

50376
1992
41

50376
1992
41

N° d'ordre : 859

THESE

présentée à

L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE
FLANDRES ARTOIS

pour l'obtention du titre de

DOCTEUR

en Productique : Automatique et Informatique Industrielle

par

François CABESTAING

DETECTION DE CONTOURS EN MOUVEMENT DANS UNE
SEQUENCE D'IMAGES. CONCEPTION ET REALISATION
D'UN PROCESSEUR CABLE TEMPS-REEL.

Soutenu le 31 janvier 1992 devant la commission d'examen :



MM.

P. VIDAL	Président	Professeur à l'U.S.T.L.F.A.
Y. DAVID	Rapporteur	Directeur du C.R.E.S.T.A.
J. GALLICE	Rapporteur	Professeur à l'U.B.P Clermont-Ferrand II.
J.G. POSTAIRE	Directeur de thèse	Professeur à l'U.S.T.L.F.A.
J. BAJON	Examineur	Professeur à l'E.N.S.E.E.I.H.T.
P. BONNET	Examineur	Maître de Conférences à l'U.S.T.L.F.A.
J.P. DEPARIS	Examineur	Chargé de recherches C.R.E.S.T.A.

SCD LILLE 1



D 030 290421 4

50376
1992
41

2

61 259 981

50376
1992
41

REMERCIEMENTS

Le travail présenté dans ce mémoire a été effectué au Centre d'Automatique de l'Université des Sciences et Techniques de Lille Flandres Artois (USTLFA), en collaboration avec l'Institut National de Recherche sur les Transports et leur Sécurité (INRETS).

J'exprime toute ma gratitude à Monsieur le Professeur P. VIDAL pour m'avoir accueilli au sein de son laboratoire. Je le remercie d'avoir bien voulu accepter la présidence de mon jury de thèse.

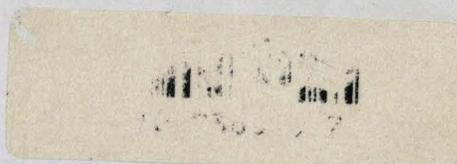
C'est en grande partie grâce à Monsieur le Professeur J-G POSTAIRE que cette étude a abouti. Ses conseils, ses constants encouragements et sa grande disponibilité m'ont permis de mener à bien ce travail. Qu'il trouve en ces quelques mots l'expression de mes remerciements les plus sincères.

Que Monsieur Y. DAVID, directeur du CRESTA, trouve ici mes plus vifs remerciements pour l'intérêt constant qu'il a porté à mes travaux et pour ses conseils avisés. Je lui suis très reconnaissant d'avoir bien voulu juger mon mémoire et participer à mon jury. Je remercie également Monsieur le Professeur J. GALLICE, de l'université Blaise Pascal de Clermont Ferrand, d'avoir accepté de juger mon travail.

J'exprime toute ma gratitude à Monsieur le Professeur J. BAJON pour l'intérêt qu'il a bien voulu porter à mon travail en acceptant de participer à mon jury.

Je tiens à remercier Monsieur P. BONNET, Maître de Conférences à l'USTLFA, et Monsieur J-P DEPARIS, chargé de Recherches au CRESTA, d'avoir accepté de faire partie de mon jury. Qu'ils trouvent ici l'expression de ma reconnaissance pour les nombreux conseils avisés qu'ils m'ont donnés lors de la réalisation de ce travail.

Je remercie enfin toute l'équipe de chercheurs de l'équipe Image et Décision du Centre d'Automatique de Lille, tout particulièrement Monsieur C. VIEREN, qui a en quelque sorte été à l'origine de mon travail de recherche, Messieurs L. DUVIEUBOURG et L. MACAIRE, pour leur présence amicale et l'aide qu'il m'ont apporté.



PRESENTATION DU CONTEXTE DE L'ETUDE

L'étude présentée dans ce mémoire s'intègre dans un programme de recherche lancé il y a quelques années par le Groupement Régional Nord/Pas-de-Calais pour la Recherche dans les Transports (GRRT), dont fait partie le Centre Régional d'Evaluation de la Sécurité des Transports Automatiques (CRESTA). Ce programme étudie la possibilité d'implanter un système de transport guidé automatique en site non protégé.

LE PROJET DE CONDUITE AUTOMATIQUE

Les transports automatiques sont devenus courants de nos jours. On peut notamment citer les métros de Lille, Londres, Vancouver, Osaka. Toutefois, ces différents systèmes évoluent en sites protégés, c'est à dire sans interaction directe avec d'autres véhicules. Cette caractéristique, permettant de renforcer les conditions de sécurité concernant les voyageurs, implique toujours des travaux de génie civil très importants (tunnels, voies aériennes ...). L'augmentation du coût global des installations est alors un facteur qui a jusqu'alors souvent limité les possibilités d'implantation de tels systèmes.

Une solution pour éviter ces coûts prohibitifs consiste à utiliser des moyens de transport évoluant en sites semi-protégés, c'est à dire sur des voies pouvant traverser localement des zones banalisées où circulent d'autres types de véhicules, ce qui permet de diminuer la complexité des différentes infrastructures nécessaires au

passage des voies en les implantant en surface. Cependant, à l'heure actuelle, il n'existe aucun système de transport de ce type circulant de façon totalement automatique, et ce principalement pour des raisons de sécurité.

Le programme de recherche lancé par le GRRT il y a une dizaine d'années doit tenter de répondre aux différentes questions que soulève l'automatisation d'un système de transport de ce genre. Au sein du groupement, le Centre d'Automatique de Lille, en collaboration avec le CRESTA, s'est vu confier une partie des études concernant la spécification, la conception et la mise au point des différents systèmes devant permettre le pilotage du véhicule au niveau des zones banalisées (carrefours urbains).

LA VISION ARTIFICIELLE

Différentes études ont été entreprises au Centre d'Automatique concernant ce projet, elles ont pour l'instant fait l'objet de trois thèses [Ste85, Bou88, Vie88]. Deux axes principaux de recherche se sont dégagés.

Tout d'abord, une étude a été réalisée concernant la mise en évidence des problèmes rencontrés lors de la conception d'un processus de pilotage automatique du véhicule [Dep84]. Le système de pilotage doit intégrer les règles élémentaires de circulation (code de la route), ce qui en soi ne constitue pas un problème insurmontable. Mais il doit aussi être capable de réagir à un non respect de ces règles par les véhicules qui circulent dans les zones traversées. Ce dernier point, difficilement appréhendable au niveau d'un simple automatisme, montre la nécessité de recourir à un système faisant intervenir des notions d'intelligence artificielle [Bou88].

D'un autre côté, afin de générer les ordres de conduite, il est nécessaire de prendre en compte de façon fiable l'environnement du véhicule au niveau des carrefours. On peut utiliser pour cela différents capteurs tels que boucles d'induction, radars, etc... Cependant, les informations que ces derniers sont susceptibles de fournir restent très locales, et ne permettent pas de prédire efficacement l'évolution de la situation.

Par utilisation d'un système de vision artificielle, il s'avère possible de disposer d'une vue d'ensemble de la zone non protégée que le véhicule automatique doit franchir. Des études précédentes [Ste85, Vie88] ont montré que par analyse d'une séquence d'images acquise par une caméra vidéo surveillant le carrefour, il est possible de détecter les obstacles présents dans la zone à traverser et de déterminer leurs trajectoires.

L'utilisation d'un système de vision en tant que capteur spécifique dans un processus de conduite automatique implique le respect de certaines contraintes de sécurité : les mobiles doivent être détectés le plus rapidement possible, et de façon fiable quelles que soient les conditions d'utilisation. Ces deux critères s'avèrent décisifs lorsqu'il s'agit de comparer deux méthodes de détection.

PLAN DU MEMOIRE

La première partie de ce document présente un aperçu des méthodes permettant d'extraire des informations concernant le mouvement des objets présents dans une séquence d'images. Une comparaison des différentes approches, effectuée selon des critères liés à leur fiabilité et leur rapidité, nous permet de montrer que les techniques de détection basées sur des différences d'images successives sont celles correspondant le mieux à nos besoins.

Une analyse plus spécifique de ces algorithmes est présentée dans le deuxième chapitre. En particulier, une méthode particulièrement efficace de détection des contours des objets en mouvement dans une séquence d'images est décrite dans cette partie. Différentes méthodes servant habituellement à l'évaluation des opérateurs de détection de contours statiques sont utilisées afin de comparer les différentes techniques de détection de contours en mouvement.

La troisième partie est consacrée à la conception et à la réalisation d'un processeur câblé de traitement de séquences d'images, basé sur une structure pipeline, réalisant l'opération d'extraction des contours mobiles en temps-réel selon les procédures décrites au deuxième chapitre. Le STREAM (Système Temps-Réel d'Extraction et d'Analyse du Mouvement) fonctionne en tant que préprocesseur indépendant de toute architecture informatique d'accueil.

Dans le quatrième chapitre, nous présentons des essais d'utilisation de ce processeur sur site réel. Les résultats obtenus montrent l'efficacité du prototype réalisé, à la fois lors de son utilisation en tant que capteur spécialisé dans le projet de conduite automatique d'un véhicule guidé, et dans d'autres domaines d'application, telle la surveillance de stations de métro ou le comptage de personnes.

CHAPITRE I

DETECTION DE MOUVEMENT DANS UNE SEQUENCE D'IMAGES

I.1) LE MOUVEMENT DANS UNE SEQUENCE D'IMAGES

En analyse d'images statiques, il est devenu courant de décrire une méthode selon le niveau d'abstraction qu'elle utilise afin de représenter l'information [Pog87, Hwa89, Wee89]. Selon ce principe de hiérarchie, on peut distinguer trois niveaux de complexité. Le "bas niveau" pour lequel l'image est un ensemble de pixels. Le "haut niveau" où l'information contenue dans l'image est décrite en termes d'objets. Le niveau intermédiaire assurant la liaison inévitable entre ces deux représentations.

En transposant cette classification au problème de l'analyse d'images dynamiques, on se rend compte que l'étude du mouvement dans une séquence d'images peut être envisagée de plusieurs façons différentes.

Haut niveau :

Certaines caractéristiques des objets présents dans les images successives (position, taille, forme ...) sont analysées en termes d'évolution dans le déroulement de la séquence.

Niveau intermédiaire :

Des traitements effectués sur la séquence d'images, pouvant combiner des caractéristiques locales (par exemple l'évolution des niveaux de gris dans une région) et des caractéristiques globales (évolution de l'illumination de la scène observée) permettent de segmenter l'image en zones homogènes vis à vis du mouvement.

Bas niveau :

Des traitements locaux, mettant en jeu tant le voisinage spatial d'un pixel que son voisinage temporel (pixels situés à la même position dans les autres images) fournissent une nouvelle image dans laquelle les pixels affectés par le mouvement se différencient simplement des pixels des zones stationnaires.

Le premier niveau regroupe principalement les approches qualifiées de méthodes structurales, ou orientées objets (token-based methods). Le deuxième niveau regroupe des méthodes basées sur l'analyse du champ des déplacements ou du flux optique. Le troisième, quant à lui, fait appel à des techniques d'analyse des variations locales de l'intensité de l'image, parfois qualifiées de méthodes orientées mouvement (motion-based methods).

Dans ce chapitre, en nous basant sur cette classification des méthodes de détection de mouvement, nous passons en revue les différentes approches présentées dans la littérature.

I.2) METHODES STRUCTURELLES (HAUT NIVEAU)

Comme nous l'avons précisé dans le paragraphe précédent, les méthodes structurelles sont basées sur l'étude de l'évolution temporelle de certaines caractéristiques d'objets présents dans les différentes images.

Si l'information de départ se résume une séquence d'images, une première phase obligatoire consiste en la détermination de certains éléments caractéristiques des objets (features).

Dans un deuxième temps, la mise en correspondance de ces éléments structurels d'une image à la suivante permet de caractériser le mouvement.

I.2.1) EXTRACTION DES ELEMENTS STRUCTURELS

Un élément structurel peut être défini comme un élément de l'image (pixel, ligne ou surface) possédant une propriété caractéristique permettant de le différencier de son voisinage. On suppose souvent que cette caractéristique reste invariante dans le temps. On peut alors assimiler l'élément structurel à un objet (token)(Figure 1.1).

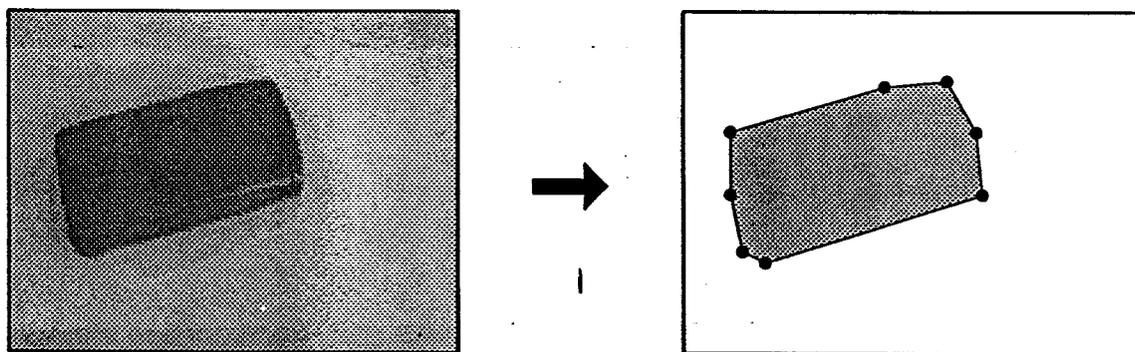


Figure 1.1 : Extraction des éléments structurels, exemple : points anguleux

De nombreuses études ont été menées sur la recherche de ces éléments structurels [Kor84]. Selon le type de détection de mouvement souhaité, ou selon la précision nécessaire, on peut choisir entre différentes catégories d'éléments caractéristiques :

- Des points d'intérêt (coins, points anguleux, points de courbure maximale ...) [Agg75, Ros75, Mor77, Asa80, Mor83, Dre83].
- Des segments de contours [Dav79, Kor80, Jac80, Sny81].
- Des régions plus ou moins étendues [Mar77, Pri75, Pri77, Wil80, Kan85, Hun87].
- Des marqueurs morphologiques [Blo87].

La recherche des éléments structurels peut se faire soit sur l'image initiale, soit sur une image transformée : image de contours [Law83, Mai84], transformée de Hough [Fen79].

1.2.2) MISE EN CORRESPONDANCE

Afin de connaître le mouvement d'un objet, il est nécessaire de définir sa position dans toutes les images successives de la séquence. Cependant la phase d'extraction des éléments structurels ne permet pas, à elle seule, d'établir les liens entre les images successives. Etant donné la position d'un objet mobile dans une image, on ne dispose en général que d'un nombre limité d'informations a priori quant à sa position dans l'image suivante.

L'association d'un couple (objet, position) dans l'image en cours d'analyse avec un autre couple (objet, position) dans l'image précédente est appelée mise en correspondance (feature matching). Cette mise en correspondance permet d'associer un déplacement à chaque objet lors du passage d'une image à la suivante. De façon générale, il s'agit de rechercher le couple (objet, position) dans la nouvelle image se trouvant à une distance minimale du couple de référence dans l'image précédente (Figure 1.2). Seule la définition de la fonction de distance, qui dépend des différents paramètres caractéristiques de l'élément structurel, va changer selon la méthode de mise en correspondance.

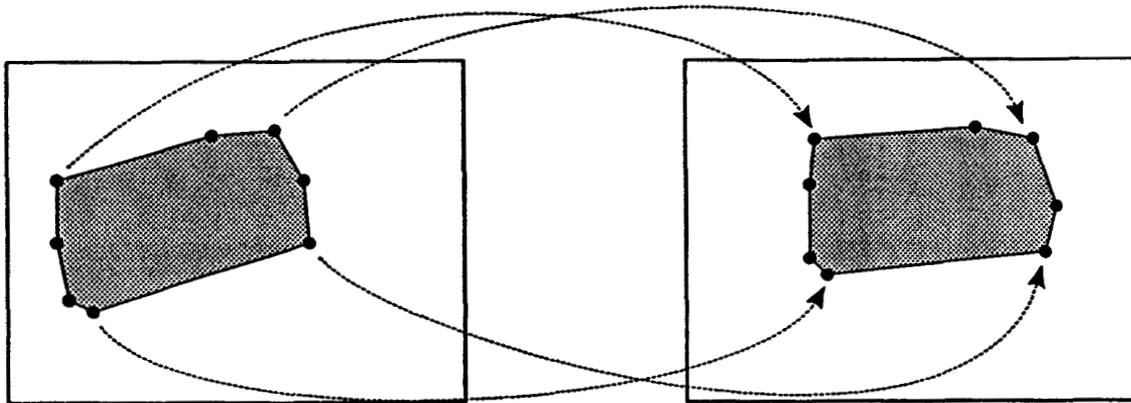


Figure 1.2 : Mise en correspondance

Si on essaye de simuler le système visuel humain [Ull78, Ull79, Ull81], la fonction de distance se résume souvent à une simple fonction métrique :

- Dans le cas de mise en correspondance de points d'intérêt isolés, de caractéristiques identiques, la distance choisie est la distance spatiale Euclidienne.
- Dans le cas de segments, il convient d'ajouter à la distance Euclidienne une certaine contribution reflétant les longueurs des segments comparés, et les orientations.

Quand les éléments structurels choisis sont plus complexes, un plus grand nombre de paramètres doit être pris en compte, d'où une plus grande complexité de la fonction distance [Roa79, Jac80]. Dans certains cas, il s'avère même impossible d'exprimer la fonction distance de manière explicite [Lev83].

D'autres approches ont été proposées, souvent fondées sur un critère de mise en correspondance optimale en fonction de l'élément structurel. On peut citer Gennery et Yachida qui utilisent l'intercorrélation [Gen79, Yac78], Jacobius qui explore un graphe [Jac80], ainsi que différentes méthodes de relaxation et de recherche par approximations successives [Ros76, Bar80, Dre83].

Des méthodes de recherche faisant intervenir des principes d'analyse d'images à différents niveaux de résolution sont exposées dans [Gen77] et [Tan78]. Une approche de ce genre permet souvent de diminuer la complexité du processus de mise en correspondance.

Une description plus synthétique des différentes méthodes de mise en correspondance est donnée dans [Agg81]. Une présentation de certains algorithmes permettant de mettre en évidence le mouvement d'un objet à partir d'éléments structurels situés dans une succession d'images (plus de deux) est disponible dans [Sha90].

1.2.3) AVANTAGES ET INCONVENIENTS

Quand les objets observés possèdent des caractéristiques facilement repérables dans l'image, les méthodes structurelles sont celles permettant d'obtenir la plus grande précision dans l'estimation du mouvement. Elles autorisent en général des déplacements importants des mobiles d'une image à la suivante sans pour cela entraîner une diminution de la fiabilité de la détection. De plus, seules les méthodes structurelles permettent en général d'appréhender le cas d'objets mobiles déformables.

La plus importante limitation des méthodes structurelles découle du fait qu'il est nécessaire de disposer a priori d'informations concernant les mobiles susceptibles d'être observés dans la scène. Les éléments structurels utilisés afin de localiser les objets doivent en effet dépendre directement de leurs caractéristiques visuelles (forme, texture, couleur ...). La fiabilité de la mise en correspondance est en outre directement liée à la complexité des éléments structurels : plus l'information a priori concernant les objets est importante, plus l'estimation de leur mouvement s'avère aisée.

I.3) ANALYSE DU FLUX OPTIQUE (NIVEAU INTERMEDIAIRE)

Comme nous l'avons dit précédemment, la détection du mouvement peut aussi être considérée en tant que moyen d'analyse du contenu informationnel d'une séquence d'images. L'étude des variations locales des caractéristiques des pixels (information brute) peut permettre, dans certains cas, le regroupement de points en classes possédant des caractéristiques similaires vis à vis du mouvement.

Dans un premier temps, il s'agit d'associer à chaque pixel d'une image de la séquence, un vecteur déplacement permettant de le situer dans l'image suivante. Le champ de vecteurs ainsi obtenu est appelé champ des déplacements (Figure 1.3). En divisant chacun de ces vecteurs par l'intervalle de temps séparant deux acquisitions d'images successives, on obtient un nouveau champ de vecteurs représentant cette fois la vitesse de chaque point : c'est le flux optique.

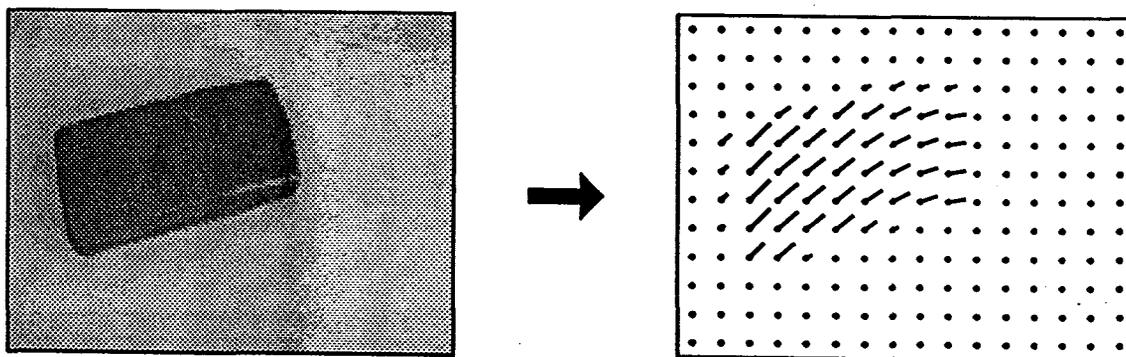


Figure 1.3 : Champ des déplacements

Bien que riche en informations, le flux optique est difficilement interprétable de façon directe. Il s'avère nécessaire de poursuivre l'analyse en segmentant certaines régions de l'image en classes de points sensés appartenir à un même objet dans la scène observée. Durant cette phase de classification, divers paramètres concernant l'objet peuvent être déterminés, selon l'application recherchée : sa forme ou sa structure, son mouvement réel en deux ou trois dimensions, etc...

I.3.1) DETERMINATION DU FLUX OPTIQUE

I.3.1.a) Corrélation

Il existe une façon intuitive de poser le problème de la détermination du champ des déplacements. Etant donné un point d'une image, la détermination du vecteur déplacement qui lui est associé revient à rechercher sa nouvelle position dans l'image suivante. Pour chaque point, plusieurs candidats sont disponibles, et un critère de choix doit être utilisé pour sélectionner le meilleur.

Cette sélection peut être réalisée par des méthodes de corrélation [Lee70, Lee71, Smi72, Sny80, Ber80, Riv86]. Une fenêtre de la première image est corrélée avec la seconde image. Un pic trouvé dans la fonction d'intercorrélation correspond à la valeur de déplacement la plus probable. Divers paramètres peuvent intervenir dans le choix de la taille de la fenêtre initiale, selon les changements d'échelle ou d'orientation des objets observés entre deux images successives, ou les variations des conditions d'observation. Dans certains cas, par utilisation d'une fonction modifiée (corrélation centrée ou filtrée), on peut diminuer la sensibilité au bruit de la méthode.

Le calcul des vecteurs déplacement étant réalisé sur certaines fenêtres, il s'avère nécessaire d'extrapoler les résultats afin d'obtenir un vecteur en tout point de l'image [Dav79, Bou86].

I.3.1.b) Gradient spatio-temporel

Un pixel d'une image numérique constitue une représentation d'un certain nombre de caractéristiques d'un élément de surface de la scène observée, caractéristiques qui peuvent varier selon le système d'acquisition utilisé. Le plus souvent, la valeur du niveau de gris d'un pixel représente une mesure de la quantité de lumière réfléchiée par un élément de surface de l'objet observé. Dans certains cas, on peut disposer d'informations supplémentaires telles que la couleur (différentes bandes spectrales d'observation) ou la distance entre l'objet et l'observateur (télémétrie).

Lorsqu'un élément de surface d'un objet se déplace dans la scène observée, le pixel de position (x,y) dans l'image qui le caractérise à l'instant t , se déplace lui aussi à une nouvelle position $(x+dx,y+dy)$ à l'instant $t+dt$. Si l'on suppose qu'il est possible de construire une fonction $f(x,y,t)$, définie en tout point (x,y) de l'image, et à tout instant t de la séquence, qui soit indépendante des conditions d'observations, c'est à dire ne dépendant que des caractéristiques intrinsèques des objets observés, alors cette fonction doit vérifier la contrainte :

$$(1) \quad f(x,y,t) = f(x+dx,y+dy,t+dt)$$

Si la fonction f est continue et dérivable en tout point, on peut alors obtenir par développement au premier ordre de (1) une équation différentielle de contrainte [Sch86b] reliant les composantes (u,v) du vecteur vitesse au point (x,y,t) aux dérivées partielles de la fonction f :

$$(2) \quad df/dx \cdot u + df/dy \cdot v + df/dt = 0$$

Souvent, la fonction f apparaissant dans l'équation différentielle de contrainte représente directement l'intensité de l'image (niveau de gris). Cependant, cela sous entend que l'illumination de la scène doit être uniforme et constante dans le temps. D'autres auteurs proposent d'utiliser des fonctions plus complexes (couleur, opérateurs spatiaux) afin de limiter l'influence des conditions d'observation sur le résultat final [Woh83, Mit84, Mit87]. Ma, Lu et Wu proposent d'utiliser directement la réflectance des objets observés afin d'autoriser des variations spatiales ou temporelles de l'intensité lumineuse [Ma87].

Afin de déterminer le vecteur vitesse (u,v) en tout point de l'image, il est nécessaire de disposer d'au moins une autre équation de contrainte [Sch86a]. Une méthode de résolution consiste à ajouter des conditions supplémentaires concernant le vecteur vitesse : continuité, invariance dans un voisinage, direction constante [Sny80, Spo83].

La résolution numérique des équations de contrainte, si elle est menée sans précaution particulière, conduit à des résultats fortement entachés d'erreurs, principalement à cause du bruit présent dans l'image initiale et du nombre limité de valeurs servant à quantifier les fonctions image. Des méthodes d'optimisation, tenant compte du voisinage de chaque point [Sny80], ou de critères de continuité plus globaux [Hor81, Ais89] permettent de s'affranchir de ce genre de problème.

I.3.1.c) Transformation de Fourier

En analysant la phase de la transformée de Fourier de chaque image de la séquence, il est possible de déterminer un terme de déplacement [Hua81, Has74].

Malgré les progrès réalisés dans le domaine du calcul rapide par utilisation de circuits spécialisés, les calculs nécessaires à la détermination du flux optique par cette approche [Sny83, Raj83] restent coûteux en temps de calcul.

I.3.2) SEGMENTATION

I.3.2.a) Analyse directe du flux optique

La méthode de segmentation la plus simple consiste en un simple seuillage de l'amplitude de chaque vecteur du flux optique, permettant d'effectuer la distinction entre régions fixes et régions mobiles (Figure 1.4). Cependant, il n'est pas possible, en général, de considérer qu'il existe une relation directe entre les "zones" détectées et les "objets" mobiles. En effet, en cas d'occlusion partielle d'un objet par un autre, cette méthode ne distinguera qu'une seule zone en mouvement, même si les directions et sens de déplacement des deux mobiles ne sont pas identiques.

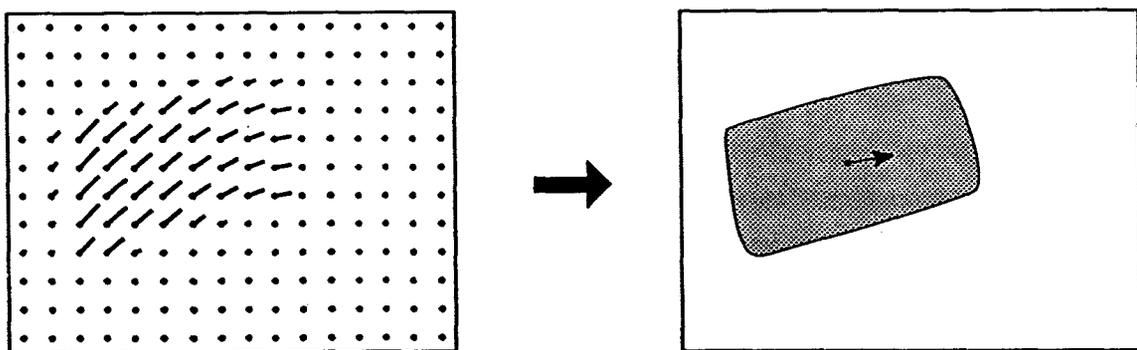


Figure 1.4 : Seuillage simple du flux optique

En combinant l'analyse du champ de déplacement avec d'autres informations disponibles a priori concernant les objets susceptibles d'être présents dans l'image, il est possible d'obtenir une segmentation plus précise de la scène observée [Yac83, Kap83, Man87].

