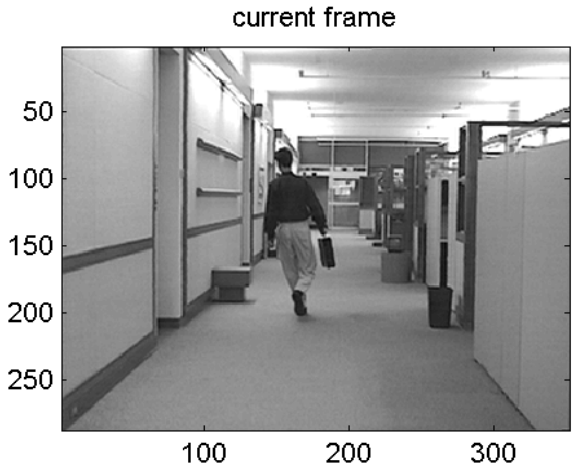


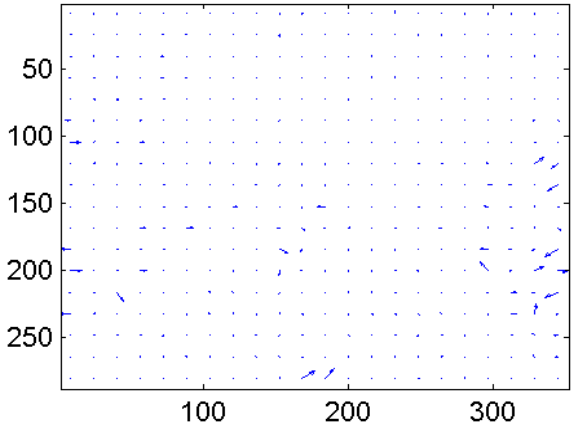




Block-matching : exemple







LIMITATIONS DU BLOCK MATCHING

- Les blocs ne peuvent correspondre aux objets réels de la scène
- Choisir la taille des blocs est périlleux:

  Petits blocs => mouvements homogènes, contours bien suivis, mais plus difficile à coder

  Plus grands blocs => réduction du temps calcul, patterns plus significatifs pour la mise en correspondance, mais mouvement mal estimé sur les bords du bloc.

- Généralement la taille choisie est 8x8, 16x16 ou 32x32.

  $\Rightarrow$ *MPEG: taille normalisée = 16x16 pixels*

  (macroblock)

CRITERES DE RESSEMBLANCE

- Maximisation de la fonction d'intercorrélation normalisée

$$\max_{(i,j)\in W} \frac{\sum_{(m,n)\in B} f(m,n,t)f(m-i,n-j,t-1)}{\sqrt{\left(\sum_{(m,n)\in B} f^2(m,n,t)\right)}\sqrt{\left(\sum_{(m,n)\in B} f^2(m-i,n-j,t-1)\right)}}$$

- Minimisation de l'erreur quadratique moyenne (MSE)

$$\min_{(i,j)\in W} \sum_{(m,n)\in B} [f(m,n,t)-f(m-i,n-j,t-1)]^2$$

- Minimisation de l'erreur absolue moyenne (MAE)

$$(\hat{i},\hat{j})=\arg \min_{(i,j)\in W} J(i,j) \qquad J(i,j)=\sum_{(m,n)\in B} |f(m,n,t)-f(m-i,n-j,t-1)|$$

CRITERES DE RESSEMBLANCE

- Régularisation des vecteur
- Pénalisation des vecteurs “irreguliers” ou “chers à coder”

