

ANALYSE MULTIRÉSOLUTION ET ESTIMATION
DENSE DU FLUX OPTIQUE



Berthollet Laurent

10 juin 2003

Table des matières

1	Introduction	1
2	Structures des données et algorithmes classiques	3
2.1	Modélisation	3
2.1.1	Représentation des images	3
2.1.2	Système de voisinage	3
2.2	Champs de Markov	5
2.2.1	Définition	5
2.2.2	Théorème de Hammersley-Clifford	6
2.3	Estimation bayésienne	6
2.3.1	Relation observations-étiquettes	6
2.3.2	Règle de Bayes	7
2.4	Caractérisation de l'énergie	7
2.5	Algorithmes classiques	8
2.5.1	Algorithmes déterministes : ICM	8
2.5.2	Algorithmes stochastiques : recuit simulé	10
3	Résolution multigrille	13
3.1	Algorithme multirésolution	13
3.1.1	Principe	13
3.1.2	Construction des pyramides	13
3.2	Algorithme multiéchelle	17
3.3	Application à divers modèles	18
3.3.1	Modèle de détection de mouvement	19
3.3.2	Modèle de mesure de déplacement	29
4	Optimisation robuste et évolutionnaire	38
4.1	Estimation dense par techniques robustes	38
4.1.1	Raffinement itératif	38

4.1.2	Combinaison multirésolution-multiéchelle	41
4.1.3	Minimisation de l'énergie	41
4.1.4	Applications	45
4.2	Estimation par exploration-sélection	46
4.2.1	Principe de l'exploration-sélection	47
4.2.2	Applications	47
5	Conclusion	49
A	Représentation du flux optique	51
B	Lecture des données pour la séquence Yosemite	53

Table des figures

2.1	Exemple de deux images consécutives : (a) Image à l'instant t . (b) Image à l'instant $t+1$	4
2.2	Relation de voisinage aux 4 et 8 plus proches voisins.	4
2.3	Cliques associées aux relations des 4 ou 8 plus proches voisins.	5
2.4	Initialisation capitale pour l'ICM.	10
3.1	Pyramide obtenue par décimations successives d'une grille.	14
3.2	Relation entre les pyramides d'étiquettes et d'observations.	15
3.3	Stratégie multirésolution descendante avec duplication.	16
3.4	Structure de la résolution multiéchelle.	18
3.5	Observations : (a) $o(k)$, (b) $o(k+1)$, (c) $\bar{o}(k)$, (d) $\bar{o}(k+1)$	23
3.6	Relaxation obtenue par ICM.	24
3.7	Evolution de l'énergie au cours de l'ICM.	24
3.8	Relaxation obtenue par recuit simulé.	25
3.9	Evolution de l'énergie au cours du recuit simulé	25
3.10	Légende des pyramides.	26
3.11	Pyramides des images et observations associées.	27
3.12	Relaxation obtenue par multirésolution à 3 niveaux.	28
3.13	Relaxation multiéchelle à 3 niveaux.	28
3.14	Positionnement relatif des grilles S et R	30
3.15	Champ de vecteurs obtenu par ICM.	32
3.16	Progression de l'énergie au cours de l'ICM.	33
3.17	Progression de l'énergie au cours du recuit simulé.	33
3.18	Champ de vecteurs obtenu par recuit simulé.	34
3.19	Champ de vecteurs obtenu par multirésolution.	35
3.20	Champ de vecteurs obtenu par résolution multiéchelle.	36
4.1	Principe de l'estimation par raffinement du flux optique.	40
4.2	Modèle multirésolution pour le raffinement.	40
4.3	Relaxation multiéchelle à chaque niveau de la pyramide.	41

4.4	Exemple de cliques chevauchant deux blocs.	43
4.5	Champ de vecteurs obtenu par raffinement itératif.	46
4.6	Champ de vecteurs obtenu après exploration-sélection.	48

Liste des tableaux

2.1	Algorithme ICM.	9
2.2	Algorithme du recuit simulé.	11
3.1	Algorithme multirésolution utilisant l'ICM à chaque niveau. .	17
3.2	Algorithme multiéchelle utilisant l'ICM à chaque niveau. . .	18
3.3	Définition de $W_{s,2}(x_s, \bar{o}_s(k), \bar{o}_s(k+1))$	21
4.1	Algorithme d'exploration-sélection.	47

Chapitre 1

Introduction

L'analyse d'images consiste en l'extraction d'informations cachées à partir d'un ensemble de données observées. En ce qui nous concerne, ces observations peuvent être une séquence d'images brutes, ou des informations préalablement dégagées des images prétraitées, à partir desquelles nous allons tirer des renseignements. Ils peuvent représenter des étiquettes binaires lors de la détection de mouvement, ou encore des vecteurs de déplacements lors de la mesure de déplacement. La résolution du flux optique réside en la détermination des vecteurs de mouvement entre deux images consécutives.

La difficulté, est que l'obtention des observations résulte la plupart du temps en la perte d'informations capitales, accrue par divers bruits se rajoutant au cours du processus. Nous faisons donc face à des problèmes mal posés. Pour outrepasser ces ennuis, nous devons formuler des hypothèses *a priori* sur les caractéristiques des étiquettes recherchées. Il s'agit, par la suite, de définir une fonction d'énergie dont il faudra obtenir le minimum.

Malheureusement, nous sommes confrontés à un problème d'optimisation globale excessivement délicat. En effet, les fonctions d'énergie à minimiser sont fortement complexes, non linéaires, et peuvent dépendre d'un grand nombre de variables. Il nous faut donc tenter d'approcher au mieux une solution du problème. Divers algorithmes classiques de relaxation existent. Il y a d'une part, les *algorithmes déterministes*, qui convergent rapidement vers des minima locaux dépendant de l'initialisation, et d'autre part les *algorithmes stochastiques*, qui convergent théoriquement vers un minimum global de la fonction d'énergie. Cependant, si leur résultat est indépendant de l'initialisation, ils sont extrêmement lents. Toutefois, ces deux types d'algorithmes arrivent à saturation lors de problèmes compliqués d'analyse du mouvement.

En effet, les premiers donnent de bien piètres résultats, alors que les seconds sont extraordinairement lents.

Pour remédier à cela, il y a quelques années, l'analyse multirésolution fut exploitée. L'un de ses plus gros avantages, est la hiérarchisation des données. En effet, à plus basse résolution, les algorithmes d'estimation sont beaucoup plus rapides. Pourtant, la majorité des techniques multirésolution utilise une modélisation déterministe. Il est excessivement difficile d'associer une analyse multirésolution avec une modélisation statistique. C'est pour cela, que Patrick Pérez [8], développa lors de sa thèse une approche multiéchelle. Cette dernière repose sur l'exploration d'ensembles de configurations emboîtés inclus dans l'ensemble de configurations initial dans lequel est défini le problème d'optimisation. Ces sous-ensembles sont tels que la minimisation de la fonction d'énergie initiale sur l'un deux, est équivalente à la minimisation d'une nouvelle fonction d'énergie définie sur un ensemble de configurations de résolutions réduites. Tout comme l'analyse multirésolution, elle fournit donc une structure pyramidale des étiquettes. A chaque niveau, elle se sert d'une modélisation markovienne des données, découlant directement de la modélisation originelle.

Par la suite, dans [5], une combinaison des deux techniques, associée à un raffinement itératif, donne une estimation robuste et de bonne qualité du flux optique. A chaque niveau de résolution, il faut déterminer des champs de raffinement réduits, selon une approche multiéchelle, qui serviront à corriger les champs de vecteurs de déplacement. Enfin, nous utiliserons le nouvel algorithme évolutionnaire de Olivier François décrit dans [2]. Il s'agit d'un algorithme d'optimisation stochastique, appelé algorithme d'*exploration-sélection*, qui comme les algorithmes génétiques, adhère à la théorie de convergence du recuit simulé généralisé. Le processus de minimisation est fondé sur deux opérateurs. Le premier, l'opérateur *exploration*, est probabiliste. Il considère chaque solution comme un individu qui tente une recherche aléatoire sur un graphe. Le second, l'opérateur *sélection*, est déterministe. Il crée des interactions entre les individus.

Chapitre 2

Structures des données et algorithmes classiques

2.1 Modélisation

Notre objectif est de déterminer et de représenter les vecteurs du champ de déplacement (le flux optique) entre deux images consécutives d'une séquence donnée. Il s'agit en fait de définir la vitesse des objets mobiles dans la scène. Pour ceci, il faut tout d'abord modéliser les images.

2.1.1 Représentation des images

Les images que nous utilisons comportent 256 niveaux de gris, et elles sont de taille $M \times N$, où M est le nombre de lignes de l'image, et N celui des colonnes (*cf. fig 2.1*). Cela signifie que notre image comporte $M \times N$ pixels. Chaque pixel correspond à un *site* s .

Soit $f(t) = \{f(s, t), s \in S\}$, la fonction de luminance, c'est-à-dire d'intensité à l'instant t , avec $S = [1, M] \times [1, N]$ correspondant à l'ensemble des sites de l'image. Il nous faut trouver des correspondances entre $f(t)$ et $f(t + 1)$. Pour cela, nous devons d'abord déterminer un ensemble de voisinages.

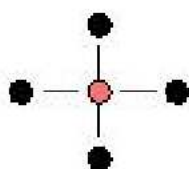
2.1.2 Système de voisinage

Les relations de voisinage sur S les plus couramment utilisées sont celles des 4 plus proches voisins, ou encore celles des 8 plus proches voisins. On notera $\mathcal{G}_s$ le système de voisinage associé au site s .

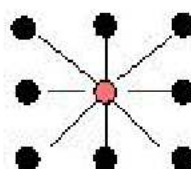
Une fois les relations de voisinage fixées, il suffit de construire un graphe non orienté en reliant tout simplement deux sites s_1 et s_2 s'ils sont voisins.



FIG. 2.1 – Exemple de deux images consécutives : (a) Image à l'instant t .
(b) Image à l'instant $t+1$.



4 voisins



8 voisins

FIG. 2.2 – Relation de voisinage aux 4 et 8 plus proches voisins.

