

Localisation d'objets mobiles dans des scènes naturelles filmées par une caméra fixe

Moving Objects Location in Complex Scenes Filmed by a Fixed Camera



Maciej ORKISZ

Laboratoire de traitement d'Images
et Reconnaissance des Formes,
Institut National Polytechnique
de Grenoble,
46 avenue Félix Viallet,
38031 Grenoble Cedex

Né en Pologne en 1961. Titulaire d'un diplôme d'ingénieur électronicien obtenu en 1985 à la Polytechnique de Varsovie, d'un DEA en automatique et traitement du signal obtenu en 1986 à l'INPG et d'un doctorat de l'INPG obtenu en 1990, thèse intitulée : « Éléments du traitement d'images et analyse du mouvement humain ». Enseignant à l'ENS d'Électronique et de Radioélectricité de Grenoble. S'intéresse à l'analyse de scènes dynamiques par image.

RÉSUMÉ

Le présent article porte sur l'analyse de scènes dynamiques filmées par une caméra immobile. Notre but est de localiser les objets en mouvement, rapidement et sans connaissance préalable de la scène et du caractère du mouvement. Les méthodes connues qui répondent le mieux à ces critères comparent les images successives afin d'exploiter les frontières variant entre images voisines. Cependant, chacune de ces méthodes interprète comme mobiles certaines frontières statiques qui ne sont pas visibles dans toutes les images comparées. Or, l'information fournie par trois images consécutives suffit pour distinguer ces frontières statiques des véritables frontières mobiles.

Nous proposons un opérateur original qui localise les frontières mobiles dans l'image courante par comparaison du gradient de cette image aux gradients des images précédente et suivante. Notre méthode permet à la fois de mieux éliminer les frontières statiques et de conserver un plus grand nombre de frontières pertinentes.

MOTS CLÉS

Analyse de séquences d'images, détection du mouvement, temps réel vidéo.

SUMMARY

In this paper, we are dealing with the analysis of dynamic scenes filmed by a fixed camera. The purpose of this analysis is a quick locating of the moving objects, without prior knowledge of the scene and of the nature of the movement. The existing methods, which best respond to these criteria, compare successive images in order to exploit edge position changes. However, each of these methods misinterprets some of the static edges which are not visible in all the compared images. As a matter of fact, the information provided by three consecutive images is sufficient to tell these static edges from the real moving edges.

We are proposing, herein, an original operator which locates the moving edges in the current image by comparing the gradient of this image to the gradients of the previous image and of the next one. Our method both improves the elimination of the static edges and makes it possible to extract a higher number of pertinent edges.

KEY WORDS

Image sequence analysis, motion detection, video real time.

Introduction

Le présent article est consacré à la localisation d'objets en mouvement dans des scènes filmées par une caméra fixe. Cette étude peut trouver de nombreuses applications telles que le contrôle du trafic routier [11], [20], [22], [23], l'analyse du mouvement humain [14], [17], la détection d'intrus, la surveillance d'une zone de sécurité (autour d'un robot par exemple), la compression d'images pour visio-phonie, etc.

A. CRITÈRES D'ÉVALUATION DES MÉTHODES

Notre travail, effectué cependant hors du cadre d'une application particulière, a été guidé essentiellement par deux critères :

• Aspect temps réel vidéo

Le traitement en temps différé ne convient pas dans bien des applications (ex. : télésurveillance, contrôle du trafic routier). De plus, dans le cas de longues séquences d'images, il nécessite l'utilisation du magnétoscope, ce qui pose des problèmes techniques difficiles à résoudre (avancement pas à pas, acquisition à l'arrêt sur image, dégradation de la qualité d'image, etc.). Afin d'éviter ces problèmes, nous cherchons un algorithme qui soit suffisamment simple et rapide pour être implantable sous forme câblée fonctionnant en temps réel vidéo.

• Absence de contraintes quant au caractère du mouvement et à la connaissance préalable de la scène

Pour permettre un large éventail d'applications, nous ne faisons pas de restrictions relatives au type d'objets, à leurs tailles, vitesses, trajectoires ou niveaux de gris.

B. TECHNIQUES DE LA VISION BAS NIVEAU

Par ailleurs, nous nous sommes résolument concentrés sur des traitements de bas niveau, en faisant abstraction de la mise en correspondance de haut niveau. En effet, les deux approches sont sensiblement différentes : la première exploite les changements temporels dans les images pour *éliminer les parties statiques avant toute interprétation*, tandis que la seconde *interprète tous les éléments des images* au cours d'une segmentation statique visant à extraire les primitives à mettre en correspondance [2]. Notre choix a été dicté par le fait que les images présentent généralement une forte proportion d'informations redondantes. L'élimination d'une partie de ces informations, afin d'interpréter uniquement l'information pertinente, permet généralement un gain de temps de calcul.

Nous nous intéressons donc à des traitements de bas niveau. Par « bas niveau » nous entendons le niveau où l'on exploite uniquement l'information locale fournie par chaque pixel et éventuellement par ses voisins les plus proches. Les méthodes évoquées dans les pages qui suivent utilisent soit directement les intensités lumineuses, soit les gradients, soit encore les contours binaires.

• Niveaux de gris

Lorsque l'on utilise directement les intensités lumineuses, il est difficile de proposer des opérations pixel à pixel donnant des résultats significatifs. On exploite plutôt les propriétés statistiques, donc plus globales. Parmi les opérations arithmétiques, seule la soustraction a un sens : le résultat est nul pour les parties invariantes de l'image. Mais l'interprétation correcte des parties non nulles demande généralement une connaissance préalable des intensités lumineuses relatives du fond et des objets en mouvement.

• Gradients

Puisque toutes les régions uniformes donnent des gradients nuls, la connaissance a priori des niveaux de gris relatifs n'est pas nécessaire. Différents éléments de la scène sont représentés par des zones de contraste où les valeurs du gradient sont relativement élevées. Pour ces zones, on peut raisonner en termes de « présence » et « absence », ce qui permet d'envisager des opérations à caractère « logique » même si les opérandes ne sont pas binaires (ex. : la multiplication peut jouer le rôle de « et »). Il faut cependant préciser que, dans ce cas, la dynamique des valeurs pose parfois des problèmes d'interprétation.

• Contours binaires

Les valeurs binaires ont l'avantage de refléter naturellement les notions de présence (« 1 ») et absence (« 0 ») des éléments auxquels elles correspondent et de se prêter à des opérations logiques simples. Leur inconvénient majeur est lié à la perte de l'information au cours de la binarisation. En particulier, on ne peut pas juger si un contour décalé d'un pixel entre deux images comparées l'est à cause d'un mouvement ou à cause du bruit. Or, la détection des contours passe toujours par une opération de dérivation qui amplifie le bruit.

C. CATÉGORIES DE MÉTHODES DE LOCALISATION DES OBJETS MOBILES

Deux catégories de méthodes sont utilisées pour localiser les objets en mouvement dans une image :

— soit l'image courante est comparée à un modèle statique de la scène, appelé image de référence, et tout élément présent dans l'image courante et absent dans l'image de référence est considéré comme mobile,

— soit l'image courante est comparée à ses voisines et il s'agit de sélectionner parmi les éléments variant d'une image à l'autre ceux qui correspondent à la position actuelle de l'objet en mouvement.

• Méthodes de construction d'une image de référence

Les méthodes appartenant à la première catégorie diffèrent essentiellement par la façon de construire le modèle statique de la scène à partir de la séquence d'images étudiée. Il est bien entendu souhaitable de pouvoir acquérir l'image de référence tout simplement en filmant la scène exempte de tout objet mobile. Cela n'est cependant pas toujours possible et, même si l'on y parvient, tout changement de la lumière ambiante (intensité, contrastes, ombres...) survenu après cette prise de vue peut rendre

l'image de référence incompatible avec les images à analyser.

Pour construire un modèle statique de la scène, deux approches sont envisageables. La première consiste à accumuler l'information contenue dans la séquence d'images, mettant ainsi progressivement en évidence les parties immobiles de la scène, et effaçant ce qui bouge [15], [17]. Un bon effacement demande cependant le traitement d'un grand nombre d'images. Dans la seconde approche, il s'agit de remplacer les objets mobiles dans une image particulière I_i par des éléments du fond cachés dans cette image, mais visibles dans une autre [11], [12], [17], [21]. Pour ce faire, on analyse l'évolution de la différence entre l'image I_i et les images consécutives. Il est toutefois difficile de fixer un seuil permettant de distinguer les différences dues au mouvement, de celles qui sont provoquées par le bruit ou par des fluctuations d'éclairage.

Les méthodes visant à construire une image de référence ne répondent pas aux critères évoqués ci-dessus. En effet, elles imposent des contraintes quant à la scène filmée : les déplacements des objets mobiles doivent être plus grands que leur taille, leurs trajectoires ne peuvent pas se croiser lors de la phase de construction, de même qu'aucun objet ne peut revenir sur ses pas. En outre, la construction d'une image de référence entraîne un retard incompatible avec l'objectif temps réel. C'est pourquoi les techniques citées ci-dessus présentent plus d'intérêt pour les applications où il s'agit d'étudier les parties statiques de la scène, que pour la détection d'objets en mouvement.

• Analyse d'éléments variant entre images voisines

Les atouts des méthodes qui exploitent les éléments variant entre images voisines peuvent être exprimés par opposition aux inconvénients de la construction d'une image de référence. D'une part, n'ayant pas besoin de construire l'image de référence, on évite à la fois le retard dû à la construction de cette image et les contraintes relatives aux trajectoires des objets mobiles pendant la phase de construction. D'autre part, ces méthodes sont moins sensibles aux variations d'éclairage. En effet, d'éventuelles variations d'éclairage sont généralement assez lentes par rapport à la fréquence d'échantillonnage. Par conséquent, les changements de caractéristiques lumineuses de la scène entre images voisines sont faibles.

De nombreuses méthodes fondées sur la comparaison d'images voisines sont connues dans le domaine d'analyse de scènes filmées par une caméra mobile. On pourrait s'en inspirer, en considérant le cas d'une caméra fixe comme un cas particulier. Cependant, dans une séquence d'images obtenue à l'aide d'une caméra mobile, « tout bouge ». Par conséquent, pour distinguer les éléments mobiles des éléments statiques de la scène, il faut évaluer les vitesses relatives de tous les objets, afin d'en déduire la composante commune due au mouvement de la caméra. Pour ce faire, les auteurs proposent généralement des solutions dérivées du calcul de flot optique [4], [5], [9]. Ces solutions ne répondent pas à nos critères, car elles demandent des calculs longs et, pour la plupart, s'appuient sur l'hypothèse de petits déplacements, ce qui limite l'éventail des mouvements qui peuvent être étudiés. Ce dernier problème peut

éventuellement être contourné grâce à l'introduction de la multirésolution [1]. Dans le cas d'une caméra fixe, l'analyse est a priori moins complexe et l'on peut envisager des méthodes spécifiques, plus simples et plus rapides que le calcul du flot optique.

Pour les mêmes raisons, nous avons dû rejeter l'approche probabiliste [13] qui conduit cependant à des méthodes robustes, permettant une interprétation appropriée même en présence d'une observation bruitée. Malheureusement, la technique de relaxation, utilisée pour attribuer à l'image un champ d'étiquettes, nécessite un temps de calcul très long. De plus, les modèles d'étiquetage existants sont fondés sur certaines hypothèses restrictives, par exemple : le déplacement de l'objet mobile entre images voisines doit être supérieur à sa taille ou, dans le cas contraire, cet objet ne doit pas être uniforme.

L'approche qui répond le mieux aux critères de rapidité et d'absence de contraintes relatives à la scène filmée, consiste à exploiter les frontières variant entre images voisines. Dans les pages qui suivent, nous nous concentrons sur cette catégorie de méthodes. La première partie de notre article est consacrée à une étude critique des opérateurs connus, tandis que la deuxième partie présente un opérateur nouveau. Nous proposons de localiser les frontières des objets en mouvement dans l'image courante par comparaison du gradient de cette image avec les gradients des images précédente et suivante. La présentation de notre méthode est suivie par une discussion de ses performances et de ses limites.

1. Techniques de détection de frontières mobiles

Un premier tri des techniques de localisation d'objets en mouvement dans des scènes filmées par une caméra fixe — tri effectué selon les critères de rapidité d'une part, et d'absence de contraintes quant au caractère du mouvement étudié et à la connaissance préalable de la scène, d'autre part — a focalisé notre recherche sur les méthodes d'extraction de frontières mobiles par comparaison d'images voisines. Nous présentons ici deux méthodes-phares dans cette catégorie.

1.1. POSITION DU PROBLÈME

Nous souhaitons localiser les frontières mobiles dans l'image courante. Sachant que dans une séquence d'images obtenue à l'aide d'une caméra fixe, les éléments statiques de la scène apparaissent d'une image à l'autre aux mêmes endroits, on peut éliminer ces éléments par une simple différence d'images pixel à pixel. Les éléments qui restent après cette opération, correspondent aux positions consécutives des objets en mouvement. En faisant coïncider les frontières de la différence d'images avec les frontières de l'image courante, on peut sélectionner les frontières correspondant à la position courante des objets en mouvement (fig. 1).

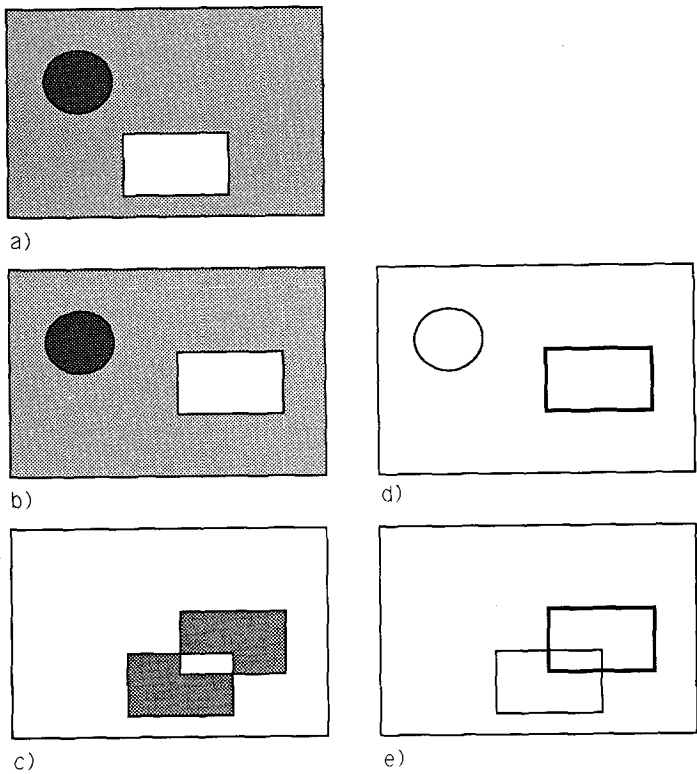


Figure 1. — Principe de la coïncidence de frontières :

- a) image précédente,
 - b) image courante,
 - c) différence absolue d'images,
 - d) frontières de l'image courante,
 - e) frontières de la différence d'images :
- les frontières coïncidentes (traits épais) correspondent à la position courante de l'objet en mouvement.