$$\mathbf{v}^* = \arg\min_{\mathbf{v}} d(\mathbf{v}) + \lambda r(\mathbf{v})$$

Exemple

Exemple

Entropie MVF: 1.675 Kbits PSNR erreur prediction: 21.69

SSD

Entropie MVF: 1.430 Kbits PSNR erreur prediction: 21.69

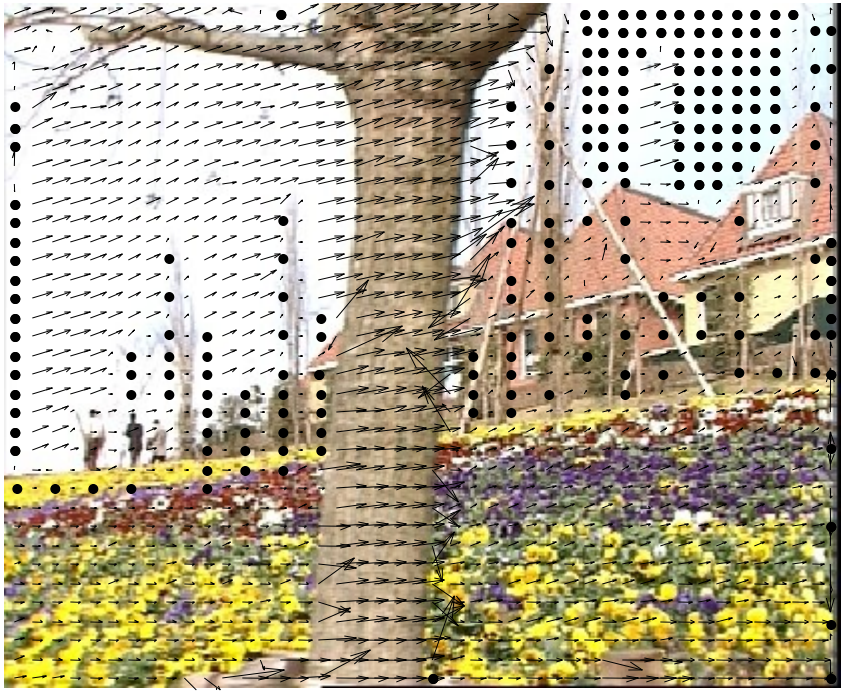
D+λR

Exemple

Entropie MVF: 8.397 Kbits PSNR erreur prediction : 22.82

SSD

Entropie MVF: 5.642 Kbits PSNR erreur prediction : 22.79



D+λR

OPTIMISATIONS DU BMA

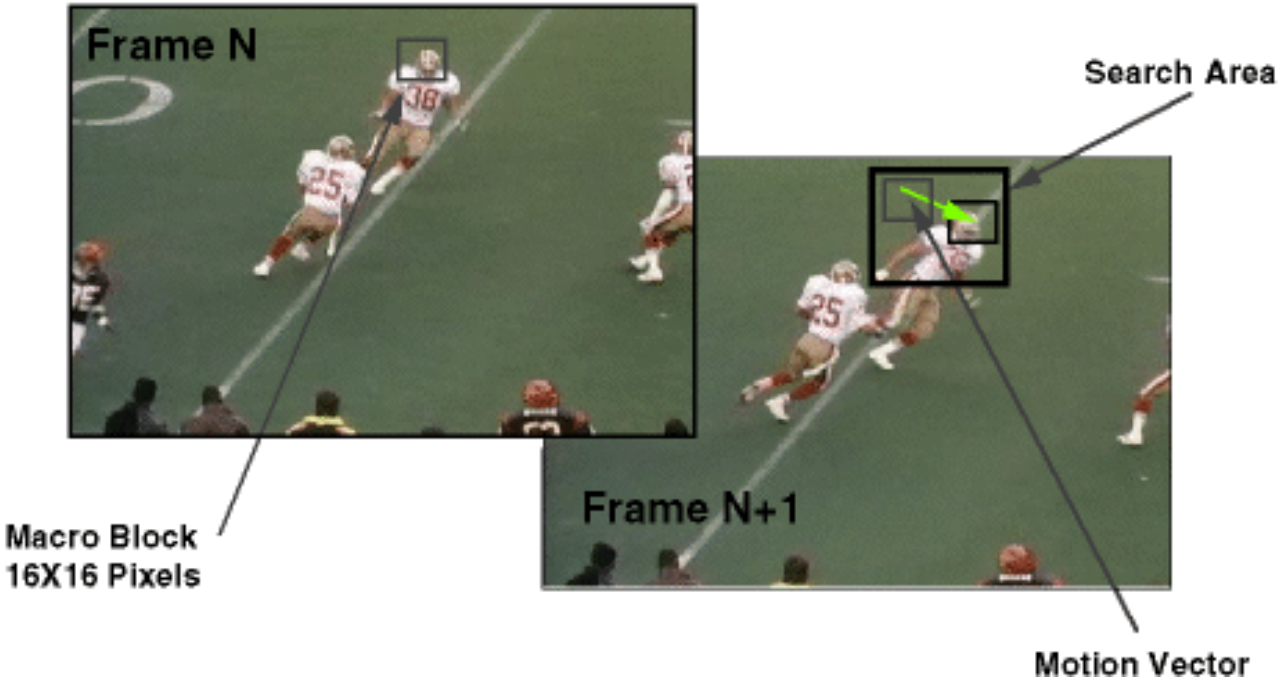
L ’estimation de mouvement est un point critique pour un système de codage temps réel, en particulier en TV numérique
=> nécessité de réduire les temps calculs.

Première alternative:
Pour accélérer le temps calcul on contraint l’estimation block matching par une valeur maximale de vecteur (plage de recherche).

Cette contrainte se justifie:

- Il est rare qu ’un objet traverse l’écran d’une image sur l’autre.
- Si un vecteur est hors plage, la prédiction sera mauvaise, l’objet mal codé. Mais le déplacement est si rapide que l’œil ne verra pas forcément les erreurs.

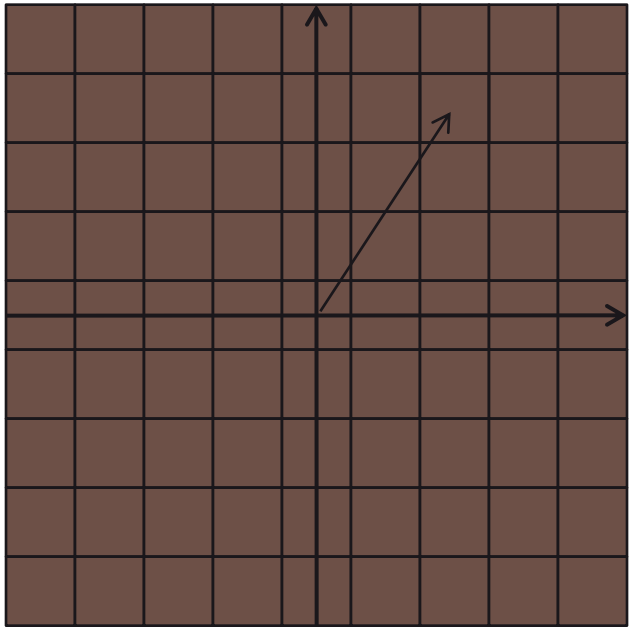
LIMITATION A UNE PLAGE DE RECHERCHE



Plusieurs raffinements possibles:

- La zone de recherche est généralement rectangulaire car statistiquement il y a plus de mouvements horizontaux que verticaux.
-la taille de la zone peut être adaptée à la distance temporelle image de référence / image courante.

Full search



n^2 calculs

RECHERCHES SOUS-OPTIMALES

L itérations (complexité réduite)

- Initialisation : $(\hat{i}_0, \hat{j}_0) = (0,0)$
- A chaque itération $l \in \{1,...,L\}$

$$W_l = \{(i,j) \in W, (i - \hat{i}_{l-1}, j - \hat{j}_{l-1}) \in \Delta W_l\}$$

$$(\hat{i}_l, \hat{j}_l) = \arg \min_{(i,j) \in W_l} J(i,j)$$

- Estimation finale : $(\hat{i}, \hat{j}) = (\hat{i}_l, \hat{j}_l)$

Exemples

- Recherche 2D logarithmique
- Algorithme en 3 coups (« three-step algorithm »)
- Recherche dans des directions conjuguées

Recherche 2D logarithmique

Fenêtre de recherche : $W = \{-A,...,A\} \times \{-A,...,A\}, \quad 2^n \le A < 2^n + 1.$

Forme des fenêtres de recherche partielle :

$$F(2^j) = \{(0,0),(2^j,0),(0,2^j),(-2^j,0),(0,-2^j)\}, \quad j > 1$$

$$F(1) = \{-1,0,1\}^2$$

Initialisation : $\Delta W_1 = F\left(2^{n-1}\right) \Rightarrow \left(\hat{i}_1, \hat{j}_1\right)$

Si $\left(\hat{i}_1, \hat{j}_1\right) \neq (0,0) \Rightarrow \Delta W_2 = \Delta W_1$

Sinon, $\Delta W_2 = F\left(2^{n-2}\right)$

On arrête l'algorithme quand $\Delta W_L = F(1) \Rightarrow$ nombre d'itérations variable

Algorithme en 3 coups (« three-step algorithm »)