Si la seule information disponible a priori concernant les objets est l'hypothèse de rigidité, il est possible de regrouper en classes les zones connexes où tous les points sont animés de la même vitesse [Jai79c, Tho80, Sny80]. Dans le cas d'objets non déformables dont le mouvement reste parallèle au plan image traité, les effets de perspective sont négligeables, et cette hypothèse est souvent vérifiée.

I.3.2.b) Analyse de la transformée de Hough

La recherche des différents mouvements présents dans la séquence d'images analysée peut se faire par transformation de Hough du champ des déplacements. Si un point d'accumulation est détecté dans la transformée, on peut supposer qu'il correspond à la transformation d'un ensemble de points connexes affectés d'un même mouvement (Figure 1.5). En supposant vérifiée l'hypothèse de rigidité, on détermine alors facilement l'objet correspondant.

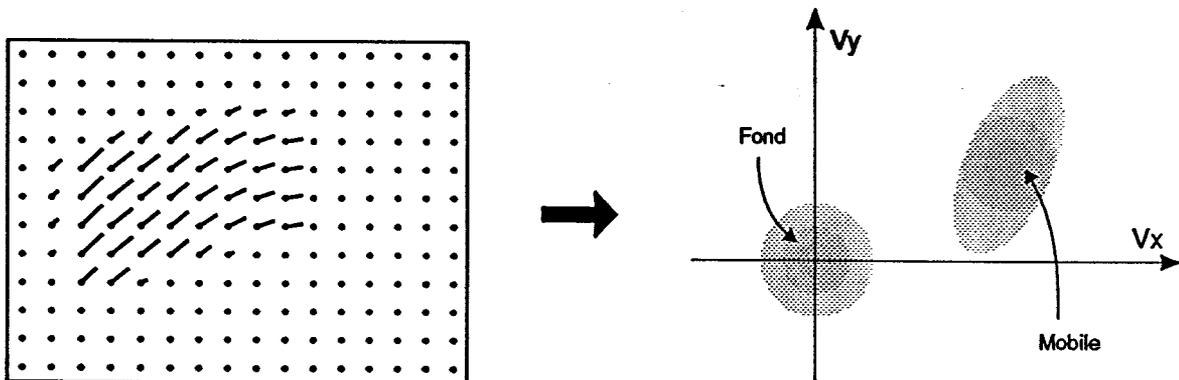


Figure 1.5 : Transformée de Hough du champ des déplacements

La détection des points d'accumulation dans l'espace de Hough peut se réaliser de façon assez efficace en employant des techniques de classification automatique non supervisée [Oro81, Sny83], ou par mise en évidence de formes fortes par utilisation du principe des nuées dynamiques [Did82, Fac87].

I.3.3) ESTIMATION DE PARAMETRES

L'analyse du flux optique, durant ou après la phase de segmentation, peut permettre de déterminer des caractéristiques concernant les objets mobiles présents dans la scène observée. De nombreuses recherches ont été menées dans ce sens, portant non seulement sur l'estimation du mouvement des objets, mais aussi concernant la possibilité d'analyser leur forme ou leur structure d'après le champ des déplacements.

I.3.3.a) Estimation du mouvement

Après segmentation du flux optique, une analyse du champ de vecteurs restreinte aux pixels supposés représenter un objet unique permet de caractériser simplement son mouvement dans le plan image [Hor81, Pra81, Adi83, Sny83]. Les paramètres déduits, qui caractérisent à la fois la translation (orientation, amplitude) et la rotation (centre, angle) de l'objet, permettent parfaitement de décrire son mouvement 2D dans le plan image.

Cependant, le mouvement réel d'un l'objet dans l'espace 3D, introduit inévitablement des composantes selon la troisième dimension. Des travaux ont été menés afin de déterminer des algorithmes de mise en évidence du mouvement 3D des objets par analyse du flux optique [Nag79, Nag81a, Nag81b, Pra81b, Tsa82, Fan83, Tsa83, Fan84, Tsa84].

Les équations liant le mouvement 3D d'un objet au champ des déplacements qu'il induit sur l'image sont essentiellement basées sur des transformations géométriques (projections avec ou sans perspective). Cependant, elles comportent des termes non linéaires qui rendent leur résolution relativement complexe. De plus, ces non linéarités entraînent souvent une grande sensibilité au bruit de mesure qui entache inévitablement le flux optique.

Dans le cas d'objets indéformables, c'est à dire sous l'hypothèse de rigidité, la résolution des équations caractérisant le mouvement peut se faire en utilisant des informations concernant un nombre limité de points d'intérêt. Le nombre de points nécessaire à cette estimation du mouvement varie selon le nombre d'images utilisées [Fan84, Mit85, Sha86, Sha90].

En imposant des hypothèses restrictives sur le type de mouvement recherché, on peut éliminer le problème de la résolution d'équations non linéaires par des méthodes itératives qui s'avèrent souvent instables. Il s'agit des hypothèses de translation pure [Cle80, Lee80], de petites rotations [Fan84, Adi85, Med85], ou de décomposition possible en mouvements simples [Bal83].

I.3.3.b) Reconstruction de formes 3D

Il est possible de mettre en évidence l'information de profondeur caractérisant la position des objets dans l'espace à trois dimensions en analysant leurs projections dans différents plans image. Les méthodes de stéréovision utilisent des paires d'images acquises par deux caméras fixes dont les axes optiques sont séparés d'une certaine distance.

D'un point de vue formel, il est assez facile d'assimiler le champ des déplacements obtenu par analyse de deux images successives d'une séquence, au champ des disparités obtenu par analyse d'une paire d'images stéréoscopiques acquises par des caméras fixes.

Sous certaines conditions, il s'avère alors possible de mettre en évidence différentes caractéristiques concernant les objets observés en analysant le flux optique déduit de la séquence. On détermine alors un système d'équations liant les caractéristiques structurelles d'un objet, ainsi que son mouvement, au flux optique [Web81, Web82, Bal83, Agg85, Mit86, Sha90]. La résolution de ce système est réalisée par des méthodes similaires à celles utilisées en reconstitution du mouvement.

I.3.4) AVANTAGES ET INCONVENIENTS

La limitation principale des méthodes de mise en évidence du mouvement basées sur l'évaluation du flux optique est leur forte sensibilité aux bruits de mesure et aux variations pouvant apparaître dans les conditions d'acquisition. Le calcul direct du flux optique à partir des variations locales des caractéristiques de l'image est toujours réalisé par des algorithmes utilisant des estimations itératives de dérivées

partielles spatiales et temporelles des signaux : ces méthodes s'avèrent souvent instables quand l'image est bruitée.

Le problème de la sensibilité au bruit est en fait inhérent à la méthode : l'évaluation du mouvement instantané d'un objet ne peut se faire que si les déplacements entre deux images successives restent très limités. Dans ce cas, l'estimation des dérivées partielles de l'image par des différences finies se réalise sur des valeurs très voisines du niveau de gris des pixels, d'où une grande sensibilité aux perturbations locales dues au bruit. Certaines approches ont été proposées afin de diminuer la sensibilité au bruit des méthodes utilisant le gradient spatio-temporel [Fen79, Yac83, Ais89], mais elles entraînent souvent une perte de précision concernant les mouvements détectés.

Toutefois, comme nous l'avons vu dans les paragraphes précédents, le flux optique est très riche en informations, et peut permettre d'obtenir une multitude de caractéristiques concernant les objets observés. A l'heure actuelle, les chercheurs travaillent sur la mise au point de méthodes permettant à la fois de diminuer la complexité algorithmique des phases d'évaluation du flux optique et d'augmenter le nombre d'informations fournies lors de la phase de segmentation.

I.4) METHODES ORIENTEES MOUVEMENT (BAS NIVEAU)

Les méthodes orientées mouvement, dites de bas niveau, reposent sur des traitements ponctuels. Leur principe consiste souvent en l'élimination des zones de l'image restant immobiles durant la séquence. On peut donc considérer cette opération comme un pré-traitement servant à diminuer la quantité d'information restant à interpréter. La détermination des paramètres régissant le mouvement des objets n'est pas réalisée durant cette opération qui ne permet de mettre en évidence que la présence ou l'absence de mouvement.

Les méthodes orientées mouvement sont presque toujours basées sur l'exploitation des différences entre images. Toutefois, la détection du mouvement peut se faire selon deux approches différentes : certains algorithmes fournissent une image résultat contenant des régions représentatives des objets mobiles, d'autres permettent d'en obtenir les contours.

I.4.1) EXTRACTION DES REGIONS MOBILES

L'extraction des régions mobiles repose sur une opération de soustraction pixel à pixel entre deux images. En supposant que l'illumination de la scène ne s'est pas modifiée durant l'intervalle de temps ayant séparé les deux acquisitions, les parties de la scène observée restées immobiles par rapport à la caméra durant cette période se trouvent alors projetées de façon identique dans le plan image : après soustraction des deux images, les zones correspondantes se trouvent effacées.

Il est possible de choisir les images opérandes de deux façons différentes, ce qui fournit deux variantes de la méthode :

- Si l'une des deux images est acquise à un instant où aucun objet mobile n'est présent dans la scène observée, la différence entre cette image et une image de la séquence contenant un ou plusieurs objets en mouvement fournit directement une localisation de ces derniers. L'image correspondant au fond est appelée image de référence.
- Si les images opérandes correspondent à deux vues successives de la séquence, le résultat de la soustraction pixel à pixel fournit une image où tout objet mobile

donne naissance, en général, à plusieurs régions différentes.

1.4.1.a) Utilisation d'une image de référence

L'image obtenue en réalisant une différence pixel à pixel entre une image contenant des objets mobiles et une image de référence comporte deux types de régions (Figure 1.6) :

- Les régions de niveaux de gris faibles correspondent aux parties de la scène non affectées par le mouvement.
- Les régions de niveaux de gris élevés correspondent aux objets mobiles détectés.

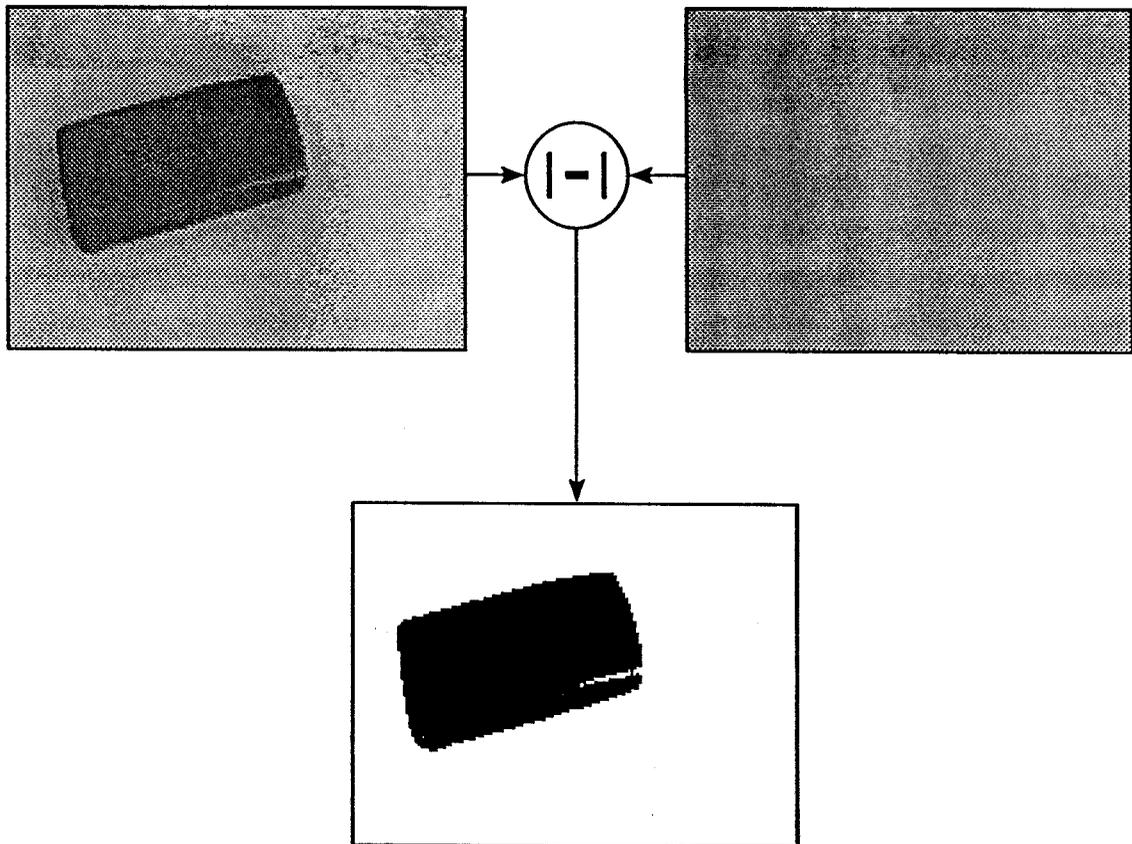


Figure 1.6 : Détection de mouvement par différence avec une image de référence

En général, la distinction entre les deux classes de pixels ne peut pas se faire de façon simple. En effet, dans les zones stationnaires de l'image, il existe toujours une faible variation du niveau de gris des pixels, pouvant être due soit au bruit présent dans les images, soit à de légères modifications des conditions d'illumination de la scène entre les deux prises de vue.

Quand la variation de niveau de gris des pixels due au bruit reste limitée, la segmentation de l'image différence en deux classes de points peut se faire par simple binarisation. La recherche du seuil optimal s'effectue par analyse de l'histogramme des niveaux de gris de l'image, en supposant qu'il comprend au moins un mode prépondérant correspondant au pixels immobiles.

Cette hypothèse ne peut plus se justifier quand les différences de niveaux de gris entre les deux images sont dues à une variation de l'éclairage de la scène entre les deux acquisitions. Dans ce cas, il est nécessaire d'actualiser l'image de référence, en utilisant les informations contenues dans l'image différence [Aok84].

Dans certaines applications, si la scène observée est complexe et le nombre d'objets en mouvement important, il est très difficile d'acquérir une image de référence exempte d'éléments mobiles. Différents auteurs se sont penchés sur le problème de la détermination d'une image de référence à partir des images successives de la séquence :

- Dreschler et Nagel [Dre78], proposent une méthode basée sur l'utilisation de deux images de la séquence pour lesquelles les zones correspondant aux objets mobiles ne se chevauchent pas. L'image de référence est reconstruite en utilisant des critères de sélection de zones de l'image selon un critère calculé sur leurs contours.
- Tang et Snyder [Tan82], utilisent des images dans lesquelles les mouvements restent faibles. Les zones de l'image initiale dans lesquelles des pixels en mouvement sont détectés sont circonscrites par des rectangles. Les pixels situés à l'extérieur de ces rectangles sont utilisés afin de reconstituer une image de référence.
- Jain [Jai84], propose un algorithme basé sur l'analyse du signe de l'image différence. En analysant l'évolution de deux images construites par accumulation des pixels positifs et négatifs des images différence obtenues à partir de la séquence, l'auteur montre qu'il est possible de reconstruire une image du fond.

I.4.1.b) Utilisation d'images successives

Comme nous l'avons précisé auparavant, il est aussi possible de réaliser une détection de mouvement en analysant les différences existant entre deux images successives d'une séquence d'images. Un objet mobile sera constitué de plusieurs régions distinctes dans l'image différence (Figure 1.7) :

- Une zone située à l'avant du mobile dans le sens du déplacement, résultant de la différence des niveaux de gris entre l'objet et le fond qu'il a recouvert.
- Une zone située à l'arrière du mobile, correspondant à la partie du fond de la scène découverte durant le déplacement.
- Une zone de niveaux de gris faibles due à l'invariance du fond entre les deux acquisitions.
- Une zone de niveaux de gris faibles correspondant aux parties du mobile communes aux deux acquisitions en cas de recouvrement.

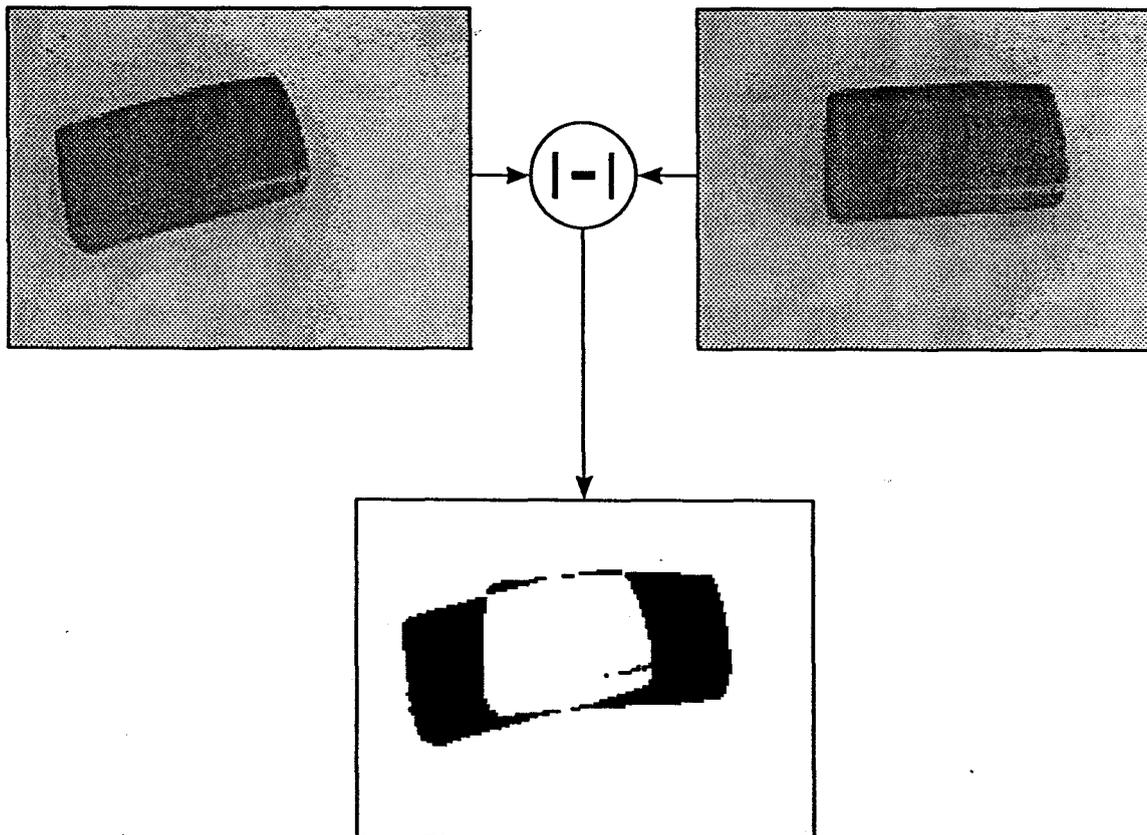


Figure 1.7 : Détection de mouvement par différence entre deux images successives

L'interprétation de ces différentes régions de l'image constitue en général un

problème complexe, les mobiles ne pouvant pas être localisés par simple binarisation de l'image différence. Divers auteurs ont cependant présenté des méthodes d'analyse de l'image différence :

- Jain a présenté un approche basée sur le regroupement des régions en classes dépendant du type de zone détecté [Jai81a].
- Yalamanchi propose une méthode de segmentation basée sur l'homogénéité et la connexité des différentes régions de l'image différence [Yal80, Yal82].
- Yachida utilise le signe de l'image différence et des connaissances a priori sur le fond et l'objet afin de déterminer le sens de déplacement des mobiles [Yac78].
- Par utilisation de plusieurs images successives, Lenz montre qu'il est possible de simplifier le processus de localisation des mobiles [Lenz83].

1.4.2) AVANTAGES ET INCONVENIENTS

Le principal avantage des méthodes de détection de mouvement basées sur la différence d'images réside dans leur simplicité de mise en oeuvre. Les traitements permettant la mise en évidence du mouvement sont appliqués à tous les points de l'image, indépendamment de leur position. Cela autorise, en général, l'utilisation d'architectures de calcul assez simples et peu coûteuses. Cette simplicité repose sur l'invariance du fond des images, ce qui limite l'utilisation des méthodes orientées mouvement aux applications où la caméra est fixe par rapport à la scène observée.

Cependant, quand les objets mobiles présents dans une séquence d'images ne sont pas constitués de zones de niveau de gris homogène, il est difficile de segmenter globalement l'image différence, comme le font les méthodes basées sur l'analyse de régions.

Plutôt que de définir la position d'un objet mobile dans l'image par le groupe des pixels qui le constituent, il suffit de disposer de son contour extérieur. Divers auteurs ont proposé des méthodes de mise en évidence du mouvement par extraction des contours mobiles de la séquence d'images.

Une présentation plus précise des différentes méthodes de détection de

mouvement basées sur l'extraction des contours mobiles, accompagnée d'une étude comparative, fait l'objet du chapitre suivant de ce mémoire.

I.5) CONCLUSION ET CHOIX D'UNE METHODE

Aucune des méthodes présentées auparavant ne s'avère utilisable en toute situation. Le choix d'une approche particulière de détection de mouvement dans une séquence d'images ne peut se faire qu'en analysant les spécificités de l'application.

Des études précédentes [Vie88, Ste85] ont montré que les méthodes orientées mouvement sont bien adaptées au problème qui nous intéresse, et ce pour plusieurs raisons :

- Les contraintes de sécurité inhérentes au processus de conduite automatique du véhicule guidé exigent une détection des obstacles présents sur tout le carrefour en un temps minimum. Cet argument nous conduit à rejeter les méthodes basées sur l'interprétation du champ des déplacements qui s'avèrent toujours très longues quand le calcul doit être mené en tous les points de l'image. De plus, la détermination fiable du flux optique nécessite une cadence d'acquisition des images incompatible avec leur traitement en temps réel.
- Le nombre et le type des objets présents dans la scène observée n'est pas connu a priori. En effet, les mobiles devant être détectés correspondent à tous les véhicules susceptibles d'emprunter le carrefour, ainsi qu'aux piétons ou aux animaux. Cet argument conduit à rejeter les méthodes structurelles nécessitant une connaissance a priori non négligeable concernant les objets en mouvement.
- La stratégie de prise de vue proposée par M^{TS} Stelmaszyk et Vieren, qui utilise une caméra située à la verticale du carrefour, est tout à fait compatible avec les méthodes orientées mouvement : la caméra est fixe par rapport au carrefour et le mouvement à détecter ne s'effectue que dans deux dimensions [Ste85, Vie88].

La description de la méthode utilisée afin de détecter le mouvement, ainsi qu'une justification de sa fiabilité, font l'objet du chapitre suivant.

CHAPITRE II

EVALUATION DES METHODES DE DETECTION DE CONTOURS EN MOUVEMENT

Nous avons vu dans le chapitre précédent qu'il est possible de mettre en évidence le mouvement d'objets dans une séquence d'images par utilisation de la différence pixel à pixel d'images successives de la séquence.

Certaines approches, basées sur l'analyse directe de ces images différence, permettent la détermination des régions affectées par le mouvement d'objets mobiles. Les algorithmes correspondants réalisent une segmentation de l'image différence en utilisant des critères d'homogénéité des niveaux de gris pour regrouper les pixels constituant une même région. Cette approche n'est en général pas utilisable dans le cas de scènes réelles pouvant contenir des objets mobiles de niveaux de gris non homogènes évoluant sur un fond quelconque, présentant également des variations spatiales de niveau de gris.

Afin de s'affranchir de cette limitation, certains auteurs ont proposé de rechercher les contours des objets en mouvement plutôt que les objets eux mêmes. Dans cette partie du mémoire sont présentés différents opérateurs basés sur ce principe de détection de contours mobiles. Nous nous proposons de comparer les différentes méthodes existantes afin d'évaluer leurs performances.

II.1) DETECTION DES CONTOURS EN MOUVEMENT

II.1.1) GENERALITES

Les opérateurs de détection des contours des objets en mouvement présentés par la suite font partie des méthodes orientées mouvement : les opérandes sont des images numériques successives extraites de la séquence et le résultat est lui aussi une image numérique. Les notations utilisées sont les suivantes :

C désigne l'image courante de la séquence, c'est à dire celle par rapport à laquelle les objets détectés sont positionnés.

P est l'image précédente, acquise juste avant l'image C dans la séquence.

S est l'image suivant immédiatement l'image courante.

G(I) est l'image obtenue par application d'un opérateur de détection de contours sur l'image I. Cette notation symbolise une fonctionnalité et aucun opérateur particulier n'est associé à ce formalisme.

D(I,J) représente l'image obtenue en prenant la valeur absolue de la différence des niveaux de gris pixel à pixel entre deux images I et J.

II.1.2) OPERATEUR DE HAYNES ET JAIN

L'opérateur proposé par Haynes et Jain en 1983 utilise deux images successives de la séquence afin de déterminer les contours des objets mobiles [Hay83]. Une détection des contours spatiaux des objets est effectuée sur l'image courante. Chaque pixel de l'image des contours est combiné avec le pixel situé à la même position dans l'image différence en utilisant une opération de multiplication. L'opérateur de détection, noté CM_J , peut donc être décrit par :

$$CM_J(C,P) = D(C,P) \times G(C)$$

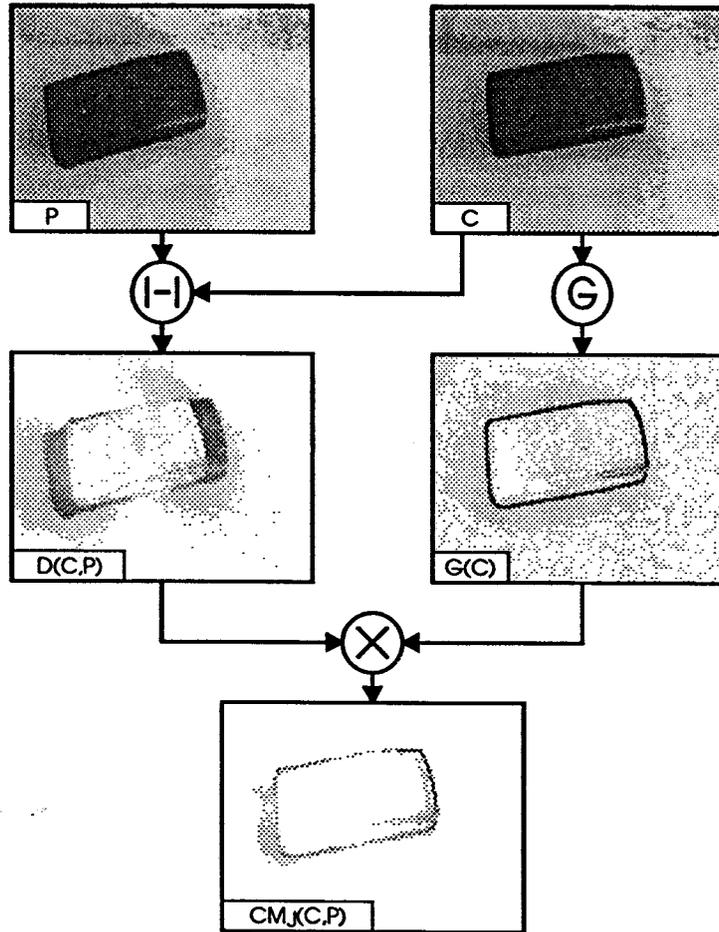


Figure 2.1 : Opérateur de Haynes et Jain

Dans l'image $CM_J(C,P)$, seuls sont détectés les contours correspondant à des objets s'étant déplacés durant l'intervalle de temps ayant séparé l'acquisition des deux images (cf. figure 2.1). En effet, le niveau de gris d'un pixel de l'image résultante ne peut avoir une valeur élevée que si le pixel de l'image de contour $G(C)$ et le pixel correspondant de l'image différence $D(C,P)$ ont simultanément des valeurs élevées, donc quand ce pixel appartient à un contour mobile.