La coïncidence de frontières peut être déterminée de plusieurs manières. Leung et Yang [14] proposent de la réaliser tout simplement par l'opération « et » logique. Pour ce faire, il faut d'abord extraire les contours binaires de l'image courante ainsi que ceux de la différence absolue entre l'image courante et l'image précédente. Cependant, l'extraction des contours de la différence d'images est, dans cette méthode, précédée par un seuillage. Par conséquent, les positions des contours obtenus dépendent de la valeur du seuil et l'on court le risque de ne trouver de coïncidence que pour une partie (parfois faible) des contours pertinents. D'une manière générale, l'utilisation de contours binaires, quelle que soit la méthode de leur extraction, présente le défaut évoqué dans l'introduction, à savoir : il est impossible de distinguer, au cours d'un traitement pixel par pixel, les fragments de contours décalés à cause d'un mouvement de ceux qui le sont à cause du bruit. Ainsi, on perdra toujours une partie des contours pertinents, bien qu'ils soient présents aussi bien dans l'image courante que dans la différence d'images.

C'est pourquoi Haynes et Jain [10] affirment qu'il est préférable de repousser à la fin du traitement les opérations fortement non linéaires, seuillage et affinement, liées à l'extraction des contours. Autrement dit, il faut déterminer

la coïncidence avant toute binarisation. Pour ce faire, les auteurs proposent de multiplier la différence absolue d'images par le gradient de l'image courante obtenue à l'aide de l'opérateur de Sobel. Leur méthode s'exprime donc par la formule suivante :

$$(1) \quad \forall_{x,y} M_n(x,y) = |I_n(x,y) - I_{n-1}(x,y)| \cdot G_n(x,y)$$

où $I_n(x,y)$ représente le niveau de gris du pixel (x,y) dans l'image n et $G_n(x,y)$ est le gradient de cette image obtenu pour le pixel (x,y) à l'aide de l'opérateur de Sobel. Notons que le gradient de l'image courante représente les frontières de cette image. La différence d'images en revanche, est utilisée directement, sans que l'on détermine ses frontières. La formule (1) ne réalise donc pas véritablement de coïncidence de frontières. Néanmoins, l'approche de Haynes et Jain a inspiré d'autres chercheurs qui ont essayé d'améliorer la formule (1) tout en conservant le principe de la coïncidence non binaire.

1.2. MÉTHODE DE STELMASZYK

Tout comme les frontières de l'image courante, celles de la différence d'images peuvent également être représentées par des gradients. Partant de cette constatation et gardant la multiplication comme opérateur de coïncidence, Stelmasyk *et col.* [20] ont obtenu la formule suivante :

$$(2) \quad \forall_{x,y} M_n(x,y) = G(|I_n(x,y) - I_{n-1}(x,y)|) \cdot G_n(x,y)$$

De toutes les méthodes connues en matière de détection des frontières mobiles à l'aide de deux images consécutives [10], [14], la méthode de Stelmasyk donne les meilleurs résultats.

Cependant, l'information fournie par deux images seulement ne permet pas de résoudre, sans connaissance préalable de la scène, le problème de désoccultation. En effet, tout objet visible dans l'image courante là où il ne se trouvait pas dans l'image précédente, est interprété comme mobile, même s'il s'agit d'un élément du fond statique qui était caché, et qui vient d'apparaître (*fig. 2*).

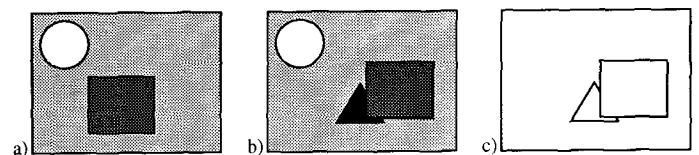


Figure 2. — Problème de désoccultation :

- a) image précédente,
- b) image courante, le triangle noir appartenant au fond statique vient d'apparaître,
- c) éléments interprétés comme mobiles.

1.3. MÉTHODE DE VIEREN

Un élément statique désocculté dans l'image courante reste généralement visible dans l'image suivante. Une comparaison de l'image courante à l'image suivante permet

donc de distinguer un tel élément des éléments mobiles qui, eux, varient aussi bien entre les images précédente et courante qu'entre les images courante et suivante. Vieren [22], [23] propose d'extraire d'abord les éléments variant dans chacun de ces couples d'images. Pour ce faire, il calcule deux différences absolues d'images. Ensuite, il fait coïncider les frontières de ces différences d'images, représentées par leurs gradients, afin de sélectionner les frontières mobiles correspondant à l'image courante. Tout comme dans les méthodes précédentes, cette opération est réalisée par la multiplication :

$$(3) \quad \forall_{x,y} M_n(x,y) = G(|I_n(x,y) - I_{n-1}(x,y)|) \cdot G(|I_n(x,y) - I_{n+1}(x,y)|).$$

Notons que le terme $G(|I_n(x,y) - I_{n+1}(x,y)|)$ peut être réutilisé lors de la détection des frontières mobiles dans l'image $n+1$, ce qui rend cette méthode plus rapide que celle de Stelmazyk. En effet, cette dernière demande pour chaque nouvelle image tous les calculs : différence absolue d'images, gradient de cette différence et gradient de l'image courante. Or, la méthode de Vieren nécessite seulement le calcul de la nouvelle différence absolue d'images et de son gradient.

Néanmoins, la formule (3) n'est pas exempte de défauts. Bien qu'elle permette d'éliminer les frontières désaccrétées, les résultats obtenus peuvent comporter des éléments parasites : les frontières statiques absentes dans l'image courante, mais visibles dans ses deux voisines. Pour s'en convaincre, il suffit d'effectuer un raisonnement mettant en œuvre des contours binaires. Les contours d'une différence absolue d'images correspondent à un OU EXCLUSIF entre les contours de ces images. La multiplication jouant le rôle de l'opérateur logique ET, la formule (3) peut être réécrite sous la forme binaire suivante :

$$(3') \quad \forall_{x,y} M_n(x,y) = (E_n(x,y) \oplus E_{n-1}(x,y)) \cdot (E_n(x,y) \oplus E_{n+1}(x,y))$$

$$= (E_n(x,y) \oplus E_{n+1}(x,y)) \cdot (E_n(x,y) \oplus E_{n-1}(x,y))$$

$$= (E_n(x,y) \cdot \overline{E_{n+1}(x,y)} + \overline{E_n(x,y)} \cdot E_{n+1}(x,y)) \cdot (E_n(x,y) \cdot \overline{E_{n-1}(x,y)} + \overline{E_n(x,y)} \cdot E_{n-1}(x,y))$$

où $E_n(x,y)$ signifie que le pixel (x,y) appartient à un contour dans l'image n , tandis que $\overline{E_n(x,y)}$ signifie que le pixel (x,y) n'appartient à aucun contour dans cette image. L'élimination des termes nuls permet d'obtenir une formule facile à interpréter :

$$(3'') \quad \forall_{x,y} M_n(x,y) = E_n(x,y) \cdot \overline{E_{n-1}(x,y)} \cdot \overline{E_{n+1}(x,y)} + \overline{E_n(x,y)} \cdot E_{n-1}(x,y) \cdot E_{n+1}(x,y).$$

Le premier terme correspond bien au résultat attendu : contours présents dans l'image courante et absents dans les deux images voisines, tandis que le second terme repré-

sente les contours indésirables, absents dans l'image courante mais présents dans ses deux voisines. Ces éléments ressortent à l'intérieur de l'enveloppe formée par les véritables frontières mobiles (fig. 3).

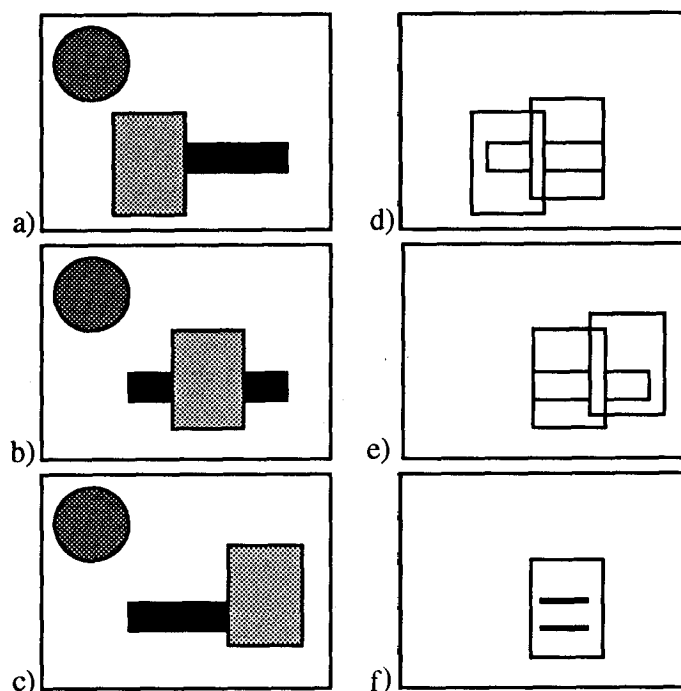


Figure 3. — Frontières communes à deux différences d'images :
 a) image précédente I_{n-1} ,
 b) image courante I_n ,
 c) image suivante I_{n+1} ,
 d) frontières de la différence $D_n = |I_n - I_{n-1}|$,
 e) frontières de la différence $D_{n+1} = |I_{n+1} - I_n|$,
 f) résultat = frontières mobiles (traits fins) + frontières parasites (traits épais) correspondant aux frontières statistiques absentes dans I_n mais visibles dans I_{n-1} et I_{n+1} .

En outre, l'utilisation d'une troisième image introduit inévitablement un retard : le résultat pour l'image courante ne peut être obtenu avant l'acquisition et le traitement de l'image suivante.

1.4. CRITIQUE

La technique de détection de frontières mobiles par comparaison d'images voisines est peu contraignante et rapide, elle répond donc aux critères évoqués dans l'introduction. La seule restriction à respecter concerne le déplacement minimum de chaque objet en mouvement entre images voisines. Lorsque ce déplacement est inférieur à l'épaisseur des frontières mobiles, la réponse du détecteur est atténuée (fig. 4). Il existe donc une vitesse minimum au-dessous de laquelle cette atténuation devient trop forte et le mouvement ne peut plus être détecté. Cependant, une telle limite existe pour toute famille de méthodes et la seule solution consiste alors à sous-échantillonner la séquence d'images selon l'axe de temps.

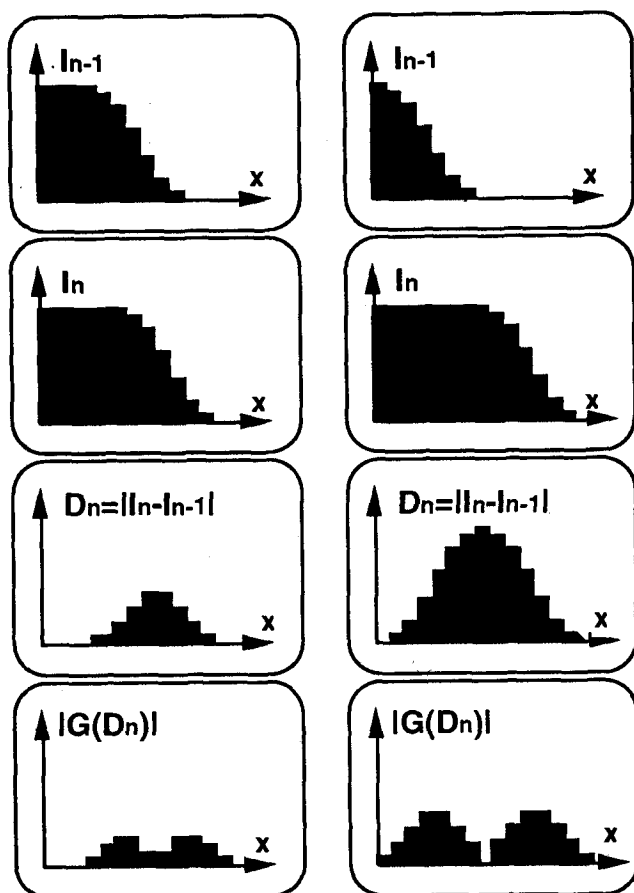


Figure 4. — Déplacement d'un objet comparé à l'épaisseur de sa frontière — exemple unidimensionnel : à gauche : déplacement inférieur à l'épaisseur de la frontière — la différence absolue et son gradient sont atténués ; à droite : déplacement supérieur à l'épaisseur de la frontière — la réponse du détecteur est plus forte.

La nécessité d'extraire les frontières des images n'est pas vraiment une contrainte supplémentaire, car cette extraction est souvent le premier pas vers la segmentation d'images, elle est alors effectuée de toute façon, que ce soit en vue de la détection du mouvement ou non. D'ailleurs, depuis plusieurs années existent des processeurs câblés capables d'effectuer cette opération en temps réel vidéo [7], [18], tout comme la plupart des opérateurs de détection de frontières mobiles sont eux-mêmes réalisables sous forme câblée. Vieren, par exemple, a implanté son opérateur sur un processeur fonctionnant à la cadence de 10 images par seconde [22].

Cependant, nous avons déjà remarqué que chacune des méthodes connues, dans la catégorie étudiée, fait ressortir certaines frontières statiques : soit désoccultées (méthode de Stelmaszyk et toutes les méthodes utilisant deux images seulement), soit visibles dans les images précédente et suivante mais cachées dans l'image courante (méthode de Vieren).

Notons également que l'opération de multiplication, utilisée pour déterminer la coïncidence de frontières, modifie très sensiblement la dynamique. Le produit de deux valeurs

élevées donnant une valeur très élevée, les contrastes forts entre les objets mobiles et le fond sont favorisés. En revanche, le produit de deux valeurs faibles donnant un résultat très faible, les frontières mobiles peu marquées sont encore atténuées. Ce dernier résultat peut être inférieur à celui que l'on obtient en multipliant le gradient correspondant à une frontière à éliminer, par le bruit du gradient de la différence d'images.

2. Méthode proposée

Guidés par les critères de rapidité et d'absence de contraintes relatives à la scène filmée, nous avons choisi d'exploiter les frontières variant entre images voisines. Considérant que le retard d'une image vidéo est acceptable dans la grande majorité des applications, nous comparons les frontières de l'image courante non seulement à celles de l'image précédente, mais aussi à celles de l'image suivante. Nous disposons alors de toute l'information nécessaire pour distinguer les véritables éléments mobiles des éléments statiques désoccultés, à l'exception d'une situation rare où ces derniers sont aussitôt recachés dans l'image suivante.