La donnée du système de voisinage est équivalente à la donnée de l'ensemble $\mathcal{C}$, dont les éléments c , appelés *cliques*, sont les parties de S vérifiant :

$$\begin{cases} \exists s \in S \mid c = \{s\} \\ \text{ou} \\ \forall \{s, t\} \subset c, \quad t \in \mathcal{G}_s. \end{cases}$$

Nous disposons de cliques unaires et binaires.

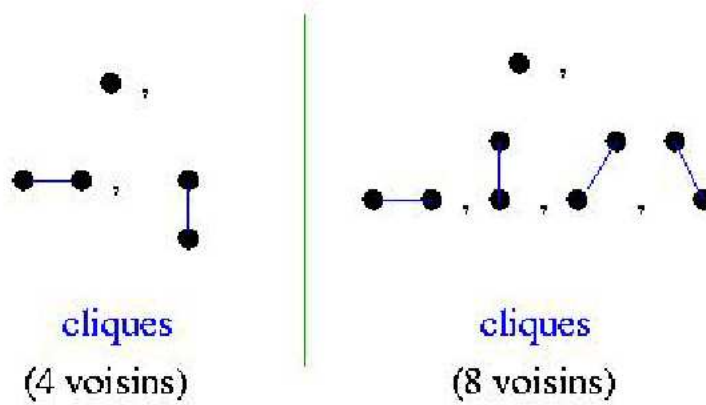


FIG. 2.3 – Cliques associées aux relations des 4 ou 8 plus proches voisins.

2.2 Champs de Markov

2.2.1 Définition

Une famille de variables aléatoires Σ est dit être un champ de Markov sur S , par rapport à un système de voisinage, en l'occurrence celui que nous venons de définir, si et seulement si :

$$P(\sigma) > 0, \quad \forall \sigma \in \Sigma$$

$$P(\sigma_i \mid \sigma_{S-\{i\}}) = P(\sigma_i \mid f_{\mathcal{G}_i}), \quad \forall \sigma \in \Sigma$$

avec : $f_{\mathcal{G}_i} = \{\sigma_{i'} \mid i' \in \mathcal{G}_i\}$ notre système de voisinage.

Cela signifie, que le comportement de la variable aléatoire portée par un site est totalement déterminé par la réalisation des variables aléatoires voisines.

2.2.2 Théorème de Hammersley-Clifford

Ce théorème démontre que Σ est un champ markovien, associé à notre système de voisinage $\mathcal{G}$, si et seulement si $P_\Sigma(\sigma)$ est une mesure de Gibbs, c'est-à-dire :

$$\forall \sigma \in \Omega, \quad P_\Sigma(\sigma) = \frac{1}{Z} \exp(-U(\sigma))$$

où U est une fonction appelée *énergie*, et Ω représente l'ensemble des configurations possibles pour Σ . Par ailleurs Z , qui s'appelle fonction de répartition, est en réalité la constante de normalisation de la probabilité $P_\Sigma(\sigma)$.

$$Z = \sum_{\sigma \in \Omega} \exp(-U(\sigma)).$$

En fait, nous avons :

$$U(\sigma) = \sum_{c \in \mathcal{C}} V_c(\sigma).$$

Les fonctions V_c sont des fonctions de potentiel, et elles ne dépendent que des éléments de σ qui appartiennent à la clique c .

2.3 Estimation bayésienne

2.3.1 Relation observations-étiquettes

L'estimation bayésienne permet de modéliser conjointement les étiquettes et les observations à l'aide d'un champ aléatoire couplé. Soit alors :

$$(X, Y) = (\Omega \times \Omega_{obs}, P_{XY})$$

avec Ω qui est l'ensemble des configurations d'étiquettes $x = \{x_s, s \in S\}$ définies sur S , et Ω_{obs} , qui est l'ensemble des configurations d'observations $y = \{y_r, r \in R\}$ définies sur R^1 .

La composante X de ce champ couplé constitue le *champ d'étiquettes*, alors que la composante Y constitue le *champ d'observations*. Comme nous

¹Les deux ensembles S et R peuvent être confondus. Mais la plupart du temps, nous avons $S \subset R$

disposons de champs discrets, pour tout $(x, y) \in \Omega \times \Omega_{obs}$, $P_{Y|X}(y|x)$ correspond à la probabilité $Pr\{Y = y \mid X = x\}$ du champ d'observations conditionnellement aux étiquettes. Cette distribution conditionnelle donne la vraisemblance des observations.

2.3.2 Règle de Bayes

Elle nous permet d'atteindre la distribution jointe P_{XY} du champ couplé :

$$P_{XY}(x, y) = P_{Y|X}(y|x)P_X(x),$$

d'où découle pour une réalisation donnée y du champ d'observations, la distribution *a posteriori* des étiquettes :

$$P_{X|Y}(x|y) = \frac{P_{Y|X}(y|x)P_X(x)}{P_Y(y)}.$$

Or, $P_Y(y)$ est constante, car la réalisation du champ d'observations, c'est-à-dire les données (images) que nous traitons, est connue. Par conséquent nous obtenons :

$$P_{X|Y}(x|y) \propto P_{Y|X}(y|x)P_X(x). \quad (2.1)$$

2.4 Caractérisation de l'énergie

Comme nous utilisons une modélisation markovienne, nous avons vu que :

$$\forall x \in \Omega, \quad P_X(x) = \frac{1}{Z} \exp \left(- \sum_{c \in \mathcal{C}} V_c(x) \right). \quad (2.2)$$

Il nous faut à présent déterminer la distribution des observations conditionnellement aux étiquettes. Cela dépend de la façon dont les observations sont formées à partir des étiquettes. La plupart du temps, le processus d'élaboration comprend la convolution par un filtre. D'après [8], on obtient l'expression suivante :

$$P_{Y|X}(y|x) = \exp \left(- \sum_{s \in S} W_s(x_s, y) \right), \quad (2.3)$$

où les fonctions W_s reflètent le processus de formation des observations en chaque site. En combinant (2.1), (2.2) et (2.3) nous avons donc :

$$P_{X|Y}(x|y) \propto \frac{1}{Z} \exp \left(- \sum_{c \in \mathcal{C}} V_c(x) \right) \exp \left(- \sum_{s \in S} W_s(x_s, y) \right)$$

$$\begin{aligned} &\propto \frac{1}{Z} \exp \left(- \sum_{c \in \mathcal{C}} V_c(x) - \sum_{s \in S} W_s(x_s, y) \right) \\ &\propto \frac{1}{Z} \exp \left(- U(x, y) \right), \end{aligned}$$

avec

$$U(x, y) = \underbrace{\sum_{c \in \mathcal{C}} V_c(x)}_{U_1(x)} + \underbrace{\sum_{s \in S} W_s(x_s, y)}_{U_2(x, y)}.$$

D'une part, $U_1(x)$ correspond au terme d'énergie a priori. Il caractérise et modélise les propriétés du champ d'étiquettes que l'on recherche. D'autre part, $U_2(x, y)$ symbolise le terme d'attache aux données, c'est-à-dire, qu'il modélise le lien entre les observations et les étiquettes. Nous voulons maximiser la probabilité conditionnelle *a posteriori*, ce qui revient en fait à minimiser la fonction d'énergie U .

2.5 Algorithmes classiques

La taille souvent imposante des données, ainsi que surtout la complexité de la fonction d'énergie, fait que nous sommes confrontés à un délicat problème d'optimisation. En effet, la plupart du temps, U est non convexe et non linéaire, ce qui entraîne l'existence de nombreux minima locaux. Il nous faut donc adopter des algorithmes itératifs pour tenter d'approcher l'un des minima globaux. Ces algorithmes se répartissent en deux grandes familles :

- Les algorithmes déterministes qui atteignent en un nombre fini d'itérations un minimum local dépendant de l'initialisation.
- Les algorithmes stochastiques qui utilisent un *recuit simulé*, et qui convergent en un temps infini vers un minimum global indépendant de l'initialisation.

2.5.1 Algorithmes déterministes : ICM

Le principe de l'ICM (*Iterated Conditional Modes*), est qu'à chaque itération, on passe de façon déterministe de la configuration courante à une configuration d'énergie inférieure. Il ne peut donc pas avoir de "remontées d'énergie". De ce fait, en un nombre fini d'étapes nous sommes assurés d'atteindre un minimum local de notre fonction d'énergie. Si x est la configuration courante, alors l'étiquette au site courant s est remplacée par l'étiquette

qui minimise U sur le système de voisinage $\mathcal{G}_s$. D'après la définition de U , l'énergie en s est :

$$U(x_s, y) = \sum_{c \in \mathcal{C}: s \in c} V_c(x) + W_s(x_s, y).$$

On effectue un balayage aller-retour de tous les sites. Le déroulement de l'algorithme ICM est alors :

- A. Initialisation du champ d'étiquettes : $x \leftarrow \hat{x}^{[0]}$.
- B. Calcul de $\hat{x}^{[k+1]}$ à partir de $\hat{x}^{[k]}$.
1. *compteur* = 0.
 2. Balayage aller-retour de tous les pixels :
 - En chaque site s on calcule les $U(x_s, y)$ pour chaque étiquette possible.
 - On choisit l'étiquette qui correspond à l'énergie minimale :

$$\hat{x}_s^{[k+1]} = \arg \min_{x_s} \left(U(x_s, y) \right)$$
 - Si nécessaire, on met à jour :

$$\hat{x}_s^{[k+1]} \neq \hat{x}_s^{[k]} \Rightarrow \begin{cases} x_s \leftarrow \hat{x}_s^{[k+1]} \\ \text{compteur} = \text{compteur} + 1 \end{cases}$$
 3. Si *compteur* > 0, $k = k + 1$ retour en B.

TAB. 2.1 – Algorithme ICM.

Le problème avec l'algorithme ICM, est que pour obtenir un bon résultat, il est capital d'avoir une initialisation qui soit assez proche du minimum global. Si tel n'est pas le cas, alors, il faut se contenter d'un minimum local. Cette contrainte n'apparaît pas pour les algorithmes stochastiques, tels que le recuit simulé. En effet, quelque soit l'initialisation, il convergera vers un minimum global. Ainsi, il n'est pas nécessaire d'effectuer un prétraitement pour déterminer la configuration initiale. Cela représente donc un gain d'effort et de temps, qui nous sera bien utile par la suite...