II.1.3) OPERATEUR DE STELMASZYK

Dans son principe, l'opérateur proposé par Stelmazyk en 1985 est similaire à celui décrit précédemment [Ste85a, Ste85b, Pos86]. Un opérateur de détection de contour est combiné pixel à pixel avec un opérateur de détection de mouvement grâce à une opération de multiplication. Cependant, plutôt que d'utiliser directement l'image différence comme détecteur de mouvement, l'auteur propose de déterminer les contours présents dans cette image. L'opérateur, noté CM_S , peut donc s'exprimer de la façon suivante :

$$CM_S(C,P) = G(D(C,P)) \times G(C)$$

L'auteur précise que le même opérateur de détection de contour $G(\cdot)$ doit être utilisé sur les deux images, C et $D(C,P)$. La figure 2.2 illustre le principe de cet opérateur.

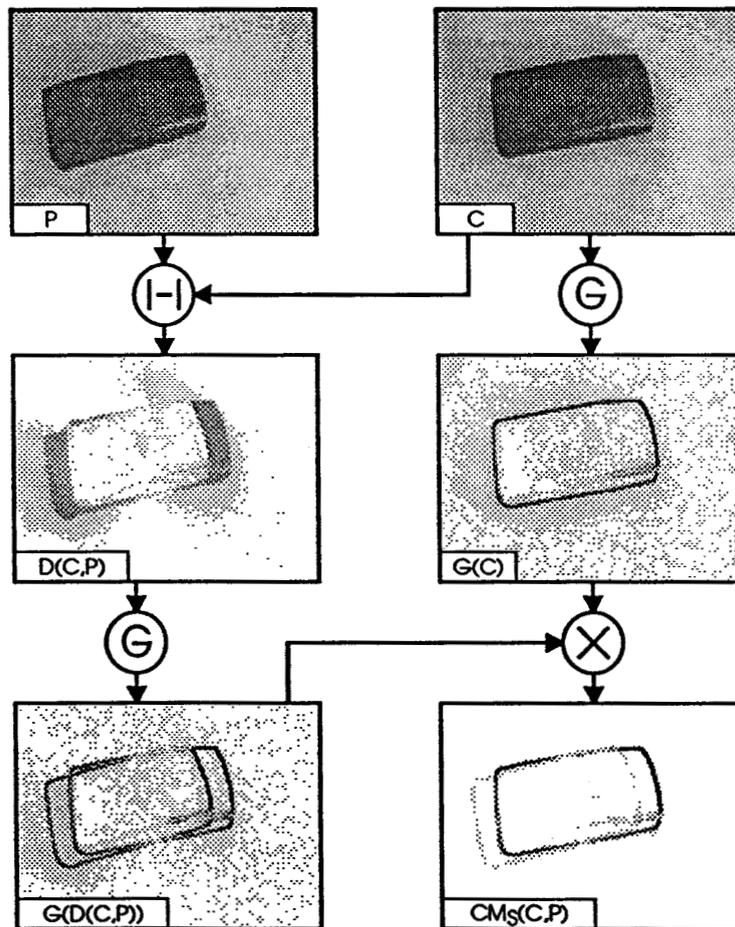


Figure 2.2 : Opérateur de Stelmazyk

II.1.4) OPERATEUR DE VIEREN

L'opérateur proposé par Vieren en 1988 correspond à une extension de l'algorithme précédent par utilisation de trois images successives [Vie87, Vie88, Vie89]. L'opération de détection de mouvement par extraction des contours présents dans l'image différence est réalisée sur les paires d'images (Courante,Précédente) et (Suivante,Courante). En multipliant les deux images résultantes, $D(C,P)$ et $D(S,C)$, on isole les contours des objets mobiles de l'image courante qui sont présents simultanément dans les deux images. L'opérateur, noté CM_V , s'exprime par :

$$CM_V(S,C,P) = G(D(C,P)) \times G(D(S,C))$$

Comme on peut le voir sur la figure 2.3, la position des contours d'un objet mobile détecté par l'algorithme correspond à sa position dans l'image courante. Cette méthode nécessite la connaissance de l'image suivante de la séquence et apporte donc un retard d'une image au niveau du résultat.

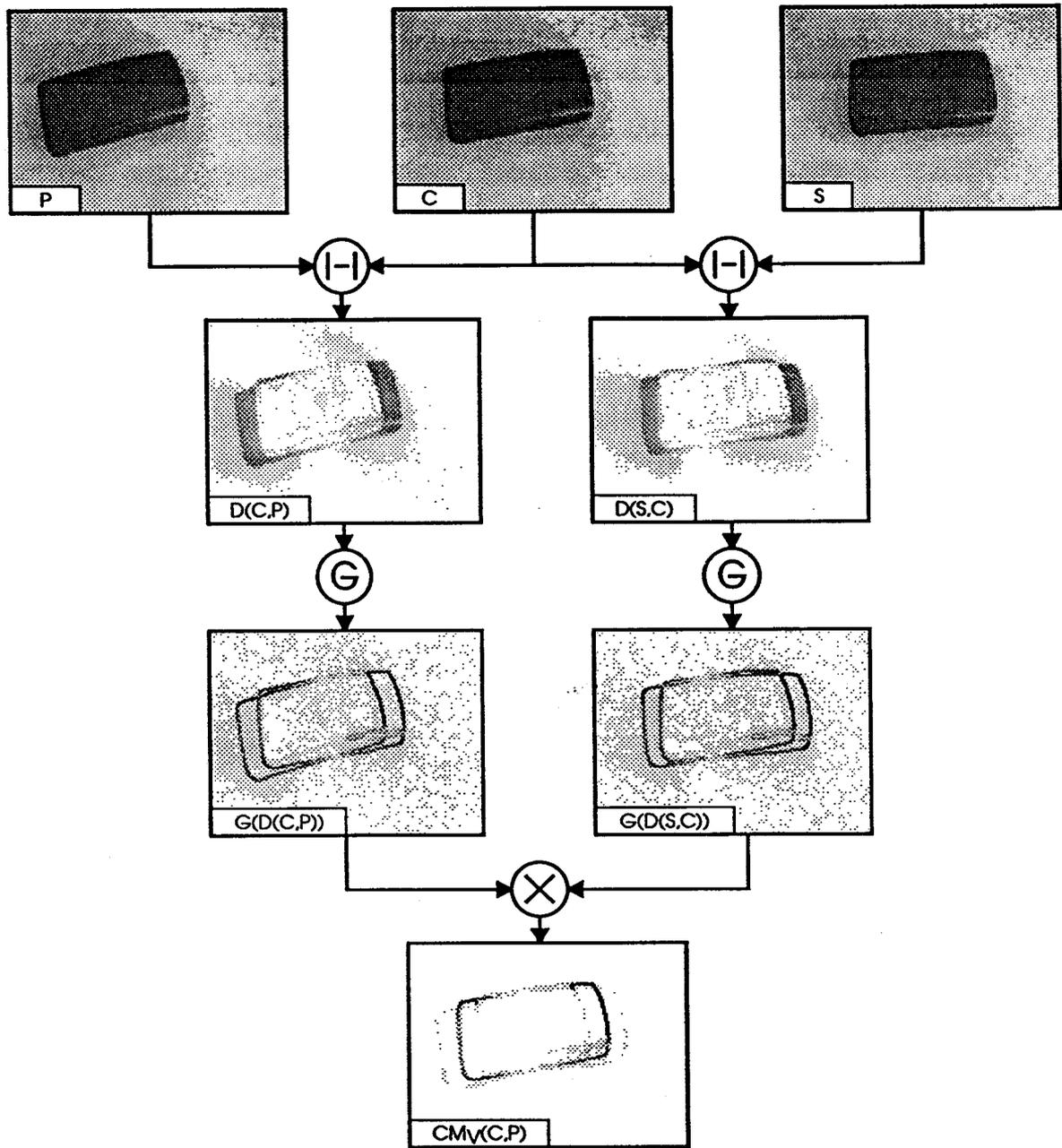


Figure 2.3 : Opérateur de Vieren

II.1.5) SPECIFICITES DES OPERATEURS DE DETECTION DE CONTOURS EN MOUVEMENT

L'opérateur de Haynes et Jain est le premier apparu dans la littérature utilisant la multiplication point à point d'images en niveaux de gris afin de détecter les contours mobiles. Cet opérateur est devenu une référence en matière de détection de contours en mouvement.

L'opération de multiplication des niveaux de gris correspond en fait à une combinaison des valeurs de pixels sensiblement équivalente à un ET logique entre les points qui se correspondent d'une image à l'autre. L'avantage de cette opération est qu'il n'est pas nécessaire de binariser les images opérandes, opération qui s'avère souvent délicate.

L'algorithme de Stelmaszyk apporte une amélioration par rapport à la méthode de Haynes et Jain en ce qui concerne la localisation des contours détectés dans l'image. Les deux opérandes de l'opération de multiplication, à savoir le gradient de l'image courante $G(C)$, et le gradient de l'image différence $G(D(C,P))$, se comportent de façon similaire en présence d'une variation locale de niveau de gris dans une image, ce qui permet d'éliminer les erreurs de positionnement du contour mobile que l'on rencontre avec l'opérateur de Haynes et Jain. Cette amélioration, qui sera vérifiée de façon quantitative par la suite, est illustrée sur des images monodimensionnelles par la figure 2.4.

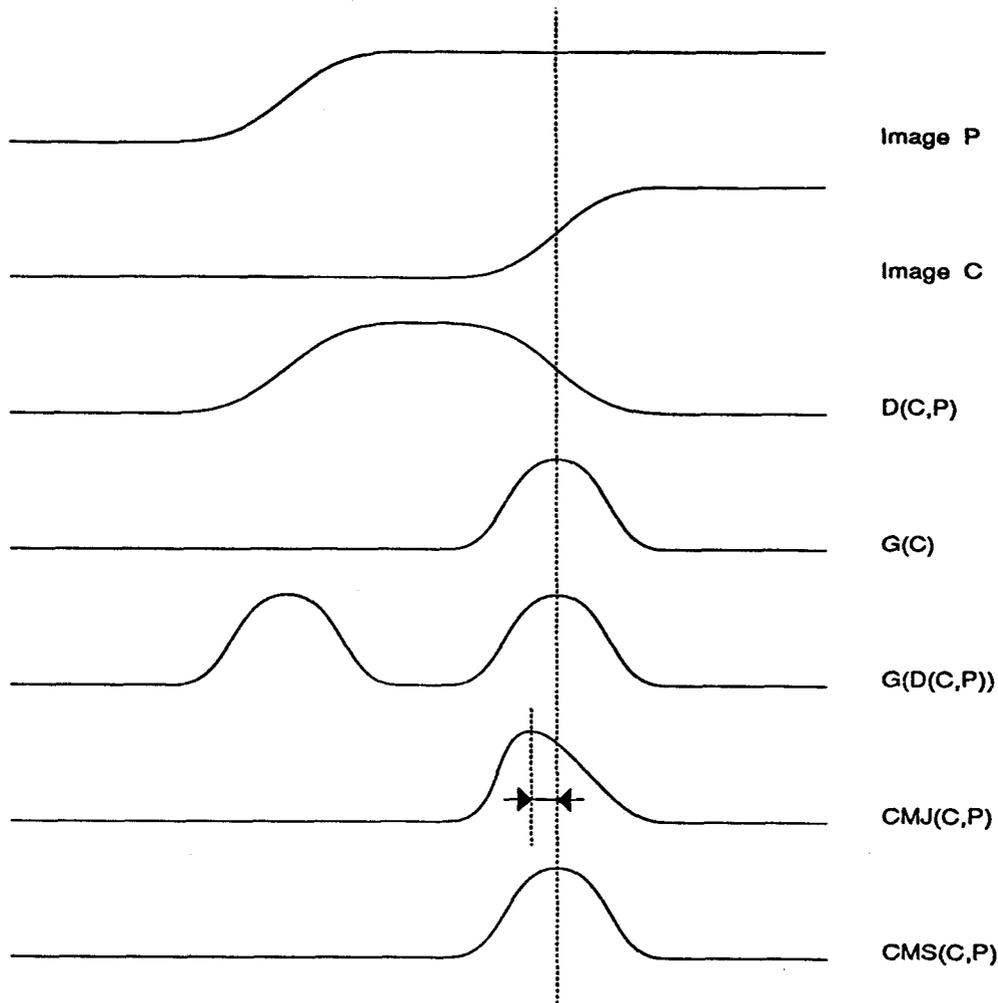


Figure 2.4 : Position du contour détecté

La procédure proposée par Vieren apporte une réelle amélioration par rapport aux opérateurs précédents quand les objets mobiles se déplacent sur un fond non uniforme. Les opérateurs de Haynes et Jain et de Stelmaszyk sont mis en défaut dès que le fond présente lui-même des contours, révélant des discontinuités de la fonction niveau de gris. En effet, par utilisation des informations contenues dans deux images successives de la séquence, il s'avère impossible de différencier un contour appartenant à un objet mobile, d'un contour du fond devenu visible après déplacement de ce même objet. L'utilisation d'une troisième image permet de lever cette ambiguïté (cf. figure 2.5).

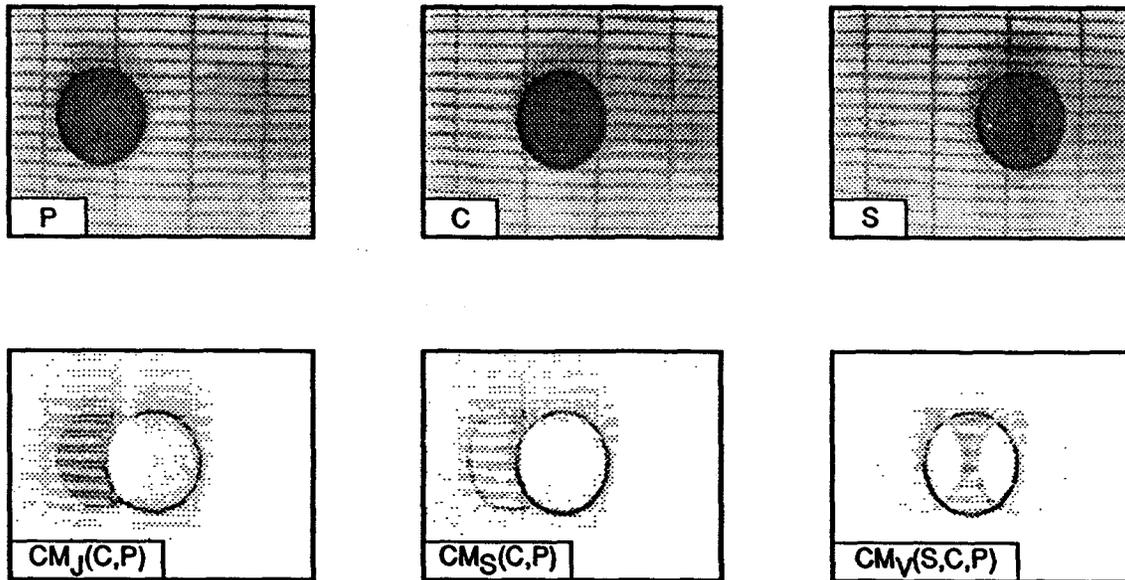


Figure 2.5 : Comparaison des détections sur fond non uniforme

Les différents points de comparaison exposés précédemment tendent à montrer la supériorité de la méthode de détection proposée par Vieren. Cependant, avant d'envisager la réalisation d'un processeur spécifique de segmentation d'images dynamiques, il convient de comparer ces différents opérateurs de façon plus quantitative.

Nous nous attachons à évaluer deux aspects fondamentaux vis à vis de l'analyse de séquences d'images, à savoir la précision du positionnement des contours détectés et la possibilité de segmenter simplement l'image contenant les contours mobiles.

Avant d'aborder l'analyse du comportement de ces opérateurs de détection de contours en mouvement, il convient de rappeler les principes d'évaluation des opérateurs de détection de contours statiques.

II.2) EVALUATION DES OPERATEURS DE DETECTION DE CONTOURS STATIQUES

II.2.1) GENERALITES

Il est assez difficile de définir précisément ce que l'on entend par contour d'un objet dans une image, cette notion faisant intervenir un grand nombre d'éléments subjectifs [Dav75]. Cependant, on suppose généralement qu'un contour peut être mis en évidence par une étude locale des variations de niveau de gris des pixels [Pre70, Hue73, Mor81, Nal86].

Partant de cette hypothèse, de nombreux auteurs ont proposé des procédés de détection et d'extraction des contours basés sur l'évaluation du gradient, ou d'une dérivée d'ordre supérieur, de la fonction image [Kir71, Rob77, Pra78, Mar79, Har81, Har84, Neu86]. La dérivée spatiale peut être estimée en chaque point de l'image numérique par utilisation de méthodes de différences finies, consistant en général à appliquer un ou plusieurs masques de convolution sur un voisinage du pixel où l'on désire évaluer cette dérivée. Le résultat de la détection effectuée selon ce principe est disponible sous la forme d'une image numérique où le niveau de gris de chaque pixel représente l'amplitude d'un contour en ce point dans la scène observée.

II.2.2) EVALUATION D'UN OPERATEUR DE DETECTION DE CONTOURS

Différentes recherches ont été menées concernant le problème de la détermination d'une mesure de l'efficacité d'un opérateur de détection de contours [Abd79, Bry79, Kit81, Pel82, Suc82, Can83, Neu86]. Certaines ont d'ailleurs abouti à la mise au point de méthodes de détection optimales selon la mesure considérée [Can86, Der87, Pet91].

La méthode d'évaluation la plus simple consiste en un contrôle visuel de la qualité des contours détectés, selon des critères de continuité ou de distorsion géométrique [Pel82]. Ce principe d'évaluation, utilisable sur des images réelles, reste

toutefois fondé sur des critères de comparaison très subjectifs.

Afin de mettre en oeuvre des mesures d'efficacité moins subjectives, il est souvent nécessaire d'avoir recours à des images de test synthétiques, contenant un ou plusieurs contours idéaux, de profils connus. La réponse de l'opérateur étudié peut alors être comparée à une détection idéale réalisée sur l'image de test.

La plupart des mesures rencontrées dans la littérature sont basées sur ce principe [Abd79, Kit81, Pel82, Suc82, Neu86]. En général, elles consistent à analyser l'évolution des performances des algorithmes de détection de contours sur des images bruitées en fonction du niveau de bruit.

Afin d'évaluer les performances des trois algorithmes de détection de contours mobiles présentés précédemment, nous nous proposons d'utiliser certaines de ces mesures pour juger de la qualité des contours fournis par ces opérateurs.

II.2.3) MESURES DE LA QUALITE D'UN OPERATEUR DE DETECTION DE CONTOURS

Nous avons retenu trois mesures pour juger de la qualité d'un opérateur de détection de contours parmi celles existant dans la littérature. Les deux premières privilégient les opérateurs donnant une bonne précision quant à la position du contour détecté par rapport au contour idéal. La troisième permet de vérifier la robustesse de l'opérateur par rapport au niveau de bruit présent dans l'image résultat.

II.2.3.a) Facteur de mérite selon Abdou et Pratt

Abdou et Pratt ont proposé une méthode d'évaluation utilisant un facteur de mérite pour comparer différentes approches de détection de contours [Abd79]. Ce critère a été utilisé par la suite par d'autres auteurs [Kit81, Pel82, Suc82]. La méthode de mesure est basée sur l'étude de la position des points de contours détectés par un opérateur.

Afin d'évaluer un facteur de mérite permettant de juger de la qualité d'un opérateur, l'algorithme de détection est appliqué à des images synthétiques contenant

un seul contour dont on connaît parfaitement les caractéristiques. La localisation de ce contour, ainsi que le nombre de pixels devant être détectés par un opérateur idéal, sont connus a priori. L'image sur laquelle le critère est calculé est une image binaire déduite par simple seuillage du résultat fourni par l'opérateur testé. Le seuil utilisé pour la binarisation est celui permettant de maximiser le facteur de mérite, qui est alors calculé de la façon suivante :

$$F_1 = \frac{I_r}{\sum_{k=1}^{I_r} (1 / (1 + \alpha \cdot d(k)^2))} / \max(I_i, I_r)$$

où I_i est le nombre de pixels qui seraient détectés par un opérateur idéal, et I_r le nombre de pixels effectivement détectés. α est un facteur permettant de pondérer l'importance des points de contour proches du contour idéal. Abdou et Pratt fixent cette valeur à 1/9.

$d(i)$ est la distance séparant le $i^{\text{ème}}$ point détecté du contour idéal. Abdou et Pratt ont utilisé des images de test contenant un contour idéal rectiligne, disposé soit verticalement, soit en diagonale. Dans ce cas, la distance d'erreur correspond à la distance, au sens Euclidien du terme, entre le pixel considéré et la droite idéale.

Un opérateur de détection de contours est d'autant meilleur que son facteur de mérite se rapproche de l'unité.

II.2.3.b) Distance moyenne

Une autre solution pour mesurer les performances d'une méthode de détection de contour, qui privilégie également la précision de localisation du contour dans l'image, consiste à mesurer la distance moyenne séparant le contour idéal des pixels détectés par l'opérateur. Ce critère de performance est alors calculé sous la forme :

$$F_2 = \frac{I_r}{\sum_{k=1}^{I_r} |d(k)|} / I_r$$

L'opérateur est d'autant plus efficace que la mesure F_2 est proche de zéro. L'image binaire sur laquelle on évalue la distance moyenne du pixel au contour idéal

est celle obtenue par seuillage de l'image résultat de la détection lors du calcul du facteur de mérite F_1 .

II.2.3.c) Facteur de bruit

Cette approche considère en fait le détecteur de contours en tant que filtre numérique appliqué sur une image bruitée. Le niveau de bruit présent dans l'image résultat dépend à la fois du niveau de bruit de l'image initiale et des caractéristiques propres du filtre.

Le facteur de bruit d'un filtre est défini comme étant la proportion de bruit ajoutée sur la sortie par le filtre lui-même. On peut le calculer de la façon suivante :

$$NF = (S_{in} / B_{in}) / (S_{out} / B_{out})$$

(S_{in} / B_{in}) et (S_{out} / B_{out}) représentant respectivement les rapports signal sur bruit des images d'entrée et de sortie. La troisième mesure d'efficacité, F_3 , correspond à l'évaluation de ce facteur de bruit pour l'opérateur étudié [Neu86]. Un opérateur est d'autant plus performant que son facteur de bruit est faible.

Ces trois mesures de performances sont adaptées à l'évaluation des détecteurs de contours statiques. Dans la suite de ce chapitre, nous allons voir qu'il s'avère tout à fait possible de les utiliser afin de comparer les différents algorithmes de détection de contours en mouvement.

II.3) APPLICATION AUX OPERATEURS DE DETECTION DE CONTOURS EN MOUVEMENT

II.3.1) MODE OPERATOIRE

Les méthodes présentées au paragraphe précédent sont destinées à évaluer des opérateurs de détection de contours statiques. Les opérateurs testés sont appliqués à des images comportant un contour idéal, de caractéristiques connues a priori, auxquelles on ajoute un bruit gaussien, sensé représenter le bruit pouvant entacher les images réelles. Les mesures sont calculées pour différentes valeurs du niveau de bruit. On obtient ainsi des courbes présentant l'évolution des mesures de performances en fonction du rapport signal sur bruit de l'image, ce qui permet de caractériser le comportement de l'opérateur sur des images bruitées.

La transposition de cette méthode d'évaluation aux algorithmes de détection de contours mobiles peut se faire par utilisation de séquences d'images synthétiques contenant un objet mobile, sur lesquelles on vient ajouter un bruit gaussien.

Afin de disposer de contours locaux possédant différentes orientations, l'objet mobile synthétique utilisé est un disque de niveau de gris homogène, se déplaçant sur un fond uniforme. Pour simuler le plus fidèlement possible le processus d'échantillonnage résultant de l'emploi des caméras CCD actuelles, nous avons généré les images d'un disque dont les valeurs des niveaux de gris des pixels situés sur sa périphérie sont calculées de telle sorte qu'elles correspondent aux proportions de surface du disque idéal échantillonné sur une grille de pixels carrés [Kit87].

Le contour idéal contenu dans ces images de test est alors un cercle de centre et de rayon connus. La distance d'erreur séparant un pixel de l'image de ce contour idéal peut donc être calculée de façon simple en soustrayant la valeur du rayon du cercle à la distance séparant le pixel considéré du centre du cercle.

Le rapport signal sur bruit des images de test est défini, conformément à [Abd79], comme le rapport du carré du contraste de l'image (différence de niveaux de gris existant entre l'objet et le fond) par la variance σ^2 du bruit gaussien :

$$\text{SNR} = h^2 / \sigma^2$$

Le rapport signal sur bruit est identique pour toutes les images d'une même séquence de test. Les images numériques synthétisées comportent 256 niveaux de gris, le disque de niveau de gris 120 se situant sur un fond de niveau de gris 136. Les valeurs des rapports signal sur bruit simulés vont de 1 à 250. Plusieurs images de test, correspondant à différents rapports signal sur bruit, sont présentées sur la figure 2.6.

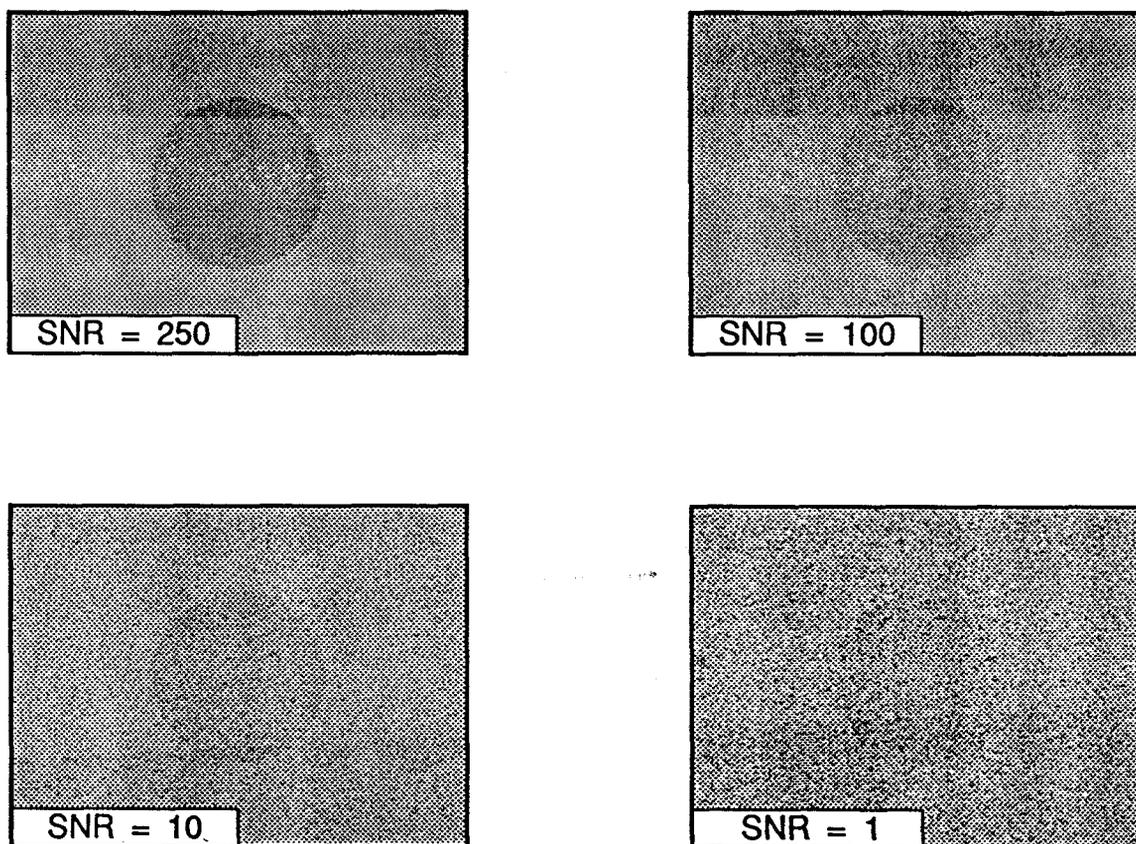


Figure 2.6 : Quelques images de test présentant différents rapports signal/bruit

Le rapport signal sur bruit dans les images résultant de l'extraction des contours mobiles, nécessaire au calcul de la mesure F_3 , est approché par l'expression suivante :

$$SNR_{out} = G_c^2 / \sigma^2$$

dans laquelle G_c représente le niveau de gris moyen du contour détecté et σ^2 la variance de l'image dans les zones ne comportant pas de contours. La distinction entre les deux zones de l'image est réalisée grâce à la connaissance a priori du contour idéal.