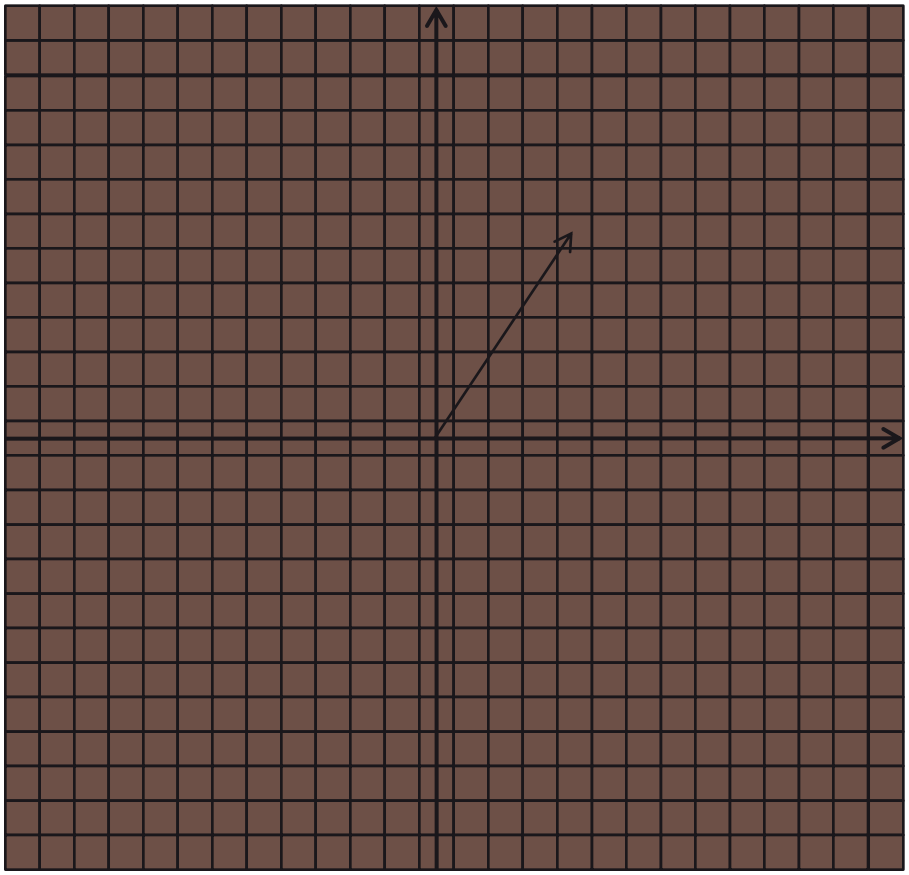
On garantit un nombre fixe d'itérations (en général, $N = 3$)

Fenêtre de recherche à l'itération j :

$$\Delta W_j = F\left(2^{N-j}\right)$$

N-step search

$\approx 9(\log_2 n-1)$ calculs



Recherche dans des directions conjuguées

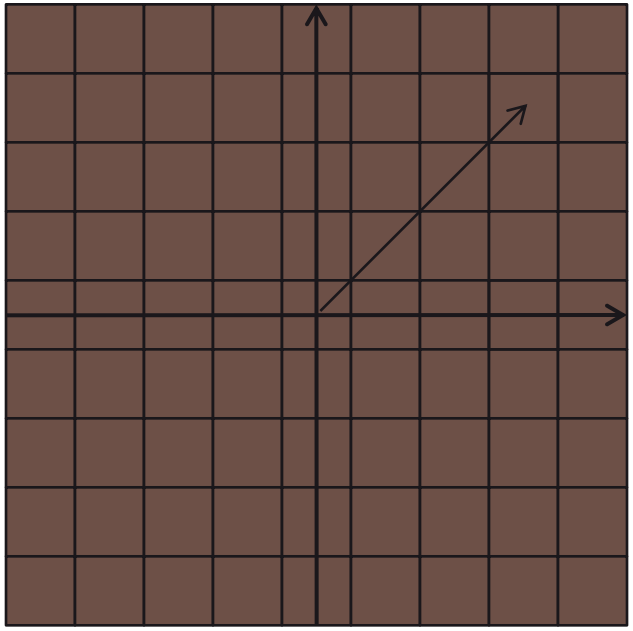
Recherche verticale : $\Delta W_1 = \{(0,0),(1,0),(-1,0)\}$

• si l'une des positions extrêmes correspond au minimum du critère J, on poursuit la recherche en conservant la même fenêtre pour les itérations suivantes
• si le minimum est localisé en (0,0) (ou sur un bord de la fenêtre maximale de recherche W) on passe à l'itération suivante

Recherche horizontale : $\Delta W_2 = \{(0,0),(0,1),(0,-1)\}$

Si $(\hat{i}_L, \hat{j}_L)$ est le vecteur estimé au terme des deux premières étapes de l'algorithme, on peut éventuellement réaliser une dernière recherche dans la direction $(\hat{i}_L, \hat{j}_L)$