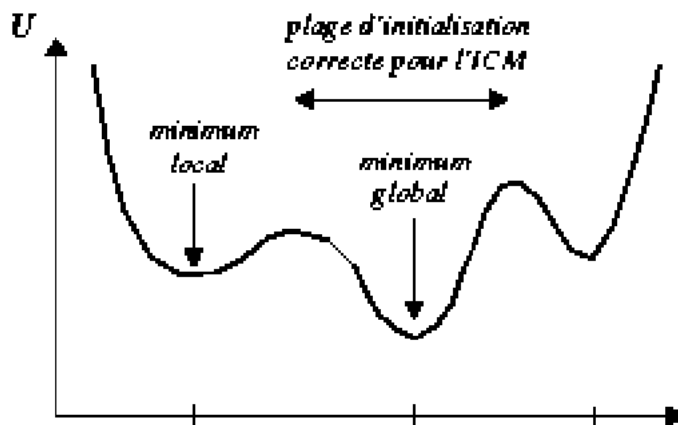


FIG. 2.4 – Initialisation capitale pour l'ICM.

2.5.2 Algorithmes stochastiques : recuit simulé

Le recuit simulé est une méthode de minimisation d'une fonction qui a la propriété de converger vers le minimum global. A l'origine cette technique s'inspire de la physique statistique et de l'étude des phénomènes chimiques liés au refroidissement de systèmes cristallins jusqu'à leur point de stabilité maximale, c'est-à-dire d'énergie minimale. En effet, lorsque la température approche zéro, la seule probabilité non nulle pour que la particule se trouve à un état d'énergie E correspond à l'état d'énergie minimale du système. Si on considère un système de particules à haute température, cet état de la matière correspond à un état de désordre total (maximum d'entropie). Si on refroidit subitement celui-ci, le solide que l'on obtiendra ne sera pas dans son état d'ordre maximal. En effet, en refroidissant brutalement, on fige les particules dans leurs positions désordonnées et aléatoires. Par contre si on abaisse la température de façon lente et si pour chaque température on laisse s'établir l'équilibre thermique, lorsque l'on approche la température nulle, le solide que l'on obtiendra sera dans son état d'énergie minimale. Donc il se trouvera dans son état d'ordre maximal. Le recuit simulé est basé sur le même principe, la fonction à minimiser joue le rôle de l'énergie.

Le passage d'une configuration à une autre se fait de manière statistique,

avec abaissement de la température². A chaque palier, les étiquettes en tout site, sont choisies selon leur probabilité à être réalisable. Tout comme pour l'ICM, on effectue un balayage aller-retour de tous les sites, et le critère d'arrêt peut être le nombre de changement qui est nul d'une étape à l'autre, ou la température qui a atteint la température minimale.

La structure de l'algorithme de recuit simulé est alors :

A. Initialisation du champ d'étiquettes et de la température :

- $x \leftarrow \hat{x}^{[0]}$.
- $T \leftarrow T_0$.

B. Calcul de $\hat{x}^{[k+1]}$ à partir de $\hat{x}^{[k]}$.

1. Balayage aller-retour de tous les pixels :

- En chaque site s on calcule les probabilités pour chaque étiquette e_i possible :

$$p_s^i = \exp \left(- \frac{U(x_s^i, y)}{T} \right)$$

- On tire l'une des étiquettes selon sa probabilité :
 - On extrait $\alpha \in [0; 1]$ selon une loi uniforme
 - On prend le plus petit j tel que $\sum_{i=1}^j p_s^i > \alpha \times \sum_i p_s^i$
 - $\hat{x}_s^{[k+1]} = x_s^j$
- Si nécessaire, on met à jour :

$$\hat{x}_s^{[k+1]} \neq \hat{x}_s^{[k]} \Rightarrow x_s \leftarrow \hat{x}_s^{[k+1]}$$

2. $T = a \times T$

3. $k = k + 1$ retour en B tant que $T \geq T_{min}$

TAB. 2.2 – Algorithme du recuit simulé.

Le recuit simulé permet d'obtenir de très bons résultats, excessivement

²En général, on multiplie la température par $a \in [0, 97; 0, 99]$.

proches des minima globaux (cela dépend du critère d'arrêt). Cela est d'autant plus intéressant, que c'est indépendant de la configuration initiale. Cependant, cet algorithme stochastique est d'une extrême lenteur.

Nous venons de voir deux algorithmes classiques de recherche de minimum. Le premier, rapide et déterministe, donne une solution qui, si l'initialisation est mauvaise, restera bloquée sur un minimum local. Le second, probabiliste, produit une excellente solution, mais cela au prix d'un abondant effort. Il faudrait donc trouver un compromis entre l'efficacité du recuit simulé, atteinte au bout d'un très long temps, et la rapidité de l'ICM, qui ne fournit qu'un minimum local.

Chapitre 3

Résolution multigrille

3.1 Algorithme multirésolution

3.1.1 Principe

Les méthodes multigrilles ont été développées en analyse numérique pour résoudre les équations aux dérivées partielles (EDP). Il s'agit de résoudre un problème initial, avec plusieurs pas de discrétisation. En analyse d'image, cela se traduit par trouver le minimum de la fonction d'énergie à différentes résolutions. Ainsi, lorsque la résolution est plus faible, nous avons à traiter des problèmes moins complexes. Par ailleurs, le résultat trouvé servira pour traiter le problème à une résolution plus élevée. Il va donc apparaître une structure pyramidale, dans laquelle chaque étage correspondra à une résolution donnée. Plus on sera haut dans la pyramide, plus faible sera la résolution, et donc plus simple sera le problème d'optimisation. Pour commencer, nous devons construire la pyramide des observations, à laquelle nous attacherons celle des étiquettes.

3.1.2 Construction des pyramides

Nous avons déterminé dans le chapitre précédent la forme de la fonction d'énergie :

$$U(x, y) = \sum_{c \in \mathcal{C}} V_c(x) + \sum_{s \in \mathcal{S}} W_s(x_s, y).$$

Pour définir des versions plus grossières du problème d'optimisation, considérons des grilles réduites S^i et R^i , pour $i \in [1; n]$, obtenues par décimations successives d'un facteur 2 dans chaque direction à partir des grilles

originales S et R^1 . De ce fait, nous avons S^{i+1} qui est quatre fois plus petite que la grille précédente S^i . Si s est un site de la sous-grille S^i , $i > 1$, on note $\mathcal{D}^i(s)$ l'ensemble de ses $2^i \times 2^i$ descendants sur la grille initiale.

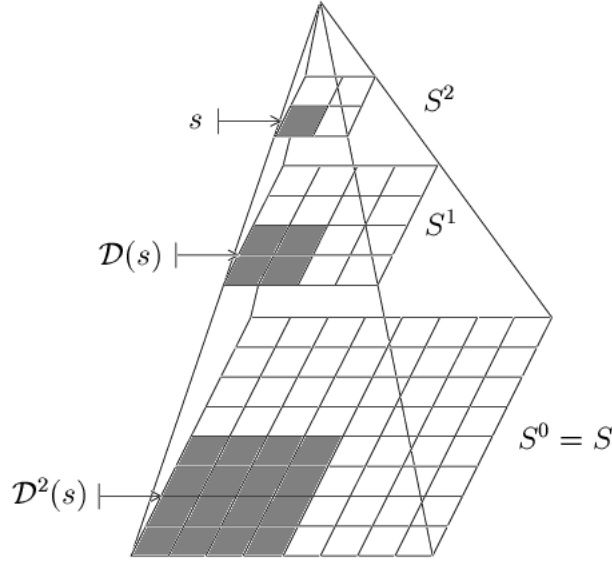


FIG. 3.1 – Pyramide obtenue par décimations successives d'une grille.

Soit Ω^i , l'ensemble des configurations réduites d'étiquettes² définies sur la grille S^i . A présent, nous devons donc minimiser sur Ω^i une certaine fonction $U^i(x^i, y^i)$. Pour cela, il faut déjà déterminer les approximations des observations y^i sur la sous-grille R^i . La séquence y^i , $i \in [1; n]$ est obtenue par filtrages passe-bas et décimations successives des observations y , soit :

$$y^{i+1} = (d \circ H)(y^i), \quad (3.1)$$

où d est une fonction de sous-échantillonnage d'un facteur deux dans chaque direction sur la sous-grille d'observations, est H est un filtre passe-bas bidimensionnel associé par exemple à une famille d'ondelettes, ou tout simplement un filtre de Gauss.

Chaque grille réduite S^i est munie du même système de voisinage que celui de S . On en déduit donc l'ensemble des cliques $\mathcal{C}^i$ associé. De ce fait,

¹On rappelle que S est la grille des étiquettes, et R celle des observations.

²On rappelle que Ω est l'ensemble des configurations des étiquettes.

le terme d'énergie a priori U_1^i , ainsi que celui d'attache aux données U_2^i , sont respectivement calqués sur U_1 et U_2 précédemment définies. Nous obtenons donc :

$$U^i(x^i, y^i) = U_1^i(x^i) + U_2^i(x^i, y^i) \quad \text{soit}$$

$$U^i(x^i, y^i) = \sum_{c \in \mathcal{C}^i} V_c^i(x^i) + \sum_{s \in S^i} W_s^i(x_s^i, y^i).$$

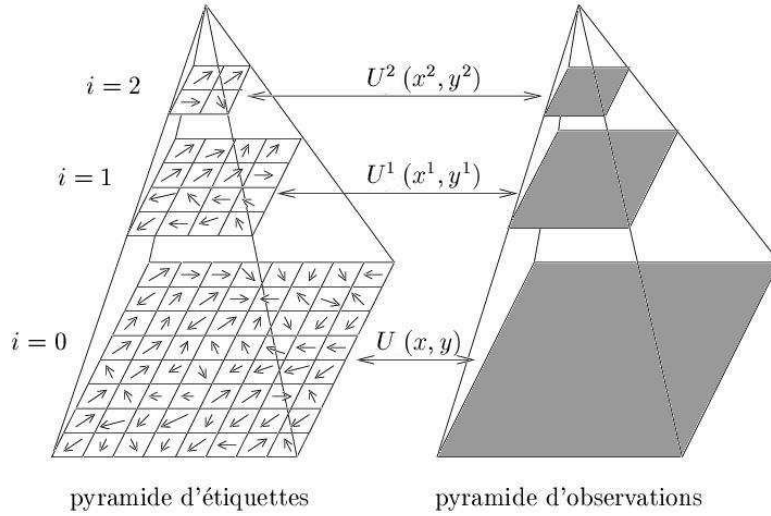


FIG. 3.2 – Relation entre les pyramides d'étiquettes et d'observations.