Parmi les méthodes connues représentant cette approche, la plus performante est celle de Vieren (cf. § 1.3). Cependant, deux défauts spécifiques à cette méthode peuvent être évoqués : changement de dynamique dû à la multiplication d'une part, et fausse interprétation des frontières statiques visibles dans les images précédente et suivante, mais cachées dans l'image courante, d'autre part. C'est pourquoi, dans notre recherche, nous nous sommes fixé deux critères supplémentaires :

- remplacer la multiplication par une opération qui ne modifie pas la dynamique et qui soit, si possible, plus rapide ;
- en comparant l'image courante aux images précédente et suivante, localiser uniquement les frontières présentes dans l'image courante et absentes dans ses deux voisines, ce qui correspond à la formule binaire suivante :

$$(4) \quad \forall_{x,y} M_n(x,y) = E_n(x,y) \cdot \bar{E}_{n-1}(x,y) \cdot \bar{E}_{n+1}(x,y)$$

La formule (4) est une formule de principe. Au lieu d'utiliser les contours binaires, nous avons cherché une opération équivalente pour les gradients, afin d'éviter les problèmes déjà évoqués, liés à la binarisation précoce. En outre, dans un souci de rapidité, nous avons favorisé la réutilisation de résultats intermédiaires. Il s'agit de pouvoir réutiliser, au cours de la détection des frontières mobiles dans l'image $n+1$, un maximum de résultats intermédiaires obtenus pendant la détection des frontières mobiles dans l'image n .

Dans les pages qui suivent, nous exposons d'abord le principe de fonctionnement de l'opérateur proposé, puis nous comparons les performances de notre méthode avec celles des méthodes connues.

2.1. PRINCIPE

Soient I_{n-1} , I_n , I_{n+1} trois images consécutives et G_{n-1} , G_n , G_{n+1} leurs gradients respectifs. Les initiales P, C et S désignent les frontières présentes respectivement dans les images précédente, courante et suivante. Les mêmes symboles complétés (\bar{P} , \bar{C} , \bar{S}) désignent les frontières absentes dans les images respectives. Nous considérons que la présence d'une frontière se manifeste par une valeur élevée du gradient alors que l'absence de frontière se traduit par une valeur du gradient proche de zéro. Ainsi les frontières du type C par exemple, sont les zones où $G_n \gg 0$. De même le type \bar{P} correspond aux zones où $G_{n-1} \approx 0$.

En outre les frontières en mouvement peuvent être présentes à un même endroit dans une seule des trois images. Elles appartiennent donc à une des trois catégories : $\bar{P}\bar{C}\bar{S}$, $\bar{P}C\bar{S}$ ou $\bar{P}C\bar{S}$. Les frontières statiques peuvent en revanche apparaître à un même endroit dans plusieurs images et conservent alors une valeur du gradient constante au bruit près. Par exemple, les frontières statiques désocculées (type $\bar{P}C\bar{S}$) se caractérisent par $G_{n-1} \approx 0$ et $G_n \approx G_{n+1} \gg 0$.

Définissons maintenant une image SV_n telle que :

$$(5) \quad \forall_{x,y} SV_n(x,y) = \max(G_{n-1}(x,y), G_n(x,y), G_{n+1}(x,y))$$

et une image ST_n telle que :

$$(6) \quad \forall_{x,y} ST_n(x,y) = \max(G_n(x,y), G_{n-1}(x,y), G_{n+1}(x,y)) - \max(G_{n-1}(x,y), G_{n+1}(x,y))$$

Prendre, pour chaque point, le maximum des gradients déterminés en ce point pour plusieurs images, revient à retenir toutes les frontières présentes dans au moins une des images comparées. Par conséquent, l'image SV_n (Somme des frontières des images Voisines) contient toutes les frontières statiques (types $\bar{P}C\bar{S}$, $\bar{P}C\bar{S}$ et $\bar{P}C\bar{S}$), ainsi que les frontières mobiles dans leurs positions précédente et suivante (types $\bar{P}C\bar{S}$ et $\bar{P}C\bar{S}$), mais ne contient pas de frontières mobiles dans leur position courante (type $\bar{P}C\bar{S}$), car elles sont absentes aussi bien dans G_{n-1} que dans G_{n+1} (fig. 5 g). En revanche, l'image ST_n (Somme des frontières des Trois images) contient toutes les frontières sans exception (fig. 5 h). La seule différence entre les images ST_n et SV_n correspond donc aux frontières mobiles dans leur position courante. On peut les faire ressortir par une simple soustraction (fig. 5 i), ce qui donne la formule suivante :

$$(7) \quad \forall_{x,y} M_n(x,y) = \max(G_n(x,y), G_{n-1}(x,y), G_{n+1}(x,y)) - \max(G_{n-1}(x,y), G_{n+1}(x,y))$$

Ainsi avons-nous abouti à un opérateur qui permet d'extraire les véritables frontières mobiles par comparaison

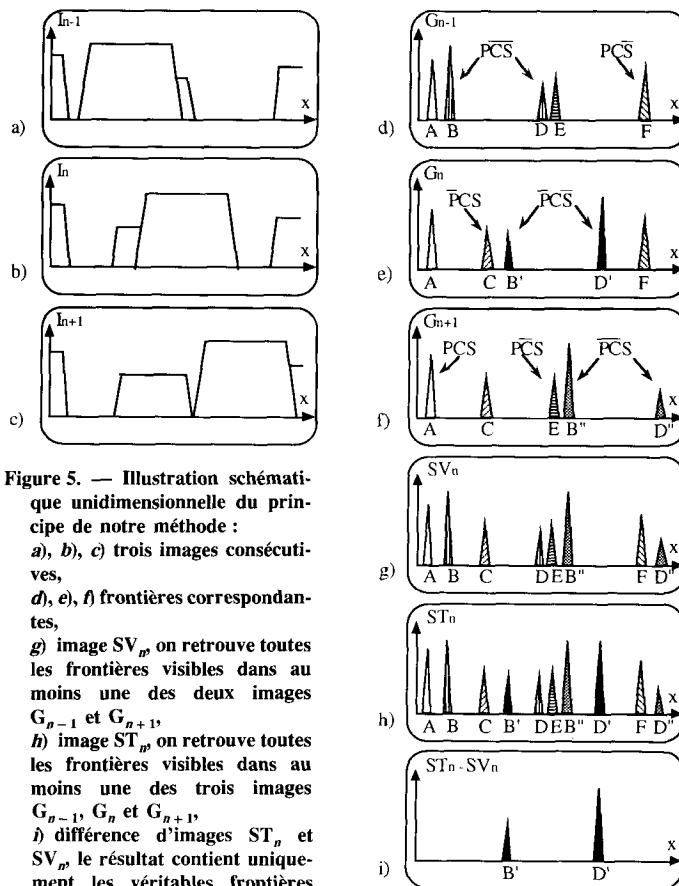


Figure 5. — Illustration schématique unidimensionnelle du principe de notre méthode :
 a), b), c) trois images consécutives,
 d), e), f) frontières correspondantes,
 g) image SV_n on retrouve toutes les frontières visibles dans au moins une des deux images G_{n-1} et G_{n+1} ,
 h) image ST_n on retrouve toutes les frontières visibles dans au moins une des trois images G_{n-1} , G_n et G_{n+1} ,
 i) différence d'images ST_n et SV_n , le résultat contient uniquement les véritables frontières mobiles correspondant à l'image courante.

directe des gradients de trois images consécutives et ceci sans avoir recours à la multiplication.

Remarque : Le même raisonnement permet de proposer un autre opérateur :

$$\forall_{x,y} M_n(x,y) = \max(G_{n-1}(x,y), G_n(x,y)) - G_{n-1}(x,y)$$

pour le cas où le retard d'une image vidéo ne serait pas acceptable. Cet opérateur, comme tout autre utilisant deux images seulement, ne permet pas de résoudre le problème de désoccultation, mais son avantage, par rapport aux opérateurs connus (celui de Stelmazyk en particulier), est de ne pas avoir recours à la multiplication.

2.2. COMPARAISON DE PERFORMANCES AVEC LES METHODES CONNUES

Les performances de ce nouvel opérateur doivent être évaluées par comparaison avec les méthodes existantes appartenant à la même catégorie, c'est-à-dire avec celles qui exploitent les gradients d'images voisines. Afin de comparer les détecteurs de frontières mobiles, sans que les résultats ne soient affectés par des méthodes de calcul du gradient différentes, il faut que ce dernier soit calculé dans tous les cas de la même manière.

Dans l'état actuel de notre recherche, aucune méthode particulière de calcul du gradient n'est favorisée. Chacune des méthodes performantes (Canny [6], Deriche [8], Shen [19]...) consiste à effectuer deux opérations, lissage et calcul de la dérivée, et c'est essentiellement le lissage qui détermine l'épaisseur des frontières ainsi extraites ainsi que leur comportement vis-à-vis du bruit. Souhaitant étudier les propriétés de différents détecteurs de frontières mobiles et non les propriétés de différentes techniques de lissage, nous avons choisi d'utiliser dans notre expérimentation les gradients calculés à l'aide d'un opérateur de dérivation simple, avec un lissage réduit au minimum. Parmi différents opérateurs envisageables (Roberts, Prewitt...), nous avons retenu celui de Sobel qui reste le meilleur dans cette catégorie.

Nous allons confronter ici notre méthode aux méthodes de Stelmaszyk et de Vieren. Les critères évoqués dans les pages précédentes ont favorisé le choix de la famille à laquelle appartiennent ces méthodes. Deux d'entre eux, la rapidité et la capacité à éliminer les frontières statiques tout en conservant les frontières pertinentes, peuvent également servir à établir un classement au sein de cette famille. Ce classement sera suivi d'une comparaison qualitative de la sensibilité au bruit des trois opérateurs.

2.2.1. Capacité à localiser les véritables frontières mobiles

Rappelons que chacun des opérateurs déjà connus fait ressortir des frontières parasites (cf. § 1.2 et 1.3) : désocculées (méthode de Stelmaszyk) ou cachées dans l'image courante mais visibles dans ses deux voisines (méthode de Vieren). Ces dernières sont beaucoup moins fréquentes, car elles apparaissent seulement lorsque le déplacement de l'objet mobile entre la première et la troisième image est plus grand que la dimension de l'objet dans le sens du mouvement. Ainsi, on peut considérer l'opérateur de Vieren comme meilleur que celui de Stelmaszyk. Notre opérateur, quant à lui, élimine ces deux catégories de frontières à la fois.

Rang	Méthode	Frontières statiques non effacées
1	proposée	$\bar{P}\bar{C}\bar{S}$
2	Vieren	$P\bar{C}\bar{S}$ et $\bar{P}C\bar{S}$
3	Stelmaszyk	$\bar{P}C\bar{S}$ et $\bar{P}\bar{C}\bar{S}$

Analysons le comportement de notre opérateur face aux différents types de frontières :

• Frontières $\bar{P}\bar{C}\bar{S}$

Pour les frontières du type $\bar{P}\bar{C}\bar{S}$, $G_n \ll \max(G_{n-1}, G_{n+1})$, donc :

$$\max(G_{n-1}, G_n, G_{n+1}) = \max(G_{n-1}, G_{n+1}).$$

Par conséquent, le résultat fourni par notre opérateur est ici obligatoirement égal à zéro. Les frontières indésirables sont donc éliminées (fig. 6). Notons au passage que l'inégalité $G_n \ll \max(G_{n-1}, G_{n+1})$ est également vérifiée pour les

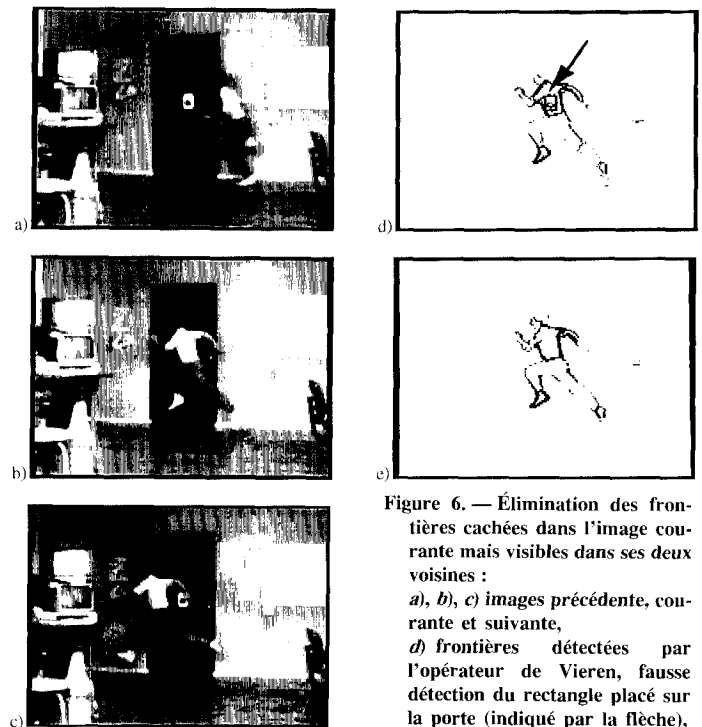


Figure 6. — Élimination des frontières cachées dans l'image courante mais visibles dans ses deux voisines :
 a), b), c) images précédente, courante et suivante,
 d) frontières détectées par l'opérateur de Vieren, fausse détection du rectangle placé sur la porte (indiqué par la flèche),
 e) frontières détectées par la différence de maxima, absence de fausse détection.

frontières mobiles dans leurs positions précédente et suivante. Ainsi, le résultat à ces endroits est aussi égal à zéro.

Remarque : La figure 6 mérite un commentaire. Rappelons que les résultats obtenus avec notre opérateur ont pratiquement la même dynamique que les gradients de départ. En revanche, les méthodes de Vieren et de Stelmaszyk consistent à calculer un produit de gradients, ce qui modifie sensiblement la dynamique. Il est difficile de comparer des frontières extraites à l'aide d'opérateurs donnant des dynamiques très différentes. Nous avons utilisé ici la technique la plus courante — le seuillage — mais l'application d'un même seuil pour les résultats fournis par des opérateurs différents n'a pas de sens. Nous avons donc choisi d'appliquer à chaque résultat un seuil différent, mais qui respecte la règle suivante : afin de visualiser le plus possible de frontières pertinentes, la valeur du seuil est fixée comme la plus petite de celles pour lesquelles le résultat du seuillage ne contient pas de bruit. Ces seuils ont été fixés manuellement.