II.3.2) TEST DES OPERATEURS DE DETECTION DE MOUVEMENT

La détermination des trois mesures d'efficacité caractérisant les opérateurs de Haynes et Jain, de Stelmaszyk et de Vieren a été réalisée sur plusieurs séquences d'images test, afin de minimiser l'effet des erreurs de calcul dues à des configurations locales particulières du bruit gaussien ajouté à ces images. L'évaluation des critères de comparaison a été réalisée sur plusieurs séquences pour chaque rapport signal sur bruit, les valeurs obtenues étant ensuite moyennées.

Les trois algorithmes de détection de mouvement utilisent un opérateur de type gradient pour mettre en évidence les contours présents dans une image. Afin de déterminer l'influence du détecteur de contours statiques intervenant dans les opérateurs testés, nous avons déterminé les facteurs de mérite des opérateurs de détection de contours mobiles en utilisant trois détecteurs de contours différents [Abd79] : le Sobel, le Prewitt, et le Roberts centré (cf Figure 2.7).

$$\begin{aligned}
 \text{Roberts} &= \text{Max} \left\{ \left| \begin{array}{|c|c|c|} \hline -1 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 1 \\ \hline \end{array} \right| * \text{Image} \right|, \left| \begin{array}{|c|c|c|} \hline 0 & 0 & -1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 0 & 0 \\ \hline \end{array} \right| * \text{Image} \right\} \\
 \\
 \text{Prewitt} &= \text{Max} \left\{ \left| \begin{array}{|c|c|c|} \hline -1 & -1 & -1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \right| * \text{Image} \right|, \left| \begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline \end{array} \right| * \text{Image} \right\} \\
 \\
 \text{Sobel} &= \text{Max} \left\{ \left| \begin{array}{|c|c|c|} \hline -1 & -2 & -1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 2 & 1 \\ \hline \end{array} \right| * \text{Image} \right|, \left| \begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array} \right| * \text{Image} \right\}
 \end{aligned}$$

Figure 2.7 : Opérateurs de détection de contours statiques

Le résultat moyen des différentes simulations est présenté sur les courbes suivantes montrant l'évolution des trois critères (facteur de mérite F_1 , distance moyenne au contour idéal F_2 et facteur de bruit F_3), pour chacun des opérateurs, en fonction du rapport signal sur bruit. La figure 2.8 correspond aux évaluations de F_1 , F_2 et F_3 sur les opérateurs utilisant un gradient de type-Sobel, la figure 2.9 montre les résultats obtenus avec un gradient de type Prewitt, et la figure 2.10 correspond aux opérateurs utilisant le Roberts centré.

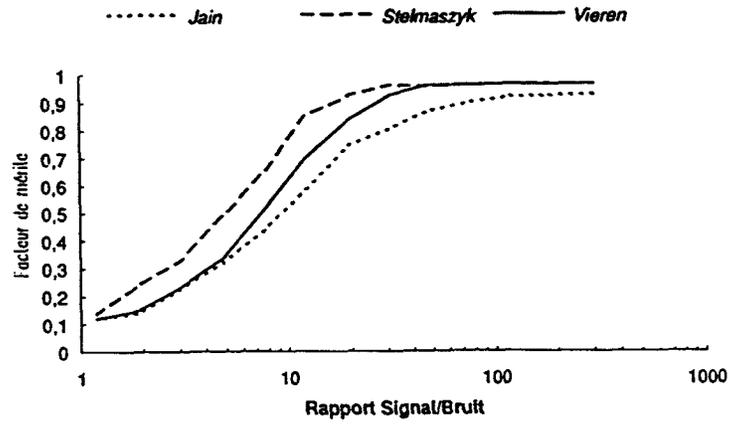
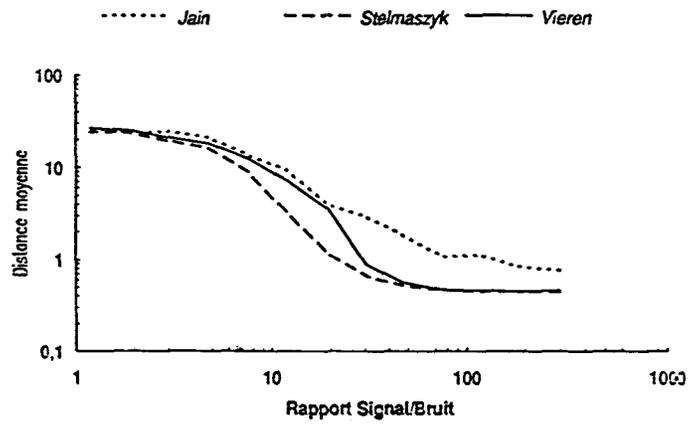
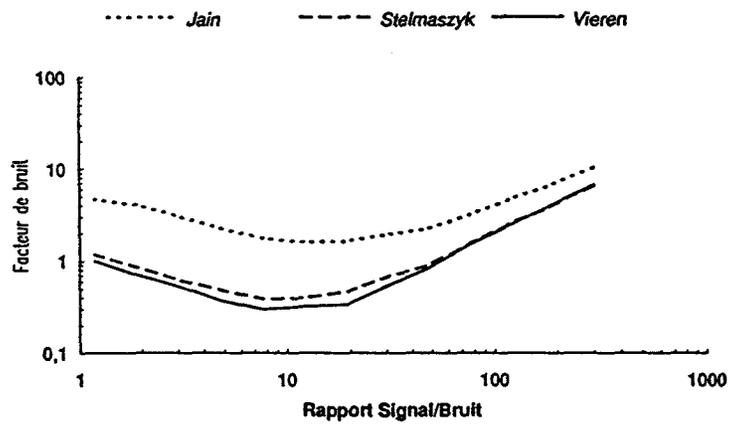
a) Facteur de mérite F_1 b) Facteur de mérite F_2 c) Facteur de mérite F_3

Figure 2.8 : Facteurs de mérite pour la détection de contours en mouvement avec l'opérateur de détection de contours statiques de Sobel

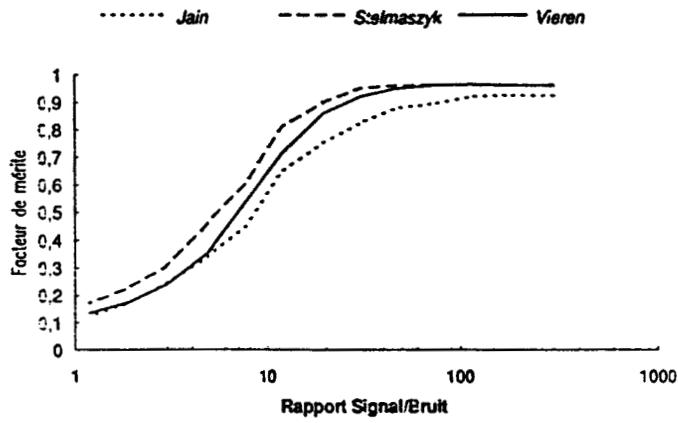
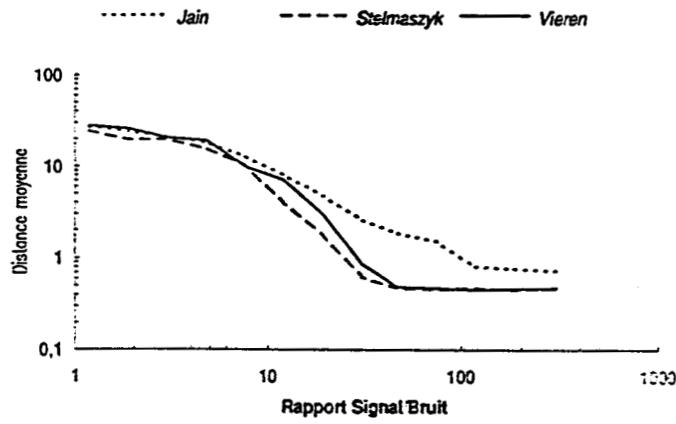
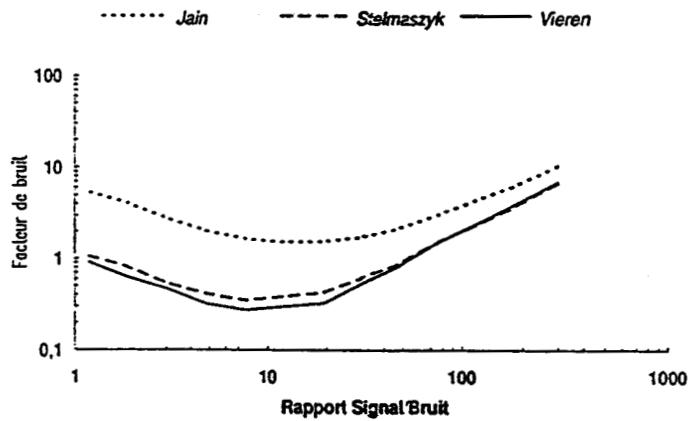
a) Facteur de mérite F_1 b) Facteur de mérite F_2 c) Facteur de mérite F_3

Figure 2.9 : Facteurs de mérite pour la détection de contours en mouvement avec l'opérateur de détection de contours statiques de Prewitt

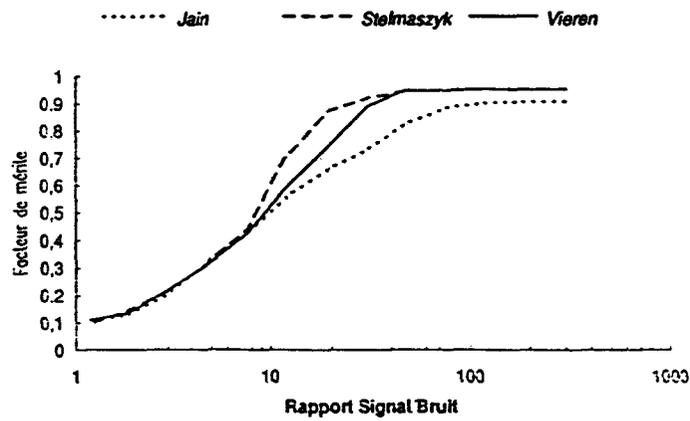
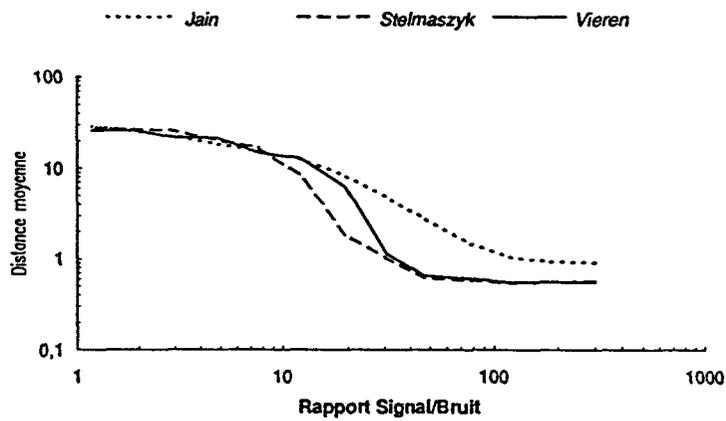
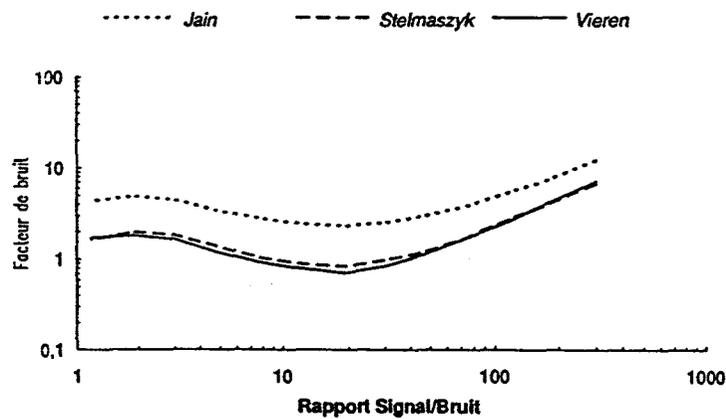
a) Facteur de mérite F_1 b) Facteur de mérite F_2 c) Facteur de mérite F_3

Figure 2.10 : Facteurs de mérite pour la détection de contours en mouvement avec l'opérateur de détection de contours statiques de Roberts (centré)

II.4) INTERPRETATION DES RESULTATS

II.4.1) PRECISION DE LA POSITION DU CONTOUR

Sur les graphiques représentant les facteurs de mérite F_1 des trois méthodes de détection de contours en mouvement, il est aisé de constater que l'opérateur le moins précis quant au positionnement des contours est celui de Haynes et Jain, et ce pour deux raisons :

Tout d'abord, sur des séquences d'images peu bruitées correspondant à des valeurs élevées du rapport signal sur bruit, le facteur de mérite F_1 pour l'opérateur de Haynes et Jain est différent de 1 (cf. figures 2.8a, 2.9a et 2.10a). Cela met en évidence le fait que les contours détectés par cet opérateur sont toujours légèrement différents des contours idéaux. On peut aussi vérifier cette particularité en considérant les courbes représentant la distance moyenne F_2 séparant un point détecté du contour idéal (cf. figures 2.8b, 2.9b et 2.10b).

Ensuite, dans la zone des graphiques correspondant aux rapports signal sur bruit susceptibles d'apparaître sur des séquences d'images réelles exploitables (entre 10 et 200), l'opérateur de Haynes et Jain présente un facteur de mérite F_1 plus faible que celui des deux autres détecteurs (cf. figures 2.8a, 2.9a et 2.10a).

Il est possible d'expliquer les moins bonnes performances de l'opérateur de Haynes et Jain en remarquant que dans les zones de l'image différence correspondant à des régions en mouvement, le niveau de gris moyen n'est pas nul, mais égal au contraste existant entre l'objet mobile et le fond. Les contours parasites de l'image courante se trouvent donc multipliés par une image de valeur moyenne non nulle : le niveau de bruit présent dans les zones correspondantes de l'image résultante est alors plus important, ce qui entraîne l'apparition d'un taux de fausses détections plus élevé. Cette constatation est d'ailleurs confirmée par la mesure du facteur de bruit F_3 , qui s'avère plus important pour l'algorithme de Haynes et Jain, quel que soit le gradient statique utilisé.

Ce phénomène est mis en évidence sur la figure 11 qui présente les images intervenant comme opérands de l'opération de multiplication lors de l'évaluation

des contours en mouvement par la méthode de Haynes et Jain. Dans chacune de ces images, l'histogramme des niveaux de gris est calculé dans une région correspondant à une partie de l'objet en mouvement. Le disque mobile et le fond étant tous deux de niveaux de gris constants, le maximum de l'histogramme correspond à la valeur moyenne du niveau de gris des pixels de la zone considérée. La largeur du mode de l'histogramme est alors directement liée au niveau de bruit présent dans l'image. Le mode principal de l'histogramme de la région extraite de l'image résultat est relativement large.

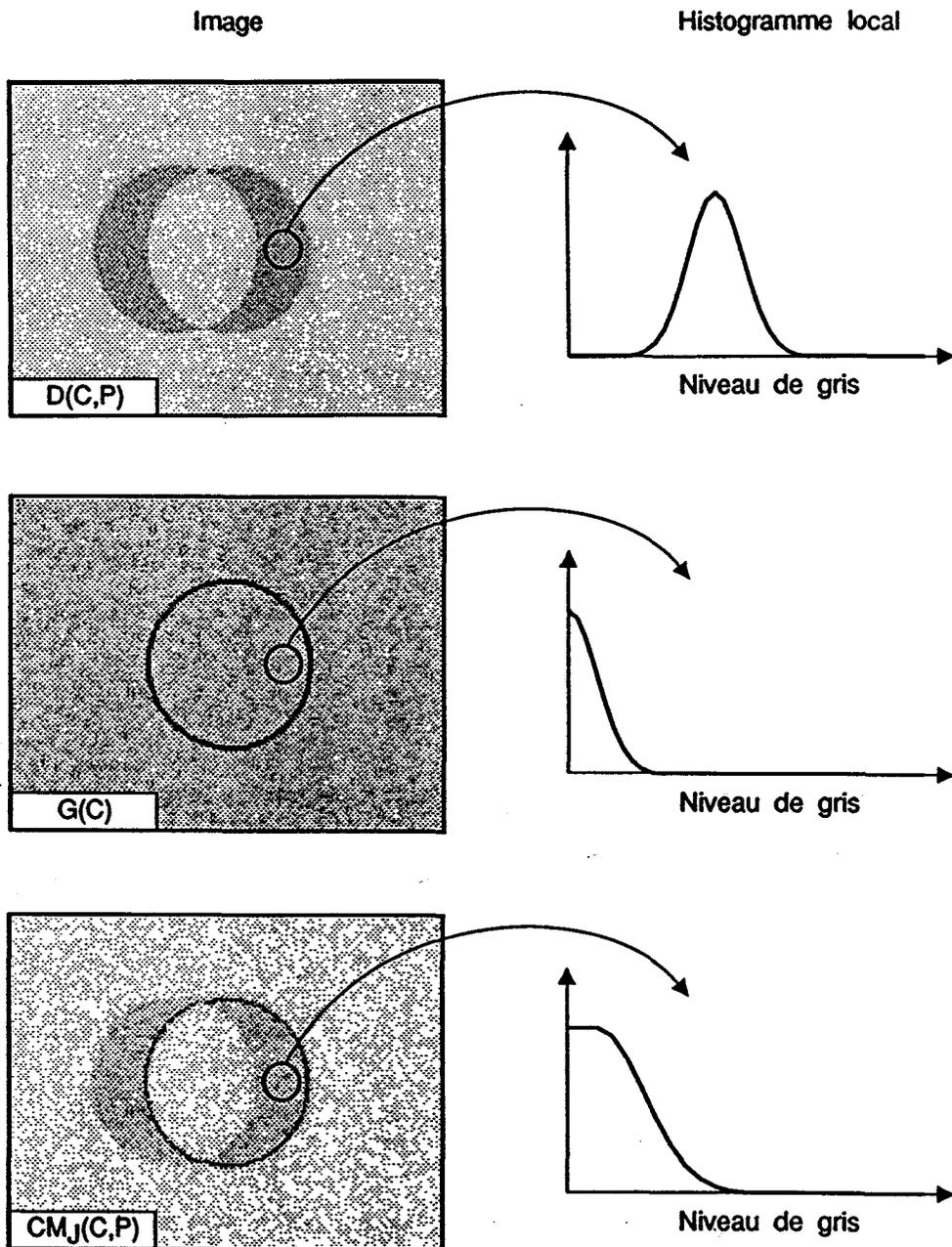


Figure 2.11 : Niveau de bruit dans les régions en mouvement, opérateur de Haynes et Jain

Dans les méthodes de Stelmaszyk et de Vieren, l'application d'un détecteur de contours sur l'image différence permet de diminuer le niveau de gris moyen à l'intérieur des régions de l'image servant à détecter le mouvement. La multiplication se réalise alors en prenant comme opérands deux images de valeur moyenne faible dans les régions ne comportant pas de contours mobiles. Le niveau de bruit présent dans l'image résultante est plus faible que dans le cas de l'utilisation directe de l'image différence comme le montre le dernier histogramme de la figure 2.12.

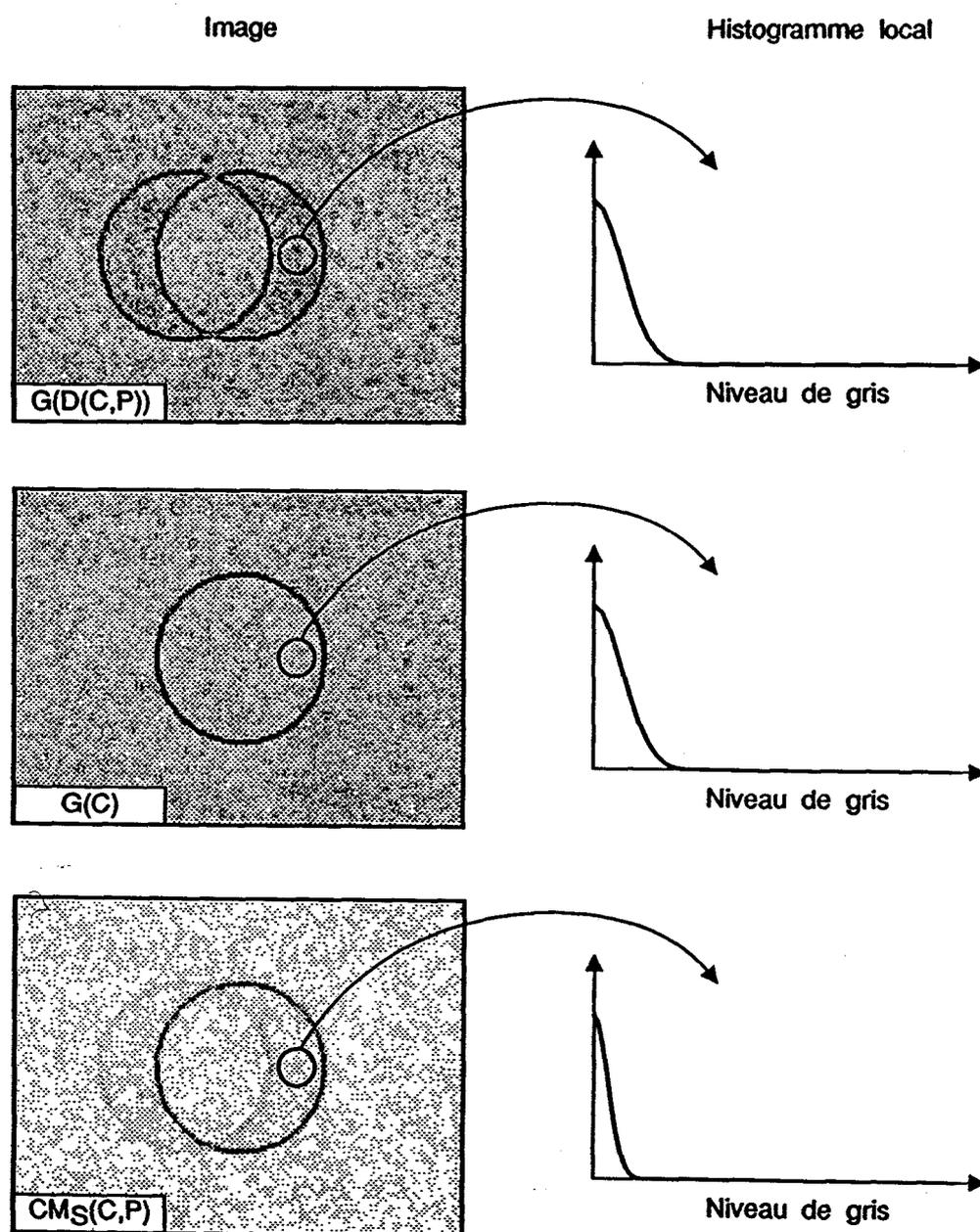


Figure 2.12 : Niveau de bruit dans les régions en mouvement, opérateur de Stelmaszyk

L'opérateur de Vieren se comporte de façon moins précise que celui de Stelmaszyk, tout au moins selon le facteur de mérite F1 (cf. figures 2.8a, 2.9a et 2.10a). Sur les images de la figure 2.13, qui montrent le résultat de la détection des contours mobiles sur une même séquence d'images bruitées, respectivement par les méthodes de Stelmaszyk et de Vieren, on peut remarquer la présence de réponses parasites au niveau des contours de l'objet dans toutes les images utilisées par l'algorithme de détection. Ce sont principalement ces contours qui entraînent une diminution du facteur de mérite des opérateurs avec le niveau de bruit.

L'algorithme de Vieren, utilisant trois images successives au lieu de deux, fait donc apparaître des contours parasites aux endroits de l'image correspondant à la position de l'objet non seulement dans l'image précédente, mais aussi dans la suivante. De ce fait, son facteur de mérite est plus faible que celui de l'opérateur de Stelmaszyk.

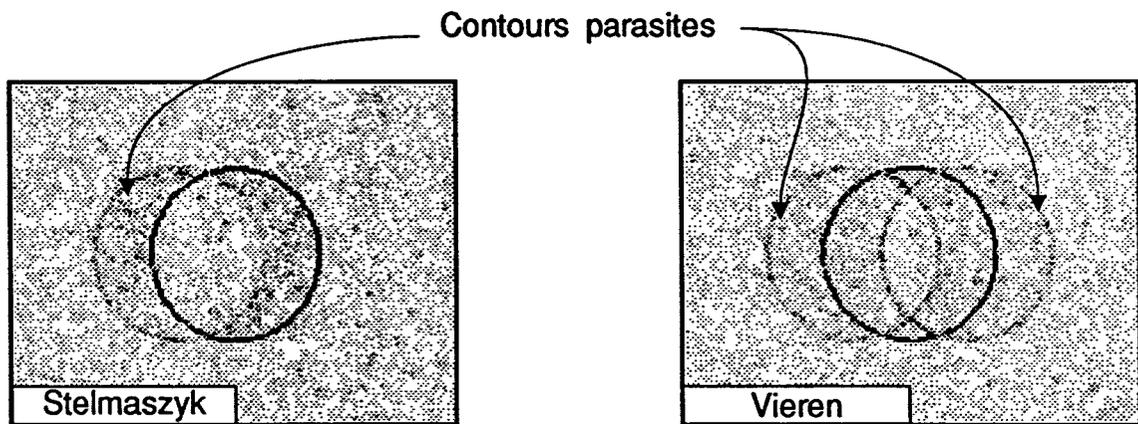
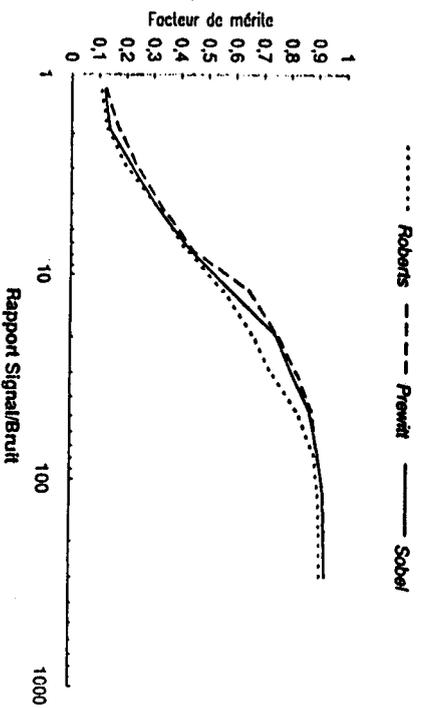


Figure 2.13 : Réponses des opérateurs de Stelmaszyk et Vieren sur une même séquence d'images bruitées

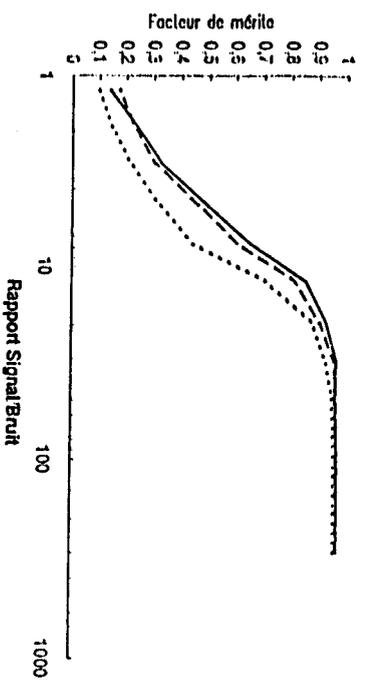
II.4.2) INFLUENCE DU DETECTEUR DE CONTOURS STATIQUES

II.4.2.a) Précision en position

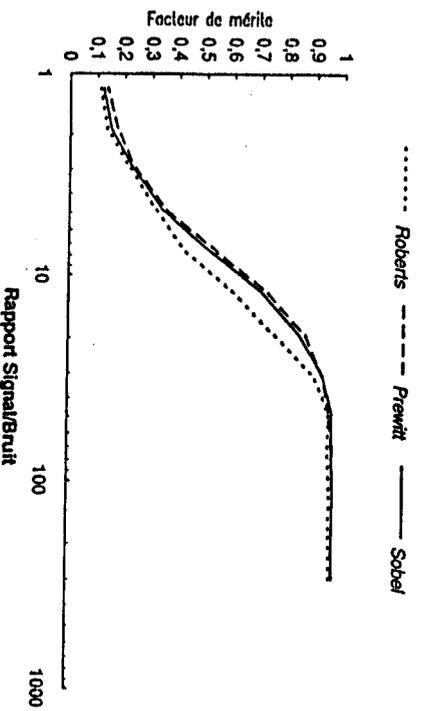
La précision propre de l'opérateur de détection de contours statiques utilisé lors de la recherche des contours en mouvement doit logiquement avoir une influence importante sur la précision globale de la méthode.



a) Opérateur de Haynes et Jain



b) Opérateur de Stelmasszyk



c) Opérateur de Vieren

Figure 2.14 : Facteur de mérite F_1 des opérateurs de détection de contours en mouvement selon le détecteur de contours statiques utilisé

Les courbes de la figure 2.14 reprennent les mêmes résultats que ceux présentés sur les figures 2.8a 2.9a et 2.10a, mais les réorganisent différemment afin de montrer l'influence du détecteur de contours statiques sur chacun des opérateurs de détection de contours mobiles.