Par conséquent, nous possédons deux pyramides : celle des observations que nous venons de construire, et celle des étiquettes. A chaque niveau, les étiquettes sont reliées aux observations qui leur correspondent. Ce modèle est généralement utilisé selon la méthode *coarse to fine*, c'est-à-dire de la résolution la plus grossière vers la plus fine³. Nous commençons donc par résoudre au niveau le plus grossier. Une fois les étiquettes obtenues, nous devons les transmettre au niveau inférieur. A l'instar de [8], nous définissons un opérateur d'interpolation d'un niveau $i + 1$ sur le niveau inférieur i . Il s'agit en fait d'une duplication de chaque site de S^i en un bloc 2×2 , soit :

$$D_{i+1 \rightarrow i}(x^{i+1}) = x^i \iff \forall s \in S^{i+1}, \forall t \in \mathcal{D}(s), x_s^{i+1} = x_t^i.$$

³donc dans le sens des i décroissants

La résolution *coarse to fine* est alors :

- Le problème d'optimisation est tout d'abord traité au niveau le plus grossier ($i = n$) à l'aide d'un algorithme de relaxation déterministe ou stochastique qui fournit une configuration $\tilde{x}^n$ de Ω^n , à partir de l'initialisation x_0^n .
- par la suite, si $\tilde{x}^i$ représente la configuration de Ω^i résultant de la relaxation au niveau i , la relaxation au niveau inférieur $i - 1$ est initialisée par l'interpolé $D_{i \rightarrow i-1}(\tilde{x}^i)$.
- On obtient enfin une configuration des étiquettes $\tilde{x}$ au niveau le plus fin ($i = 0$).

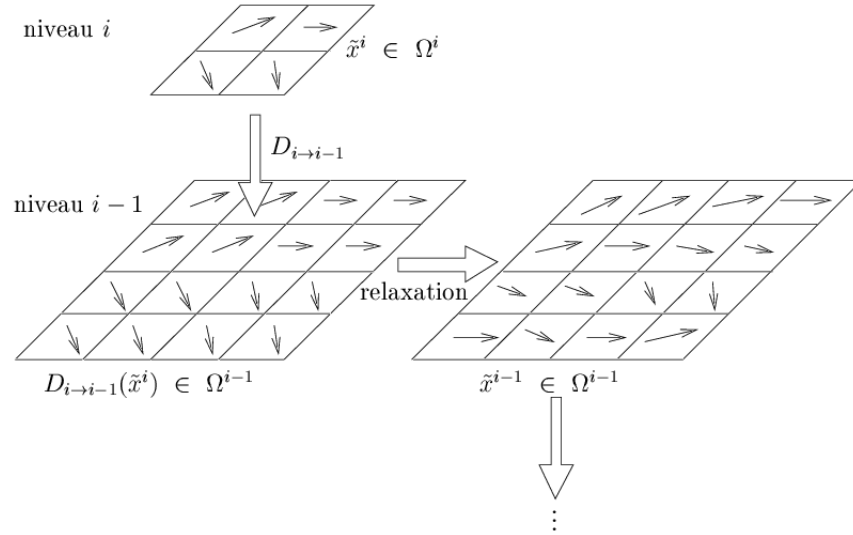


FIG. 3.3 – Stratégie multirésolution descendante avec duplication.

Le fonctionnement de l'algorithme multirésolution est donc tout simplement :

- $i \leftarrow 0$
- Tant que $i < n$ répéter :
 - $i \leftarrow i + 1$
 - $y^i \leftarrow (d \circ H)(y^{i-1})$
- $i \leftarrow n + 1$
- Tant que $i > 0$ répéter :
 - $i \leftarrow i - 1$
 - Si $i = n$ alors $x^i \leftarrow x_0^n$, sinon $x^i \leftarrow D_{i+1 \rightarrow i}(x^{i+1})$
 - Relaxation par ICM avec les observations y^i pour obtenir x^i

TAB. 3.1 – Algorithme multirésolution utilisant l'ICM à chaque niveau.

3.2 Algorithme multiéchelle

Pour l'algorithme multiéchelle développé dans [8] et [3], il y a seulement une seule pyramide. En effet, on ne retrouve que la pyramide des étiquettes. C'est-à-dire qu'il y a toujours une résolution par stratégie *coarse to fine*, mais les observations restent à pleine résolution. Ainsi, cela évite de créer, puis de stocker une pyramide des observations.

De ce fait, l'expression de l'énergie ne peut plus être la même. Il ne suffit plus ici, comme dans le modèle multirésolution, de calquer l'énergie initiale à chaque niveau de la pyramide. En fin de compte, pour un site $s \in S$, il correspondra un ensemble de sites dans R , dont le cardinal dépendra du niveau dans lequel se situe l'étiquette. Si s est un site de la sous-grille S^i , $i > 1$, on notera $\mathcal{B}^i(s, x_s^i)$ l'ensemble des sites de R lié au site s portant l'étiquette x_s^i .

Par conséquent, l'expression de l'énergie au niveau i devient :

$$\begin{aligned}
 U^i(x^i, y) &= U_1^i(x^i) + U_2^i(x^i, y) \quad \text{soit} \\
 U^i(x^i, y) &= \sum_{c \in \mathcal{C}^i} V_c^i(x^i) + \sum_{s \in S^i} W_s^i(x_s^i, y_{\mathcal{B}^i(s, x_s^i)}). \quad (3.2)
 \end{aligned}$$

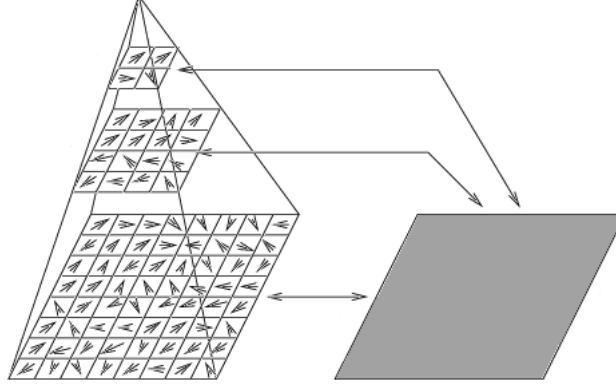


FIG. 3.4 – Structure de la résolution multiéchelle.

Le fonctionnement de l'algorithme multiéchelle est donc tout simplement :

- $i \leftarrow n + 1$
- Tant que $i > 0$ répéter :
 - $i \leftarrow i - 1$
 - Si $i = n$ alors $x^i \leftarrow x_0^n$, sinon $x^i \leftarrow D_{i+1 \rightarrow i}(x^{i+1})$
 - Relaxation par ICM avec les observations y pour obtenir x^i

TAB. 3.2 – Algorithme multiéchelle utilisant l'ICM à chaque niveau.

A présent que nous venons de décrire les fondements des algorithmes de multirésolution et multiéchelle, nous allons les appliquer sur divers problèmes de traitement d'images. Tout d'abord, nous allons nous intéresser à la détection de mouvement, puis ensuite à la mesure de déplacement. Par la suite, notamment lorsque nous verrons d'autres algorithmes, nous nous n'intéresserons qu'au second, qui est un problème de détermination du flux optique.

3.3 Application à divers modèles

Aussi bien pour la détection de mouvement, que pour le mesure de déplacement, nous nous servons des modèles présentés dans [8]. Les observations,

quantifiées sur 256 niveaux de gris, sont extraites de séquences d'images numériques. On notera $f_k(s)$ la luminance associée au site s pour la $k^{\text{ième}}$ image. Nous avons donc :

$$f_k(s) \in \{0, \dots, 255\}.$$

3.3.1 Modèle de détection de mouvement

La détection de mouvement est fréquemment utilisée en surveillance périmétrique, ou encore en télésurveillance du trafic routier. Son objectif est d'associer à chaque pixel, une étiquette binaire indiquant l'appartenance ou non du point considéré à une zone en mouvement. Il faut tout d'abord, déterminer le modèle des observations, ainsi que celui des étiquettes.

Les observations

La détection de mouvement repose sur l'analyse des variations de la fonction de luminance entre deux instants consécutifs $k - 1$ et k . De ce fait, la différence inter-images apparaît naturellement comme l'une des composantes du champ d'observations. Ainsi, en un site s , l'une des observations sera définie selon :

$$o_s(k) = |f_k(s) - f_{k-1}(s)|.$$

A ces observations quantitatives, sont ajoutées des observations symboliques binaires indiquant la présence ou l'absence de variations significatives de luminance au site s entre les instants $k - 1$ et k , soit :

$$\bar{o}_s(k) = \begin{cases} 1 & \text{si variations suffisantes} \\ 0 & \text{sinon} \end{cases}$$

Par ailleurs, les observations aux instants k et $k + 1$ sont indispensables pour la détection des mouvements à l'instant k . Une réalisation du champ d'observation est donc de la forme :

$$y = (o(k), o(k + 1), \bar{o}(k), \bar{o}(k + 1)).$$

Les étiquettes

L'ensemble S des sites à étiqueter est confondu avec la grille R des observations. Les étiquettes, binaires, indiquent la présence ou l'absence de mouvement en chacun des sites s à l'instant k , soit :

$$x_s = \begin{cases} 0 & s \in \text{zone statique} \\ 1 & s \in \text{zone en mouvement} \end{cases}$$

La fonction d'énergie

Auparavant, nous avons vu que la fonction d'énergie était définie en un site s par :

$$U = \sum_{c \in \mathcal{C}: s \in c} V_c(x) + W_s(x_s, y).$$

Le premier terme représente les potentiels sur les cliques binaires. On prend des potentiels isotropes, afin de favoriser des masques de détection compacts. On a donc :

$$\forall \{s, t\} \in \mathcal{C}, V_{s,t}(x) = \begin{cases} +\beta_1 & \text{si } x_s \neq x_t \\ -\beta_1 & \text{si } x_s = x_t \end{cases}$$

où β_1 est un paramètre réel positif.