• Frontières $P\bar{C}\bar{S}$, $\bar{P}C\bar{S}$ et $P\bar{C}\bar{S}$

Pour les frontières statiques visibles dans l'image courante, on devrait théoriquement avoir : $G_n = \max(G_{n-1}, G_{n+1})$. En pratique, la présence du bruit fait que deux cas de figure sont possibles :

$G_n \ll \max(G_{n-1}, G_{n+1})$, notre opérateur donne alors un résultat égal à zéro ;

$G_n > \max(G_{n-1}, G_{n+1})$, le résultat est alors non nul, mais il reste du même ordre de grandeur que le bruit du gradient d'image.

Ce mécanisme permet d'éliminer aussi bien les frontières statiques visibles dans les trois images (type PCS) que les frontières statiques cachées dans G_{n-1} (type $\bar{P}CS$) ou dans G_{n+1} (type PCS), donc en particulier les frontières désoccultées (fig. 7). Seules les frontières désoccultées et recouvertes aussitôt dans l'image suivante (type $\bar{P}CS$) ne peuvent pas être éliminées, mais il en est de même pour tout opérateur utilisant seulement les images voisines.

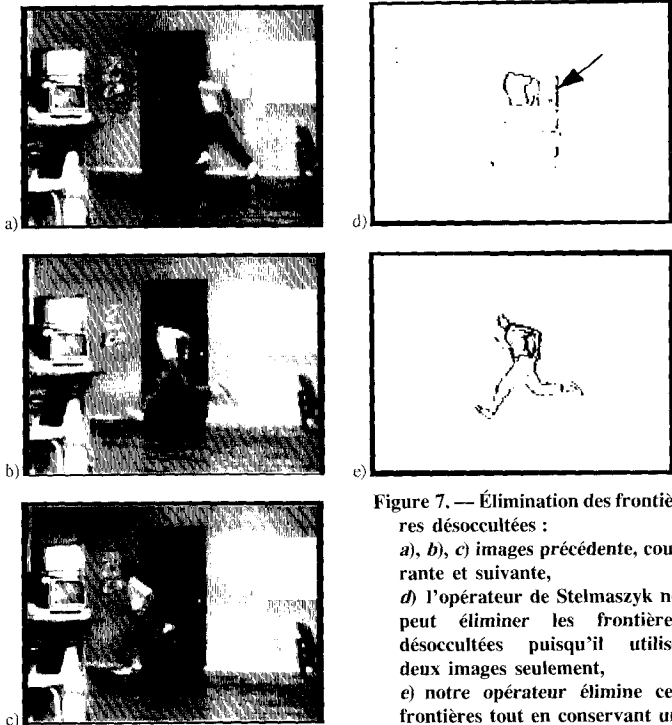


Figure 7. — Élimination des frontières désoccultées :
 a), b), c) images précédente, courante et suivante,
 d) l'opérateur de Stelmaszyk ne peut éliminer les frontières désoccultées puisqu'il utilise deux images seulement,
 e) notre opérateur élimine ces frontières tout en conservant un plus grand nombre de frontières pertinentes.

2.2.2. Rapidité

Avant cette comparaison, remarquons que la réalisation pratique de notre opérateur — aussi bien par logiciel que par une éventuelle structure pipe-line — demande la décomposition suivante :

$$(8) \quad S = \max (G_{n-1}, G_{n+1})$$

$$\text{si } G_n > S \text{ alors } M_n = G_n - S$$

$$\text{sinon } M_n = 0 .$$

L'évaluation de la rapidité des opérateurs étudiés doit être effectuée dans le contexte d'un traitement « à la chaîne » de toute une séquence d'images, le cas d'une seule détection, pour deux ou trois images isolées, ayant peu d'applications pratiques. En particulier, le cas du traitement « à la chaîne » correspond à d'éventuelles réalisations câblées destinées à fonctionner en temps réel. Le nombre d'opérations par pixel dépend alors non seulement de la complexité de la méthode de détection, mais aussi de la possibilité de réutiliser les résultats intermédiaires obtenus

lors de la détection des frontières mobiles dans l'image I_{n-1} , afin de diminuer le nombre de calculs effectués pour détecter les frontières mobiles dans l'image I_n . La méthode de Vieren permet de réutiliser le gradient $G(|I_n - I_{n-1}|)$, tandis que la nôtre réutilise les gradients $G(I_{n-1})$ et $G(I_n)$.

Rang	Méthode	Calculs par pixel
1	proposée	1 gradient, 2 comparaisons, 1 différence simple
2	Vieren	1 gradient, 1 produit, 1 différence absolue
3	Stelmaszyk	2 gradients, 1 produit, 1 différence absolue

Sachant qu'une plus grande rapidité s'obtient souvent au prix d'une plus grande taille mémoire nécessaire pour le stockage des résultats intermédiaires, il est intéressant de comparer la place mémoire demandée par chacun de ces opérateurs. L'occupation de la mémoire dépend du type de réalisation : logicielle ou câblée à structure pipe-line. En version pipe-line, il faut pouvoir stocker autant d'images qu'utilise l'opérateur, moins une, car l'image la plus récente est traitée au fur et à mesure de sa numérisation. Dans une version logicielle, l'image la plus récente doit être entièrement acquise avant d'être traitée, ce qui demande un plan-image de plus.

Version pipe-line :

Rang	Méthode	Nombre de plans-images	Utilisation
1	Stelmaszyk	1	pour l'image I_{n-1}
2 ex æquo	Vieren	2	pour l'image I_n et le gradient $G(I_n - I_{n-1})$
2 ex æquo	proposée	2	pour les gradients $G(I_{n-1})$ et $G(I_n)$.

Les méthodes utilisant trois images nécessitent évidemment plus de mémoire que la méthode de Stelmaszyk. Cependant, comme elles sont plus performantes sur tous les autres plans, il est surtout important de les comparer entre elles. Or, on constate qu'elles utilisent, toutes les deux, la même taille mémoire.

2.2.3. Sensibilité au bruit

Au cours du seuillage des résultats fournis par les opérateurs comparés, on rencontre inévitablement le problème de sensibilité au bruit. En effet, plus on baisse le seuil, afin d'augmenter le nombre de frontières pertinentes visualisées, plus on risque de faire ressortir le bruit (fig. 8). On constate cependant que la répartition de ce bruit varie selon l'opérateur utilisé :

- **L'opérateur de Stelmaszyk** amplifie le bruit à tous les endroits correspondant à des frontières statiques visibles dans l'image courante, car le bruit du gradient de la différence d'images y est multiplié par des valeurs élevées du gradient de l'image courante, ce qui justifie le mauvais résultat obtenu par cet opérateur dans la figure 7 d. Il en est

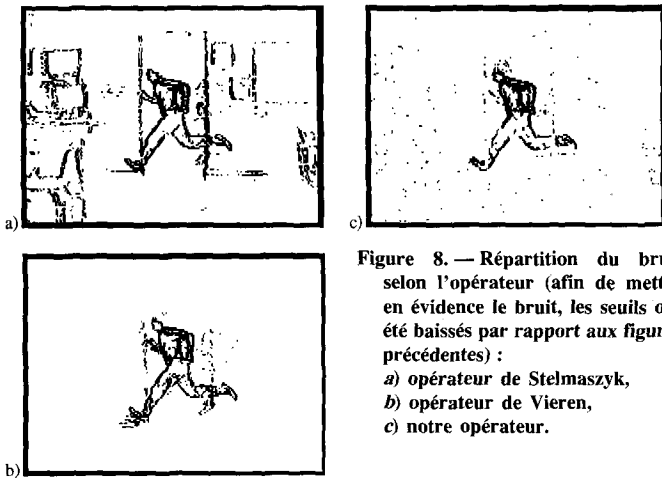
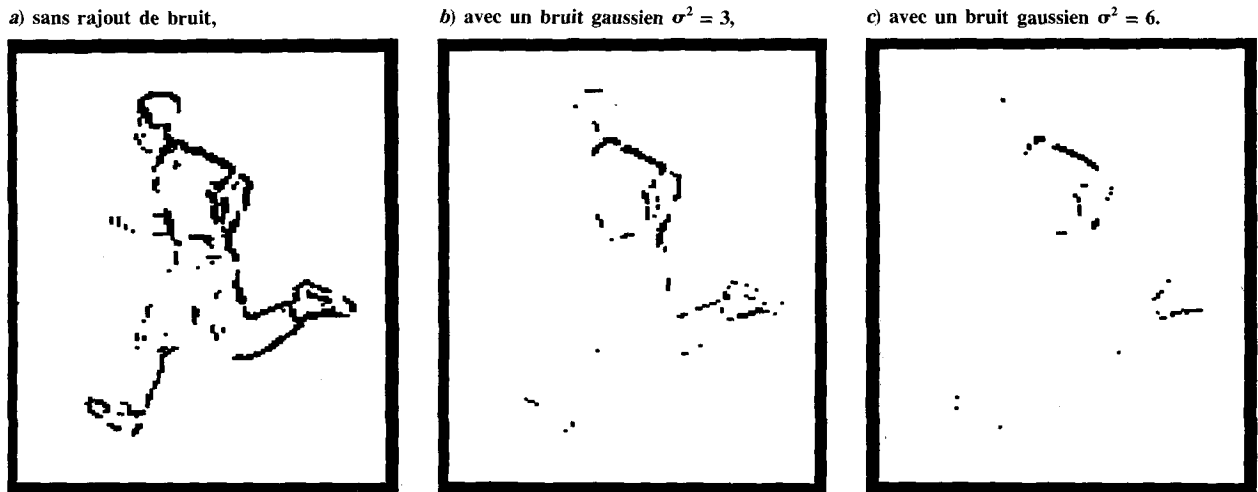


Figure 8. — Répartition du bruit selon l'opérateur (afin de mettre en évidence le bruit, les seuils ont été baissés par rapport aux figures précédentes) :
 a) opérateur de Stelmaszyk,
 b) opérateur de Vieren,
 c) notre opérateur.

de même pour les frontières mobiles de l'image précédente où le bruit du gradient de l'image courante est multiplié par des valeurs élevées du gradient de la différence d'images. Partout ailleurs le bruit est bien atténué par rapport aux amplitudes des frontières pertinentes.

• L'opérateur de Vieren amplifie le bruit par le même mécanisme de multiplication, mais seulement là où la présence de l'objet en mouvement dans l'image précédente ou dans l'image suivante donne une différence non nulle entre une de ces images et l'image courante, à savoir : pour les frontières mobiles dans leurs positions précédente et suivante, ainsi que pour les frontières statiques du type $\bar{P}CS$ et $PC\bar{S}$ (ex. : les montants de la porte dans la figure 8 b). Partout ailleurs, y compris pour les frontières statiques visibles dans les trois images, le bruit est bien atténué.

RÉSULTATS OBTENUS PAR L'OPÉRATEUR PROPOSÉ :



RÉSULTATS OBTENUS PAR L'OPÉRATEUR DE VIEREN :

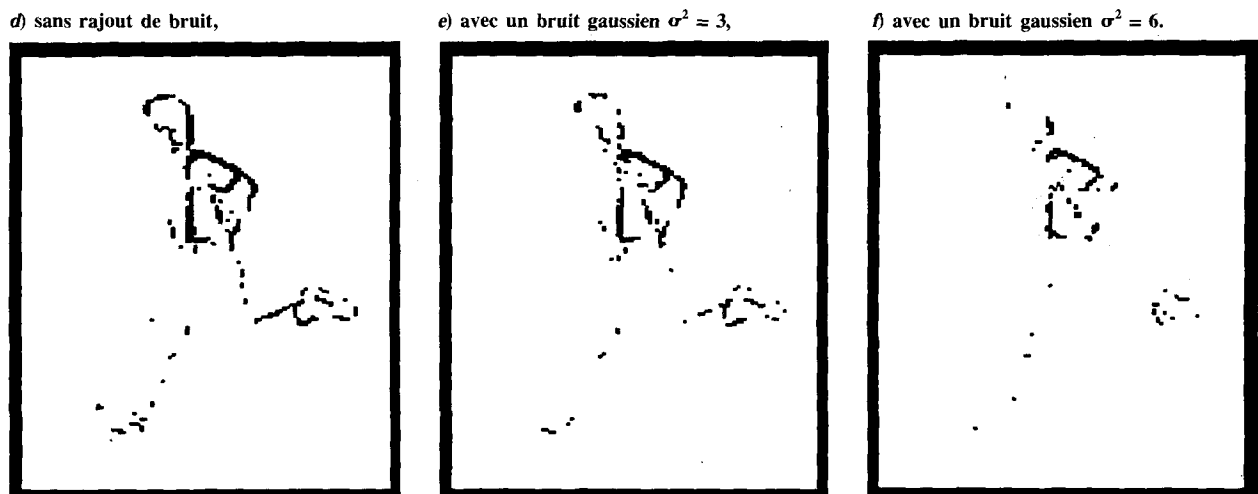


Figure 9. — Dégradation des résultats en fonction de la présence du bruit. N'utilisant pas la multiplication, notre opérateur n'atténue pas les frontières peu marquées et fournit ainsi initialement un plus grand nombre de frontières pertinentes. Avec l'augmentation de la présence du bruit ses résultats se dégradent plus rapidement que les résultats obtenus par l'opérateur de Vieren, mais les résultats fournis par les deux opérateurs deviennent alors inexploitable. Précisons que le trait vertical sur la poitrine du coureur, dans le résultat fourni par l'opérateur de Vieren, est une fois de plus une frontière parasite. Il s'agit d'un montant de porte caché dans l'image courante mais visible dans les images précédente et suivante.

• **Notre opérateur** élimine parfaitement les frontières mobiles dans leurs positions précédente et suivante, ainsi que les frontières statiques visibles dans les deux images voisines mais cachées dans l'image courante, comme il a été démontré précédemment. En revanche, pour toutes les autres frontières statiques (et pour toutes les régions statiques, en général), la différence des maxima a tendance à amplifier le bruit. Par exemple, si le bruit du gradient d'image était gaussien avec une variance σ^2 et indépendant d'une image à l'autre, la variance du bruit du résultat pourrait être majorée par $2 \cdot \sigma^2$.

A titre indicatif, nous avons évalué expérimentalement la sensibilité au bruit relative des trois opérateurs, dans le cas où le gradient est calculé à l'aide de l'opérateur de Sobel. Pour ce faire, nous avons comparé le nombre de points pertinents extraits par la méthode de seuillage décrite plus haut (cf. § 2.2.1 *Remarque*) et ceci avant et après un bruitage artificiel des images. Avant le bruitage notre opérateur a toujours donné les meilleurs résultats, l'opérateur de Stelmaszyk étant de loin le moins performant. L'écart entre notre opérateur et l'opérateur de Vieren diminue lorsque la présence du bruit dans les images augmente. L'opérateur de Vieren prend la tête du classement pour les images très bruitées, mais les trois opérateurs fournissent alors des résultats inexploitable (fig. 9).