Cross search



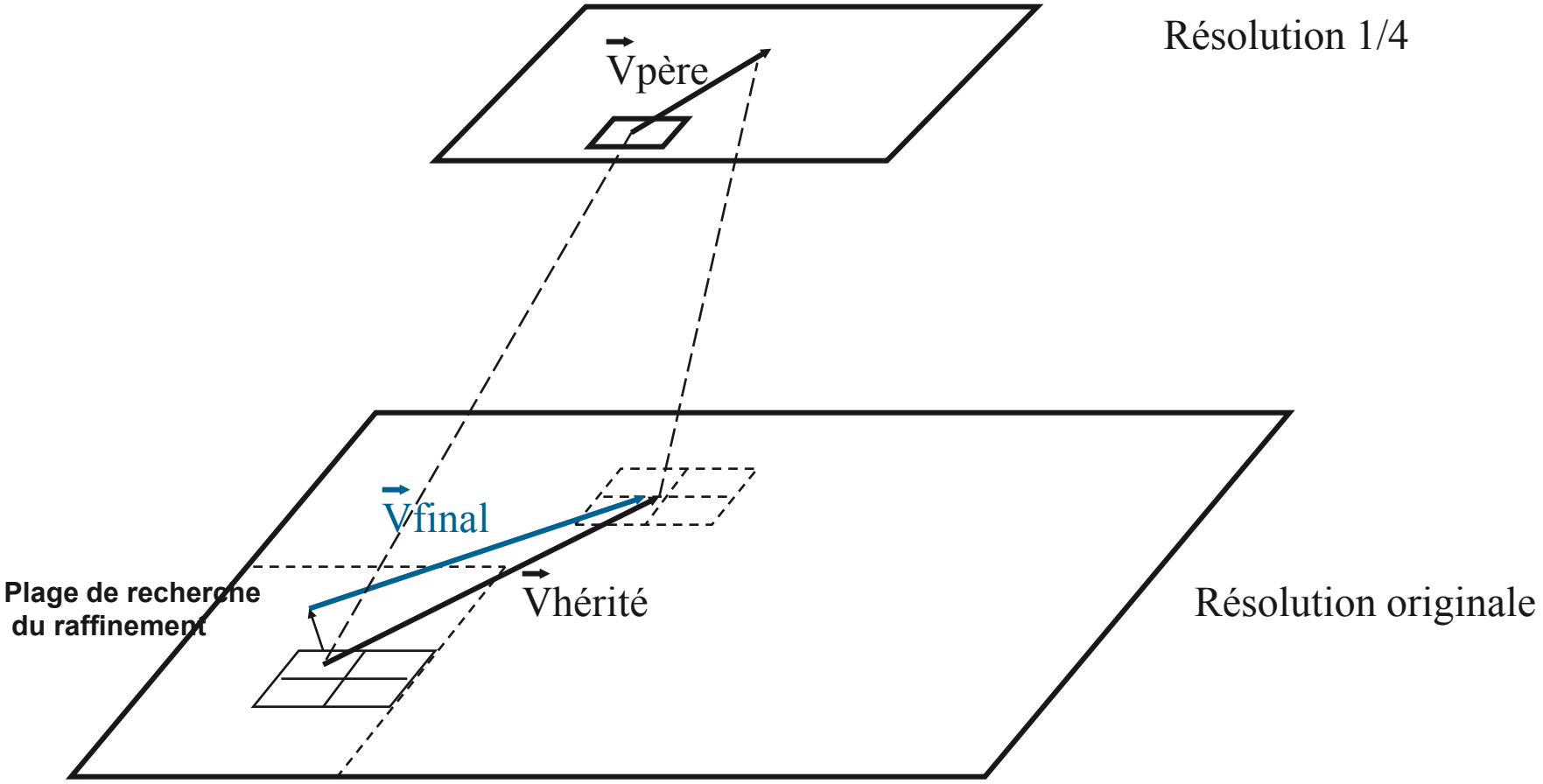
2n-1 calculs

SECONDE ALTERNATIVE: ESTIMATION HIERARCHIQUE MULTIRESOLUTION

- **Consiste à effectuer d'abord l'estimation sur une version sous-échantillonnée de l 'image. Ensuite chaque bloc fils de l 'image hérite du vecteur estimé pour son père et un raffinement local a lieu.**

- Ce procédé mélange :
  - les aspects hiérarchique (recherche multi-étape) avec héritage de vecteurs
  - et multi-résolution (recherche à différentes résolutions).

HIERARCHICAL BMA



ESTIMATION HIERARCHIQUE ET ESTIMATION MULTI-ECHELLE

- Outils complémentaires pour l'amélioration des méthodes classiques :

  - accélération de la recherche

  - amélioration des résultats

- A condition de … :

  - bâtir un schéma global cohérent et adapté

  - avoir une bonne stratégie de propagation des paramètres

Méthodes différentielles: Méthode du gradient

- Utilise le gradient de l'image
- Optimisation globale (par régions)
- Méthode pel-récursive (algorithme de Cafforio-Roca):
  - estimation *a priori* du déplacement
  - test de fiabilité de l'estimation
  - mise à jour de l'estimation

Méthodes différentielles: Méthode du gradient

$$f(x,y,t+T) = f(x-c(x,y),y-d(x,y),t)$$

$$\mathbf{D}(x,y) = \begin{bmatrix} c(x,y) \\ d(x,y) \end{bmatrix}$$

$$f(x,y,t+T) = f(x,y,t) - [c(x,y)f_x(x,y,t) + d(x,y)f_y(x,y,t)] + o[\|\mathbf{D}(x,y)\|]$$

$$\mathbf{V}(x,y) = \lim_{T\to 0} \frac{\mathbf{D}(x,y)}{T} = \begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix}$$

$$f_t = -\mathbf{V}\nabla f$$

$$uf_x + vf_y + f_t = 0$$

Méthodes différentielles: Méthode du gradient

- L'équation $uf_x + vf_y + f_t = 0$ ne permet de trouver que la projection du champ de mouvement sur le gradient
- En plus, la présence du bruit et les variation de luminosité peuvent faire en sorte que l'équation ne soit parfaitement respectée
- Alors on relaxé l'équation et on ajoute alors une contrainte de régularité pour le champ de mouvement

Méthodes différentielles: Méthode du gradient

$$\iint_{\mathbb{R}} (uf_x + vf_y + f_t)^2 \, dx\,dy = \min$$

$$\iint_{\mathbb{R}} \|\nabla u\|^2 + \|\nabla v\|^2 \, dx\,dy \leq \tau$$

$$J = \iint_{\mathbb{R}} (uf_x + vf_y + f_t)^2 \, dx\,dy + \iint_{\mathbb{R}} \lambda \left(\|\nabla u\|^2 + \|\nabla v\|^2\right) dx\,dy$$

Méthodes différentielles: Méthode du gradient

$$J = \iint_{\mathbb{R}} (uf_x + vf_y + f_t)^2 + \lambda(\|\nabla u\|^2 + \|\nabla v\|^2)\,dx\,dy$$

Pour minimiser : $$\iint_{\mathbb{R}} \Theta(x, y, \omega, \omega_x, \omega_y)\,dx\,dy$$

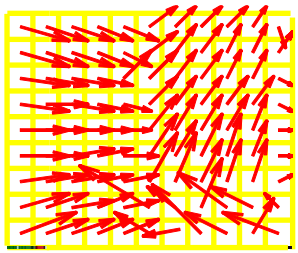
On doit imposer : $$\frac{\partial \Theta}{\partial \omega} - \frac{\partial^2 \Theta}{\partial x \partial \omega_x} - \frac{\partial^2 \Theta}{\partial y \partial \omega_y} = 0$$

$$\lambda \nabla_2 u = (uf_x + vf_y + f_t) f_x$$
$$\lambda \nabla_2 v = (uf_x + vf_y + f_t) f_y$$