Pour déterminer le potentiel en un site s , il suffit donc de regarder les étiquettes de ses voisins.

Par ailleurs, on décompose la fonction W_s en deux termes :

$$W_s(x_s, y_s) = W_{s,1}(x_s, o_s(k), o_s(k+1)) + W_{s,2}(x_s, \bar{o}_s(k), \bar{o}_s(k+1)).$$

$W_{s,1}$ exprime localement le lien entre les variations temporelles de luminance dans la paire $(o(k), o(k+1))$, et le champ d'étiquette x . Cependant, la détection des variations temporelles de luminance n'est pas suffisante en soi. En effet, on aura des variations à la fois dans les zones découvertes par l'objet en mouvement, et dans celles recouvertes par ce même objet. cette énergie s'exprime sous la forme :

$$W_{s,1}(x_s, o_s(k), o_s(k+1)) = \frac{1}{2\sigma^2} \left(\left(o_s(k+1) - \Psi(x_s) \right)^2 + \left(o_s(k) - \Psi(x_s) \right)^2 \right)$$

avec

$$\Psi(x_s) = \begin{cases} 0 & \text{si } x_s(k) = 0 \\ \alpha & \text{si } x_s(k) = 1, \end{cases}$$

où σ^2 est une variance connue, et α un paramètre réel positif.

Le second terme $W_{s,2}$ permet d'éviter la prise en compte des zones d'occlusion, qui sont des zones apparaissant ou disparaissant entre deux instants. De ce fait, on ne considéra réellement que l'objet qui est en mouvement. Les valeurs de cette fonction sont représentées dans la table 3.3, avec β_2 qui est un paramètre réel positif.

Nos images sont étalonnées sur 256 niveaux de gris, allant du noir, valeur 0, au blanc, valeur 255. Si une zone est statique, alors $x_s = 0$, et la couleur associée sera le noir. Par contre, si elle est en mouvement, $x_s = 1$, et la couleur associée sera le blanc. Informatiquement, elle vaudra donc 255.

x_s	$\bar{o}_s(k)$	$\bar{o}_s(k+1)$	$W_{s,2}$
0	0	0	$-\beta_2/2$
0	0	1	$-\beta_2/2$
1	0	0	β_2
1	0	1	β_2
0	1	0	$-\beta_2/2$
0	1	1	β_2
1	1	0	β_2
1	1	1	$-\beta_2/2$

TAB. 3.3 – Définition de $W_{s,2}(x_s, \bar{o}_s(k), \bar{o}_s(k+1))$.

Par conséquent, le calcul de $W_{s,2}$ donne :

```

int val = 0;

val += matbap[i][j];
val += matb[i][j] << 1;
val += etiquette << 2;

switch (val)
{
    case 0      : return (double)(-BETA2/2);
                  break;

    case 255    : return (double)(-BETA2/2);
                  break;

    case 510    : return (double)(-BETA2/2);
                  break;

    case 765    : return (double)BETA2;
                  break;

    case 1020   : return (double)BETA2;
                  break;

    case 1275   : return (double)BETA2;
                  break;
}

```

```

case 1530 : return (double)BETA2;
           break;

case 1785 : return (double)(-BETA2/2);
           break;

default : cout << "Cela est impossible!" << endl;
          exit(0);
}

```

`matbap` et `matb` sont respectivement les matrices de stockage de $\bar{o}(k+1)$ et $\bar{o}(k)$.

La fonction d'énergie au niveau i du modèle multiéchelle

Pour le modèle multiéchelle, d'après la définition de la fonction d'énergie au niveau i donnée en (3.2), nous obtenons en décomposant W_s , et en différenciant les potentiels sur les cliques unaires et binaires :

$$\begin{aligned}
U^i(x^i, y) = & \sum_{s \in S^i} V_{\{s\}}^i(x^i) + \sum_{\{s,t\} \in \mathcal{C}^i} V_{\{s,t\}}^i(x^i) \\
& + \sum_{s \in S^i} W_{s,1}^i(x_s^i, o_{b_s^i}(k), o_{b_s^i}(k+1)) + W_{s,2}^i(x_s^i, \bar{o}_{b_s^i}(k), \bar{o}_{b_s^i}(k+1)),
\end{aligned}$$

où b_s^i correspond à l'ensemble des $2^i \times 2^i$ sites de la grille d'observations liés au site s^4 .

Les potentiels sur les cliques de $\mathcal{C}^i$ valent :

$$\left\{ \begin{array}{ll} \text{pour } s \in S^i & V_{\{s\}}^i(x^i) = -p^i \beta_1 \\ \text{pour } \{s, t\} \in \mathcal{C}_{hv}^i & V_{\{s,t\}}^i(x^i) = \begin{cases} +q_{hv}^i \beta_1 & \text{si } x_s^i \neq x_t^i \\ -q_{hv}^i \beta_1 & \text{si } x_s^i = x_t^i \end{cases} \\ \text{pour } \{s, t\} \in \mathcal{C}_d^i & V_{\{s,t\}}^i(x^i) = \begin{cases} +q_d^i \beta_1 & \text{si } x_s^i \neq x_t^i \\ -q_d^i \beta_1 & \text{si } x_s^i = x_t^i \end{cases} \end{array} \right.$$

avec $p^i = 2(2^i - 1)(2^{i+1} - 1)$, $q_{hv}^i = 3 \times 2^i - 2$ et $q_d^i = 1$.

⁴On a $\mathcal{B}^i(s, x_s^i) = b_s^i$, car l'étiquette au site s de S^i est rattachée aux seules observations situées dans b_s^i .

Par ailleurs, le lien entre les étiquettes x^i et les observations y est défini par :

$$W_{s,1}^i(x_s^i, o_{b_s^i}(k), o_{b_s^i}(k+1)) = \frac{1}{2\sigma^2} \sum_{r \in b_s^i} \left(\left(o_r(k+1) - \Psi(x_s^i) \right)^2 + \left(o_r(k) - \Psi(x_s^i) \right)^2 \right)$$

et

$$W_{s,2}^i(x_s^i, \bar{o}_{b_s^i}(k), \bar{o}_{b_s^i}(k+1)) = \sum_{r \in b_s^i} W_{s,2}(x_s^i, \bar{o}_r(k), \bar{o}_r(k+1)).$$

A présent que nous venons de caractériser les diverses fonctions d'énergie, nous allons détecter les mouvements dans la séquence dont sont extraites les images de la figure (2.1) avec les quatre algorithmes que nous venons de présenter. Les images, de taille 256×256 , proviennent d'une caméra stationnaire placée au **Durlacher-Tor-Platz à Karlsruhe (Allemagne)**. Cinq voitures tournent sur la gauche à une intersection. La voiture en bas à gauche, à cause de sa couleur foncée, apparaît beaucoup plus homogène que les quatre autres. En effet, pour ces dernières, on peut aisément distinguer les vitres de la carrosserie.

Le champ d'observations ressemble à :

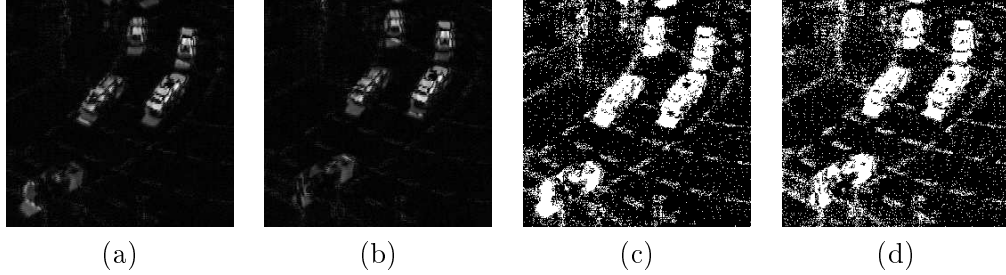


FIG. 3.5 – Observations : (a) $o(k)$, (b) $o(k+1)$, (c) $\bar{o}(k)$, (d) $\bar{o}(k+1)$

Les paramètres sont choisis selon :

$$\beta_2 = 20 \quad \beta_1 = 100 \quad \alpha = 10.$$

Relaxation déterministe :

Comme l'initialisation est capitale pour l'ICM, il faut prendre une configuration qui indique déjà certaines zones où il y a du mouvement. A cet effet, on choisit $x_0 \equiv \bar{o}(k)$. On obtient un minimum local au bout de $3s$, avec une énergie valant -51159400 .



FIG. 3.6 – Relaxation obtenue par ICM.

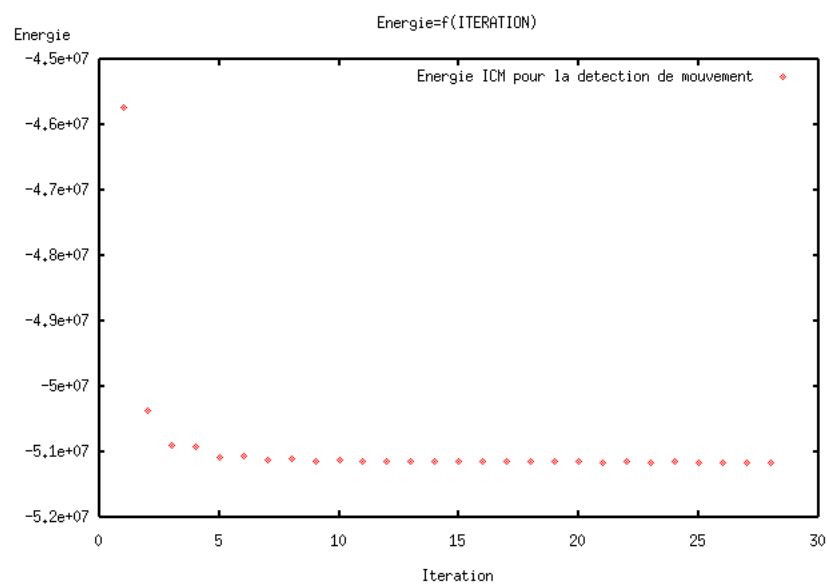


FIG. 3.7 – Evolution de l'énergie au cours de l'ICM.