2.2.4. Bilan

L'avantage de notre opérateur est particulièrement visible en matière d'élimination de frontières statiques, car il est le seul à éliminer à la fois les frontières désoccultées et les frontières cachées dans l'image courante mais visibles dans ses deux voisines. Même pour les images où ces frontières parasites sont absentes, notre opérateur permet d'extraire un plus grand nombre d'éléments pertinents, à l'exception des images très bruitées. En effet, dans les conditions expérimentales « normales », une plus grande sensibilité au bruit de notre opérateur est compensée par la non-utilisation de la multiplication, ce qui évite l'atténuation des frontières pertinentes peu marquées. On peut considérer que notre opérateur est également le plus rapide, puisque le cas de traitement par séquence d'images est le plus intéressant et correspond, en particulier, à une éventuelle réalisation câblée. Quant à la place mémoire nécessaire, il est à égalité avec l'opérateur de Vieren et derrière celui de Stelmaszyk, mais celui-ci occupe la dernière place dans tous les autres classements.

2.3. LIMITES DE LA MÉTHODE PROPOSÉE

Notre opérateur, plus performant que les opérateurs connus jusqu'à présent, ne résout tout de même pas tous les problèmes liés à l'utilisation de trois images consécutives d'une part et à l'exploitation des frontières extraites de ces images d'autre part.

Tout d'abord, le résultat pour l'image courante n'étant pas disponible avant l'acquisition de l'image suivante, il faut pouvoir admettre le retard d'une image vidéo. Cette contrainte semble cependant être acceptable dans la grande majorité des applications.

La deuxième limite concerne les frontières statiques désoccultées dans l'image courante et recachées aussitôt dans l'image suivante. L'information fournie par l'image courante et ses voisines est alors insuffisante pour qu'un traitement de bas niveau puisse distinguer de telles frontières des véritables frontières mobiles. Cependant, ce cas est relativement rare. Il peut se produire seulement pour un objet mobile fortement concave ou articulé, ou encore suivi de très près par un autre objet mobile.

Rappelons également que l'on ne peut pas détecter des mouvements arbitrairement lents (cf. § 1.4). Dans le cas des méthodes qui utilisent les frontières, le déplacement minimum entre deux images consécutives permettant encore une détection correcte dépend de l'épaisseur des frontières, donc du contraste dans les images et de la méthode d'extraction des frontières.

En outre, puisque l'extraction des frontières amplifie le bruit et que seule l'information locale est exploitée, les résultats obtenus pour des images très bruitées peuvent être inexploitable. L'influence du bruit peut être réduite par un préfiltrage approprié des images à traiter, mais il faut trouver un compromis entre l'amélioration du rapport signal/bruit et l'épaississement des frontières. Ces problèmes, étroitement liés au choix de la méthode de calcul du gradient, ne sont pas abordés dans le présent article.

En revanche, nous nous penchons sur le problème de discontinuités que peuvent comporter les frontières mobiles extraites. Les fragments manquants correspondent alors soit aux frontières peu marquées, soit aux frontières situées dans l'image courante là où, dans une des images voisines, il y a également une frontière, statique (fig. 10 ab) ou mobile (fig. 10cd). Dans les paragraphes qui suivent, nous essayons d'apporter une solution à ce problème.

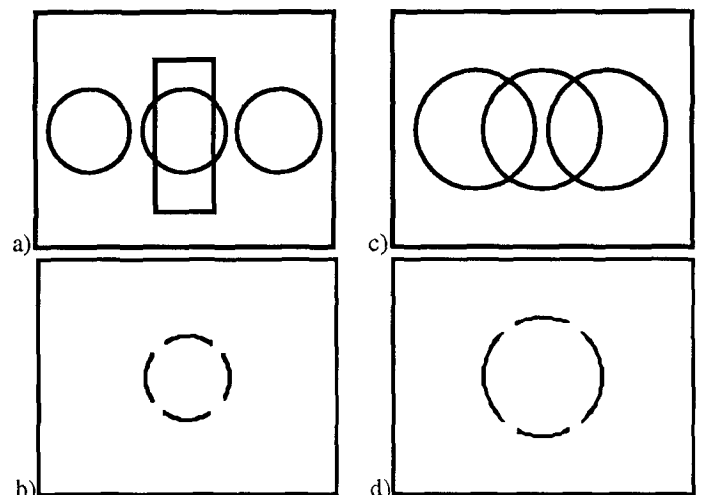


Figure 10. — Discontinuités dues à des superpositions de frontières :
 a) frontières mobiles (cercle) dans trois positions consécutives et frontières statiques (rectangle),
 b) frontières mobiles détectées, les fragments manquants correspondent aux intersections avec les frontières statiques,
 c) frontières mobiles dans trois positions consécutives,
 d) frontières mobiles détectées, les fragments manquants correspondent aux intersections avec les frontières mobiles dans leurs positions précédente et suivante.

3. Extension de la méthode : amélioration de la continuité des frontières détectées

L'amélioration de la continuité des frontières mobiles détectées peut être envisagée à deux niveaux : au cours de l'extraction des frontières mobiles — par l'introduction d'un critère supplémentaire qui permettrait d'éviter la suppression de fragments de frontières pertinentes — ou par un post-traitement visant à compléter les fragments manquants. Les deux solutions proposées ci-dessous correspondent respectivement à ces deux approches.

3.1. PRISE EN COMPTE DE L'ORIENTATION DU GRADIENT AU COURS DE L'EXTRACTION DES FRONTIÈRES MOBILES

Selon la définition introduite plus haut, un pixel (x, y) placé sur une frontière dans l'image courante est classé comme appartenant à une frontière mobile seulement si aucune frontière ne passe par ce même pixel dans les images précédente et suivante. La présence d'une frontière au même endroit, dans une des images voisines, entraîne automatiquement l'élimination du pixel (x, y) qui est interprété comme appartenant à une frontière statique. Or, nous voulons que cette élimination se produise uniquement lorsque la frontière visible dans l'image voisine est la même que celle dans l'image courante. Pour distinguer les frontières entre elles, nous allons comparer les orientations du gradient.

Une frontière statique, visible dans deux images consécutives, conserve (à epsilon près) non seulement son amplitude (module du gradient), mais aussi son orientation. Prenons un pixel appartenant à une frontière dans l'image courante et dans une des images voisines. Lorsque l'orientation du gradient, pour ce pixel, est très semblable dans les deux images, on peut considérer qu'il s'agit d'une même frontière. Inversement, lorsque les orientations du gradient sont très différentes, on peut affirmer qu'il s'agit de deux frontières distinctes (fig. 11).

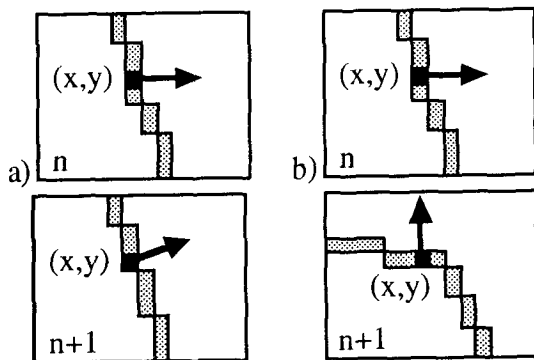


Figure 11. — Frontières distinguées grâce à leurs orientations :
 a) même frontière visible dans deux images consécutives, l'orientation locale est légèrement altérée par le bruit,
 b) les frontières qui passent par le pixel (x, y) dans les images n et $n + 1$, ont des orientations sensiblement différentes, on en déduit qu'il s'agit de deux frontières distinctes.

Nous proposons donc de modifier la définition des frontières mobiles. Un pixel placé sur une frontière dans l'image courante sera considéré comme appartenant à une frontière mobile si :

- soit aucune frontière ne passe par ce pixel dans les images précédente et suivante,
- soit, lorsqu'une frontière passe par ce même pixel dans une des images voisines, son orientation est incompatible avec l'orientation de la frontière visible dans l'image courante.

Reste à préciser ce que l'on entend par « incompatible ». Pour ce faire, on détermine un secteur autour de l'orientation courante. Les orientations appartenant à ce secteur sont considérées comme compatibles et celles qui n'y appartiennent pas, comme incompatibles avec l'orientation courante.

Notre opérateur peut être adapté à cette nouvelle définition de la manière suivante : pour chaque pixel, on compare d'abord l'orientation du gradient dans l'image courante ($orient_n$) à celles dans les images précédente ($orient_{n-1}$) et suivante ($orient_{n+1}$). Lorsque $orient_{n-1}$ ou $orient_{n+1}$ est compatible avec $orient_n$, on procède à la comparaison des modules du gradient selon la formule (7). Dans le cas contraire, on conserve inchangé le module du gradient de l'image courante.

En discrétisant les orientations on peut déterminer leur compatibilité rapidement, par indirection, en faisant appel à un tableau bidimensionnel rempli d'avance avec des « 1 » là où les indices correspondent à des orientations compatibles et avec des « 0 » partout ailleurs. Appelons ce tableau comp et prenons deux orientations a et b qui sont compatibles entre elles, ainsi que deux autres orientations c et d qui ne le sont pas. Nous avons alors : $comp[a][b] = comp[b][a] = 1$ et $comp[c][d] = comp[d][c] = 0$. Voici la formule utilisée dans notre expérimentation :

$$(9) \quad \begin{aligned} &\text{si } comp[orient_n(x, y)][orient_{n-1}(x, y)] = 1 \\ &\text{ou } comp[orient_n(x, y)][orient_{n+1}(x, y)] = 1 \\ &\text{alors } M_n(x, y) = \\ &\quad \max(G_n(x, y), G_{n-1}(x, y), G_{n+1}(x, y)) \\ &\quad - \max(G_{n-1}(x, y), G_{n+1}(x, y)) \\ &\text{sinon } M_n = G_n(x, y). \end{aligned}$$

La prise en compte de l'orientation du gradient permet une amélioration de la continuité des frontières mobiles détectées (fig. 12). Cette amélioration est d'autant plus sensible que le secteur des orientations compatibles est étroit. Cependant, plus on limite la largeur de ce secteur, plus il est difficile d'éliminer les fragments de frontières statiques dont l'orientation a été altérée par le bruit. En outre, il faut préciser que la modification proposée concerne les discontinuités dues à la présence, dans les images voisines, de frontières situées au même endroit que les frontières mobiles dans l'image courante. On ne peut pas améliorer ainsi la continuité des frontières mobiles peu prononcées.

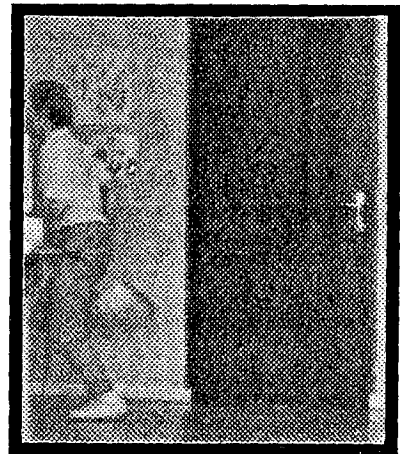
IMAGES DE DÉPART :



a) précédente,



b) courante,



c) suivante,

FRONTIÈRES MOBILES EXTRAITES (même seuil partout) :



d) sans prise en compte des orientations du gradient,



e) avec un secteur de compatibilité assez large (cinq orientations : courante ± 1 et ± 2),



f) avec un secteur de compatibilité étroit (trois orientations : courante et ses deux voisines),



g) points présents dans la figure e et absents dans la figure d,



h) points présents dans la figure f et absents dans la figure e.

Figure 12. — Frontières mobiles détectées sans et avec la prise en compte des orientations du gradient

L'utilisation de l'information concernant l'orientation du gradient demande évidemment un temps de calcul et une place dans la mémoire supplémentaires.

- Temps de calcul

Pour chaque pixel, il faut effectuer deux opérations supplémentaires : déterminer l'orientation du gradient et la compatibilité de cette orientation avec les orientations dans les images voisines. Cependant, l'orientation du gradient, tout comme la compatibilité, peut être déterminée très rapidement, en faisant appel à un tableau rempli d'avance (une ROM dans une version câblée) et en utilisant comme indices (adresses) les composantes horizontale et verticale qui sont de toute façon nécessaires pour calculer le module du gradient [7], [18]. De plus, sans prise en compte des orientations, la différence de maxima est calculée pour tous les pixels, tandis qu'avec la prise en compte des orientations ce calcul est limité aux pixels pour lesquels le test de compatibilité donne le résultat « 1 ». Le gain de temps ainsi obtenu est plus ou moins sensible suivant le nombre de tels pixels et il compense en partie le temps passé à déterminer les orientations et la compatibilité.

- Place mémoire

Précisons d'abord que les orientations du gradient peuvent être codées à l'aide des nombres entiers de 0 à 15, donc sur 4 bits. Compte tenu des incertitudes sur les orientations observées expérimentalement, une résolution plus fine n'est pas justifiée. Ainsi le tableau de compatibilité a une taille très réduite $16 \times 16 \times 1 \text{ bit} = 32 \text{ octets}$. Le tableau qui sert à déterminer les orientations à partir des composantes du gradient a la taille $NG \times NG \times 4 \text{ bits}$, où NG est le nombre de niveaux gris sur lesquels sont codées les images. Pour les nombres de niveaux de gris les plus courants, 64 et 256, on obtient respectivement 2 kilo-octets et 32 kilo-octets.

Enfin, il faut stocker les orientations correspondant aux images traitées. En pratique il n'est pas utile de les stocker toutes. Prenons une version pipe-line. Les orientations correspondant à l'image $n + 1$ sont calculées au fur et à mesure de la numérisation de cette image. Les orientations correspondant à l'image n ont été calculées en même temps que le module du gradient de cette image — elles doivent donc être stockées, ce qui demande 4 bits par pixel dans le cas du codage de 0 à 15. Pour l'image $n - 1$ en revanche, il n'est pas utile de stocker les orientations — il suffit de savoir si elles étaient compatibles avec celles de l'image n , ce qui demande la mémorisation d'un seul bit par pixel. Ainsi la mémoire supplémentaire peut être évaluée à 5 bits par pixel, donc 40 kilo-octets pour les images de taille $256 \times 256 \text{ pixels}$.

L'évaluation ci-dessus a été effectuée en considérant l'extraction des frontières mobiles comme une opération isolée. Or, une telle extraction est rarement une fin en soi. Elle fait généralement partie d'un ensemble d'opérations dont certaines, telle l'extraction de segments de droite ou l'approximation polygonale, font également appel aux orientations du gradient. Les orientations sont alors déterminées et mémorisées de toute façon et la prise en compte de cette information au cours de la détection des frontières mobiles augmente d'autant moins le temps de calcul et la

place occupée dans la mémoire. En effet, l'augmentation du temps de calcul est alors égale à la durée d'une indirection par pixel et partiellement compensée par le fait que la différence de maxima n'est pas calculée pour tous les pixels. Quant à l'augmentation de la place mémoire, elle est alors égale à la taille du tableau de compatibilité (32 octets), donc négligeable.