Sur les figures 2.14a et 2.14c, on remarque que le facteur de mérite des opérateurs de Haynes et Jain et de Vieren ne dépend que faiblement du type de détecteur de contours statiques utilisé. La précision de ce dernier ne constitue donc pas un critère de sélection. En fait, les contours parasites dus à l'algorithme de détection de mouvement et situés à une distance importante du contour idéal, contribuent de façon prépondérante à la diminution du facteur de mérite. L'influence du gradient statique utilisé reste donc faible.

Par contre, sur la figure 2.14b correspondant à l'opérateur de Stelmaszyk, on voit nettement que le résultat n'est satisfaisant que si on utilise les gradients précis de Prewitt ou de Sobel. Les facteurs de mérite obtenus correspondent d'ailleurs tout à fait à ceux de l'opérateur de détection de contours statiques utilisé (cf. figure 2.15).

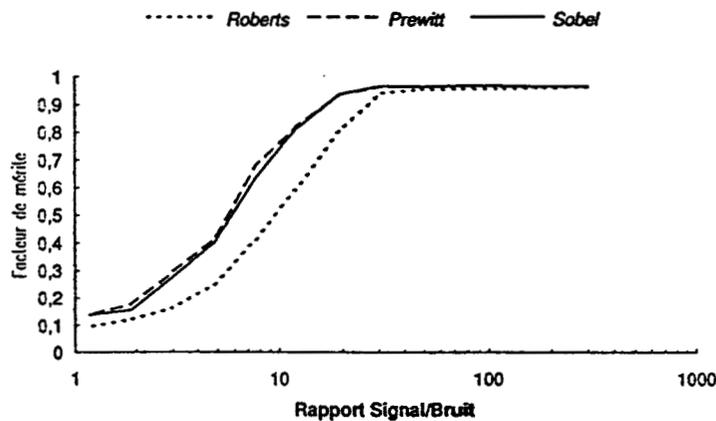
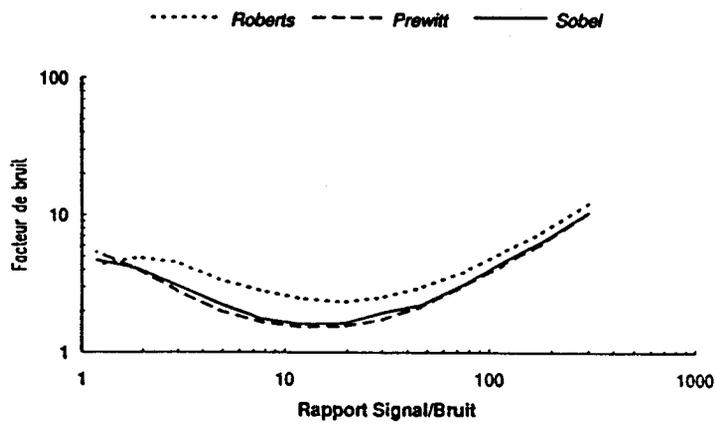


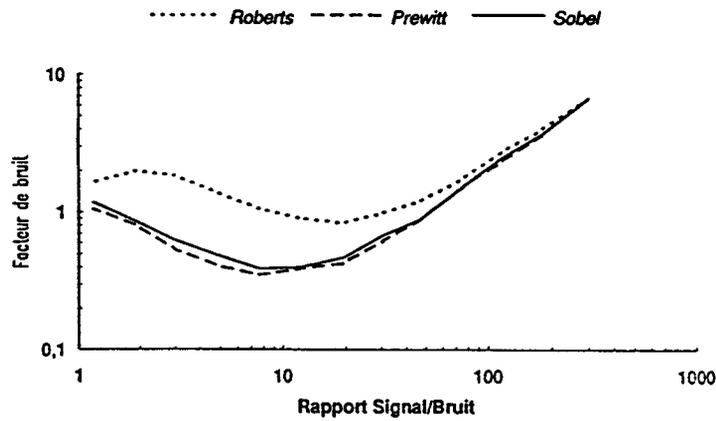
Figure 2.15 : Facteur de mérite F_1 des opérateurs de détection de contours statiques

II.4.2.b) Facteur de bruit

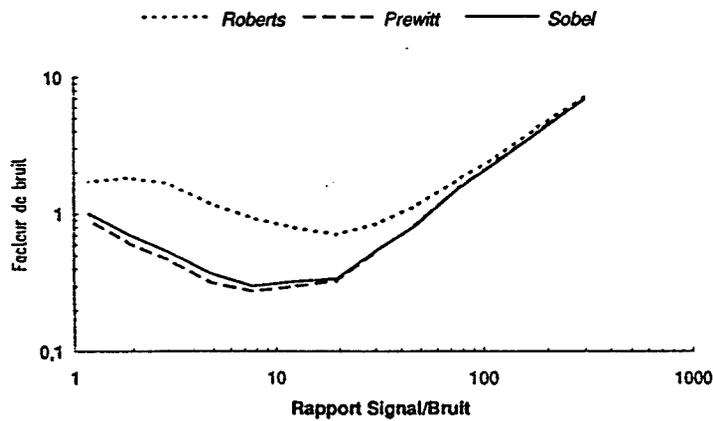
Comme dans le paragraphe précédent, afin de déterminer l'influence de la méthode de détection de contours statiques sur le facteur de bruit global de l'opérateur de détection de contours mobiles, nous avons représenté les résultats en les regroupant de façon différente.



a) Opérateur de Haynes et Jain



b) Opérateur de Stelmazyk



c) Opérateur de Vieren

Figure 2.16 : Facteur de mérite F_3 des opérateurs de détection de contours en mouvement selon le détecteur de contours statiques utilisé

Les courbes correspondantes sont présentées sur la figure 2.16. Pour chacun des trois opérateurs de détection des contours mobiles, on constate que le facteur de bruit global de la méthode est le plus élevé lorsqu'on extrait les contours présents dans les images avec le gradient de Roberts. Les opérateurs utilisant le gradient de Prewitt se comportent de façon similaire à ceux utilisant le gradient de Sobel.

Ce classement des facteurs de bruit correspond tout à fait à celui relatif aux opérateurs de détection des contours statiques. En effet, l'opérateur de Roberts est celui qui possède le facteur de bruit propre le plus élevé, et les opérateurs de Prewitt et Sobel se comportent de façons très similaires selon ce critère [Abd79].

Il apparaît donc que le choix de l'opérateur de détection de contours statiques a une influence importante sur la qualité des images fournies par les opérateurs de détection de contours mobiles, en ce qui concerne le niveau de bruit présent dans les images résultat.

II.5) CONCLUSION

L'extension aux contours mobiles de certaines mesures d'efficacité des détecteurs de contours statiques nous a permis de caractériser la précision des méthodes de mise en évidence des contours mobiles au niveau du positionnement des contours détectés. Des trois algorithmes testés, celui de Haynes et Jain s'avère être le moins performant quant au positionnement des contours mobiles contenus dans une séquence d'images bruitées.

Par évaluation du facteur de bruit caractéristique de chacun des opérateurs de détection de contours mobiles, nous avons pu mettre en évidence le fait que les algorithmes de Stelmaszyk et de Vieren fournissent une image moins bruitée que celle calculée par la méthode de Haynes et Jain sur une même séquence d'images.

Nous avons vu, d'autre part, que la méthode de Vieren présente l'avantage déterminant d'éliminer les détections parasites dues au déplacement des objets sur un fond non uniforme. Compte tenu de ces différents éléments, cet algorithme apparaît donc comme le mieux adapté au traitement de scènes réelles.

CHAPITRE III

CONCEPTION ET REALISATION D'UN PROCESSEUR CABLE POUR L'EXTRACTION DES CONTOURS MOBILES EN TEMPS REEL

La méthode de mise en évidence des contours en mouvement proposée par C. Vieren s'est avérée être la mieux adaptée au traitement de séquences d'images issues de scènes réelles. Elle se situe dans la catégorie des méthodes orientées mouvement, et nécessite à chaque estimation du déplacement des objets le traitement de trois images successives de la séquence par des opérateurs de voisinage spatiaux et temporels.

Comme nous l'avons précisé au début de ce mémoire, l'intégration d'un système de vision artificielle dans le processus de conduite automatique d'un véhicule implique, non seulement une fiabilité de fonctionnement élevée, mais aussi une vitesse de calcul très importante. C'est pourquoi nous avons décidé d'étudier la possibilité d'implanter l'algorithme de détection de mouvement sur un processeur rapide, permettant le traitement en temps-réel de séquences d'images issues d'une caméra standard.

Dans cette partie du mémoire, nous présentons tout d'abord une description des différentes architectures de machines de traitement d'images permettant un traitement rapide des données. Nous montrons ensuite comment l'algorithme de C. Vieren a été partitionné en unités de calcul indépendantes pouvant s'intégrer dans une architecture de traitement spécifique. Nous terminons par la description d'un processeur câblé réalisant cette intégration, et permettant l'extraction des contours des objets mobiles en temps-réel : le STREAM (Système Temps-Réel d'Extraction et d'Analyse du Mouvement).

III.1) LES ARCHITECTURES D'ORDINATEUR ADAPTEES AU TRAITEMENT D'IMAGES

Une première solution au problème du traitement rapide de données consiste à améliorer la vitesse des circuits électroniques, ce qui permet d'implanter les traitements sur des ordinateurs standards. En une dizaine d'années, les performances de ces machines se sont vues multipliées par un facteur cent. Toutefois, cette approche ne permet pas de résoudre à elle seule le problème, et les perspectives d'évolution ne sont plus aussi importantes qu'il y a quelques années.

Une autre approche, plus prometteuse, consiste en la mise au point de structures d'ordinateurs plus adaptées au traitement de grandes quantités d'informations. Ces ordinateurs sont appelés machines parallèles.

III.1.1) CLASSIFICATION DES MACHINES PARALLELES

Diverses approches ont été proposées dans la littérature afin de classer les architectures d'ordinateurs [Ree84, Cha88, Err91]. La plus utilisée est sans doute celle proposée par Flynn [Hwa85] qui effectue une distinction entre les machines selon le nombre d'instructions différentes pouvant être exécutées simultanément et le nombre de données traitées simultanément. Cette classification conduit à la définition de quatre catégories d'ordinateurs.

III.1.1.a) Machines SISD

Single Instruction stream, Single Data stream.

Ce type d'architecture correspond en fait aux machines courantes de traitement de données, fonctionnant selon le modèle de Von Neumann. Les données sont traitées les unes à la suite des autres par une seule unité de traitement. Lors de chaque cycle élémentaire de calcul, une seule instruction simple est exécutée par l'unité de traitement.

Les données ne pouvant pas toutes être traitées simultanément, il est nécessaire de les mémoriser durant les périodes où elles ne sont pas utilisées. La gestion

mémoire nécessaire est réalisée par une unité de contrôle agissant en synchronisme avec l'unité de traitement.

L'avantage de ce type d'architecture réside évidemment dans sa simplicité matérielle et dans sa souplesse concernant l'implantation des algorithmes de calcul. Cependant, le temps de traitement est à la fois proportionnel au nombre total de données à traiter et au nombre d'instructions élémentaires nécessaires au traitement d'une donnée.

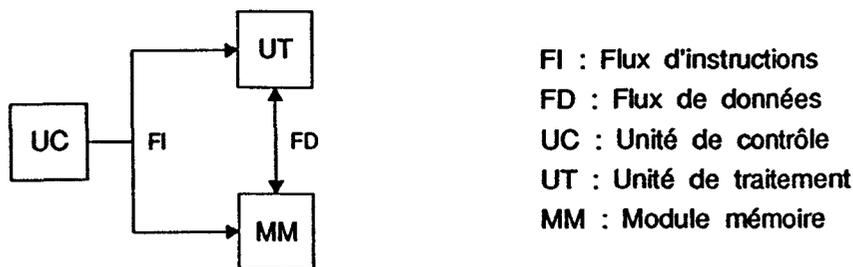


Figure 3.1 : Architecture SISD

III.1.1.b) Machines MISD

Multiple Instruction stream, Single Data stream.

Une machine appartenant à cette catégorie est susceptible de réaliser plusieurs instructions différentes durant un cycle élémentaire de calcul. Les unités de traitement agissent toutes sur un même flot de données, le résultat fourni par l'une servant d'opérande à la suivante. Ce type d'organisation des unités de traitement est la base des machines de type pipeline.

Le parallélisme utilisé dans les machines MISD est qualifié de parallélisme de fonction, ou parallélisme temporel, car une même donnée est traitée successivement par différentes unités de calcul. Le temps de traitement d'une donnée particulière reste donc toujours proportionnel au nombre d'instructions exécutées.

Toutefois, lors de chaque cycle élémentaire de traitement, une donnée est placée en entrée du processeur, et ce dernier fournit simultanément un résultat. Le temps de traitement global d'un ensemble important de données n'est donc plus proportionnel au nombre d'instructions nécessaires au traitement d'une donnée isolée, mais seulement au nombre total de données.

Lorsque les données sont traitées sous forme de flot continu, il n'est pas nécessaire d'avoir recours à un système de mémorisation des résultats intermédiaires entre les différentes phases de calcul. Néanmoins, certaines architectures MISD utilisent une mémoire de données et réalisent un calcul global en plusieurs phases successives. Elles sont appelées machines à recirculation de données. Dans ce cas, le flot continu d'entrée est interrompu, et le résultat de la phase précédente est réinjecté en entrée. Le temps de traitement global dépend du nombre de recirculations.

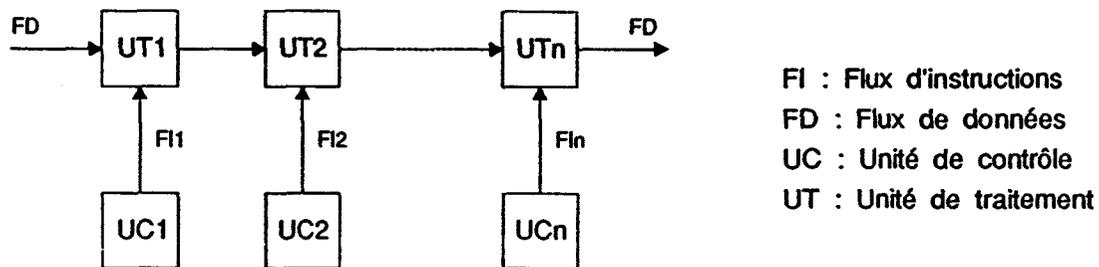


Figure 3.2 : Architecture MISD

III.1.1.c) Machines SIMD

Single Instruction stream, Multiple Data stream.

Les machines de ce type comportent elles aussi plusieurs unités de traitement, mais lors d'un cycle élémentaire, toutes les unités de traitement réalisent la même instruction sur des données différentes.

Le cas idéal d'architecture SIMD correspond à une machine dans laquelle on dispose d'une unité de traitement pour chaque donnée à traiter. Dans ce cas, le temps de traitement global est directement proportionnel au nombre d'instructions nécessaire à traiter une donnée isolée.

Quand le nombre de données à traiter ne permet pas d'attribuer une unité de traitement à chacune d'entre elles, il est nécessaire de partitionner l'ensemble des données et de réaliser le calcul en plusieurs phases successives. Le temps de calcul global peut donc aussi dépendre du nombre d'unités de traitement disponibles.

Comme dans le cas des machines SISD, chaque unité de traitement doit comporter une zone mémoire permettant de stocker les résultats intermédiaires. Les

données initiales fournies à chacune des unités de traitement sont placées dans cette mémoire avant que ne débute la phase de calcul. Cette phase d'initialisation constitue en général un des points faibles des architectures SIMD, car elle nécessite souvent de recourir à des techniques n'exploitant pas le parallélisme des machines.

Les données ne pouvant en général pas être traitées indépendamment les unes des autres, les machines de type SIMD peuvent comporter un réseau de communication reliant les différentes unités de traitement entre elles. Plus le nombre de connexions entre unités est important, plus la réalisation matérielle des ordinateurs se trouve compliquée.

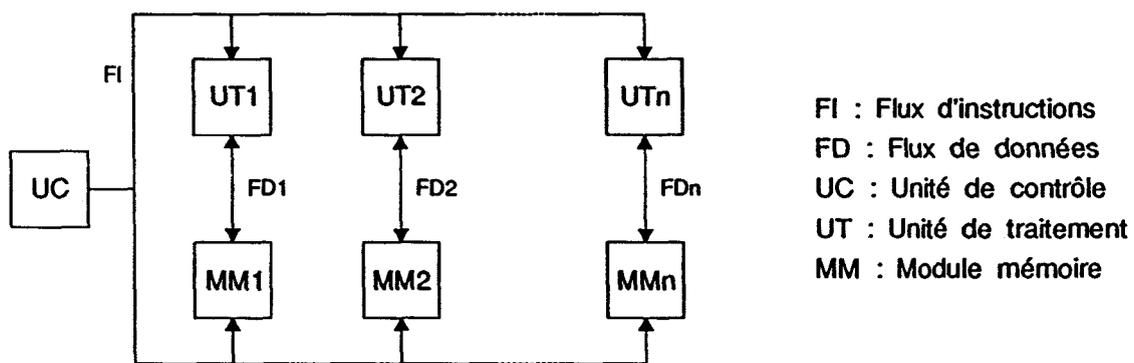


Figure 3.3 : Architecture SIMD

III.1.1.d) Machines MIMD

Multiple Instruction stream, Multiple Data stream.

Un processeur de ce type est capable de traiter plusieurs données simultanément, en utilisant des unités de traitement indépendantes, exécutant des instructions différentes.

Il est impossible de caractériser le fonctionnement et l'efficacité de ce type de machines sans apporter de précisions supplémentaires concernant leur architecture. Toutefois, comme dans le cas des machines de type SIMD, les performances sont en général proportionnelles au nombre de processeurs disponibles et aux nombre de connexions reliant ces derniers.

Un cas particulier de machine MIMD est constitué des architectures appelées MSIMD (Multiple SIMD), dans lesquelles plusieurs étages fonctionnant chacun en mode SIMD sont connectés par un réseau de communication inter-étages. Une

classification plus fine des architectures MIMD peut être réalisée par mise en évidence des différents types de réseaux de communication de données utilisés [Swa82].

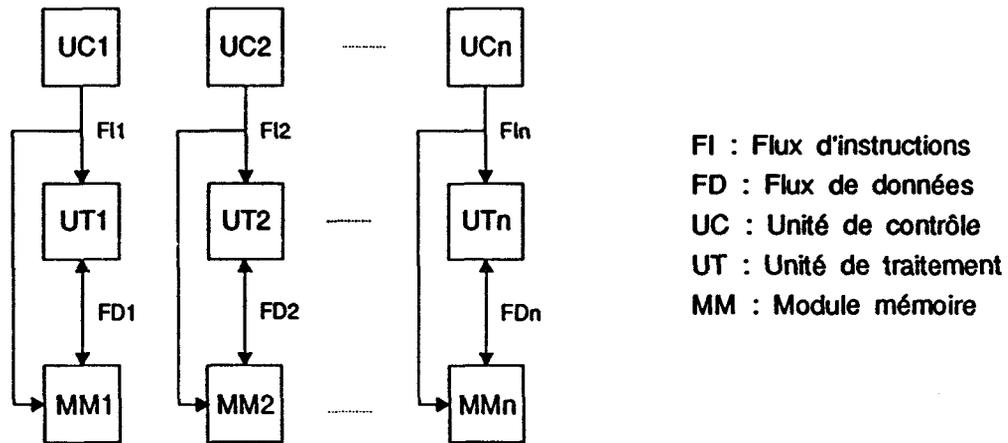


Figure 3.4 : Architecture MIMD

III.1.2) UTILISATION EN TRAITEMENT D'IMAGES

L'architecture d'une machine parallèle devant être utilisée pour implanter un algorithme particulier dépend fortement du type de données devant être traitées. En traitement numérique d'images, les types de données traitées peuvent s'étendre du simple tableau à deux dimensions représentant les valeurs de niveau de gris d'une image numérique, jusqu'aux listes d'objets caractérisés par des dizaines de paramètres de tous types [Cas85, Pra87].

En fait, le type d'architecture parallèle pouvant être utilisé afin d'implanter un algorithme particulier de traitement d'image dépend du niveau de complexité algorithmique de ce dernier : bas, moyen ou haut niveau [Ree82, Com83, Lee85, Gue86, Shu86, Hwa89, Wee89, Lee90].

III.1.2.a) Traitements de bas niveau

Les traitements qualifiés de bas niveau utilisent comme opérandes l'image numérique elle-même, c'est à dire un tableau bidimensionnel contenant la valeur du

niveau de gris des pixels. De plus, ces traitements exploitent souvent des propriétés de stationnarité de l'image dans ce sens qu'ils agissent de façon similaire en tout point de l'image quelle que soit sa position.

Machines SIMD

Les architectures de type SIMD semblent donc parfaitement adaptées à l'implantation de traitements de ce genre. En effet, il est possible d'associer une unité de traitement à chaque pixel, et de traiter ainsi l'image dans sa totalité. De nombreux exemples de machines fonctionnant selon ce principe sont décrits dans la littérature. On peut citer :

Les machines CLIP (Cellular Logic Image Processor), organisées sous forme de réseau carré de processeurs binaires élémentaires, chacun étant connecté à ces huit voisins [Duf86]. La quatrième version de ce processeur (CLIP4) comporte 96x96 processeurs élémentaires. Ces machines hautement parallèles sont largement utilisées pour le traitement rapide d'images numériques, et de nombreux algorithmes de tous genres ont été adaptés afin de pouvoir fonctionner sur cette architecture [Duf86, Hoo87].

Le MPP (Massively Parallel Processor), développé à l'origine par la NASA afin de traiter les images fournies par le satellite LANDSAT [Bat80, Pot83]. Les processeurs élémentaires sont organisés sous forme de matrice carrée, chacun étant relié à ses quatre voisins directs. Le nombre de processeurs binaires élémentaires implantés est de 128x128 (16384), chacun disposant d'une mémoire de 1K bits.

Machines MISD

Une autre approche permettant une implantation parallèle des traitements de bas niveau est basée sur l'utilisation du mode particulier d'adressage des pixels utilisé en vidéo. Dans ce cas, l'image numérique comportant deux dimensions se trouve ramenée à une seule dimension par une technique de balayage ligne par ligne.

Le flot de données issu des capteurs fonctionnant selon ce principe peut alors être traité directement par des machines de type MISD [Bas85, Dou86]. Cette méthode d'implantation matérielle des algorithmes de bas niveau est largement

utilisée à l'heure actuelle, car elle s'avère souvent simple et efficace. On peut citer :

Le Cytocomputer [Lou80] qui est l'une des premières machines dédiée au traitement fonctionnant selon le principe MISD. Elle est constituée de deux niveaux de pipeline, l'un comprenant des unités de traitement travaillant en binaire, l'autre permettant de traiter des niveaux de gris. Les unités sont connectées les unes à la suite des autres et échangent des données avec une mémoire d'image balayée en permanence. Le principal défaut de cette machine réside dans l'aspect figé de son réseau pipeline qui n'est pas configurable.

La machine IDATEN (nom d'une divinité Japonaise) qui est conçue selon une architecture pipeline [Sas85]. Cependant, le réseau de communication reliant les différents modules est configurable de façon dynamique. Les images converties en données numériques sont injectées à la cadence vidéo dans un réseau de commutation assurant la circulation des données au sein du système. Ce système, très souple et efficace, est bien adapté au traitement de séquences d'images [Oza89].

Les cartes de traitement d'images numériques utilisées en tant que coprocesseurs spécialisés sous divers environnements informatiques fonctionnent souvent selon le principe des machines MISD à recirculation de données : les images fournies par une source vidéo sont enregistrées dans une mémoire de trame pouvant être balayée afin de servir de source de données à des processeurs pipeline de traitement de voisinage.

III.1.2.b) Traitements de moyen et de haut niveau

Les traitements de niveau intermédiaire et de haut niveau rencontrés en analyse d'images sont souvent non localisables. En effet, la plupart des algorithmes permettant le passage d'une représentation brute de l'information contenue dans la matrice de pixels à des notions plus complexes de type régions ou objets, utilisent la totalité de l'image et non pas un voisinage restreint de chacun de ses pixels.

C'est pour cette raison que les algorithmes de cette catégorie ne peuvent en général pas être implantés sur des machines de type SIMD ou MISD. Il est alors nécessaire de recourir à une implantation sur des machines de type MIMD, qui doit permettre à chaque processeur élémentaire d'avoir accès à la totalité de l'image.

Machines MIMD

Il n'existe pas d'architecture MIMD optimale permettant d'implanter tous les algorithmes de traitement d'image de moyen et de haut niveau, et le choix d'une structure de connexion entre les processeurs doit être dicté par l'algorithme à implanter. Divers exemples existent cependant dans la littérature :

Le FLIP (FLexible Image Processor) dispose de 16 processeurs individuels comportant chacun une unité de traitement et une unité de contrôle [Lue80]. Les unités, disposant chacune de deux entrées et d'une sortie, sont connectées entre elles par un réseau de bus de données. Une unité d'échange réalise les transferts de données entre les unités de traitement et un processeur hôte.

Le PASM (PARTitionable SIMD/MIMD) est une machine configurable de façon dynamique en ensembles d'unités élémentaires fonctionnant en mode SIMD ou en mode MIMD [Sie81]. Chaque unité de traitement dispose d'une mémoire locale pouvant stocker des données quand le mode de travail est le SIMD, ou des instructions en mode MIMD. Cependant, la grande souplesse de cette architecture s'accompagne d'une complexité de programmation très importante.

La machine IUA (Image Understanding Architecture) est un processeur de traitement d'images organisé de façon hiérarchique, chaque niveau comportant des unités de traitement différentes [Shu86, Hwa89]. La couche située au niveau pixel, appelée CAPP (Content Adressable Parallel Processor) fonctionne en mode SIMD (512x512 processeurs), et est adressable par les autres couches en mode associatif (accès par valeur). Au niveau supérieur, on trouve le NPA (Numeric Processor Array), permettant un traitement parallèle à haut rendement (64x64 processeurs). Le niveau suivant est constitué par le GPPA (General Purpose Processor Array) disposant d'unités de traitement en virgule flottante (8x8 processeurs). Le dernier niveau est constitué d'un ordinateur standard de type VAX.