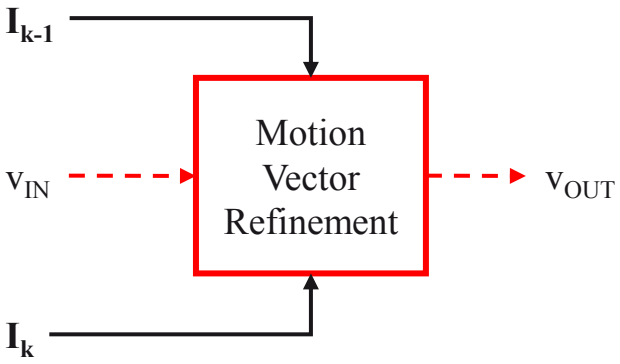
Méthodes différentielles: Algorithme de Cafforio-Rocca

- **Entrée:**
  - Image de référence
  - Image courante
  - Champ d'initialisation
- **Sortie**
  - Champ raffiné, dense

I_{k-1}

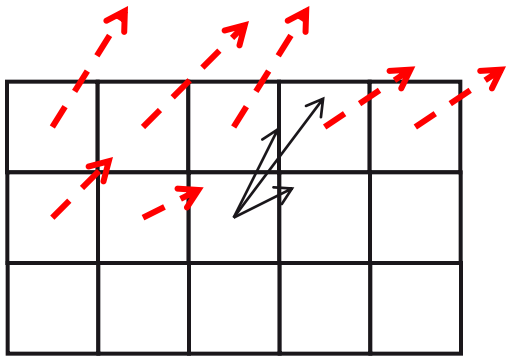


I_k



Algorithme de Cafforio-Rocca
Initialisation

- **Le vecteur de mouvement peut être initialisé :**
  - Par un vecteur associé au bloc
  - Par une fonction des vecteurs voisins



Vector associated to previous pel

Mean of neighbors

Vector from input MVF

$$\mathbf{D}^0_{n,m} = \begin{bmatrix} d^0_{n,m} \\ c^0_{n,m} \end{bmatrix}$$

Algorithme de Cafforio-Rocca
Validation

- **Validation**

Le vecteur sorti de l'inisialisation est comparé au vecteur nul en termes d'erreur de prédiction

$$A = \left| f_{nL_1, mL_2, k+1} - f_{nL_1 - c^0_{n\,m}, mL_2 - d^0_{n\,m}, k} \right|$$

$$B = \left| f_{nL_1, mL_2, k+1} - f_{nL_1, mL_2, k} \right|$$

$$\mathbf{D}^1_{n,m} = \begin{cases} \mathbf{D}^0_{n,m} & \text{si } A < B + \gamma \\ \mathbf{0} & \text{si } A \geq B + \gamma \end{cases}$$

Algorithme de Cafforio-Rocca
Raffinement

Le vecteur validé est raffiné par une petite modification.
On veut minimiser l'erreur de prédiction sous contrainte que la modification du vecteur soit petite

Le vecteur final est donc :

$$\mathbf{D}^2_{n,m} = \mathbf{D}^1_{n,m} + \delta\mathbf{D}$$

La correction est trouvée en minimisant le critère :

$$J(\delta\mathbf{D}) = [f(nL_1, mL_2, k+1) + \\ - f(nL_1 - c^1_{n,m} - \delta c_{n,m}, mL_2 - d^1_{n,m} - \delta d_{n,m}, k)]^2 \\ + \lambda \|\delta\mathbf{D}\|^2$$

Regularisation

Erreur de prédiction

Algorithme de Cafforio-Rocca
Raffinement

$$J(\delta\mathbf{D}) \approx [e + \delta\mathbf{D}^T\varphi]^2 + \lambda\|\delta\mathbf{D}\|^2$$

$$\frac{\partial J}{\partial \delta\mathbf{D}} = 2[e + \delta\mathbf{D}^T\varphi]\varphi + 2\lambda\delta\mathbf{D}$$
$$= 2e\varphi + 2(\lambda\mathbf{I} + \varphi\varphi^T)\delta\mathbf{D}$$

$$\delta\mathbf{D} = -e\left(\lambda\mathbf{I} + \varphi\varphi^T\right)^{-1}\varphi$$

Algorithme de Cafforio-Rocca
Raffinement

$$\delta \mathbf{D} = -e\left(\lambda \mathbf{I} + \varphi\varphi^T\right)^{-1}\varphi$$

$$\left(\mathbf{M} + \mathbf{x}\mathbf{x}^T\right)^{-1} = \mathbf{M}^{-1} - \frac{\mathbf{M}^{-1}\mathbf{x}\mathbf{x}^T\mathbf{M}^{-1}}{1 + \mathbf{x}^T\mathbf{M}^{-1}\mathbf{x}}$$

$$\left(\lambda \mathbf{I} + \varphi\varphi^T\right)^{-1} = \frac{1}{\lambda}\mathbf{I} - \frac{\frac{1}{\lambda^2}\varphi\varphi^T}{1 + \frac{1}{\lambda}\|\varphi\|^2}$$

$$= \frac{1}{\lambda}\left[\mathbf{I} - \frac{\varphi\varphi^T}{\lambda + \|\varphi\|^2}\right]$$

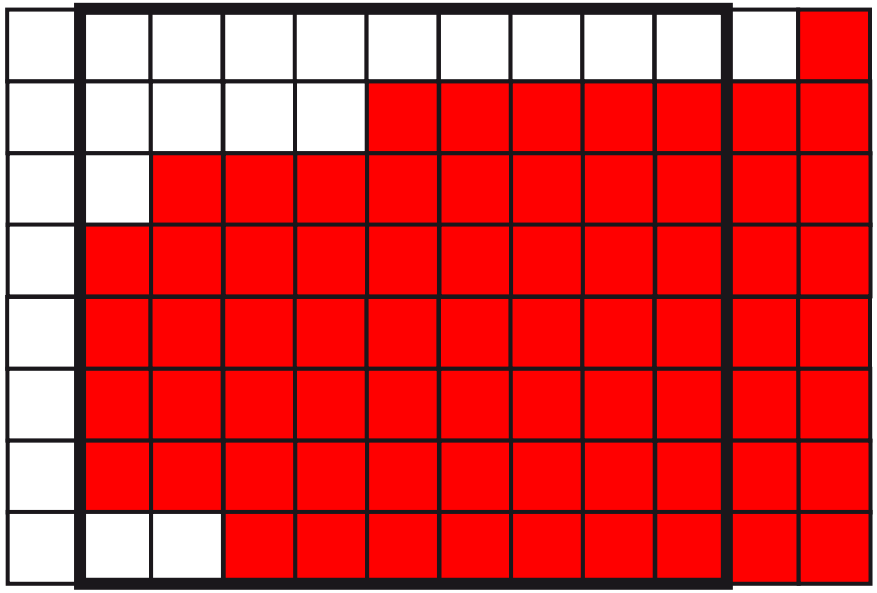
Algorithme de Cafforio-Rocca
Raffinement

$$\delta \mathbf{D} = -e\left(\lambda \mathbf{I} + \varphi\varphi^T\right)^{-1}\varphi$$

$$\delta \mathbf{D} = -e\frac{1}{\lambda}\left[\mathbf{I} - \frac{\varphi\varphi^T}{\lambda + \|\varphi\|^2}\right]\varphi$$

$$= \frac{-e\varphi}{\lambda + \|\varphi\|^2}$$

Algorithme de Cafforio-Rocca
Exemple



Mouvement du fond (blanc)