3.2. ... ET AU COURS DE LA BINARISATION

Grâce à la réalisation d'opérations à caractère logique sur les gradients, la binarisation a pu être repoussée à la fin du traitement. Néanmoins, quel que soit l'opérateur utilisé, il faut finalement répondre pour chaque pixel à la question : appartient-il, oui ou non, à une frontière mobile ? Le moyen le plus simple et le plus souvent utilisé est le seuillage à un seul seuil. Il remplit bien son rôle lorsque l'objet en mouvement présente partout un bon contraste avec le fond. Dans le cas contraire, le choix du seuil devient délicat, car un seuil haut provoque de nombreuses discontinuités correspondant à des frontières peu marquées, alors qu'un seuil bas fait ressortir le bruit (*fig. 13*).

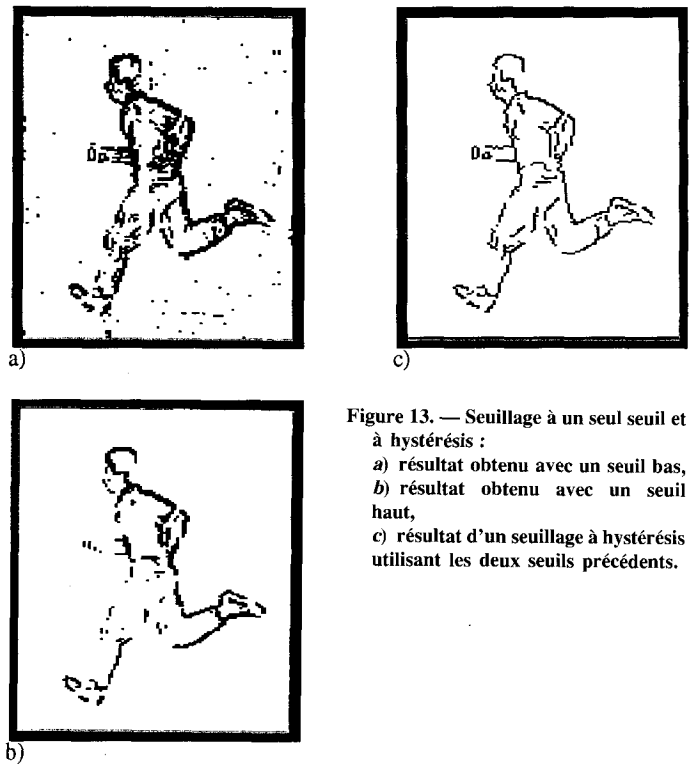


Figure 13. — Seuillage à un seul seuil et à hystérésis :
 a) résultat obtenu avec un seuil bas,
 b) résultat obtenu avec un seuil haut,
 c) résultat d'un seuillage à hystérésis utilisant les deux seuils précédents.

Il est donc préférable d'utiliser un seuillage à hystérésis [6] : un seuil haut sert à localiser les frontières bien marquées qui sont ensuite poursuivies (au sens de la connexité 2D) en dessous de ce seuil, tant que leurs amplitudes dépassent un autre seuil, plus bas. Dans [18], Chehikian et Rungsunseri proposent un algorithme fondé sur ce principe et implantable en temps réel. On y distingue deux phases : extraction des points de crête et propagation de connexité. Au cours de la première phase, on utilise l'orientation et le module du gradient pour sélectionner les

points susceptibles d'appartenir à un contour. Le module du gradient pour chacun de ces points doit remplir deux conditions : être supérieur au seuil bas et être un maximum local dans la direction du gradient. Afin de vérifier cette dernière condition, le module du gradient du pixel courant est comparé aux modules du gradient des deux pixels voisins, les plus proches de la direction du gradient. Les points retenus sont étiquetés comme « points de contour », si le module y est supérieur au seuil haut, ou comme « candidats » sinon. Par propagation on entend un changement d'étiquette d'un point « candidat » lorsqu'un de ses voisins déjà traités est un « point de contour ». Le sens de propagation est déterminé par le sens de balayage. Pour permettre la propagation dans tous les sens, on effectue quatre balayages consécutifs, chacun partant d'un coin d'image différent.

La technique citée ci-dessus est conçue dans l'optique de la segmentation d'images statiques. Néanmoins, on peut l'appliquer à un résultat de détection de frontières mobiles M_n en considérant que les orientations du gradient de l'image courante sont représentatives pour les frontières mobiles qui ont été extraites de cette image. Ainsi peut-on combler bon nombre de discontinuités dues à un faible contraste entre l'objet mobile et le fond (fig. 13 c).

En revanche, les discontinuités qui se produisent là où il y a « intersection » ou « superposition » d'une frontière mobile dans l'image courante et d'une autre frontière dans une des images voisines, non seulement ne peuvent pas être ainsi comblées, mais elles empêchent la propagation sur les points situés au-delà de telles interruptions (fig. 14 a). Remarquons que des discontinuités de ce genre peuvent subsister même dans le cas où la détection des frontières mobiles tient compte des orientations, car on ne peut pas exclure la présence, dans les images voisines, des frontières ayant des orientations très proches de celles de frontières mobiles dans l'image courante.

Nous proposons de compléter les fragments manquants à l'aide de frontières de l'image courante (G_n), en faisant l'hypothèse selon laquelle la plupart des discontinuités en question peuvent être assimilées à des segments de droite. En effet, dans le cas des « intersections », les interruptions sont généralement courtes et l'orientation du gradient varie peu entre les deux extrémités du fragment manquant. De même, les « superpositions » concernent avant tout des frontières rectilignes, car la superposition exacte de deux frontières curvilignes est moins probable.

Compte tenu de ces observations, l'algorithme cité ci-dessus peut être modifié comme suit :

Phase d'extraction des points de crête

Les points susceptibles d'appartenir à un contour sont sélectionnés dans l'image-gradient G_n selon les mêmes critères que dans la méthode de Chehikian et Rungsunseri. Les étiquettes initiales sont cependant attribuées en fonction des valeurs de l'image-résultat de la détection des frontières mobiles M_n . Ainsi, un point (x, y) appartenant à une crête dans G_n est étiqueté comme « contour mobile » lorsque la valeur de $M_n(x, y)$ dépasse le seuil haut, comme « candidat M » pour les valeurs comprises entre les seuils haut et bas et comme « candidat G » pour les valeurs

inférieures au seuil bas. Cette partie de l'algorithme se présente donc comme suit :

$\forall_{x,y}$ si $G_n(x, y) >$ seuil bas alors

début

1° sélectionner les pixels voisins p et q les plus proches de la direction du gradient en (x, y)

2° si $G_n(x, y) > G_n(x_p, y_p)$ et $G_n(x, y) \geq G_n(x_q, y_q)$

alors (c'est un maximum local dans G_n)

si $M_n(x, y) \leq$ seuil bas alors étiquette $(x, y) =$ « candidat G »

sinon si $M_n(x, y) >$ seuil haut

alors étiquette $(x, y) =$ « contour mobile »

sinon étiquette $(x, y) =$ « candidat M »

fin

Le choix de seuils dépend évidemment du rapport signal/bruit et de la dynamique des gradients, donc de la méthode de calcul du gradient. C'est pourquoi nous ne donnerons pas de recette pour un choix automatique.

Phase de propagation

Au cours de la propagation il faut traiter deux sortes de discontinuités. Pour les discontinuités dues à un faible contraste entre l'objet en mouvement et le fond, c'est-à-dire pour les points étiquetés « candidat M » connexes à des points étiquetés « contour mobile », la propagation doit se dérouler comme dans l'algorithme de Chehikian-Rungsunseri. En revanche, les discontinuités dues aux « intersections » ou « superpositions » de frontières demandent un traitement particulier.

L'objectif visé est de compléter les fragments manquants, dans le cas schématisé par la figure 14 a, tout en évitant la propagation sur des frontières statiques adjacentes. Conformément à l'hypothèse évoquée plus haut, parmi les chaînes de points initialement étiquetés « candidat G » il faut sélectionner celles qui remplissent les conditions suivantes :

1° début de chaîne connexe à un point étiqueté « contour mobile » (point A dans la figure 14 a),

2° fin de chaîne connexe à un point étiqueté « contour mobile » ou « candidat M » (point B),

3° orientation du gradient de chaque point de la chaîne compatible avec les orientations du gradient :

a) du point « contour mobile » de départ et

b) du point « contour mobile » (ou « candidat M ») d'arrivée.

En rencontrant un point « candidat G » au cours du balayage, il n'est pas possible de savoir tout de suite s'il appartient à une chaîne qui remplit toutes ces conditions. C'est pourquoi nous proposons d'introduire une étiquette temporaire « conditionnel » attribuée aux points qui satisfont les conditions 1° et 3° a, mais pour lesquels il n'est pas encore possible de dire s'ils remplissent les conditions 2° et 3° b. La gestion des chaînes de points « conditionnels » demande fondamentalement trois procédures : ouverture, prolongement et validation.

L'ouverture d'une nouvelle chaîne se produit lorsque le point courant est un « candidat G » connexe à un point étiqueté « contour mobile » (condition 1°) et que les orientations du gradient pour les deux points sont compatibles (condition 3° a). Cette procédure consiste à enregistrer

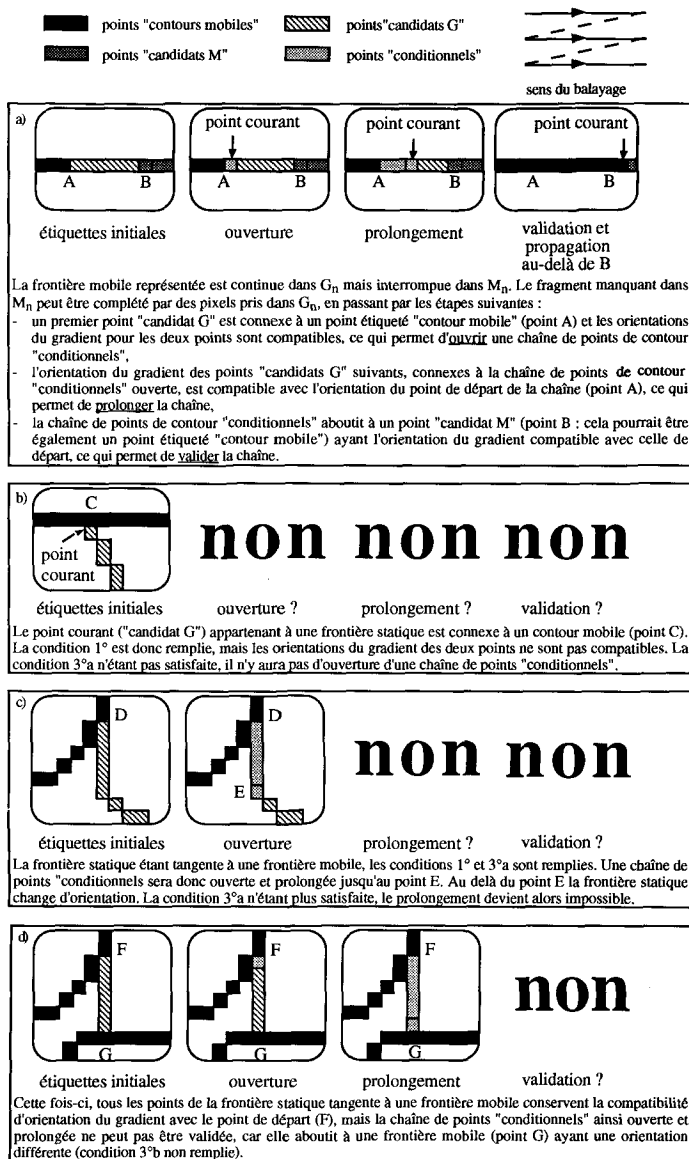


Figure 14. — Exemples de propagation sur des points « candidats G ». Les niveaux de gris correspondent aux étiquettes (le blanc étant réservé aux points qui n'appartiennent à aucun contour statique ni mobile).

l'orientation du gradient pour le point de départ, ainsi que les coordonnées du premier point de la chaîne.

Le prolongement d'une chaîne de points « conditionnels » existante peut être effectué lorsque le point « candidat G » rencontré est connexe à la queue de cette chaîne (condition 1°) et que l'orientation du gradient pour le point « candidat G » en question est compatible avec l'orientation du gradient pour le point de départ de la chaîne (condition 3° a). Cette procédure consiste à enregistrer les coordonnées du nouveau point « candidat G » devenu « conditionnel ».

La validation d'une chaîne de points « conditionnels » est possible lorsque le point courant est étiqueté « contour mobile » ou « candidat M » et qu'il est connexe à la queue

de cette chaîne (condition 2°). De plus l'orientation du gradient pour ce point doit être compatible avec l'orientation du gradient pour le point de départ de la chaîne (condition 3° b). Cette procédure consiste à rempalcer, pour tous les points de la chaîne concernée, l'étiquette « conditionnel » par l'étiquette « contour mobile ». Le fait d'avoir validé une chaîne permet de propager la connexité au-delà de la discontinuité correspondante. Ainsi l'étiquette initiale du point courant (rappelons : « contour mobile » ou « candidat M ») devient obligatoirement « contour mobile ».

Précisons encore que l'étiquette « conditionnel » est valable uniquement pour un sens de balayage donné. Autrement dit, les points « conditionnels » non validés au cours d'un balayage redeviennent des simples « candidats G » pour le balayage suivant.

Voici cette partie de l'algorithme :

```

pour quatre sens de balayage (partant successivement des quatre coins de l'image) faire
  pour tous les pixels faire
    si étiquette du pixel courant = « contour mobile » ou « candidat M »
      alors
        pour les quatre pixels voisins déjà traités faire
          si étiquette du voisin = « contour mobile » alors
            étiquette du pixel courant = « contour mobile »
          sinon
            si étiquette du voisin = « conditionnel » et orientation courante compatible avec l'orientation de la tête de la chaîne de points « conditionnels » à laquelle appartient le pixel voisin alors
              début
                étiquette du pixel courant = « contour mobile »
                validation de la chaîne de points « conditionnels »
              fin
            sinon
              si étiquette du pixel courant = « candidat G » alors
                pour les quatre pixels voisins déjà traités faire
                  si étiquette du pixel voisin = « contour mobile » alors
                    si orientation courante compatible avec l'orientation du pixel voisin alors
                      début
                        étiquette du pixel courant = « conditionnel »
                        ouverture d'une chaîne de points « conditionnels »
                      fin
                    sinon
                      si étiquette du pixel voisin = « conditionnel » alors
                        si orientation courante compatible avec l'orientation de la tête de la chaîne de points « conditionnels » à laquelle appartient le pixel voisin alors
                          début
                            étiquette du pixel courant = « conditionnel »
                            prolongement de la chaîne de points « conditionnels »
                          fin
                    fin
          fin
        fin
      fin
    fin
  fin
  
```

L'algorithme présenté ci-dessus permet d'obtenir une bonne continuité des contours mobiles détectés. L'amélioration par rapport au seuillage à hystérésis « classique » dépend cependant du caractère des discontinuités à compléter. Souvent leur longueur est de l'ordre de quelques pixels seulement et l'amélioration est alors difficilement perceptible à l'œil nu (fig. 15). Elle devient mieux visible lorsque le fait d'avoir complété une interruption de crête permet de propager la connexité sur les points « candidats M » situés au-delà de cette interruption (ex. : trait vertical indiqué par la flèche dans la figure 15 c).