Relaxation stochastique :

Nous savons que la configuration initiale n'influe pas sur le résultat du recuit simulé. Nous initialisons donc aléatoirement. La température de départ est $T = 300$, et le taux de descente vaut $a = 0.97$. On arrête, lorsque d'une étape à l'autre, il n'y a aucun changement. On obtient une énergie finale de -51892930 au bout de 262 itérations en 42s.



FIG. 3.8 – Relaxation obtenue par recuit simulé.

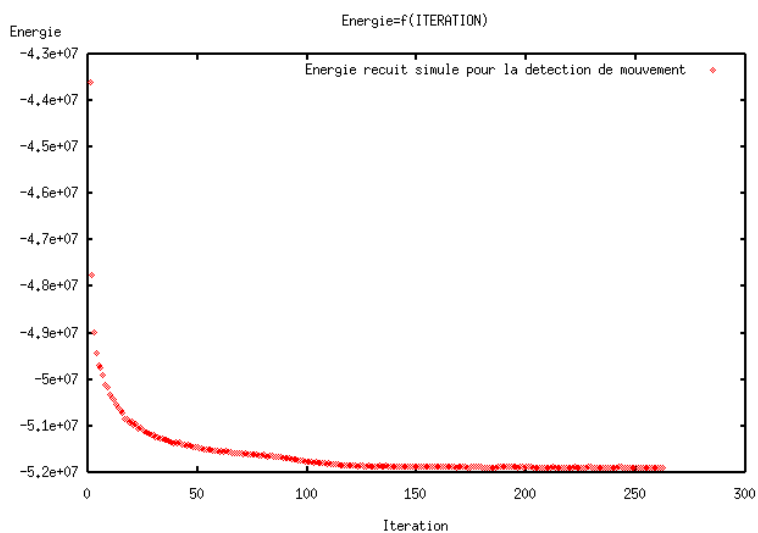


FIG. 3.9 – Evolution de l'énergie au cours du recuit simulé

Relaxation multirésolution :

Pour l'algorithme multirésolution, nous prenons une pyramide à $n = 3$ niveaux. A chaque étage, l'énergie est telle que nous l'avons définie auparavant. La configuration initiale au niveau n est confondue avec $\bar{o}^n(k)$, qui est le résultat de la montée de la troisième composante du champ d'observations dans la pyramide. D'ailleurs, dans la formule (3.1), qui indiquait comment on passait d'une résolution donnée à une plus grossière, pour H , on prend un filtre QMF séparable de support 9×9 , dont les coefficients valent :

$$[0.01995, -0.04271, -0.05224, 0.29271, 0.056458, \\ 0.29271, -0.05224, -0.04271, 0.01995].$$

Le filtrage se fait dans le domaine spectral, par l'intermédiaire d'une transformée de Fourier. La figure (3.11) montre les trois images consécutives de la séquence nécessaires à l'algorithme, ainsi que les pyramides relatives aux observations, dont la légende est :

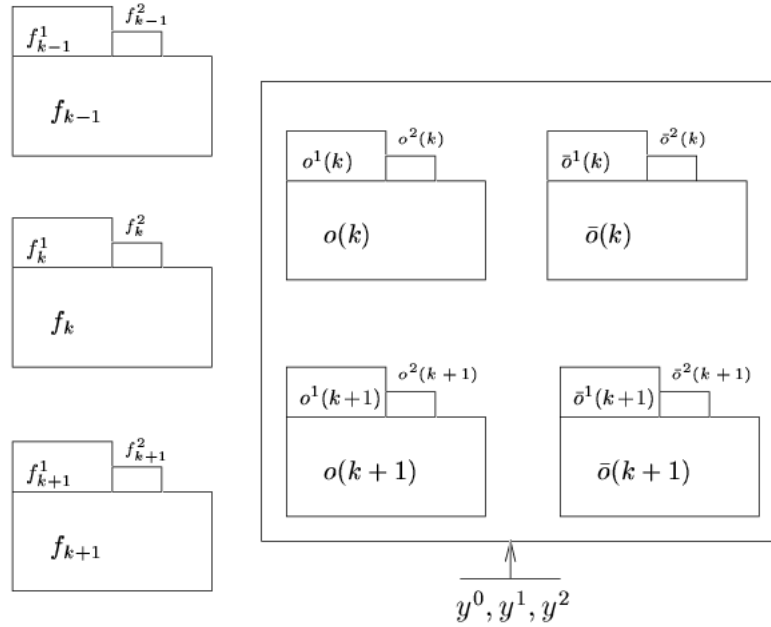


FIG. 3.10 – Légende des pyramides.

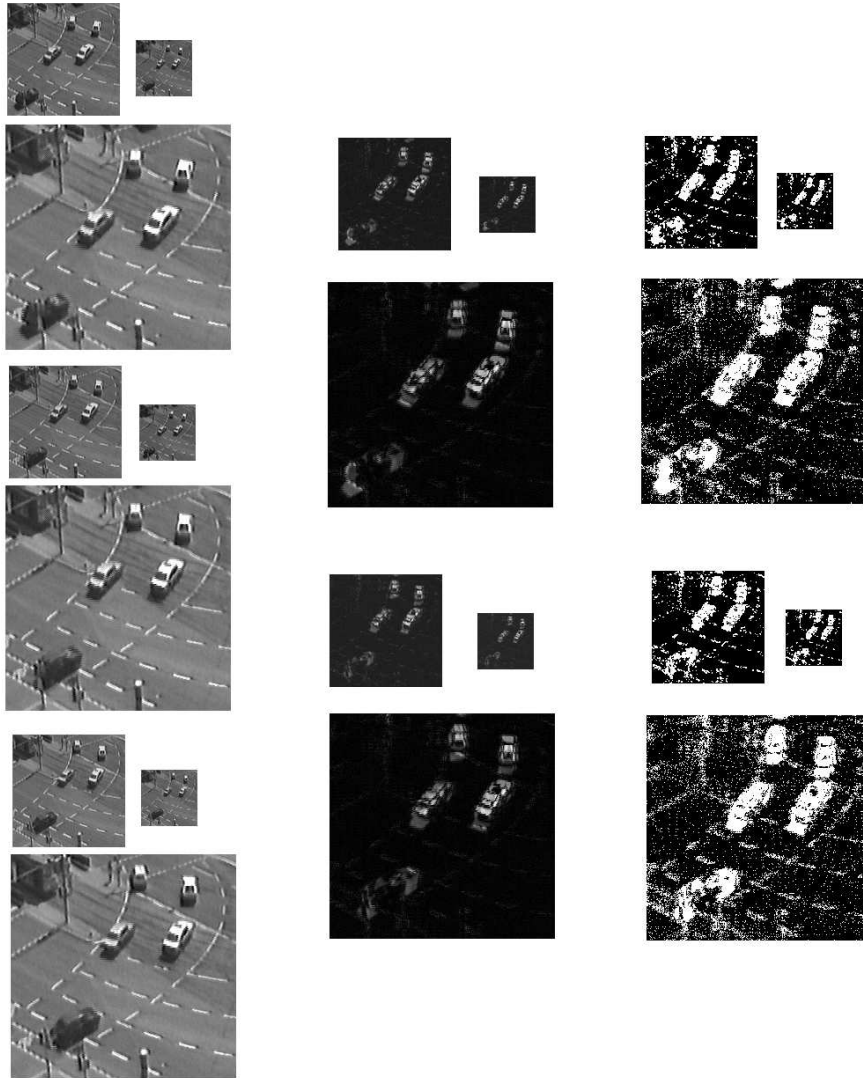


FIG. 3.11 – Pyramides des images et observations associées.

L'énergie finale obtenue en $3s$ vaut -51735710 .



FIG. 3.12 – Relaxation obtenue par multirésolution à 3 niveaux.

Relaxation multiéchelle : Pour la résolution multiéchelle, nous prenons également une pyramide des observations à trois étages. Par contre, l'initialisation au 3^{ième} niveau est aléatoire. A chaque palier, l'énergie est comme que nous l'avons définie auparavant. Au bout de $3s$, nous avons une énergie finale de -50858513 , avec des étiquettes selon :



FIG. 3.13 – Relaxation multiéchelle à 3 niveaux.

Nous pouvons remarquer que les différentes énergies obtenues, sont sensiblement les mêmes. Cela signifie sans doute que la fonction d'énergie ne présente pas de minima locaux prononcés. De plus, la convergence est plutôt rapide. En effet, nous n'avons le choix qu'entre deux classes, ainsi à chaque étape de l'ICM, comme du recuit simulé, pour chaque site s , il n'y a que deux énergies à évaluer. Par ailleurs, il faut souligner l'initialisation de bonne qua-

lité fournie par le champ binaire $\bar{o}(k)$.

En outre, comme on pouvait se l'attendre, la voiture en bas à gauche, n'est pas classée comme étant en mouvement par tous les algorithmes. Le recuit simulé, qui a l'énergie la plus faible, donc le meilleur résultat, ainsi que l'approche multirésolution, qui a la deuxième meilleure énergie, ne la voient pas du tout, alors que la résolution multiéchelle n'en voit que l'avant. L'ICM, étant bloqué sur un minimum local, classe certaines zones en mouvement, alors qu'elles ne le sont pas. C'est pour ceci, que les véhicules paraissent plus larges, et que l'on a l'arrière de la voiture en bas à gauche. Cette auto est très homogène, entre deux images, il est difficile de s'apercevoir si elle a avancé, ce qui complique drôlement la tâche. Effectivement, les variations temporelles détectées sont excessivement faibles, il suffit de regarder le champ d'observations pour s'en rendre compte. De surplus, il y a un panneau qui est devant la voiture. Tout ceci, explique amplement le fait, que ce véhicule ne soit pas totalement classé comme étant en mouvement. Par contre, les quatre autres véhicules sont dans l'ensemble bien détectés. Il faut juste remarquer, que les deux voitures les plus à droite, font partie d'une classe connexe, car elles se suivent de trop près.

3.3.2 Modèle de mesure de déplacement

Nous venons de voir le fonctionnement de quatre algorithmes de relaxation sur un problème de détection de mouvement. Maintenant, nous allons nous intéresser à la résolution du flux optique, c'est-à-dire à la détermination, pour chaque site s , d'un vecteur de déplacement $2D$. Cette estimation dense d'un champ de déplacement nous permet d'accéder à une importante information sur le contenu dynamique de la scène. En effet, non seulement elle nous donne accès aux zones qui sont en mouvement, mais en plus elle nous indique la direction, ainsi que l'intensité du mouvement. Ces données sur la nature du mouvement des objets, sont fortement utilisées en robotique mobile, en télésurveillance, ou encore en codage d'images.