Machines pyramidales de type MSIMD

Lorsque les traitements à paralléliser utilisent une représentation hiérarchique de l'information, ou une représentation de l'image à différents niveaux de résolution, une structure de machine MIMD de type MSIMD est bien adaptée [Swa82, Cha88]. Dans ce cas, les sous ensembles de la machine fonctionnant en mode SIMD

correspondent aux différents niveaux de résolution, chaque niveau étant relié aux niveaux adjacents. Comme le nombre de processeurs nécessaire au traitement décroît avec le niveau de résolution, les machines correspondantes sont souvent qualifiées de machines pyramidales. Différentes machines fonctionnant selon ce principe ont été décrites dans la littérature :

La machine PAPIA, comporte 8 couches successives de processeurs élémentaires (de tailles variant de 128×128 unités de traitement à 1 unité), chacune d'entre elles fonctionnant en mode SIMD [Can85]. Chaque processeur élémentaire de la $i^{\text{ème}}$ couche est connecté à 4 processeurs de la $(i-1)^{\text{ème}}$ couche appelés fils, ainsi qu'à ces 4 voisins directs (dans la même couche). Les données sont introduites à la base de la pyramide, et les résultats récupérés en son sommet.

La machine SPHINX (Système Pyramidal Hiérarchisé pour le traitement d'Images Numériques) utilise un réseau de connexions entre couches dans lequel un processeur est relié à deux fils dans la couche précédente, et à ses quatre voisins directs dans la même couche [Mer89]. Cette architecture simplifiée par rapport à celle des autres machines pyramidales classiques permet d'utiliser des processeurs élémentaires plus simples, à deux entrées au lieu de quatre, mais, en contre-partie, en nombre plus important.

III.1.2.c) Conclusion

De toutes les architectures décrites précédemment, il n'en est aucune qui soit universelle, et qui permette de résoudre le problème très général de l'implantation des procédures de traitement d'images. En outre, les machines les plus polyvalentes (par exemple le IUA) sont toujours très complexes et donc très coûteuses.

L'implantation de l'algorithme de détection d'objets en mouvement sur une machine rapide devait se faire en respectant deux contraintes contradictoires, à savoir le suivi d'un cahier des charges ambitieux, et la réalisation d'un processeur de faible coût. L'utilisation d'une machine de traitement d'images polyvalente ne permettait pas de satisfaire à cette deuxième condition. C'est pourquoi nous nous sommes orientés vers la conception d'une architecture spécifique.

III.2) CHOIX D'UNE ARCHITECTURE SPECIFIQUE

III.2.1) CAHIER DES CHARGES

Le cahier des charges retenu pour le processeur d'extraction des contours mobiles découle de son utilisation future en tant que capteur spécialisé dans un environnement banalisé : la simplicité de mise en oeuvre constitue un critère de conception primordial.

III.2.1.a) Interfaçage

Tout d'abord, le processeur doit être interfaçable avec tout matériel standard de traitement d'images. Le moyen le plus simple pour satisfaire cette condition est d'utiliser des signaux vidéo normalisés à la fois en entrée et en sortie du processeur.

Le système doit donc être conçu en tant que préprocesseur spécialisé, sorte de "boite noire" réalisant la transformation d'un signal vidéo. Le signal de sortie doit présenter le résultat de l'extraction des contours en mouvement, sous la forme d'une image où toutes les parties immobiles ont été éliminées.

III.2.1.b) Aspect temps-réel

La vitesse de traitement doit permettre de suivre la cadence imposée par les normes vidéo. Le processeur doit pouvoir traiter les trois images utilisées par l'algorithme de détection durant l'intervalle de temps séparant deux images du signal d'entrée, c'est à dire un vingt-cinquième de seconde. Les résultats de détection doivent être fournis en sortie à la même cadence, chaque image en sortie du processeur résultant du traitement d'un triplet de trois images successives.

Le retard existant entre la prise en compte d'une image en entrée et le résultat fourni par le processeur concernant cette image doit être minimal. Nous avons vu que la méthode de détection apporte déjà par elle même un retard de une image.

III.2.1.c) Fonctionnement autonome

Le processeur doit être indépendant de toute architecture d'ordinateur hôte. En effet, il sera amené à fonctionner de façon autonome, sans support informatique particulier.

Le traitement des données doit autant que possible être réalisé par des circuits simples, autorisant une maintenance aisée du matériel.

La flexibilité de l'architecture de calcul n'est pas un critère important dans notre application. Elle ne doit en aucun cas prévaloir sur la simplicité lorsqu'un choix doit être effectué entre différentes solutions matérielles.

III.2.2) PARTITIONNEMENT D'UN ALGORITHME EN OPERATEURS SIMPLES

III.2.2.a) Généralités

Le partitionnement d'un algorithme de traitement d'images en fonctions simples permettant une implantation optimale sur une architecture parallèle est un problème très complexe, comme en témoigne une littérature abondante consacrée à ce sujet [Yal85, Pra86, Pra87, Ros87]. La plupart des méthodes proposées permettent l'affectation optimale des procédures à un certain nombre d'unités de traitement, la structure et le nombre de ces unités étant connus a priori.

Lee et Aggarwal ont proposé une méthode permettant de déterminer une structure de calcul adaptée à la résolution d'un problème particulier [Lee85, Lee90]. Ils montrent que l'implantation d'un algorithme sur une structure de calcul rapide peut être envisagée de deux façons différentes, selon les contraintes retenues par le concepteur :

Dans un premier cas, le type de machine utilisé est connu a priori ("architecture-driven approach"). Le partitionnement de l'algorithme en fonctions simples est alors réalisé afin de minimiser le temps de calcul global. Cette approche suppose une flexibilité importante du matériel correspondant, qui s'accompagne souvent de

performances moindres.

Une autre possibilité consiste en la mise au point d'une structure de calcul spécifique ("problem-driven approach"). L'architecture dédiée correspondante sera bien adaptée au problème, mais en contrepartie moins évolutive. Cette approche est très intéressante quand il s'agit d'implanter des algorithmes pour lesquels le temps de calcul est un point crucial.

Lee et Aggarwal [Lee85, Lee90] proposent une méthode de partitionnement d'un algorithme adaptée à la mise en place d'architectures dédiées. Cette méthode permet d'introduire des notions d'indivisibilité spatiale et temporelle des fonctions.

III.2.2.b) Parallélisme de fonction

La suite d'opérations constituant le traitement à implanter en parallèle est tout d'abord scindée en fonctions indivisibles temporellement. Par fonction indivisible, les auteurs entendent un traitement ne pouvant plus être partitionné en sous-fonctions agissant successivement sur les données et communiquant entre elles par un certains nombre de résultats intermédiaires. Un problème se représente alors de la façon suivante :

$$S = f_n \circ f_{n-1} \circ \dots \circ f_2 \circ f_1 (E)$$

où E est l'entrée du système, S sa sortie, et les f_i la série de fonctions indivisibles caractérisant l'algorithme à implanter (o est l'opérateur de combinaison de fonctions). Cette première simplification permet alors d'envisager l'utilisation du parallélisme de fonction, ou parallélisme temporel (pipeline). Chaque fonction f_i est implantée sur une unité de traitement individuelle, les données circulant d'une unité à la suivante, depuis l'entrée jusque la sortie :

$$E \Rightarrow f_1 \Rightarrow f_2 \Rightarrow \dots \Rightarrow f_{n-1} \Rightarrow f_n \Rightarrow S$$

III.2.2.c) Parallélisme spatial

L'étape suivante consiste en un fractionnement de chacune des fonctions f_i en sous fonctions indivisibles spatialement, lorsque cela est possible. Une fonction est

indivisible spatialement lorsqu'elle ne peut pas être divisée en deux sous-fonctions agissant de manière identique sur des ensembles de données différents. En fait, en traitement d'images, la notion d'indivisibilité spatiale est très liée à la possibilité de localiser le traitement sur des voisinages de chaque pixel de l'image.

Une fonction f_i divisible spatialement est alors remplacée par un ensemble de sous-fonctions identiques agissant chacune sur un groupe de données différentes :

$$E_i \Rightarrow \left\{ \begin{array}{l} E_i^1 \Rightarrow f_i^1 \Rightarrow S_i^1 \\ E_i^2 \Rightarrow f_i^2 \Rightarrow S_i^2 \\ \dots \\ E_i^m \Rightarrow f_i^m \Rightarrow S_i^m \end{array} \right\} \Rightarrow S_i$$

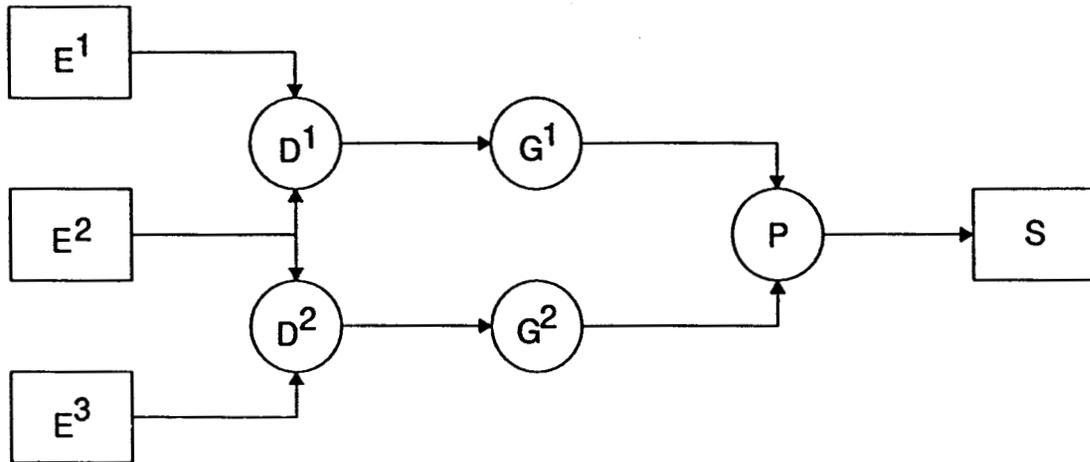
Il est alors possible de faire intervenir une unité de traitement pour chacune des sous-fonctions f_i^j , ce qui correspond à un traitement par des méthodes de parallélisme spatial (structure SIMD). La représentation finale de l'algorithme est donc la suivante :

$$S = \left\{ \begin{array}{l} f_1^1 \\ f_1^2 \\ \dots \\ f_1^m \end{array} \right\} \circ \left\{ \begin{array}{l} f_2^1 \\ f_2^2 \\ \dots \\ f_2^m \end{array} \right\} \circ \dots \circ \left\{ \begin{array}{l} f_n^1 \\ f_n^2 \\ \dots \\ f_n^m \end{array} \right\} (E)$$

où chacune des fonctions f_i^j est réalisée par une unité de traitement individuelle. Dans la pratique, il n'est pas nécessaire de fractionner les fonctions jusqu'à les rendre indivisibles, car le gain de temps apporté par un partitionnement est parfois très faible vis à vis de l'augmentation de complexité qu'il entraîne.

III.2.3) APPLICATION A L'OPERATEUR DE VIEREN

Nous avons appliqué la méthode de décomposition des algorithmes décrite précédemment à l'opérateur de détection de contours en mouvement. Les données d'entrée sont constituées des trois images successives de la séquence, la sortie est elle aussi une image. Un premier partitionnement est réalisable de la façon suivante :



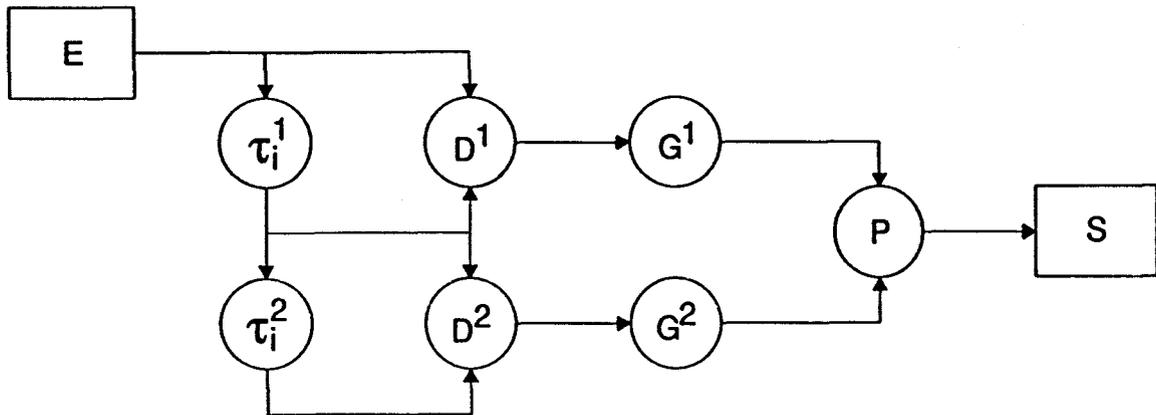
où E^1 , E^2 et E^3 sont les trois images successives de la séquence, D^1 et D^2 les deux opérateurs de soustraction en valeur absolue agissant respectivement sur (E^1, E^2) et (E^2, E^3) , G^1 et G^2 deux opérateurs de gradient identiques agissant sur les deux images différence, et P un opérateur de multiplication combinant les résultats d'extraction des contours fournis par G^1 et G^2 .

Ce partitionnement, bien qu'il soit tout à fait réalisable, ne tient pas compte du fait que les images opérandes se succèdent à l'intérieur d'une séquence. En fait, en prenant en compte cet aspect, on peut considérer qu'une image de la séquence est le résultat d'une fonction de retard appliquée à une autre image de la même séquence. En numérotant les images successives, on peut définir cette fonction retard image de la façon suivante :

$$E^{n-1} = \tau_i(E^n)$$

où E^n représente la $n^{\text{ème}}$ image de la séquence d'entrée.

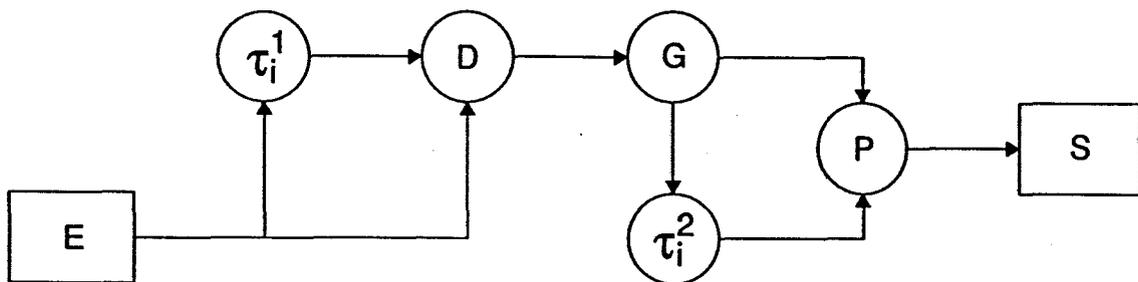
L'opérateur de détection de contours peut alors se représenter de la façon suivante :



Il est encore possible de simplifier l'implantation en remarquant que le fait de retarder les images opérantes avant un calcul est équivalent à retarder le résultat du calcul réalisé sur les images initiales. En fait, cela revient à constater que l'opérateur de retard commute avec toutes les autres fonctions :

$$\tau_i \circ f = f \circ \tau_i$$

L'opérateur de détection des contours fait intervenir deux calculs similaires sur les couples (Image Courante, Image précédente) et (Image courante, Image suivante). Le résultat d'un calcul effectué sur le premier couple est identique au résultat qui a été obtenu par un traitement identique sur le second couple lors du calcul mettant les trois images précédentes de la séquence. On peut alors décrire l'opérateur de détection de contours de la façon suivante :



On s'aperçoit alors que par utilisation judicieuse d'opérateurs de retard sur les images, il est possible d'implanter le détecteur en utilisant uniquement un opérateur de multiplication, un opérateur de détection de contours, et un opérateur de

différence d'images.

La deuxième partie de la méthode présentée par Lee et Aggarwal prévoit un partitionnement de l'algorithme en sous-fonctions indivisibles spatialement. Les opérateurs de multiplication, de calcul du gradient, et de soustraction point à point sont parfaitement localisables, c'est à dire qu'ils ne mettent en jeu que des voisinages restreints de chaque pixel de l'image. Afin de paralléliser le traitement au niveau spatial, il conviendrait d'affecter une unité de calcul à chacun des pixels de l'image. Cette augmentation de la complexité n'est ici aucunement justifiable, et ce pour deux raisons.

La première concerne le cahier des charges que nous avons retenu. Nous avons décidé de favoriser la simplicité globale de l'implantation au détriment d'autres critères tels que l'optimalité de la solution.

Ensuite, la division des opérateurs fonctionnels en éléments simples et localisés ne se justifie que si elle apporte un gain de temps. Dans le cas particulier de l'algorithme qui nous intéresse, les opérateurs mis en jeu restent simples, et peuvent parfaitement être implantés sous forme de circuits pipeline tout en respectant les contraintes dues au traitement en temps-réel.

Le fractionnement de l'algorithme en éléments simples conduit donc à une architecture globale de processeur de type pipeline dont le synoptique est présenté à la figure 3.5.

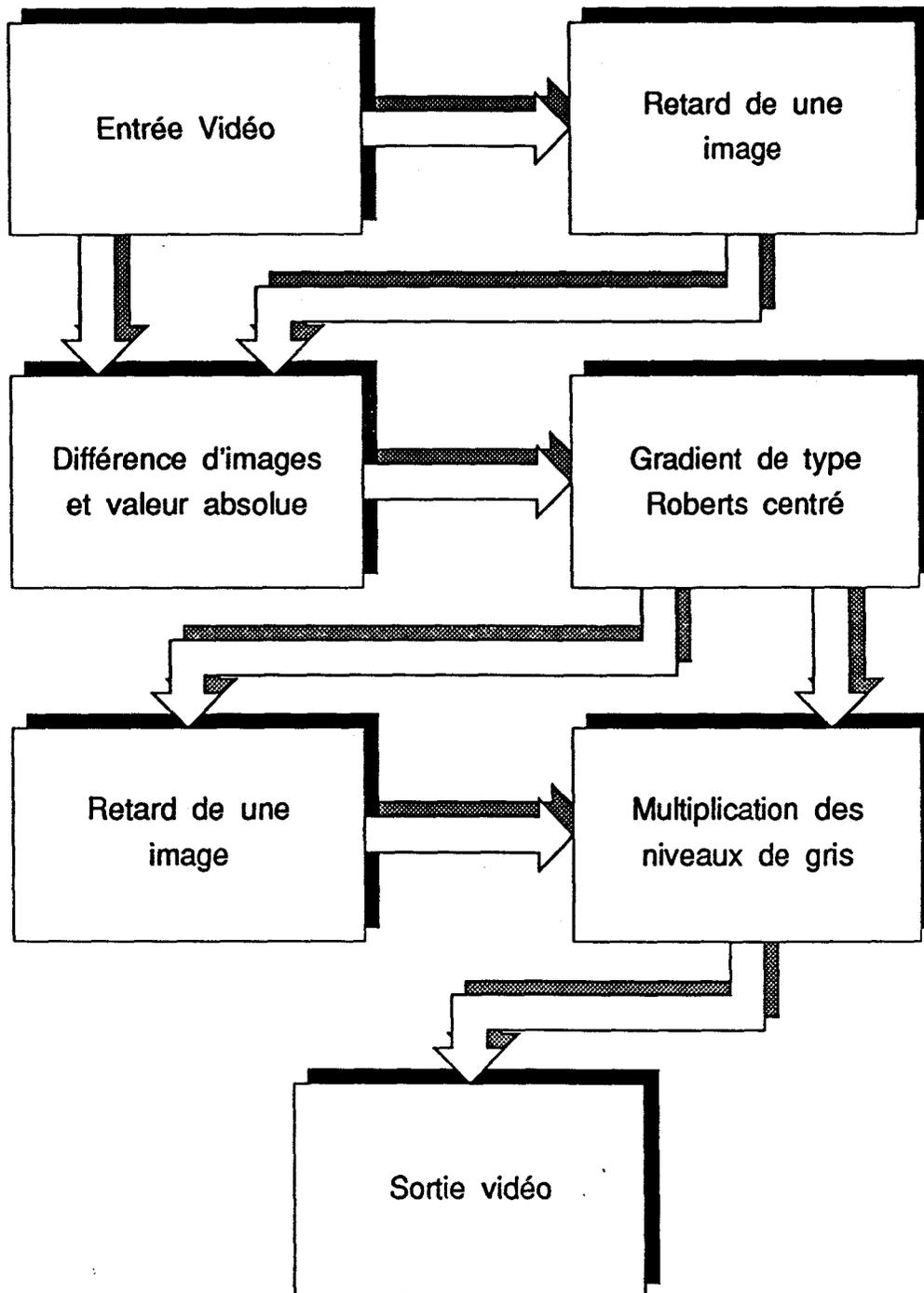


Figure 3.5 : Synoptique du processeur pipeline

III.3) IMPLANTATION MATERIELLE

Les différentes fonctions nécessaires à l'implantation du processeur de détection de contours mobiles en temps-réel ont été implantées sur une série de cartes électroniques. Chaque carte réalise un traitement élémentaire, et les différents modules sont reliés les uns aux autres par un réseau de connexions fixes permettant d'acheminer le flot vidéo numérique. Les schémas électroniques des cartes réalisées sont disponibles en annexe.

III.3.1) MODULE D'ACQUISITION ET DE RESTITUTION VIDEO

Ce module constitue l'interface entre le processeur fonctionnant en technologie numérique et les signaux vidéo analogiques extérieurs répondant à la norme CCIR. Il est chargé principalement de la transformation de ces signaux en flot de données numériques et vice-versa. Ses caractéristiques électroniques sont précisées en annexe A.

La mise en correspondance des coordonnées spatio-temporelles d'un pixel (position dans l'image, et numéro d'image dans la séquence) avec l'instant où il est disponible dans le flot numérique d'entrée est assurée par un certain nombre de signaux de synchronisation extraits du signal vidéo d'entrée par la carte d'acquisition. Ces différents signaux sont utilisés par tous les modules présents dans le système afin d'assurer la synchronisation des traitements avec le flot vidéo numérique.

La résolution spatiale de l'image numérique traitée a été fixée à 256 lignes de 256 pixels, ce qui correspond à un compromis entre la quantité de données traitées à chaque itération et la précision des résultats obtenus. Chaque pixel d'un flot vidéo numérique est codé sur huit bits, ce qui permet de disposer d'une dynamique de 256 valeurs, tant pour les valeurs correspondant aux pixels des images d'entrée et de sortie, que pour toutes celles échangées entre les modules de traitement.

La carte d'acquisition vidéo se charge du positionnement d'une fenêtre de calcul dans la séquence d'images initiale. Ce fenêtrage permet de limiter la zone active de l'image au niveau spatial, par inhibition des pixels situés en dehors de cette fenêtre spatiale.

Un fenêtrage temporel, opère d'autre part en désactivant les autres modules pendant la durée de certaines images de la séquence d'entrée, ce qui permet d'obtenir une séquence de cadence plus basse. Cette dernière fonctionnalité a été ajoutée afin de permettre d'analyser des déplacements d'objets très lents qui ne seraient pas perceptibles entre les images d'une séquence vidéo saisie à la cadence standard de vingt-cinq images par seconde.

Afin de conserver la continuité du signal vidéo de sortie, le module de restitution comporte une mémoire d'image permettant de restituer le dernier résultat de l'estimation des contours en mouvement durant les périodes d'inhibition du signal pilotant la fenêtre temporelle.

III.3.2) MODULES DE RETARD D'IMAGE

Les modules de retard d'image ne réalisent pas à proprement parler de traitements sur le flot vidéo numérique. Ils servent à implanter l'opérateur de retard τ_i décrit au paragraphe III.2.3. Comme nous l'avons vu précédemment, deux modules de ce genre sont nécessaires à l'implantation du détecteur de contours mobiles.

Un flot vidéo numérique pris en compte par un module de retard d'image est restitué par ce dernier sous forme d'un autre flot vidéo numérique, synchrone, mais où tous les pixels sont décalés temporellement d'une image, soit un vingt-cinquième de seconde. Le module n'introduit pas de retard spatial au niveau de l'image, le pixel restitué en sortie à un instant donné étant situé à la même position que le pixel présent en entrée, mais dans l'image active précédente.

La fonction est implantée par utilisation d'une mémoire d'image balayée en continu. Cela correspond en fait à une structure FIFO (First In, First Out) de grande taille. L'adresse mémoire est incrémentée à chaque cycle élémentaire de traitement, et le pixel situé en entrée du module prend la place de celui présent dans la case mémoire correspondante. Ce dernier, qui avait été mémorisé lors du balayage de l'image précédente est restitué en sortie durant le même cycle élémentaire de calcul. La description électronique de ce module fait l'objet de l'annexe B.

III.3.3) MODULES ARITHMETIQUES

Les modules arithmétiques sont utilisés afin de combiner les valeurs issues de deux flots vidéo numérique par une opération ponctuelle. Ils permettent l'implantation de l'opération de soustraction suivie du calcul de la valeur absolue, et de l'opération de multiplication.

Les flots numériques envoyés aux deux entrées de ce module doivent correspondre à des pixels situés à une même position spatiale dans l'image.

L'opération proprement dite est implantée par l'intermédiaire de tables de correspondance [Ull81], ou Look Up Tables (LUT). Les deux opérandes arrivant en entrée du module de combinaison arithmétique sont groupés afin de constituer une adresse de 16 bits permettant de sélectionner le résultat de l'opération parmi les 2^{16} combinaisons possibles. La seule différence existant entre deux modules arithmétiques réside dans le contenu de cette table de correspondance. Nous avons choisi ce principe de fonctionnement autant pour sa simplicité de mise en oeuvre (utilisation de mémoires EPROM de grande taille), que pour la possibilité de faire évoluer simplement les opérateurs implantés. Les schémas électroniques de ce module sont contenus dans l'annexe C.

III.3.4) MODULE DE TRAITEMENT DE VOISINAGE

Ce module permet actuellement de traiter un flot vidéo numérique afin d'en extraire une estimation du gradient de l'image par l'opérateur de Roberts centré décrit au chapitre II. Nous avons vu que cet opérateur est loin d'être le plus efficace, cependant il présente l'avantage de pouvoir être implanté en utilisant un nombre limité d'opérateurs arithmétiques simples.

L'opérateur de Roberts centré est divisible au point de vue temporel, c'est à dire qu'il peut être construit en utilisant plusieurs fonctions arithmétiques simples reliées entre elles par un réseau pipeline. Une littérature abondante présente les différentes possibilités permettant de réaliser l'implantation matérielle des opérateurs de voisinage, et en particulier les opérations de convolution [Ull81, Ber85, Pra87, Hei89, Asa90].

Les techniques correspondantes utilisent en général les propriétés particulières du flot vidéo numérique issues du balayage ligne par ligne utilisé en télévision, qui permet de ramener l'image à une seule dimension. Une image est transformée en un flot continu de données, dans lequel chaque pixel est disponible pendant une durée Δt , le balayage étant réalisé de haut en bas, ligne par ligne (axe des y), et de gauche à droite sur chacune des lignes (axe des x). Si chaque ligne de l'image initiale comporte N colonnes, la relation liant la coordonnée temporelle T d'un pixel dans le flot à ses coordonnées spatiales (ligne,colonne) est la suivante :

$$T = ((n^\circ \text{ de ligne}) \times N + (n^\circ \text{ de colonne})) \cdot \Delta t$$

Afin d'accéder de façon simultanée à différents pixels d'un voisinage spatial d'une image transformée en flot vidéo, il suffit de retarder ce dernier en synchronisme avec la cadence de balayage. En effet, la relation liant la coordonnée temporelle d'un pixel à ses coordonnées spatiales permet de montrer qu'une translation dans l'espace image est similaire à un retard dans le flot numérique. Les translations élémentaires T_x et T_y permettant d'accéder aux pixels situés immédiatement à la gauche et au dessus d'un pixel de l'image correspondent à deux fonctions de retard du flot vidéo appelés respectivement retard pixel (τ_x) et retard ligne (τ_y).