L'amélioration la plus appréciable se produit lorsque les discontinuités complétées sont dues à la superposition, dans les images voisines, de longues frontières plus ou

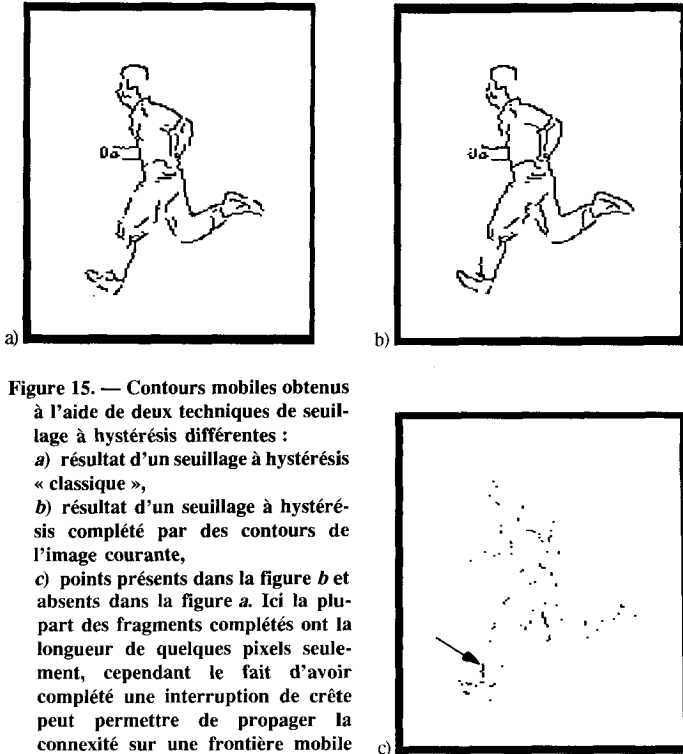


Figure 15. — Contours mobiles obtenus à l'aide de deux techniques de seuillage à hystérésis différentes :

a) résultat d'un seuillage à hystérésis « classique »,
 b) résultat d'un seuillage à hystérésis complété par des contours de l'image courante,
 c) points présents dans la figure b et absents dans la figure a. Ici la plupart des fragments complétés ont la longueur de quelques pixels seulement, cependant le fait d'avoir complété une interruption de crête peut permettre de propager la connexité sur une frontière mobile peu marquée, située au-delà de cette interruption (la flèche indique l'exemple de résultat d'une telle propagation).

moins rectilignes. En particulier, il est possible de restituer ainsi des contours mobiles quasiment parallèles à la direction du mouvement à condition d'avoir, pour chacun de ces contours, un point de départ permettant d'ouvrir une chaîne de points de contour « conditionnels » et un point d'arrivée permettant de la valider (fig. 16, 17 et 18). Ces points de départ et d'arrivée existent naturellement dans le cas de contours légèrement arrondis à leurs extrémités. Mais ils peuvent également avoir des sources plus aléatoires : variations de contraste le long du contour, reflets, ombres...

Comparons encore la place mémoire nécessaire dans le cas d'un seuillage à hystérésis selon l'algorithme de Chehikian-Rungsunseri et selon l'algorithme proposé ci-dessus. La gestion des chaînes de points « conditionnels » demande bien entendu une place mémoire supplémentaire. Il s'agit d'une mémoire tampon destinée à stocker les chaînes ouvertes (coordonnées de chaque point de la chaîne, orientation de la tête de la chaîne, pointeurs vers la tête et vers la queue etc.). Lorsqu'une chaîne est validée ou qu'elle ne peut plus être prolongée, la mémoire correspondante est libérée et peut être réutilisée. Pour éliminer les chaînes qui ne peuvent plus être prolongées, il suffit de supprimer après le balayage de chaque ligne (ou colonne, suivant le sens du balayage) les chaînes qui n'ont pas été prolongées au cours de ce balayage. Ainsi le nombre de chaînes ouvertes simultanément est réduit. La place mémoire nécessaire est par conséquent relativement modeste : dans notre expérimentation, les allocations dyna-

miques ont occupé quelques kilo-octets au maximum. Cependant, il faudrait encore trouver une architecture qui permette la réalisation câblée de la gestion des chaînes de points « conditionnels », car en version logicielle cette gestion ralentit sensiblement le fonctionnement de l'algorithme.

3.3. BILAN

Nous avons cherché à combler les discontinuités des contours mobiles détectés ; discontinuités que l'on peut regrouper dans deux catégories, suivant leurs origines : faible contraste d'une part et « intersections » ou « superpositions » de frontières dans les images voisines d'autre part. Le seuillage à hystérésis, méthode bien connue en segmentation statique, permet de faire ressortir les frontières mobiles peu marquées (donc appartenant à la première catégorie), à condition qu'elles soient connexes à des frontières mobiles plus nettes. Or cette connexité est souvent rompue par des discontinuités de la deuxième catégorie qui, elles, ne peuvent pas être comblées par le seuillage à hystérésis « classique ».

La première solution proposée (3.1), consistant à prendre en compte les orientations du gradient au cours de la détection des frontières mobiles, permet de limiter sensiblement le nombre de discontinuités dues aux « intersections » de ces frontières avec des frontières présentes dans les images voisines. Cette méthode a également le mérite d'être simple et rapide. Notons cependant qu'elle ne permet pas d'améliorer la continuité là où l'orientation d'un fragment de frontière mobile dans l'image courante est très proche de l'orientation d'une autre frontière présente au même endroit dans une des images voisines. En outre, elle s'appuie sur la notion de « secteur de compatibilité » dont l'utilisation passe par une opération fortement non linéaire qui rend la méthode sensible au bruit. Ce problème se pose essentiellement aux endroits où l'orientation du gradient est mal définie, angles en particulier, et demande encore la recherche d'une solution efficace.

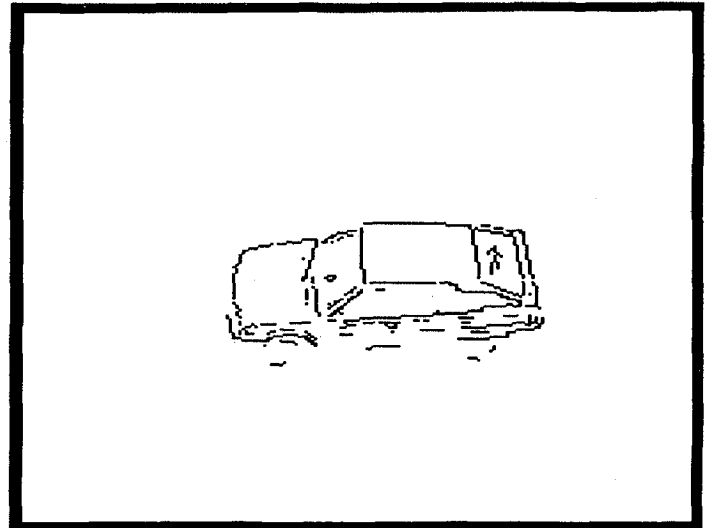
La deuxième solution, visant à restituer la continuité des frontières mobiles détectées, lorsqu'elle existe dans l'image courante (§ 3.2), intervient au cours du seuillage à hystérésis. Elle est plus générale que la première, puisqu'elle s'attaque avec succès aux deux catégories de discontinuités à la fois. Au prix d'une complexité plus grande que dans le cas du seuillage à hystérésis « classique », la technique de seuillage proposée permet de compléter de nombreux fragments de contours mobiles, même *quasiment* parallèles à la direction du mouvement. Néanmoins, on peut toujours trouver des contre-exemples, étant données les hypothèses de départ nécessaires pour éviter la confusion avec les frontières statiques adjacentes à des frontières mobiles.

En particulier, aucun traitement local ne permet de détecter les frontières mobiles :

- temporairement immobilisées (ex. : un pied qui repose par terre dans une phase de la foulée),
- *rigoureusement* parallèles à la direction du mouvement et ayant une longueur supérieure au déplacement de l'objet mobile entre deux images voisines.



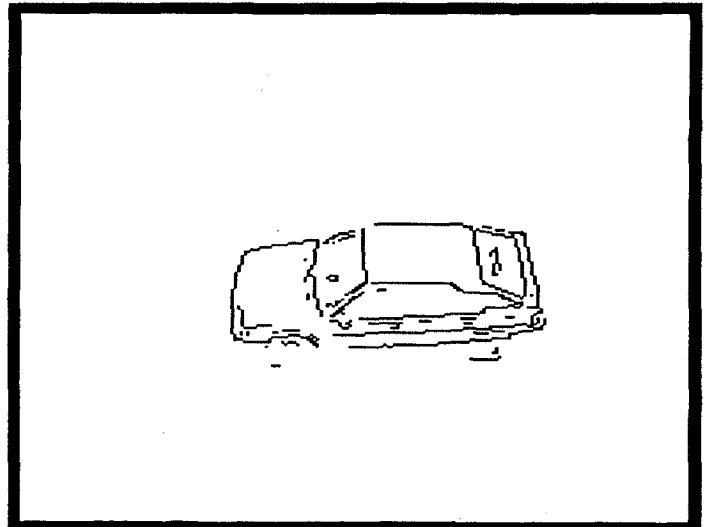
a)



d)



b)



e)

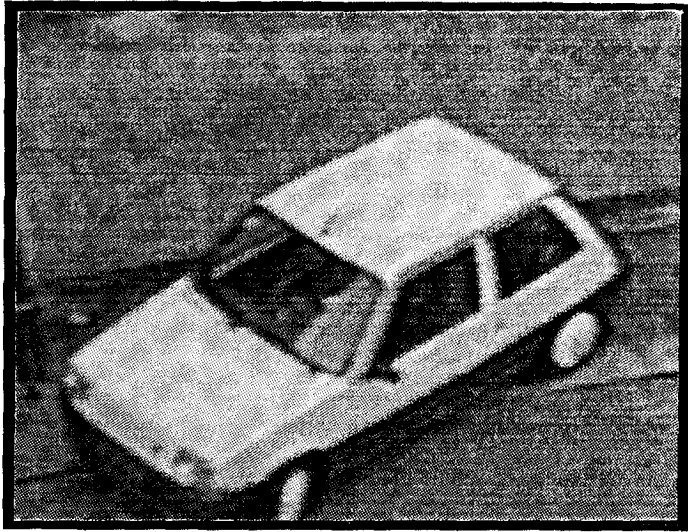


c)

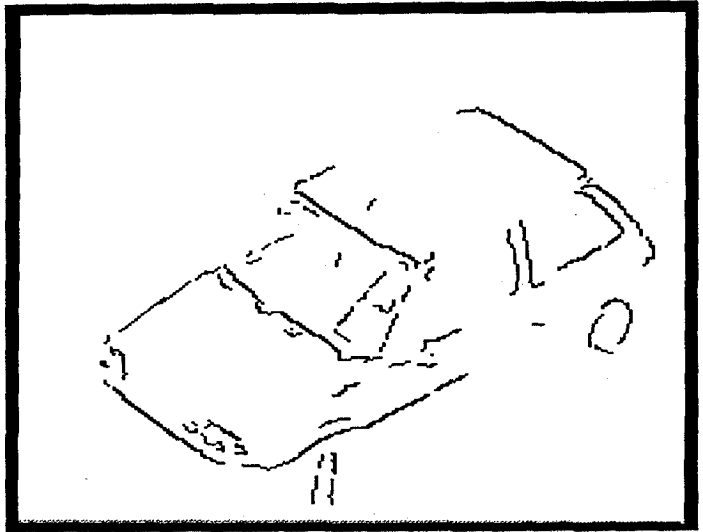
Figure 16. — Séquence CAR : exemple de contours mobiles presque parallèles à la direction du mouvement :

a), b), c) trois images consécutives de la séquence,
d) contours mobiles obtenus par seuillage à hystérésis « classique »,
e) contours mobiles obtenus par notre méthode de seuillage.

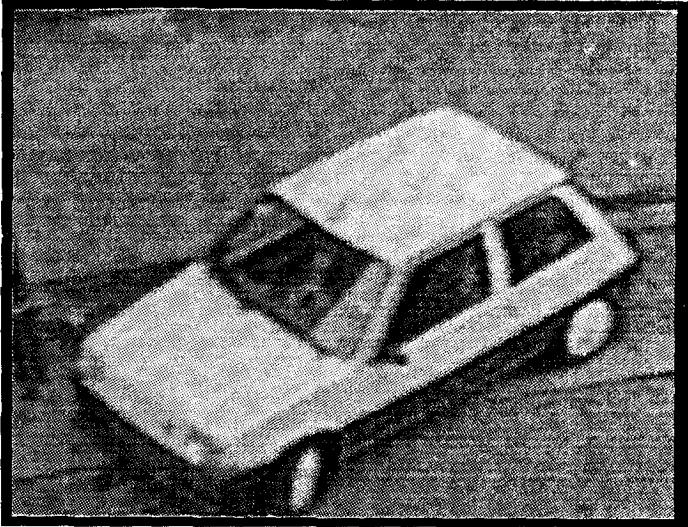
A cause des formes « aérodynamiques » de la voiture, ses contours longitudinaux ne sont pas exactement rectilignes et présentent par conséquent des parties qui n'ont pas été effacées au cours de la détection des frontières mobiles. Ces parties ont servi de points de départ et d'arrivée pour des chaînes de « points de contour conditionnels » et ont ainsi permis de combler plusieurs discontinuités de longueur importante.