Les observations

Dans ce modèle, nous allons utiliser la *différence inter-images déplacée*. Les luminances observées en chaque site s , $f_k(s)$, sont les seules observations dont nous allons nous servir pour l'estimation. De ce fait, le champ d'observations à un instant k est constitué des images aux instants k et $k + 1$, soit :

$$\forall s \in R, \quad y_s = (f_s(k), f_{k+1}(s)).$$

Les étiquettes

Ce qui nous intéresse, c'est le champ de déplacement $x = x_s, s \in S$ entre les images k et $k+1$. Le vecteur $x_s \in \mathbb{Z}^2$, exprime le fait que le point physique situé au site s dans l'image k , soit au temps t , se retrouve dans l'image $k+1$, soit à $t+1$, au site $s+x_s$. On émet l'hypothèse que entre les instants t et $t+1$, les objets se déplacent dans un rayon maximal de d_{max} , c'est-à-dire que $x_s \in [-d_{max}; d_{max}]^2$ où $d_{max} \in \mathbb{N}^*$. Les observations étant liées au site s , et au site déplacé $s+x_s$, il faut donc que pour tout déplacement les deux sites appartiennent à la grille R des observations, d'où :

$$\forall s \in S, \quad s+x_s \in R.$$

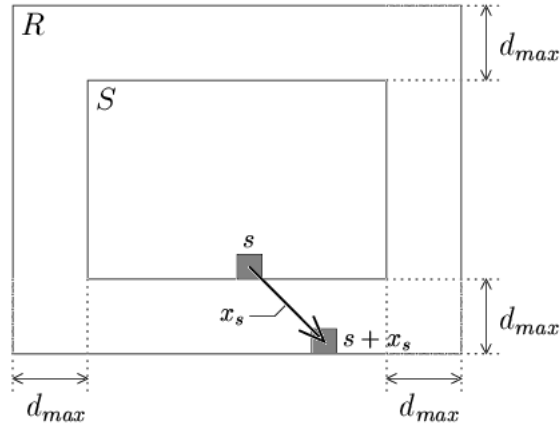


FIG. 3.14 – Positionnement relatif des grilles S et R .

En chaque site, nous avons $(2d_{max} + 1)^2$ classes possibles pour les étiquettes.

La fonction d'énergie

Les potentiels sur les cliques binaires de $\mathcal{C}$ sont des potentiels quadratiques. Cela favorise les champs de déplacement lisses, en évitant du coup les variations brutales entre deux sites voisins, ce qui donne :

$$\forall \{s, t\} \in \mathcal{C}, \quad V_{\{s, t\}}(x) = \beta \|x_s - x_t\|^2,$$

avec β qui est un paramètre réel positif, et $\|\cdot\|$ qui désigne la norme euclidienne sur $\mathbb{Z}^2$.

Par ailleurs, les fonctions W_s sont choisies de manière à ce qu'elles favorisent les champs de déplacement pour lesquels la luminance d'un point ne varie pas de manière significative entre deux images consécutives⁵, soit :

$$W_s(x_s, y) = \left(f_k(s) - f_{k+1}(s + x_s) \right)^2.$$

La fonction d'énergie au niveau i du modèle multiéchelle

Pour le modèle multiéchelle, au niveau i , les potentiels sur les cliques binaires de $\mathcal{C}^i$ valent :

$$\begin{cases} \text{pour } \{s, t\} \in \mathcal{C}_{hv}^i & V_{\{s,t\}}^i(x^i) = q_{hv}^i \beta \|x_s^i - x_t^i\|^2 \\ \text{pour } \{s, t\} \in \mathcal{C}_d^i & V_{\{s,t\}}^i(x^i) = q_d^i \beta \|x_s^i - x_t^i\|^2 \end{cases}$$

avec $q_{hv}^i = 3 \times 2^i - 2$ et $q_d^i = 1$.

De plus, le lien entre les étiquettes x^i et les observations y est défini selon :

$$W_s^i(x_s^i, y_{\mathcal{B}^i(s, x_s^i)}) = \sum_{r \in b_s^i} \left(f_k(r) - f_{k+1}(r + x_s^i) \right)^2,$$

dans ce modèle, nous avons $\mathcal{B}^i(s, x_s^i) = b_s^i \cup b_{s+x_s^i}^i$ car l'étiquette au site $s \in S^i$ est liée aux observations situées dans b_s^i et $b_{s+x_s^i}^i$, qui comportent à elles deux $2 \times (2^i \times 2^i)$ sites⁶.

Contrairement à la détection de mouvement étudiée précédemment, nous faisons face à une optimisation extrêmement difficile. En effet, le second terme de l'énergie, que l'on appelle la *différence inter-image déplacée*, est non linéaire. Il y a donc de nombreux minima locaux, et par conséquent les simples algorithmes déterministes devraient pêcher.

Pour comparer à nouveau les quatre algorithmes, nous nous servons de la même séquence d'images qu'auparavant. Nous utilisons $\beta = 20$, et $d_{max} = 8$. Les vecteurs sont tracés par l'intermédiaire de Matlab. Par souci de lisibilité, nous ne représentons qu'un vecteur sur 10. (*cf. Annexe A*).

⁵On retrouve ici la même hypothèse que dans l'équation de contrainte du flot optique : $\frac{Df}{Dt}(s, t) = \frac{\partial f}{\partial t}(s, t) + \vec{\nabla} f(s, t) \cdot \vec{\omega}(s, t) = 0$, où $\vec{\omega}(s, t)$ est le vecteur de déplacement.

⁶On rappelle que b_s^i correspond à l'ensemble des $2^i \times 2^i$ sites de la grille d'observations liés au site s .

Relaxation déterministe :

Nous prenons une configuration initiale aléatoire. Au bout de $3mn05s$, nous atteignons une énergie de 5414656. Comme nous savons que l'initialisation est capitale pour l'ICM, nous obtenons de bien piètres résultats :



FIG. 3.15 – Champ de vecteurs obtenu par ICM.

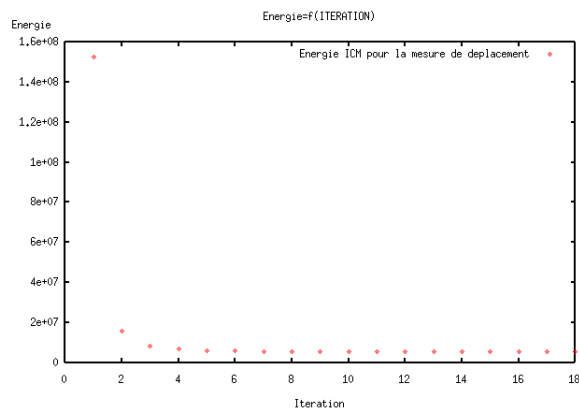


FIG. 3.16 – Progression de l'énergie au cours de l'ICM.

Relaxation stochastique :

Pour le recuit simulé, la température de départ est $T = 500$, et le taux de descente est $a = 0.99999$. On choisit $T_{min} = 0.001$, et ensuite on enchaîne par un ICM⁷. Au bout de *4jours15h7mn33s*, ce qui correspond à 2624464 itérations, nous obtenons une énergie finale de 4209839.

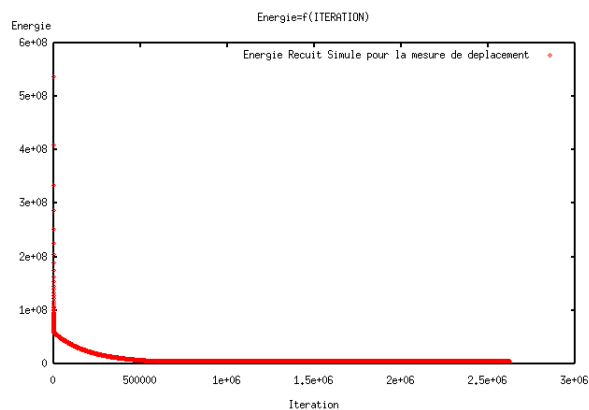


FIG. 3.17 – Progression de l'énergie au cours du recuit simulé.

⁷Effectivement, c'est le seul moyen d'obtenir un bon résultat, car sinon on a des erreurs d'approximations numériques lorsque T devient trop petit.



FIG. 3.18 – Champ de vecteurs obtenu par recuit simulé.

Relaxation multirésolution :

Comme pour la détection de mouvement, nous utilisons une pyramide à $n = 3$ niveaux. La configuration initiale x_0^n , est tirée aléatoirement. A chaque étage i , l'énergie U^i est obtenue par simple duplication de U , qui est telle que nous l'avons définie auparavant. De plus, nous adoptons le même filtrage pour construire la pyramide des observations. Nous obtenons une énergie finale de 4324072 en $2mn47s$.



FIG. 3.19 – Champ de vecteurs obtenu par multirésolution.

Relaxation multiéchelle.

Nous prenons une pyramide à quatre étages, avec initialisation aléatoire au niveau le plus grossier. A chaque palier i , l'énergie est celle que nous avons déterminée. Cette relaxation nous procure une énergie de 4211601 au bout de $6mn29s$.