En ajoutant à ces deux fonctions de retard spatial la fonction de retard temporelle τ_i décrite au paragraphe III.2.3, permettant de disposer simultanément de deux pixels situés à la même position dans deux images d'une séquence, on dispose de tous les moyens nécessaires afin d'accéder au voisinage spatio-temporel de chacun des pixels du flot vidéo. Les relations suivantes résument l'effet de ces fonctions de retard élémentaires sur le flot vidéo :

$$P(x,y,i-1) = \tau_i (P(x,y,i))$$

$$P(x,y-1,i) = \tau_y (P(x,y,i))$$

$$P(x-1,y,i) = \tau_x (P(x,y,i))$$

$P(x,y,i)$ représentant le pixel de coordonnées spatiales (x,y) situé dans la i^{ème} image de la séquence.

L'opérateur de Roberts centré utilisé par le détecteur de contours en mouvement combine les valeurs des quatre voisins du pixel $P(x,y)$ en cours de

traitement selon l'équation suivante :

$$G_r(P(x,y,i)) = \max(|P(x-1,y-1,i) - P(x+1,y+1,i)|, |P(x+1,y-1,i) - P(x-1,y+1,i)|)$$

En considérant que les voisins du pixels central sont obtenus à partir de ce dernier en utilisant les translations élémentaires T_x et T_y précédemment définies, on peut transformer l'expression précédente en :

$$G_r(P(x,y,i)) = \max(|T_x \circ T_y \cdot T_x^{-1} \circ T_y^{-1}|, |T_x^{-1} \circ T_y \cdot T_x \circ T_y^{-1}|)(P(x,y,i))$$

où l'exposant T^{-1} désigne la translation réciproque (ou inverse) de T , T_x^{-1} et T_y^{-1} permettant d'obtenir les pixels situés respectivement à droite et en dessous du pixel considéré.

En transformant les translations du domaine spatial en retards dans le domaine flot de données, l'opérateur de Roberts centré utilisé par le détecteur de contours en mouvement peut alors être noté :

$$G_r = \max(|\tau_x^{-1} \circ \tau_y^{-1} - \tau_x \circ \tau_y|, |\tau_x \circ \tau_y^{-1} - \tau_x^{-1} \circ \tau_y|)$$

Cependant, l'accès aux pixels situés en dessous ou à droite du pixel courant du flot vidéo est physiquement impossible, ces derniers n'ayant pas encore été atteints par le balayage. Les fonctions de retard τ_x^{-1} et τ_y^{-1} ne sont donc pas réalisables. Afin de lever cette impossibilité, il convient d'accepter un retard global de une ligne et de un pixel au niveau du résultat, ce qui revient à décaler spatialement le centre du masque caractérisant l'opérateur de détection de contours. Le résultat de l'estimation du gradient au point $(x-1,y-1)$ devient alors disponible lors de l'apparition du pixel (x,y) sur le flot vidéo en utilisant le détecteur suivant :

$$G_r' = \max(|I - \tau_x^2 \circ \tau_y^2|, |\tau_x - \tau_y|)$$

où I représente la fonction identité.

Le module de traitement de voisinage réalise l'opérateur de Roberts centré en utilisant ce principe de fonctionnement. L'accès simultané aux différents pixels d'un voisinage 3×3 de l'image est rendu possible par utilisation de retards pixel (implantés par l'intermédiaire de simples bascules) et de retards ligne (réalisés de façon similaire aux retards d'image). Les pixels adéquats sont ensuite combinés par des opérateurs arithmétiques simples. La description de ce module est fournie dans l'annexe D.

III.4) CONCLUSION

L'algorithme de détection de contours en mouvement défini par Vieren a été partitionné en unités de calcul indépendantes organisées en structure MISD pipeline. Les connexions entre les différentes unités de traitement véhiculent des données correspondant à un flot vidéo numérique obtenu par balayage des images selon le standard utilisé en télévision.

Nous avons implanté ces unités de traitement sur différentes cartes électroniques, chacune d'entre elles réalisant un traitement spécifique. Les modules sont reliés les uns aux autres par un réseau de connexions véhiculant les informations sous forme de flots cadencés de données.

L'architecture du processeur STREAM s'avère être relativement flexible, bien que cette caractéristique n'ait pas été retenue comme critère de choix lors de la conception de l'architecture dédiée.

CHAPITRE IV

APPORT DU PROCESSEUR STREAM POUR L'AMELIORATION DE LA SECURITE DANS LES SYSTEMES DE TRANSPORT GUIDES

Le processeur STREAM, permettant la détection en temps réel des contours en mouvement contenus dans une séquence d'images, a été testé sur site réel lors de plusieurs expérimentations concernant les systèmes de transport guidés.

Tout d'abord, il a été utilisé afin de détecter en temps réel les objets mobiles se déplaçant sur un carrefour urbain. L'objectif de l'étude est de vérifier l'efficacité de la méthode de détection de contours en vue de la réalisation d'un capteur spécifique pour le projet de conduite automatique d'un tramway, présenté au début de ce mémoire (cf. chapitre I).

Le STREAM a ensuite été utilisé en tant que système de surveillance des voies d'une station de métro. Le but de l'expérimentation est de montrer qu'il est possible de détecter automatiquement les chutes accidentelles de passagers sur les voies de métro non isolées des quais par des portes palières.

IV.1) DETECTION ET LOCALISATION DES OBJETS MOBILES ÉVOLUANT DANS UN CARREFOUR URBAIN

IV.1.1) CONDITIONS D'EXPERIMENTATION

Le processeur STREAM a été utilisé afin de traiter des séquences d'images visualisant la situation des véhicules présents sur un carrefour urbain. L'intersection choisie est traversée par les voies de la ligne de tramway qui relie Lille à Roubaix. Certaines séquences d'images ont été enregistrées sur magnétoscope, puis traitées en laboratoire.

Lors de l'expérimentation, la caméra était placée dans un immeuble situé en bordure du carrefour, à environ 15m de hauteur. Les conditions idéales d'acquisition précisées par M^{TS} Stelmaszyk et Vieren [Ste85, Vie88], qui prévoyaient une caméra située à la verticale du carrefour, n'étaient donc pas parfaitement respectées. En fait, il est très difficile de disposer d'un support de caméra satisfaisant la contrainte de parfaite stabilité. En effet, les pylônes d'éclairage ou même des mâts spécifiques utilisés pour fixer la caméra au dessus du carrefour, se sont avérés trop sensibles aux effets du vent.

Lors de l'enregistrement des séquences d'images, aucun dispositif d'éclairage artificiel n'a été utilisé. Les variations globales de la luminosité ambiante étaient atténuées par un dispositif d'auto-iris présent sur la caméra CCD utilisée.

IV.1.2) RESULTATS DE L'EXPERIMENTATION

Les résultats obtenus lors de cette expérimentation sur site réel du processeur sont présentés ci-après sous la forme de trois séquences de seize images extraites de la bande vidéo. L'algorithme de détection des contours en mouvement a été appliqué à tous les groupes possibles de trois images consécutives d'une séquence. Cela explique la présence de seulement quatorze images résultat pour seize images initiales, l'estimation des contours mobiles étant impossible sur la première et la dernière image de chaque séquence.

Les images résultat présentées ont été filtrées par un opérateur de moyennage des niveaux de gris. L'objectif de ce traitement est d'améliorer la connexité des contours [Vie88]. Le niveau de gris de chaque pixel de l'image de contours fournie par le STREAM est additionné à celui de ses voisins directs, le résultat étant ensuite divisé par le nombre de points présents dans le voisinage. Cette opération permet à la fois de diminuer le taux de fausses détections dues au bruit, et d'augmenter la continuité des contours.

IV.1.2.a) Séquence n° 1

La séquence d'images numéro 1, présentée sur les figures 4.1a et 4.1b, montre une scène sur laquelle se déplacent deux véhicules automobiles, l'un de couleur noire, l'autre de couleur blanche. Le résultat de la détection des contours en mouvement correspondant à cette séquence est présenté sur les figures 4.2a et 4.2b.

On remarque tout d'abord que la détection des contours en mouvement est réalisée sans ambiguïté quelle que soit la couleur des mobiles. Cette caractéristique de l'algorithme découle du fait que la procédure de détection de contours mobiles utilise des opérateurs différentiels, sensibles uniquement au contraste existant entre un objet et le fond, et non pas au niveau de gris propre de cet objet.

Il importe aussi de noter que le fond de l'image, constitué de la chaussée revêtue de macadam, et des deux voies de la ligne de tramway, comporte des zones de niveaux de gris très différents. En dépit de ce fond non homogène et de sa forte texture, la détection des objets mobiles a été réalisée sans erreur.

Au devant de chacun des deux véhicules, on peut constater la présence d'une zone d'ombre due aux propriétés directionnelles de l'éclairage ambiant (lumière du jour). Il s'avère impossible de différencier cette zone d'ombre des autres parties du véhicule en utilisant uniquement les informations contenues dans la séquence d'images. Sur les images résultat, cette zone souvent très contrastée, entraîne l'apparition de contours mobiles parfois plus prononcés que ceux créés par le véhicule. Dans l'application qui nous concerne, cette détection parasite ne constitue pas un défaut majeur.

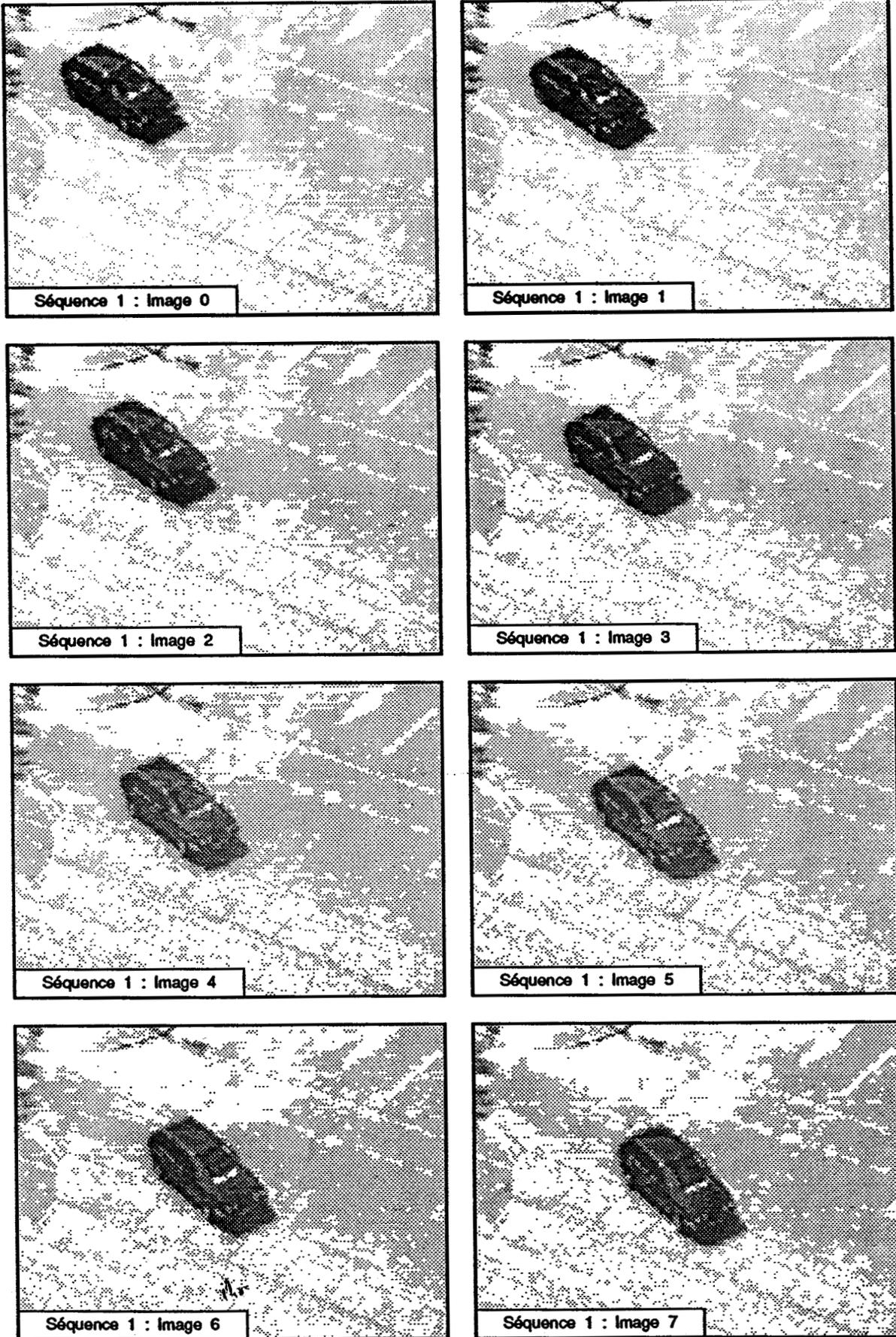


Figure 4.1a : Séquence 1, images initiales de 0 à 7

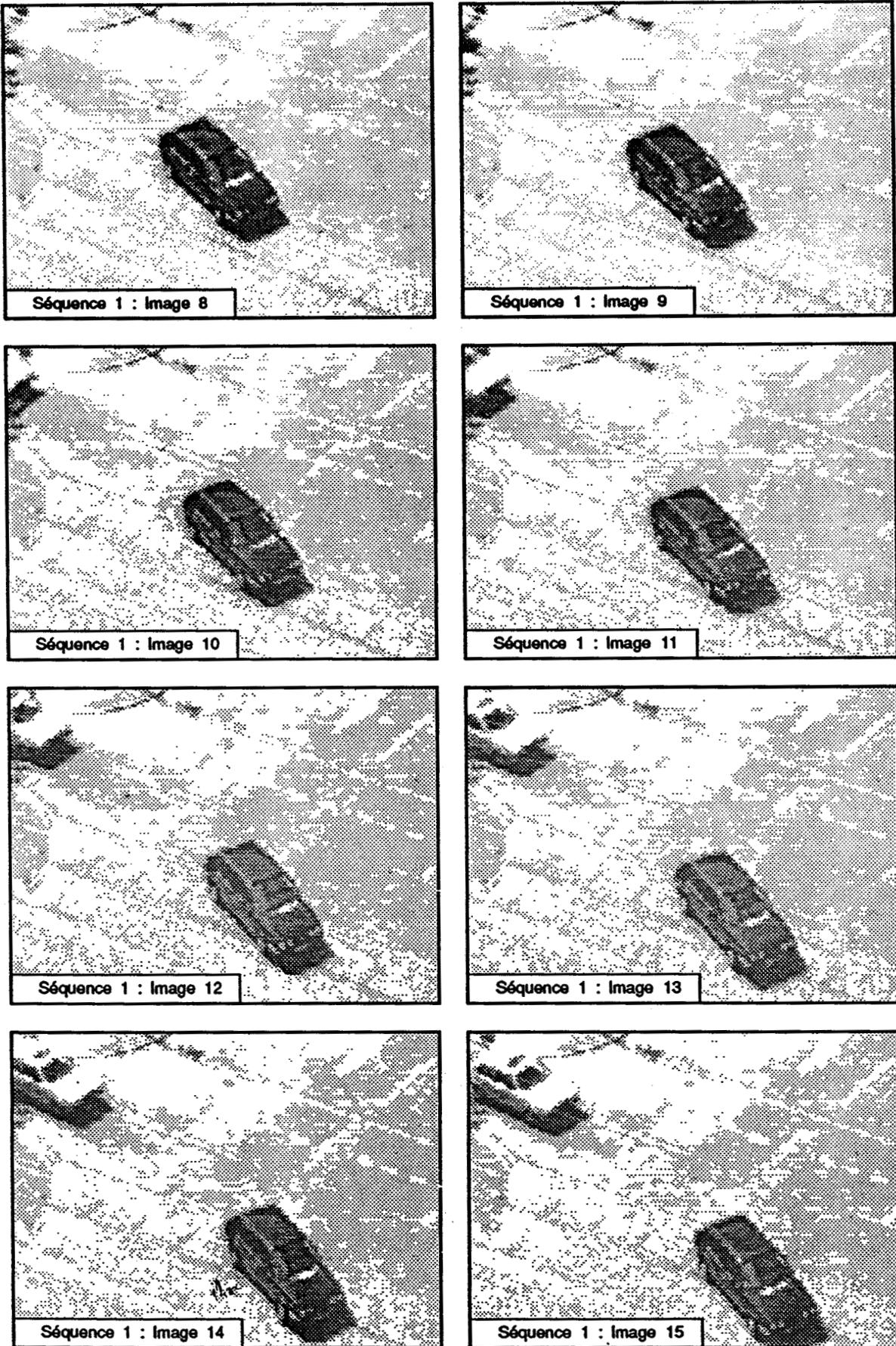


Figure 4.1b : Séquence 1, images initiales de 8 à 15

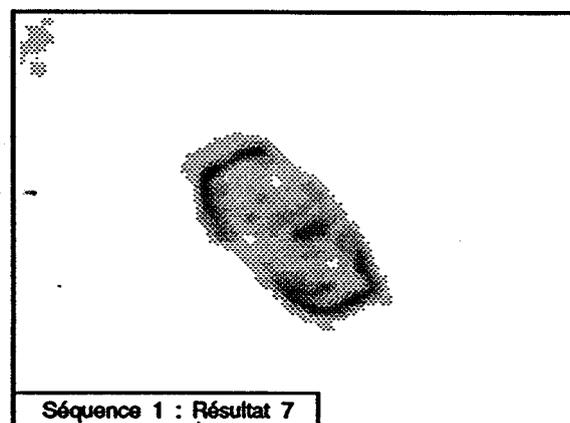
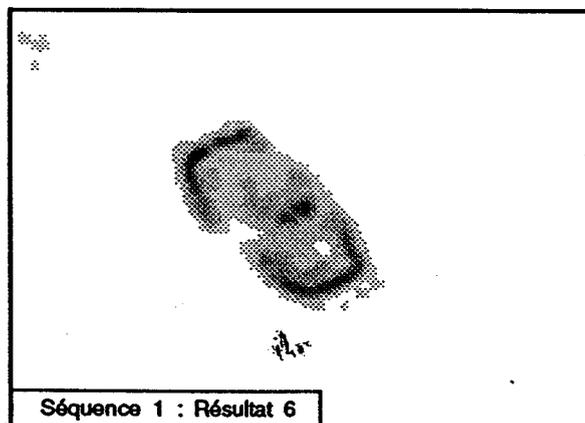
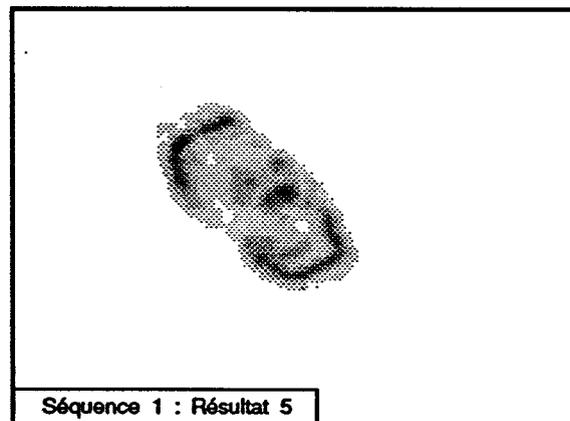
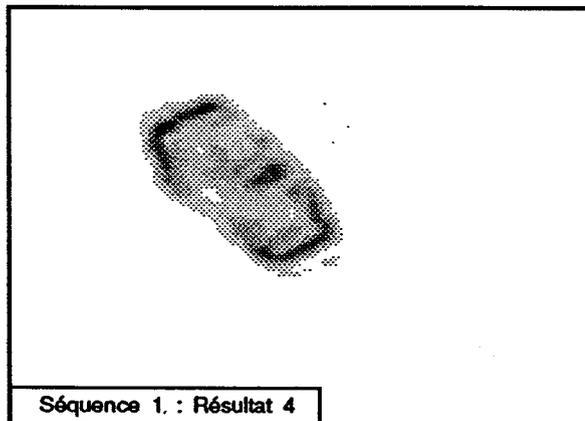
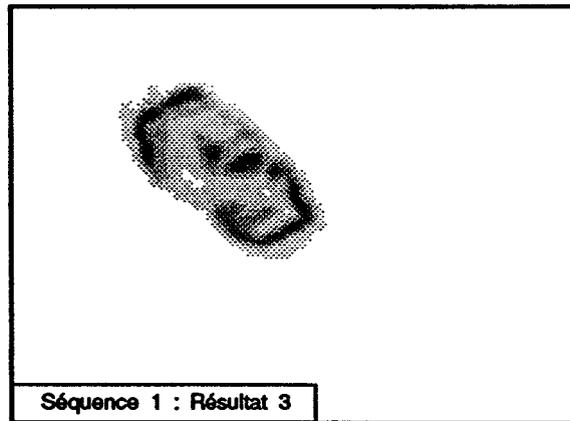
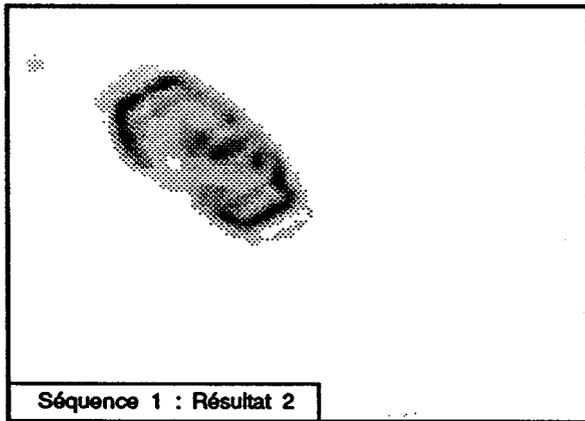
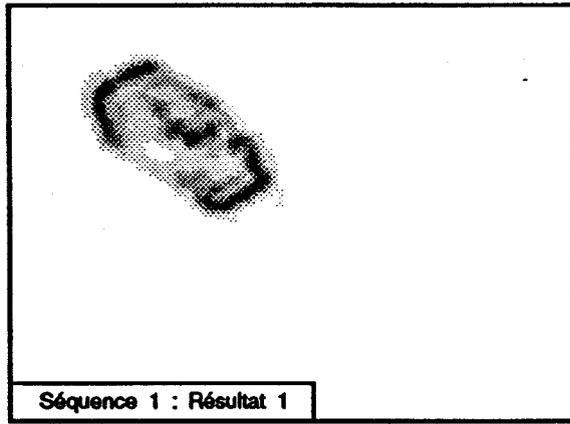
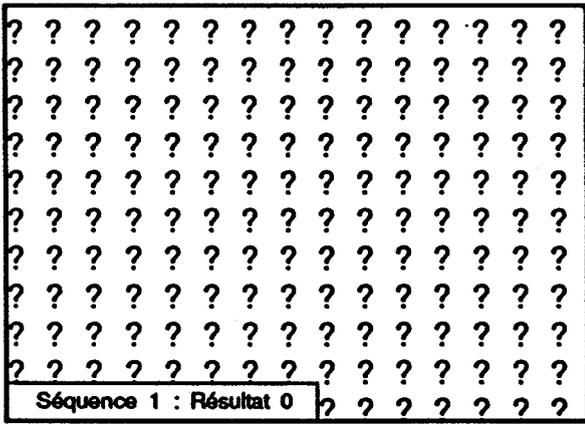


Figure 4.2a : Séquence 1, images résultat de 1 à 7

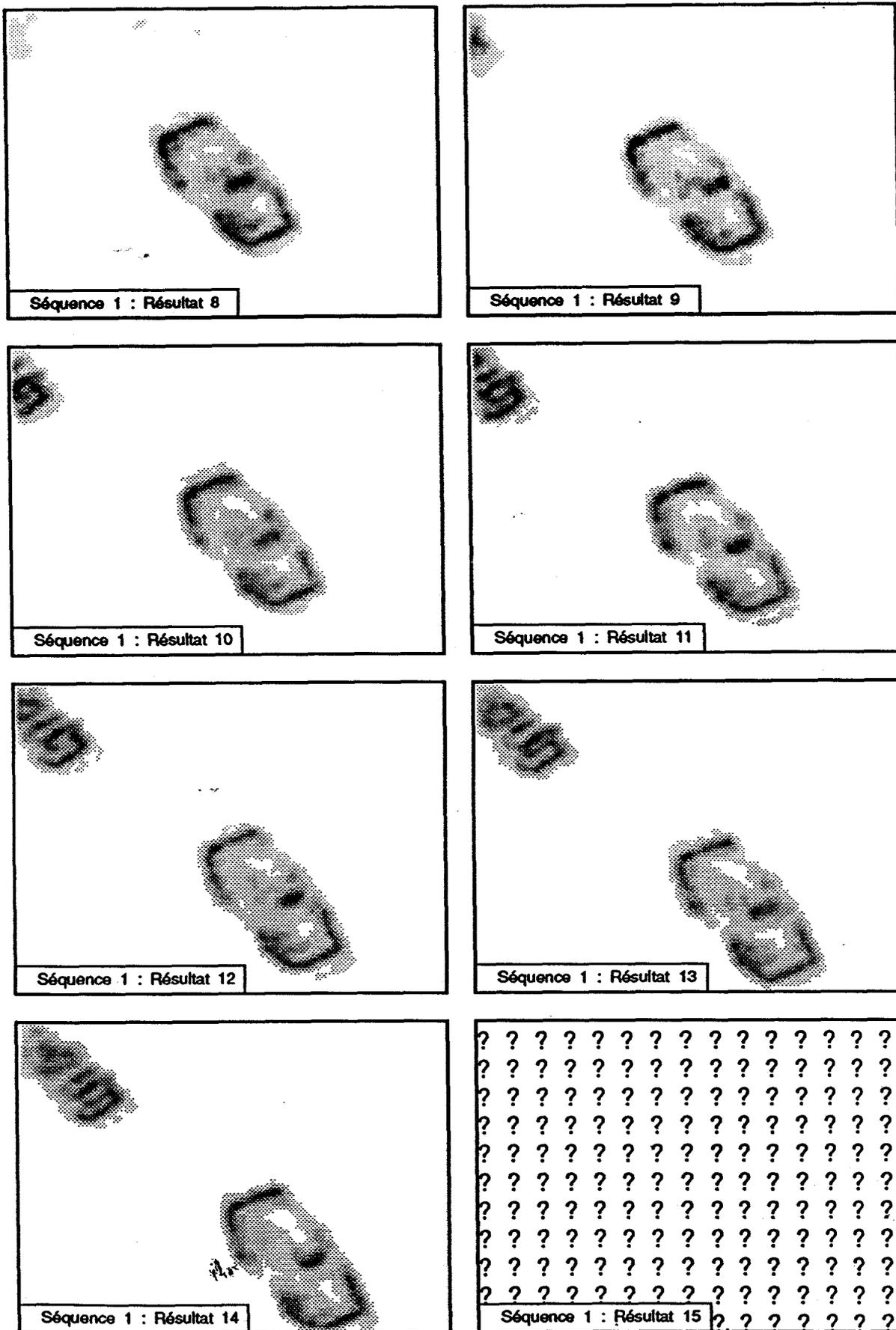


Figure 4.2b : Séquence 1, images résultat de 8 à 14

IV.1.2.b) Séquence n° 2

Dans la séquence numéro 2, présentée sur les figures 4.3a et 4.3b, plusieurs véhicules de forme et de taille variées circulent sur le carrefour. Les mobiles ne se déplacent pas tous dans le même sens. Le résultat de la détection des contours en mouvement sur cette séquence d'images est présenté sur les figures 4.4a et 4.4b.

Comme dans la séquence précédente, tous les véhicules en mouvement présents dans la scène sont parfaitement détectés.

Sur les trois premières images résultat (résultats 1, 2 et 3 de la figure 4.4a), il est impossible de différencier les contours en mouvement dus à la fourgonnette, circulant vers l'avant de la scène, de ceux dus à la voiture blanche qui se déplace vers l'arrière de l'image. Si l'on considère qu'il est impossible de différencier l'ombre de la fourgonnette du véhicule lui-même, on constate que sur la séquence initiale, les images des deux mobiles se touchent. Bien que les sens de déplacement soient différents, il est alors impossible de distinguer les deux véhicules.