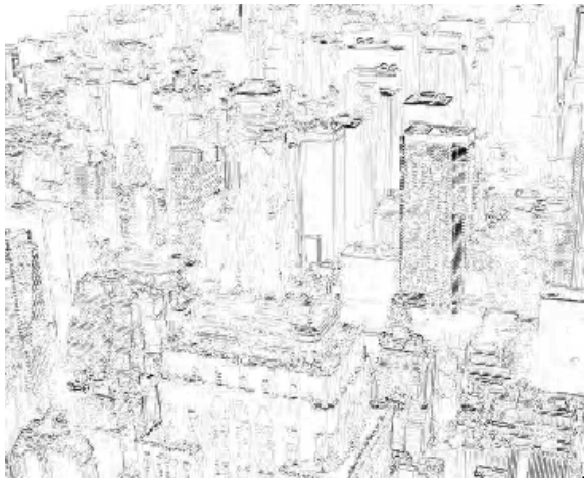
Mouvement de l'objet rouge

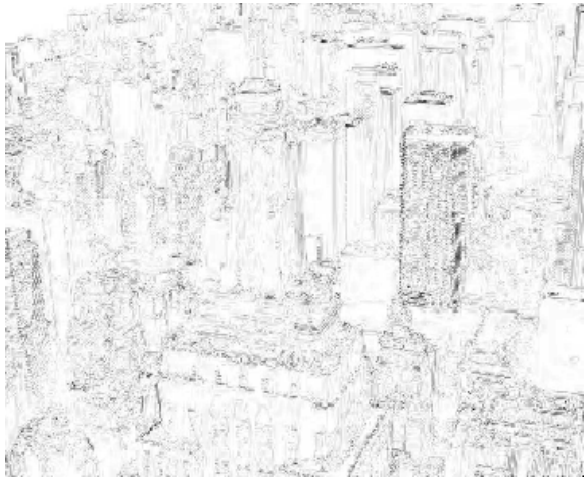
Initialisation (= objet rouge)

Initialisation : mouvement associé au bloc (objet rouge)
Le vecteur n'est pas validé : il est mis à zéro
Le raffinement permet de retrouver le mouvement du fond

Quand on arrive au premier pixel rouge, l'initialisation n'est pas validé
Le raffinement suivant permet de retrouver le mouvement de l'objet