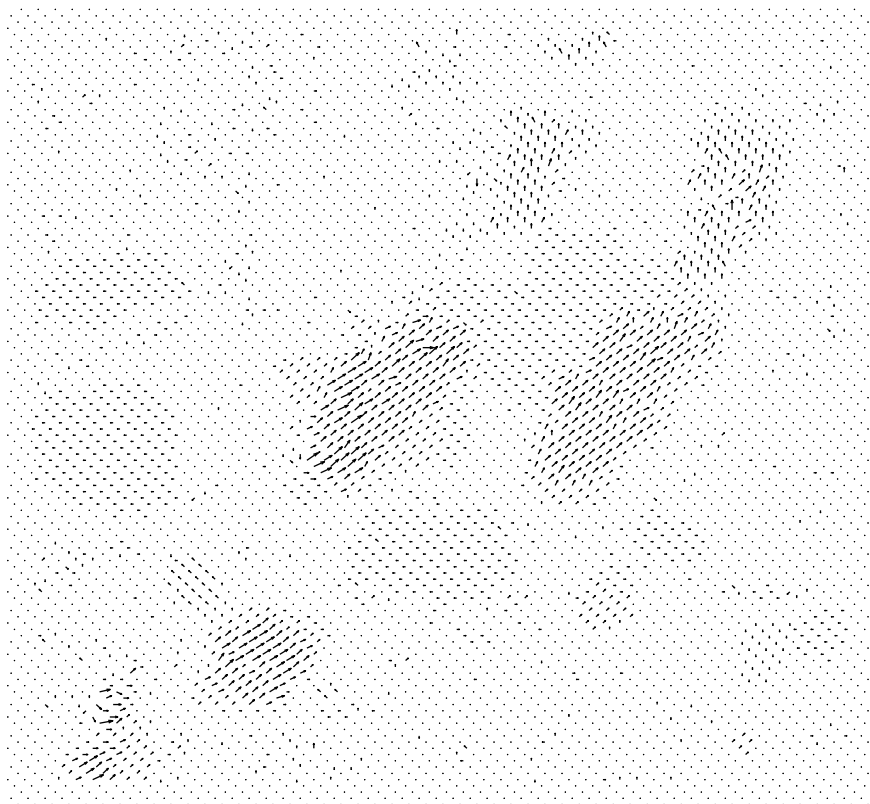


FIG. 3.20 – Champ de vecteurs obtenu par résolution multiéchelle.

Au contraire de la détection de mouvement, les résultats obtenus, c'est-à-dire en fait les énergies finales, sont assez éloignés. Comme nous l'avons dit précédemment, cela est dû au caractère non linéaire de la fonction d'énergie. En effet, la résolution par ICM s'achève sur une énergie relativement élevée par rapport aux autres relaxations. D'ailleurs, le fait que l'on se trouve sur un minimum local est très bien confirmé par l'allure du champ de déplacement trouvé (*fig.* 3.15). Le mouvement des véhicules n'est pas clairement défini, les vecteurs vitesses ne s'orientent pas dans une direction suffisamment générale pour que l'on puisse évaluer le déplacement. Certes, on peut situer les objets mobiles, mais cela concerne la détection de mouvement que nous avons réalisée auparavant. La relaxation déterministe à pleine résolution est donc loin d'être recommandée pour le calcul du flux optique.

En ce qui concerne la relaxation multirésolution, si le résultat semble légèrement meilleur, il n'est pas encore convaincant. Le fait que l'on travaille à des résolutions moindres n'arrange pas spécialement le problème. Assurément le temps de calcul est plus faible par rapport à un ICM à pleine résolution, mais les champs estimés aux niveaux les plus grossiers, procurent de mauvaises initialisations pour les paliers inférieurs. Le résultat d'un niveau de résolution inférieure apporte des connaissances sur la nature des étiquettes, mais elles ne sont pas suffisamment de bonne qualité pour que l'algorithme de multirésolution ne reste pas bloquer, au final, sur un minimum local.

Par contre, il faut bien reconnaître que l'algorithme multiéchelle fournit une bonne estimation du champ de déplacement. En effet, l'énergie finale est très proche de l'énergie de référence donnée par le recuit simulé. Surtout, on obtient un excellent résultat en peu de temps. Alors que le recuit simulé prend quasiment cinq jours, au bout de seulement quelques minutes, la relaxation multiéchelle rend une estimation du flux optique remarquable. Il semble donc assez clair que toute analyse rapide du flux optique devrait empreinter le raisonnement développé par la résolution multiéchelle.

Chapitre 4

Optimisation robuste et évolutionnaire

Nous venons de voir quatre algorithmes de relaxation appliqués à la détection de mouvement et à l'estimation du flux optique. Après le recuit simulé, qui est excessivement lent, la résolution multiéchelle s'impose comme être la méthode la plus efficace en ce qui concerne le calcul du champ de déplacement entre deux images successives. Les résultats trouvés par cette technique, vont donc être utilisés dans d'autres procédures de recherche de minimum plus complexes, et par la même plus efficaces.

4.1 Estimation dense par techniques robustes

En 1998, [5] introduisait une méthode multigrille déterministe efficace pour la résolution du flux optique. Ce procédé combine l'analyse multirésolution et la relaxation multiéchelle, que nous venons toutes les deux de décrire dans le chapitre précédent.

4.1.1 Raffinement itératif

Dans la définition de la fonction d'énergie pour le calcul des vecteurs de déplacement, nous avons vu que si on suppose que la luminance d'un point physique ne varie pas trop entre deux images consécutives¹ alors l'énergie est de la forme :

$$U(x, f) = \sum_{s \in S} \left(f(s + x_s, t + 1) - f(s, t) \right)^2 + \beta \sum_{\{s, r\} \in \mathcal{C}} \|x_s - x_r\|^2.$$

¹ $f_k \equiv f(\cdot, t)$.

Comme nous allons nous servir des résultats de la relaxation multiéchelle, nous connaissons déjà l'allure du champ de déplacement. Nous allons plutôt chercher à le raffiner, c'est-à-dire, que nous allons déterminer en chaque site s , la correction dx_s de chacune des étiquettes.

Grâce à un développement de Taylor au premier ordre, nous obtenons :

$$U(x + dx, f) \approx \sum_{s \in S} \left(\nabla f(s + x_s, t + 1)^T dx_s + f_t(s, x_s) \right)^2 + \beta \sum_{\{s, r\} \in \mathcal{C}} \|(x_s + dx_s) - (x_r + dx_r)\|^2,$$

avec $\nabla f = [f_x \ f_y]^T$ qui est le gradient spatial de la fonction de luminance, et $f_t(s, x_s) = f(s + x_s, t + 1) - f(s, t)$ qui est la différence inter-image déplacée.

Le problème, est que l'hypothèse sous-jacente que la luminance est quasiment constante d'une image à l'autre pour un point donnée est souvent fausse. En effet, cela ignore les cas d'occlusions, de transparence ou encore de réflexion spéculaire... De plus, le caractère quadratique de la fonction a plutôt tendance à ne pas saisir les discontinuités du champ de déplacement mais au contraire à le lisser et l'uniformiser.

Pour lutter contre ces effets, [5] utilise des estimateurs robustes ρ_1 et ρ_2 à la place des termes quadratiques. Cela nous donne donc une nouvelle fonction à minimiser $H = H_1 + \beta H_2$, où :

$$H_1(dx, f, x) = \sum_{s \in S} \rho_1 \left(\nabla f(s + x_s, t + 1)^T dx_s + f_t(s, x_s) \right), \quad (4.1)$$

$$H_2(dx, x) = \sum_{\{s, r\} \in \mathcal{C}} \rho_2 \left(\|(x_s + dx_s) - (x_r + dx_r)\| \right). \quad (4.2)$$

L'estimation est déterminée par l'approche multirésolution que nous avons vu auparavant. Nous construisons la pyramide des observations de la même façon. De plus, nous avons vu que pour chaque niveau de résolution, l'énergie est tout simplement calquée sur l'énergie définie à pleine résolution. De (4.1) et (4.2), nous avons donc pour chaque niveau k de la pyramide, avec les notations du chapitre précédent :

$$H_1^k(dx^k, f^k, x^k) = \sum_{s \in S^k} \rho_1 \left(\nabla f^k(s + x_s^k, t + 1)^T dx_s^k + f_t^k(s, x_s^k) \right),$$

$$H_2^k(dx^k, x^k) = \sum_{\{s,r\} \in \mathcal{C}^k} \rho_2 \left(\| (x_s^k + dx_s^k) - (x_r^k + dx_r^k) \| \right).$$

A chaque étage, nous déterminons le champ de raffinement, puis nous le projetons par duplication. Le résultat projeté constitue le champ de déplacement au niveau inférieur que nous allons raffiner. Sur une pyramide à trois étages par exemple, cela donne² :

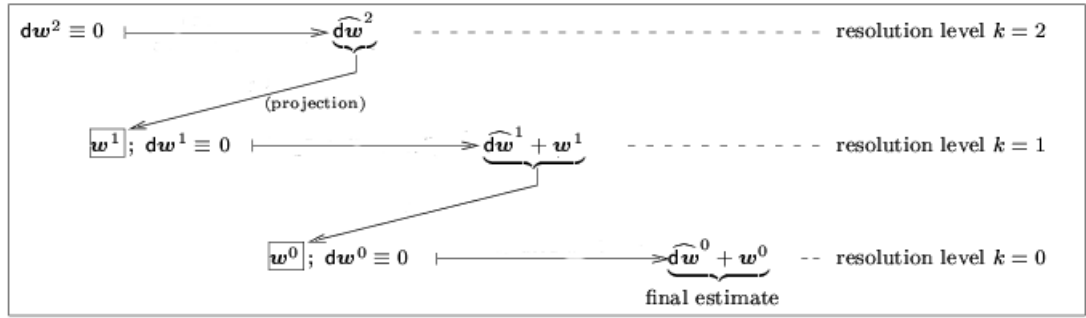


FIG. 4.1 – Principe de l'estimation par raffinement du flux optique.

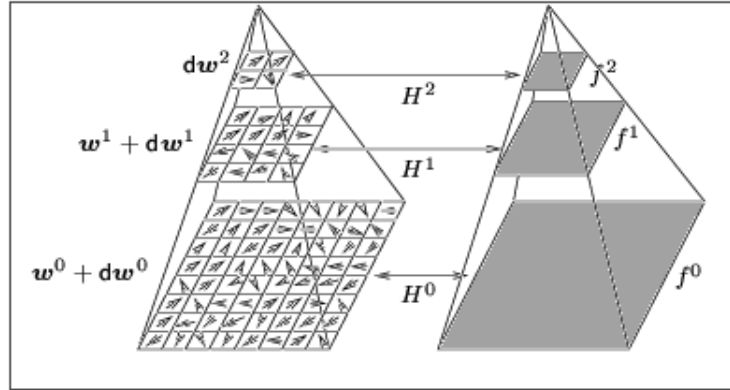


FIG. 4.2 – Modèle multirésolution pour le raffinement.

Il reste à présent à voir comment on détermine à chaque niveau le champ de raffinement.

² $x \equiv w$.