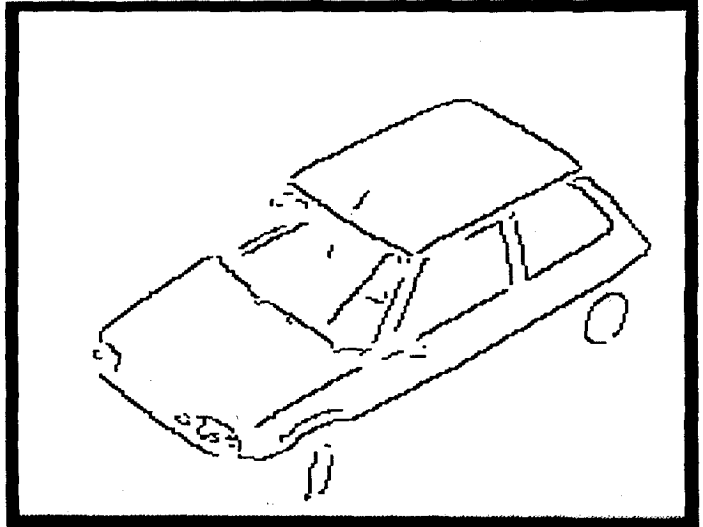
a)



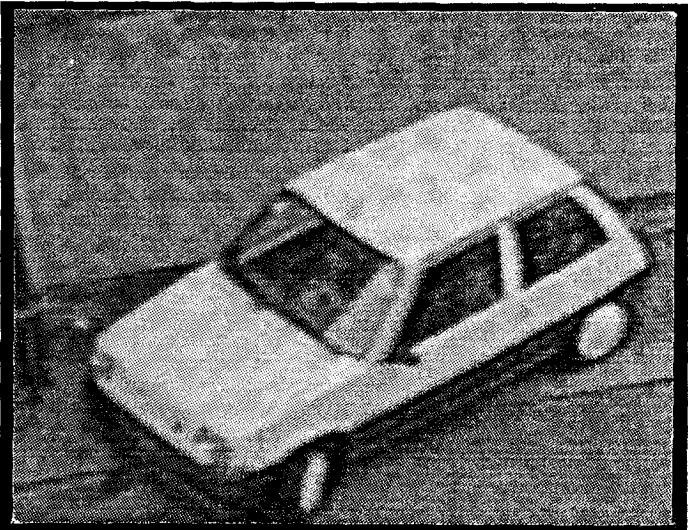
d)



b)



e)



c)

Figure 17. — Séquence CRENEAU présentant deux difficultés : contours mobiles presque parallèles à la direction du mouvement et très petit déplacement entre images voisines.

a) b), c) trois images consécutives de la séquence,

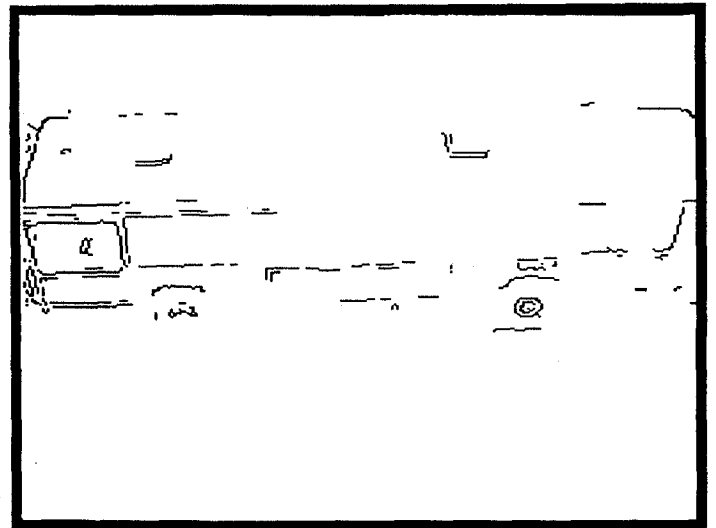
d) contours mobiles obtenus par seuillage à hystérésis « classique »,

e) contours mobiles obtenus par notre méthode de seuillage.

Les contours longitudinaux ne sont pas exactement rectilignes. Par conséquent, ils ont pu être complétés grâce au même mécanisme que dans le cas de la séquence CAR. Les contours transversaux de la voiture ont été correctement détectés malgré leur très petit déplacement entre images voisines.



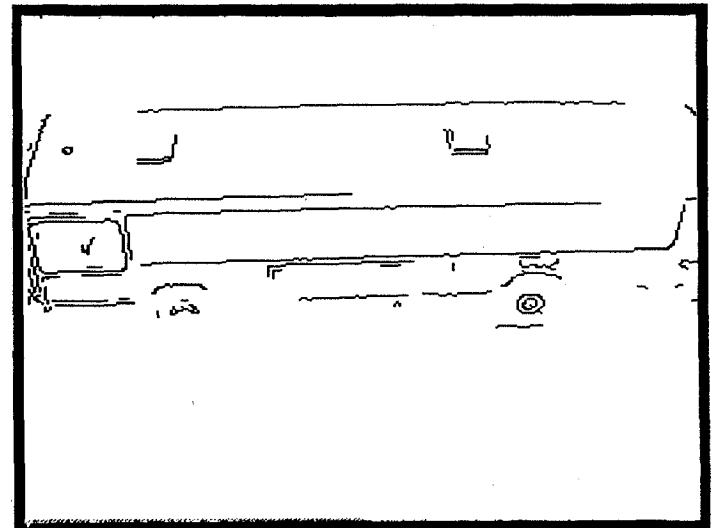
a)



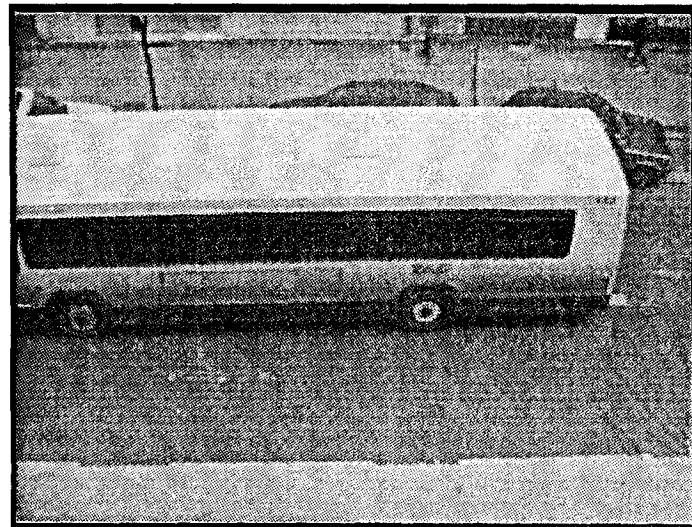
d)



b)



e)



c)

Figure 18. — Séquence BUS : un autre exemple de contours mobiles rectilignes, quasiment parallèles à la direction du mouvement.

a), b), c) trois images consécutives de la séquence,
d) contours mobiles obtenus par seuillage à hystérésis « classique »,
e) contours mobiles obtenus par notre méthode de seuillage.

Les contours longitudinaux du bus sont rectilignes mais, malgré les apparences, ne sont pas parfaitement parallèles à la direction du mouvement. En effet, une analyse « à la loupe » révèle l'existence d'une composante verticale du mouvement de la partie antérieure du bus. Cette composante est probablement due à la configuration de la chaussée.

Afin de résoudre les problèmes de ce genre, il faudrait développer un algorithme fondé sur le principe de la segmentation coopérative [3], dans lequel l'extraction des régions serait guidée par les frontières mobiles détectées à l'aide des méthodes présentées dans les pages précédentes. Cependant, la technique citée a un caractère global et, par conséquent, sort du cadre de la présente étude.

Conclusions

Nous nous sommes fixé comme objectif de trouver un opérateur permettant de localiser les objets en mouvement dans des scènes naturelles filmées par une caméra immobile, et ceci sans introduire de contraintes relatives à la connaissance préalable de la scène filmée et au caractère du mouvement étudié. Notre souci principal était la rapidité, car il est fort souhaitable que cet opérateur puisse être réalisé sous forme câblée fonctionnant en temps réel vidéo.

Ces critères ont focalisé notre recherche sur les méthodes qui exploitent les éléments variant entre images voisines et, en particulier, sur celles qui s'intéressent aux frontières. En effet, la réduction de la quantité de l'information, par rapport aux images originales à niveaux de gris, permet de distinguer plus facilement les éléments statiques des éléments mobiles de la scène.

Cependant, les opérateurs connus, appartenant à cette catégorie, interprètent comme mobiles certaines frontières statiques qui ne sont pas visibles dans toutes les images comparées. Or, trois images consécutives suffisent pour distinguer ces frontières statiques des véritables frontières mobiles.

Nous proposons de considérer comme mobile toute frontière qui, présente à un endroit dans l'image courante, est absente à ce même endroit dans l'image précédente et dans l'image suivante. Afin d'extraire les frontières mobiles selon cette définition, nous avons développé un opérateur original qui compare les gradients de l'image courante et de ses deux voisines. Les frontières mobiles extraites seulement par comparaison des modules du gradient peuvent présenter des discontinuités. La prise en compte des orientations du gradient améliore la continuité des frontières mobiles détectées. Cette voie mérite d'être explorée.

Par ailleurs, pour toute méthode de détection de frontières mobiles, la qualité des résultats dépend de la présence du bruit et de l'épaisseur des frontières. Or, la présence du bruit et l'épaisseur des frontières dépendent à leur tour de la méthode de calcul du gradient et en particulier de la méthode de lissage. Par conséquent, à l'issue de l'étude présentée ci-dessus il convient de chercher un opérateur de calcul du gradient qui satisferait au mieux les critères suivants : rapport signal/bruit, finesse des frontières et rapidité de l'ensemble des opérations.

REMERCIEMENTS

Les critiques constructives du professeur Alain Chehikian ont grandement contribué à la forme et au contenu de cet article, qu'il en soit cordialement remercié.

BIBLIOGRAPHIE

- [1] P. ANANDAN, « A Computational Framework and an Algorithm for the Measurement of Visual Motion », *International Journal of Computer Vision*, vol. 2, 1989, pp. 283-310.
- [2] A. AOUN, S. CASTAN, L. MASSIP-PAILHES, « Moving Objects Analysis », *Proceedings of the Scandinavian Conference on Image Analysis*, Oulou, Finland, June 1989.
- [3] P. BONNIN, B. ZAVIDOVIQUE, « Principles and Applications of a New Cooperative Segmentation Methodology », *Actes du congrès SPIE Visual Communications and Image Processing*, Philadelphia, November 1989, pp. 677-688.
- [4] P. BOUTHEMY, « Estimation et structuration d'indices spatio-temporels pour l'analyse du mouvement dans une séquence d'images », *Traitement du Signal*, 1987, vol. 4, n° 3, pp. 239-257.
- [5] P. BOUTHEMY, « Modèles et méthodes pour l'analyse du mouvement dans une séquence d'images », 2^e Atelier Scientifique CNRS Traitement d'images : du Pixel à l'Interprétation, Aussois, Savoie, 19-22 avril 1988, pp. XXV-1 à XXV-20.
- [6] J. F. CANNY, « Finding Edges and Lines in Images », *Technical Report No. 720*, MIT Artificial Intelligence Laboratory, June 1983.
- [7] A. CHEHIKIAN, J. HÉRAULT, J. E. MENDEZ, « Un processeur de détection de contours en temps réel dans une image de télévision », *Annales des Télécommunications*, tome 38, n°s 3-4, mars-avril 1983, pp. 91-98.
- [8] R. DERICHE, O. MONGA, J. P. COCQUEREZ, G. MALANDAIN, « Recursive Filtering and Edge Closing : Two Primary Tools for 3D Edge Detection », *Proceedings of the Computer Vision ECCV90*, Antibes, France, April 1990, pp. 56-65.
- [9] W. ENKELMANN, « Obstacle Detection by Evaluation of Optical Flow Fields », *Proceedings of the Computer Vision ECCV90*, Antibes, France, April 1990, pp. 134-138.
- [10] S. M. HAYNES, R. JAIN, « Detection of Moving Edges », *Computer Vision, Graphics and Image Processing*, 1983, vol. 21, pp. 345-367.
- [11] R. JAIN, H. N. NAGEL, « On the Analysis of Accumulative Difference Pictures from Image Sequences of Real World Scenes », *IEEE Transactions on Pattern Recognition and Machine Intelligence*, April 1979, vol. 1, n° 2, pp. 206-214.
- [12] R. JAIN, « Différence and Accumulative Difference Pictures in Dynamic Scene Analysis », *Image & Vision Comput.*, May 1984.
- [13] P. LALANDE, P. BOUTHEMY, « Détection de zones en mouvement dans une séquence d'images », *Colloque GRETSI Traitement du Signal et ses Applications*, Juan-les-Pins, 12-16 juin 1989, pp. 707-710.
- [14] M. K. LEUNG, Y. H. YANG, « Human Body Motion Segmentation in a Complex Scene », *Pattern Recognition*, 1987, vol. 20, n° 1, pp. 55-64.
- [15] M. ORKISZ, « Image Difference based Methods for Moving Objects Detection in Real World Scenes », *Proceedings of 7th International Symposium IASTED Applied Informatics*, Grindelwald, February 1989, pp. 80-83.
- [16] M. ORKISZ, « Détection d'objets mobiles dans des scènes naturelles », *Colloque GRETSI Traitement du Signal et ses Applications*, Juan-les-Pins, 12-16 juin 1989, pp. 677-680.
- [17] M. ORKISZ, « Éléments du traitement d'images, et analyse du mouvement humain », *Thèse de docteur de l'INP de Grenoble*, 1990.
- [18] Y. RUNGSUNSERI, A. CHEHIKIAN, « A Real Time System for Extracting Edges and Lines in Images », *Proceedings of the 7th Scandinavian Conference on Image Analysis*, Aalborg (Denmark), August 13-16, 1991, pp. 839-846.

- [19] J. SHEN, S. CASTAN, J. ZHAO, « Optimal Filter for Edge Detection Methods and Results », Proceedings of the Computer Vision ECCV90, Antibes, France, April 1990, pp. 13-17.
- [20] P. STELMASZYK, P. BONNET J. G. POSTAIRE, « Analyse de scènes dynamiques par recherche des contours en mouvement », 5^e Congrès Reconnaissance de Formes et Intelligence Artificielle AFCET/ADI/INRIA, Grenoble (France), November 1985, pp. 1181-1189.
- [21] I. S. TANG, W. E. SNYDER, S. A. RAJALA, « Extraction of Moving Objects in Textured Dynamic Scenes », IEEE Comput. Soc. Conf. on Pattern Recogn. Image Proc., Las Vegas, June 1982, pp. 231-236.
- [22] Ch. VIEREN, « Segmentation de scènes dynamiques en temps réel. Application au traitement de séquences d'images pour la surveillance de carrefours routiers », Thèse de docteur de l'Université des Sciences et Techniques de Lille Flandres Artois, 1988.
- [23] Ch. VIEREN, J. G. POSTAIRE, P. BONNET, P. DEPARIS, « Détection du contour extérieur d'objets en mouvement sur fond non uniforme », Colloque GRETSI Traitement du Signal et ses Applications, Juan-les-Pins, 12-16 juin 1989, pp. 621-624.

Manuscrit reçu le 21 septembre 1990, version révisée le 11 septembre 1991.