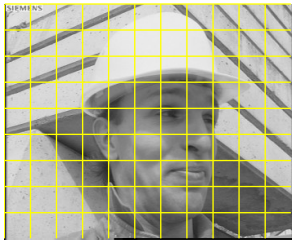
Algorithme de Cafforio-Rocca
Résultats

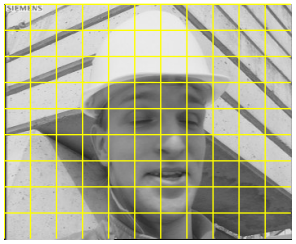


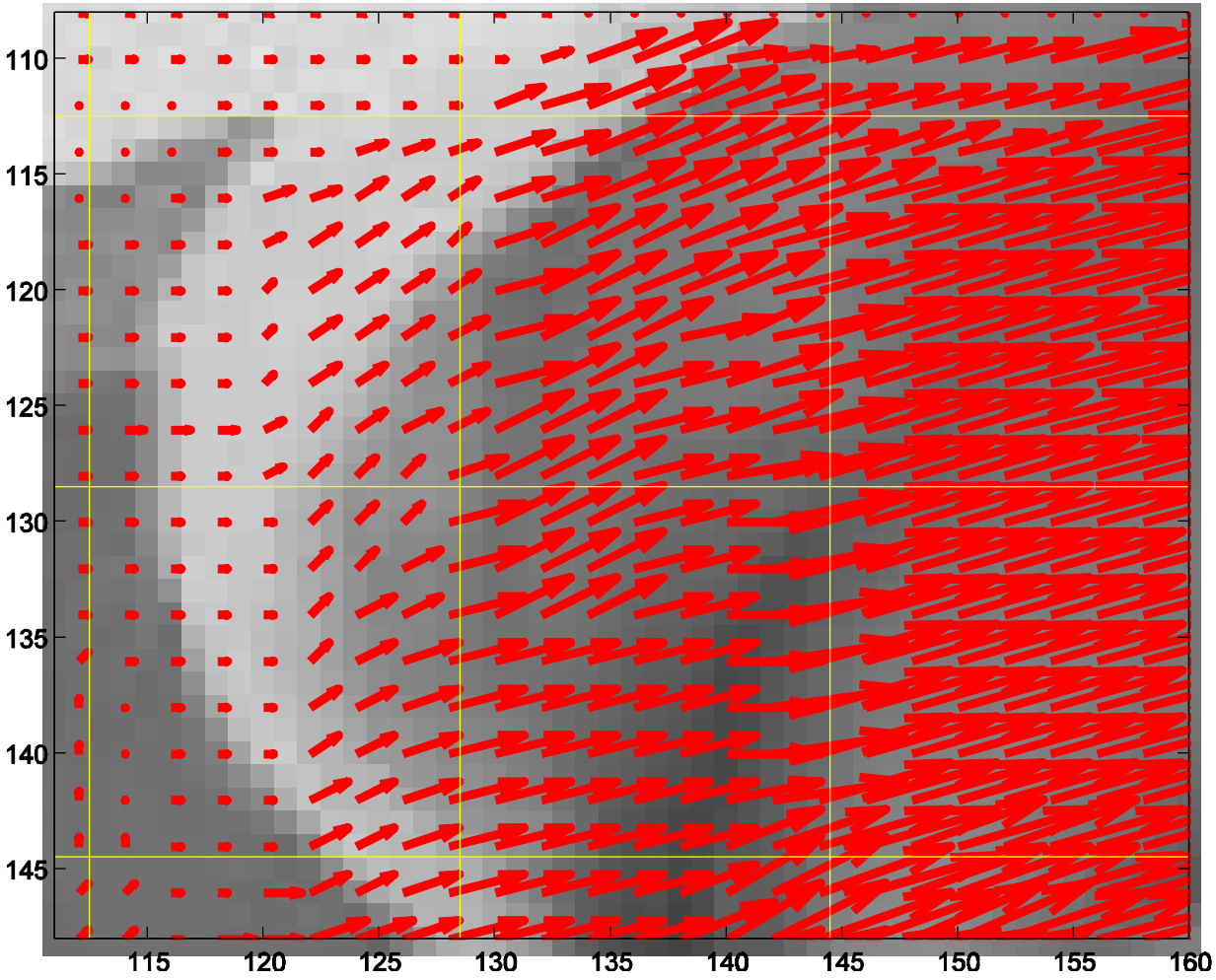


PSNR before CR: 24.1
PSNR after CR: 27.1

Algorithme de Cafforio-Rocca
Résultats







CARACTERISTIQUES DES METHODES DIFFERENTIELLES (2)

- **Avantages :**

  - bon compromis dimension espace de recherche / vitesse de recherche

  - possibilité d'utiliser des modèles à plus de paramètres, donc plus génériques, donc de les appliquer sur de grandes régions, menant à un débit d'information de mouvement inférieur.

  - flexibilité, et en particulier bonne adaptation aux méthodes orientées objet.

- **Inconvénients :**

  - difficile à maîtriser

  - convergence non assurée/divergence

  - reste complexe, et notamment difficile à implanter en HW

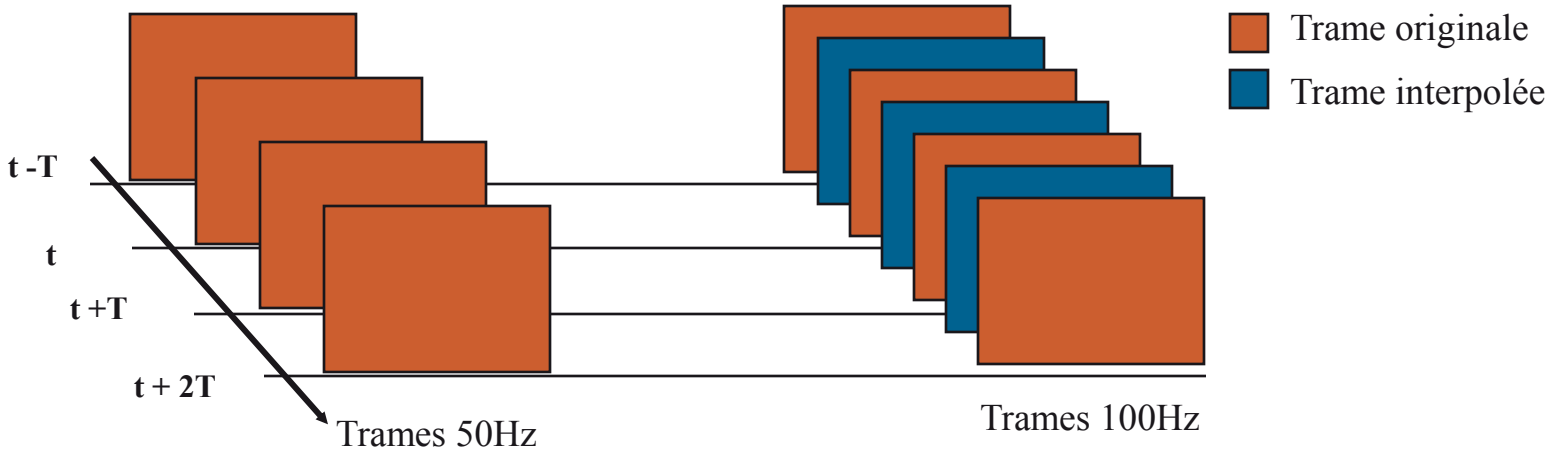
Plan

- Introduction
- La représentation du mouvement
- Les différentes techniques
- **Applications et exemples**

SCAN RATE CONVERSION

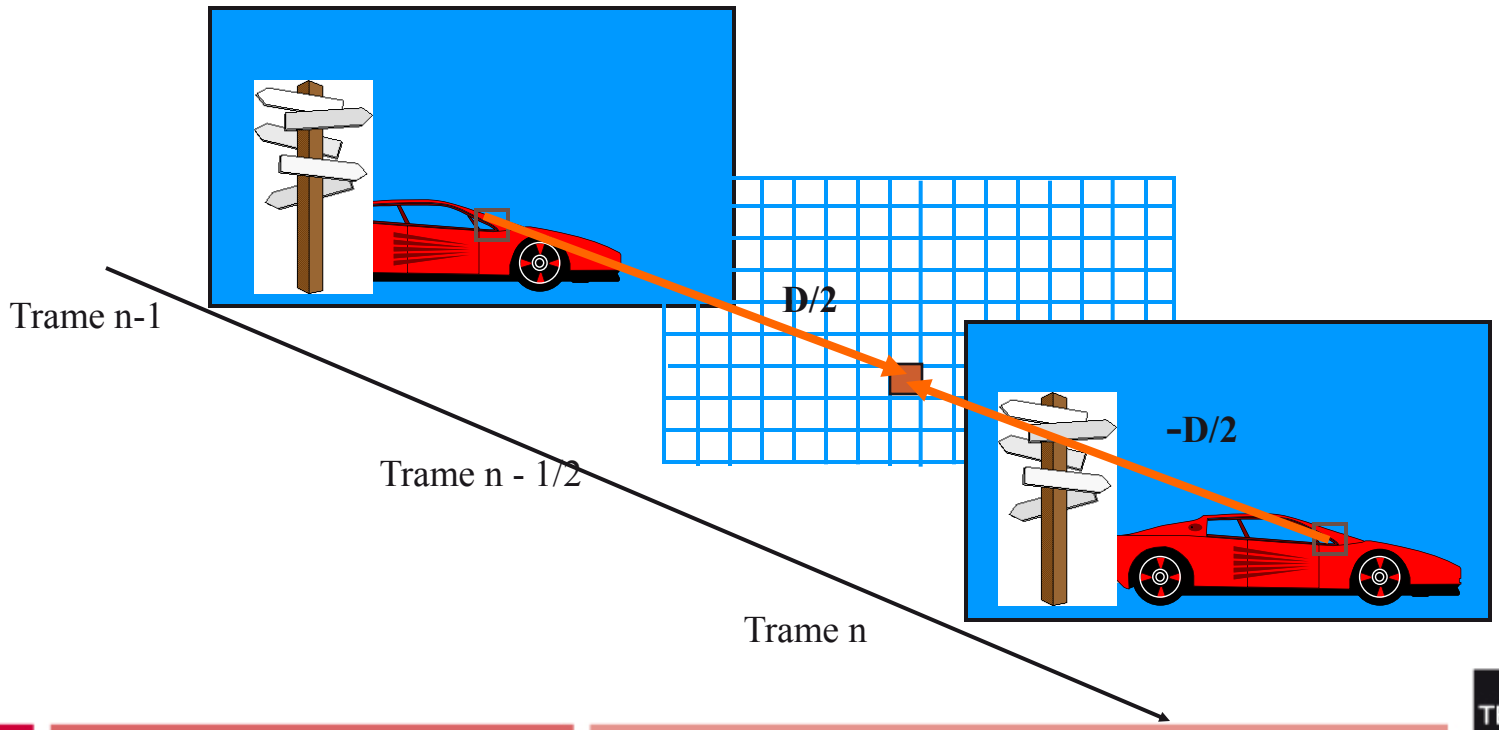
- Exemple du téléviseur 100Hz

Dans un téléviseur 100Hz, le nombre d ’images affichées par seconde est doublé. L’idée est de reconstruire artificiellement les images qui n’existaient pas à la transmission, pour améliorer la qualité d’image.



SCAN RATE CONVERSION (2)

- Pour créer les trames absentes, on va faire de l'estimation de mouvement et construire une image interpolée compensée en mouvement.



SCAN RATE CONVERSION (3)

- TV 100Hz, => marché grand public, la notion de coût est très importante. Techniquement, des chips dédiés sont intégré dans le téléviseur.

- Coût réduit => traitement simple, limitation de la mémoire image. On utilisera du block matching (solution bien maîtrisée, optimisée) avec des simplifications spécifique à l'application.

- Corrélation très forte entre les trames => on limite le nombre de blocs candidats, en fonction du mouvement précédemment déterminé (algorithme récursif) . Typiquement si l'objet se déplace sur le plan horizontal, on testera les 3 ou 4 blocs dans cette direction privilégiée + un petit nombre de blocs dans des direction aléatoires.

REDUCTION DE BRUIT

- But :

  - amélioration de la qualité de la source (principalement des films)
  - amélioration de la qualité de compression (le bruit oblige à dépenser beaucoup de bits au codage)

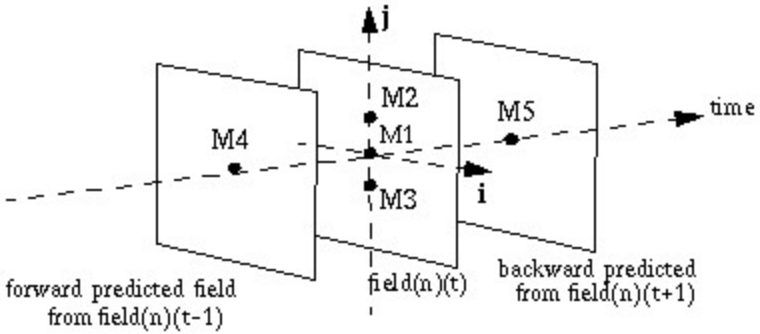
- Solutions :

  - filtre passe-bas => flou dans l 'image
  - filtrer sur la différence inter-image => low end

Réduction de bruit professionnelle:
filtrage médian adaptatif + filtre temporel récursif, compensés en mouvement.

FILTRE MEDIAN

- Idée de base du filtrage médian: supprimer les pics de bruit haute fréquence tout en conservant les contours.



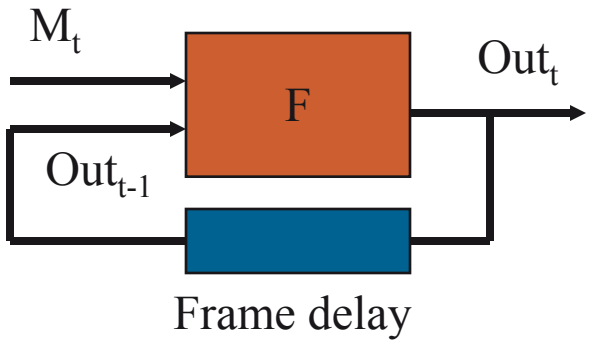
M1 = Med(M1,M2,M3,M4,M5)

- L'utilisation d 'images compensées en mouvement améliore le résultat car le filtrage est effectué sur 3 champs représentant le même instant temporel: on ne filtre que le bruit, le mouvement ne rentre pas en compte.

FILTRE RECURSIF

- En l ’absence de mouvement, le filtrage temporel est le traitement optimal pour la réduction de bruit. Mais il y a toujours du mouvement entre les images. Dans le cas de mouvement non compensé, un flou est généralement introduit par le traitement. Idée de base du filtre temporel récursif: filtrer le bruit entre les images. Pour chaque pixel:

$$Out(i)_t = F(M(i), Out(i)_{t-1})$$



- En utilisant Out_{t-1} image prédite forward de Out_t on obtient une estimation plus fine du bruit intrinsèque de l’image. On effectue généralement une moyenne pondérée adaptative, dépendant des discontinuités entre les niveaux de chaque pixel. Ceci permet de préserver la qualité du filtrage sur les zones en mouvement.

UTILISATION

- Tendance: ce type de réducteur de bruit est désormais directement implanté dans les codeurs vidéo, les vecteurs issus de l'estimateur de mouvement étant disponibles. Le type d'estimateur utilisé est donc un block matching.

- De telles structures de réducteur de bruit sont aussi utilisées en imagerie médicale, par nature, très bruitées. Les techniques les plus appropriées dans ce cas sont aussi les techniques de matching.

CONCLUSION

- Les applications de l ’estimation de mouvement sont multiples: codage, reconnaissance, amélioration de la qualité d ’image, segmentation...
- Avant toute estimation de mouvement il convient de se fixer un modèle mathématique du mouvement
- On cherche à déterminer des vecteurs représentant le flot optique, différent du mouvement réel 3D.
- Plusieurs méthodes: matching, méthodes différentielles...
- En codage vidéo, le block matching reste la solution appropriée en terme de complexité, temps calcul, cohérence des résultats.