IV.1.2.c) Séquence n° 3

Dans la séquence numéro 3, présentée sur les figures 4.5a et 4.5b, deux voitures se déplacent sur le carrefour. De plus, on peut remarquer les deux piétons qui marchent sur le trottoir situé à gauche de l'image. Le résultat de la détection des contours en mouvement sur cette séquence d'images est présenté sur les figures 4.6a et 4.6b.

On constate que les voitures, ainsi que les deux piétons sont parfaitement détectés. Il est possible de suivre le déplacement des deux personnes de façon fiable tout au long de la séquence, malgré leur petite dimension et leur faible vitesse de déplacement.

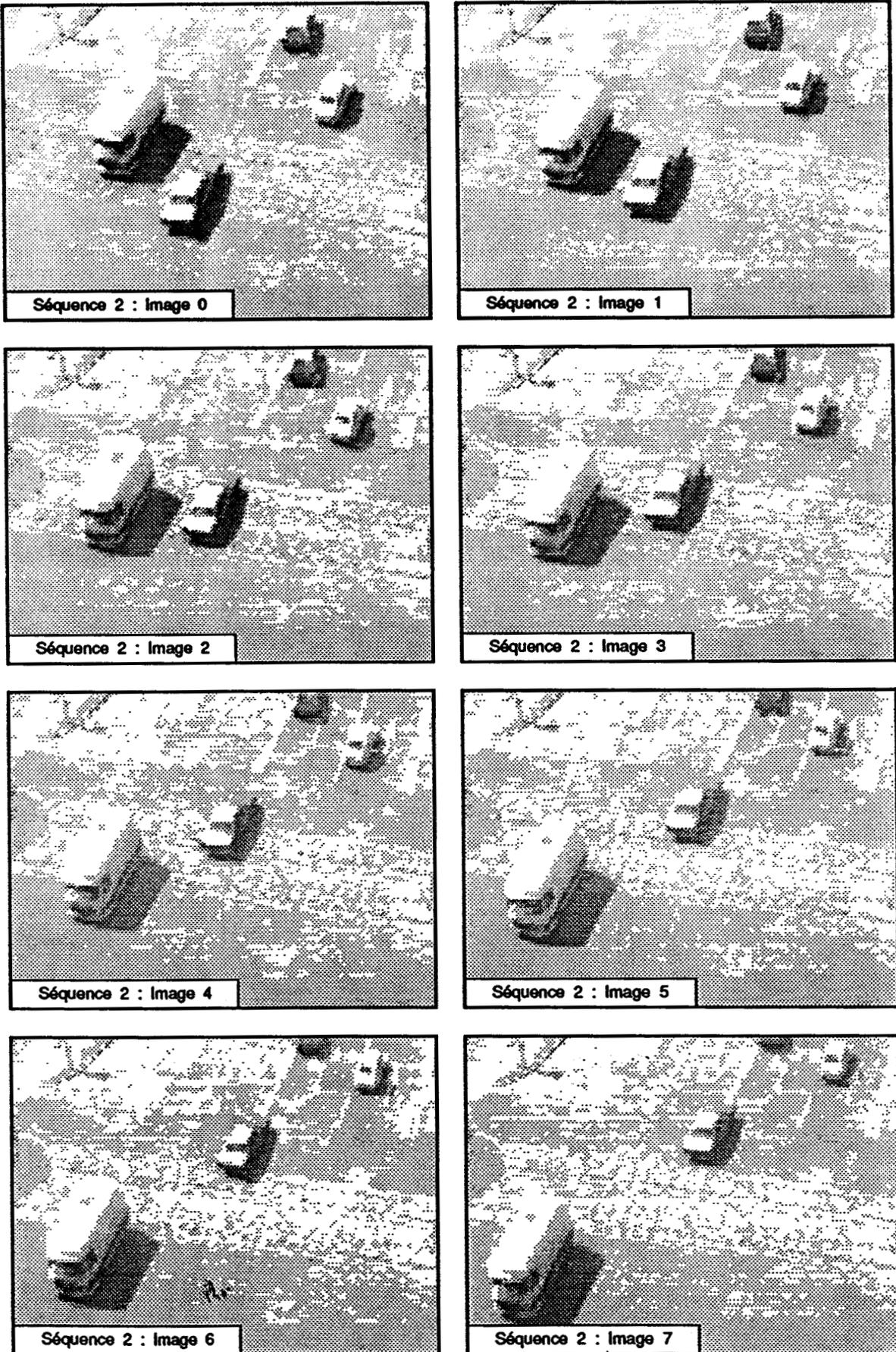


Figure 4.3a : Séquence 2, images initiales de 0 à 7

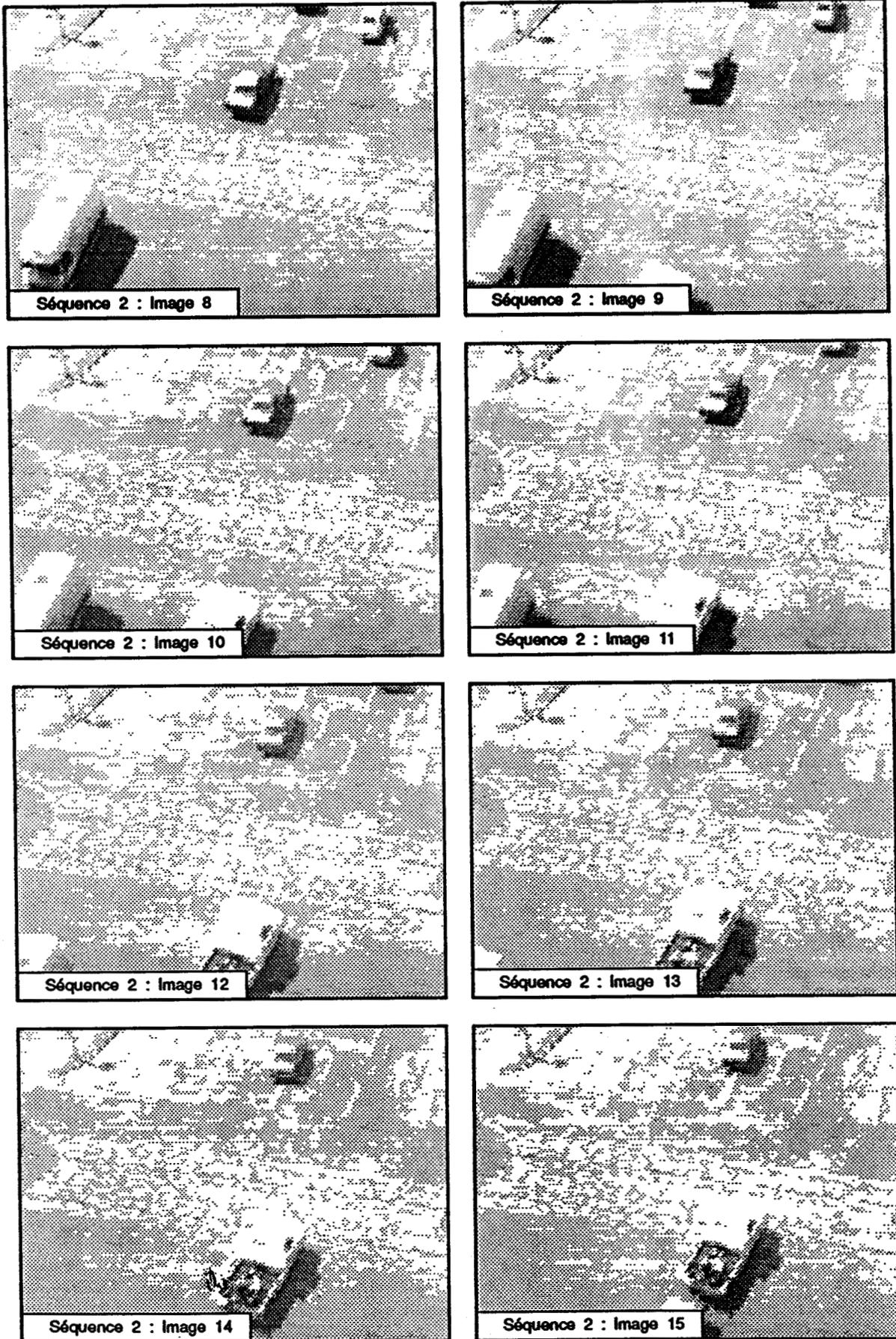


Figure 4.3b : Séquence 2, images initiales de 8 à 15

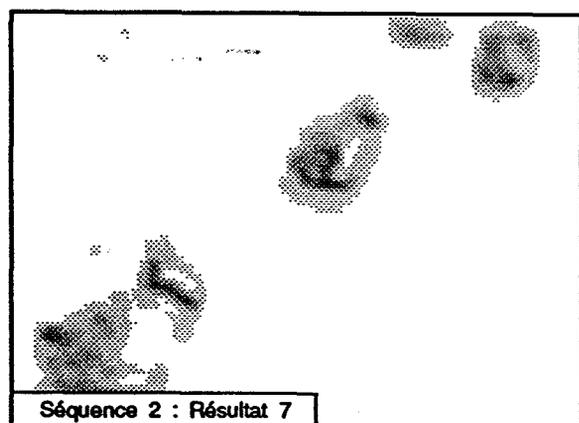
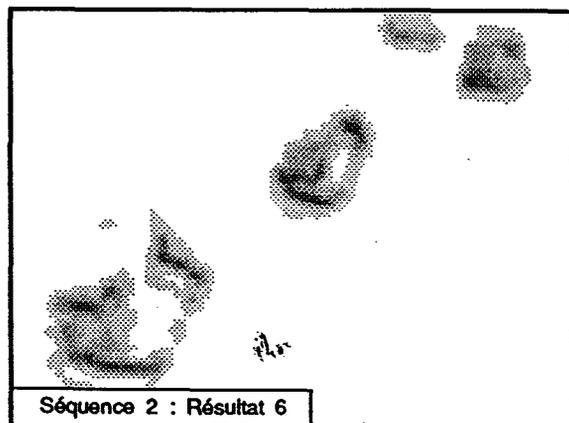
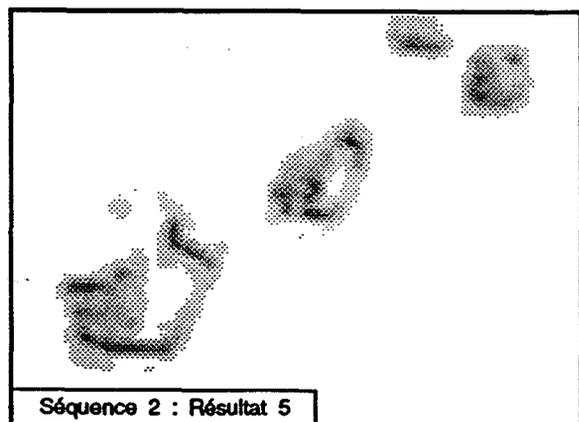
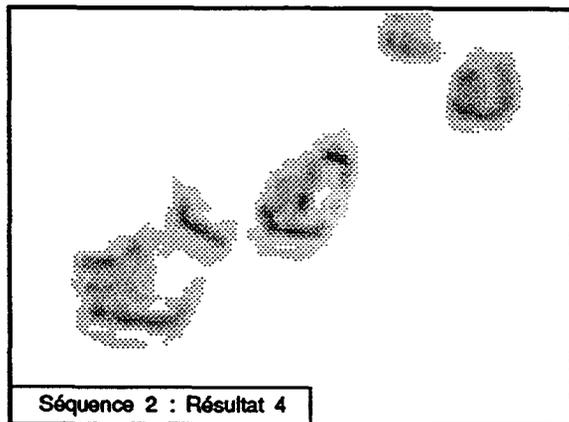
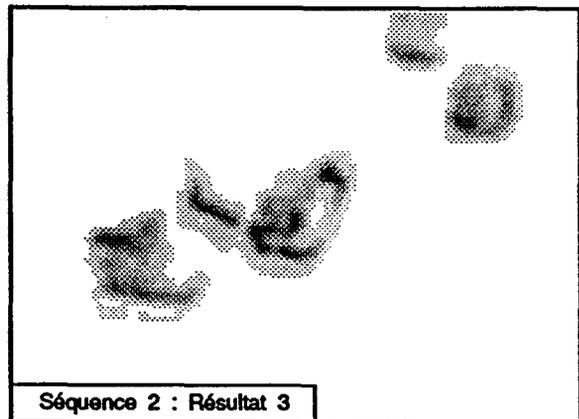
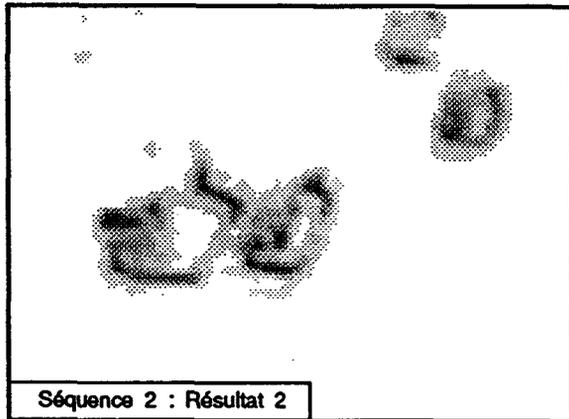
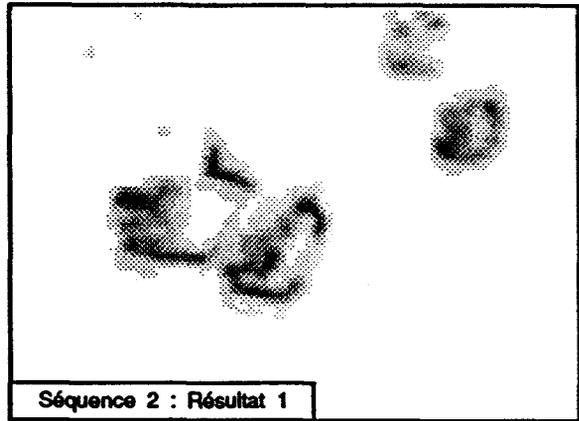
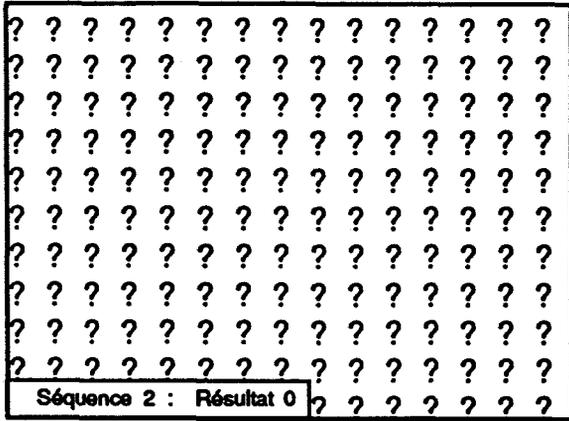


Figure 4.4a : Séquence 2, images résultat de 1 à 7

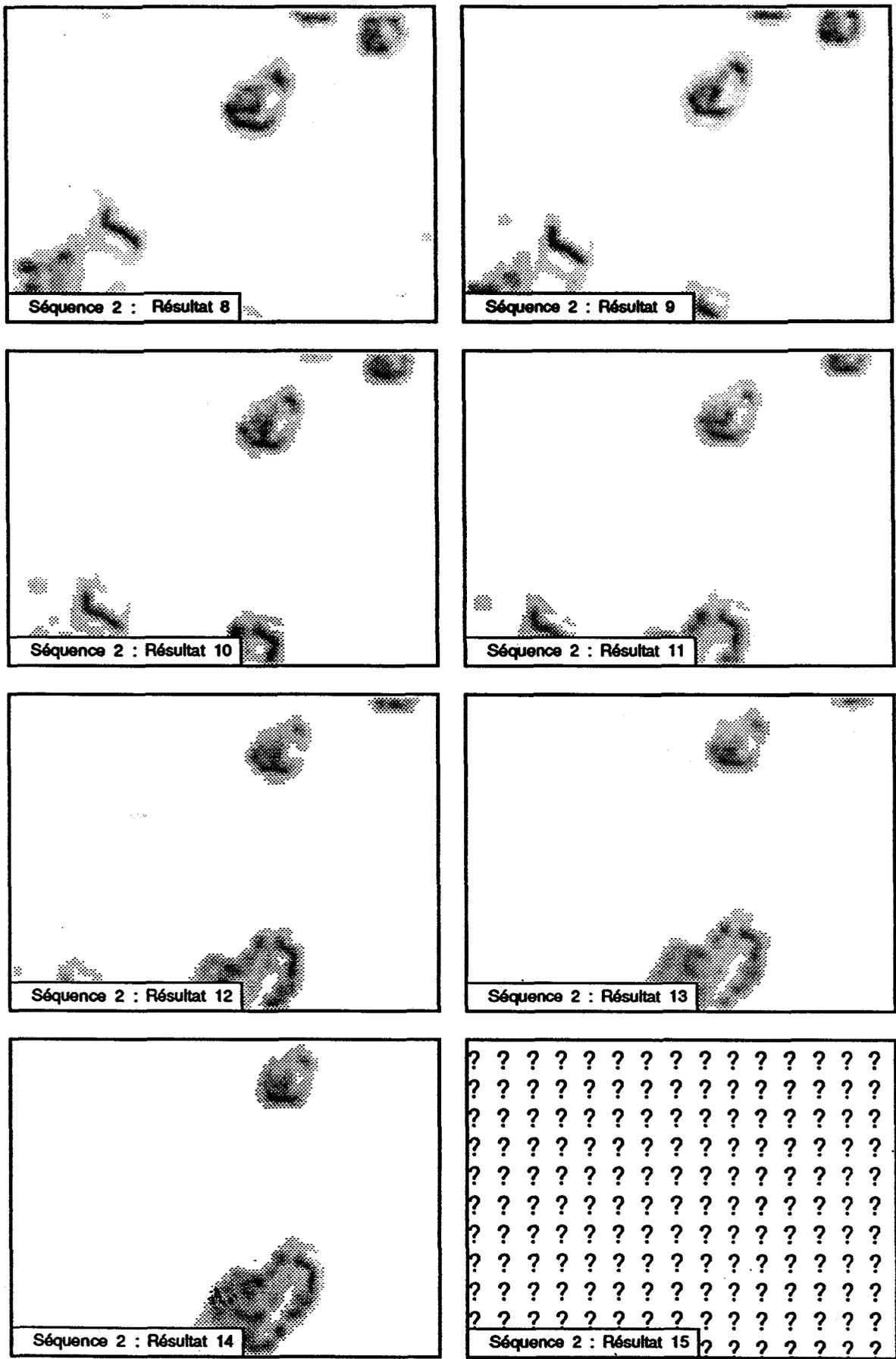


Figure 4.4b : Séquence 2, images résultat de 8 à 14

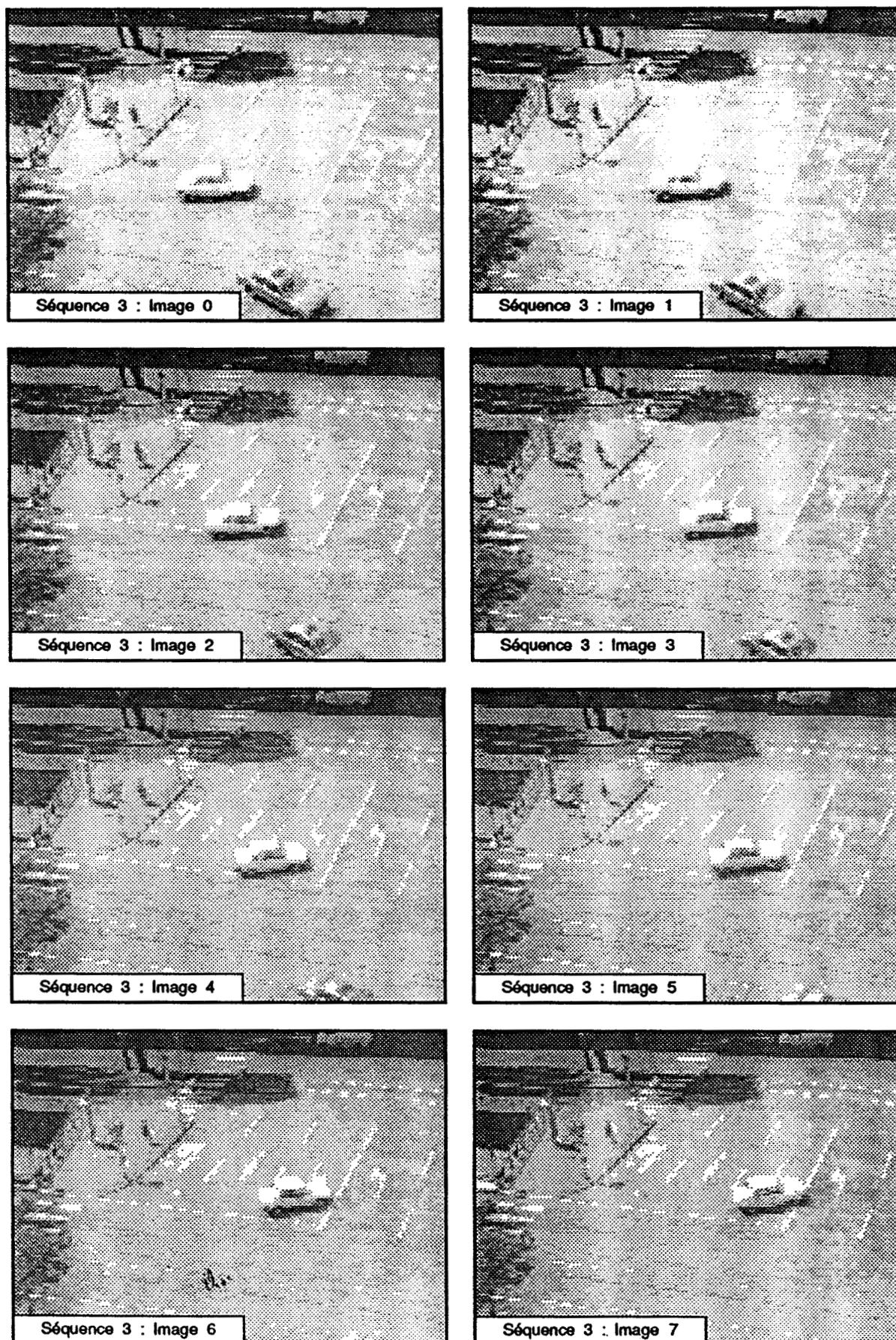


Figure 4.5a : Séquence 3, images initiales de 0 à 7

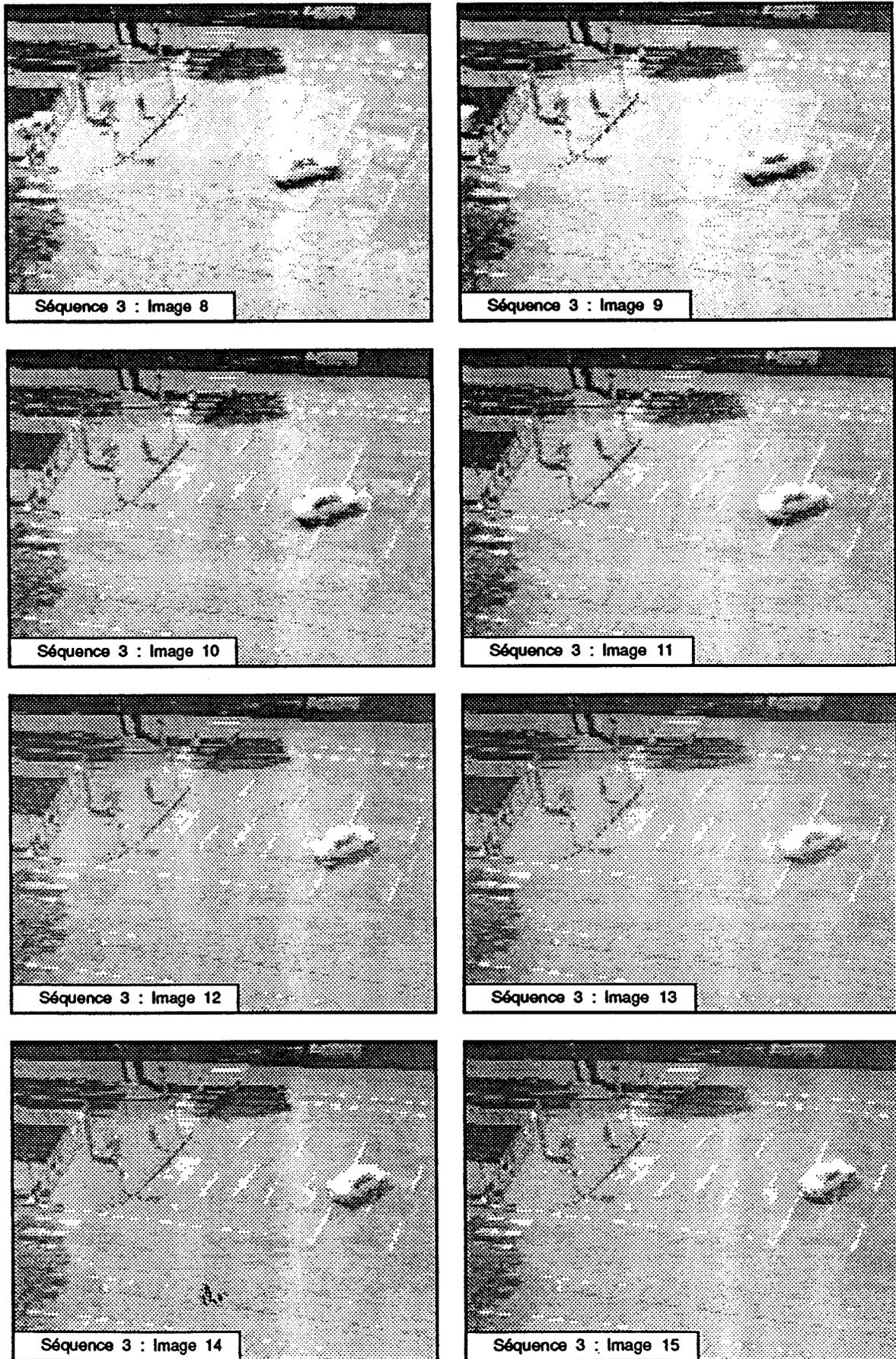


Figure 4.5b : Séquence 3, images initiales de 8 à 15

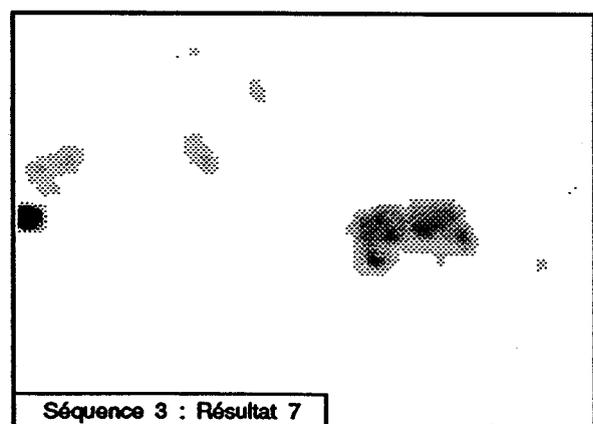
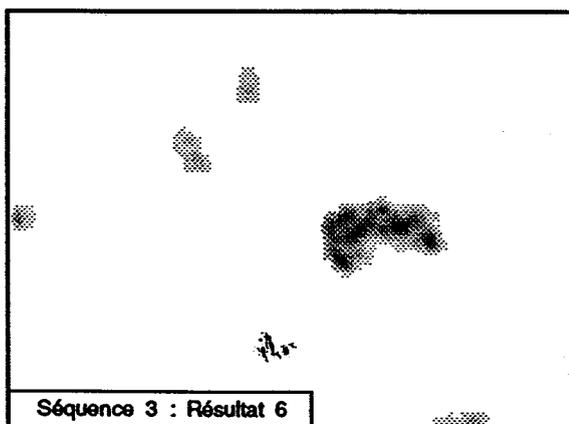
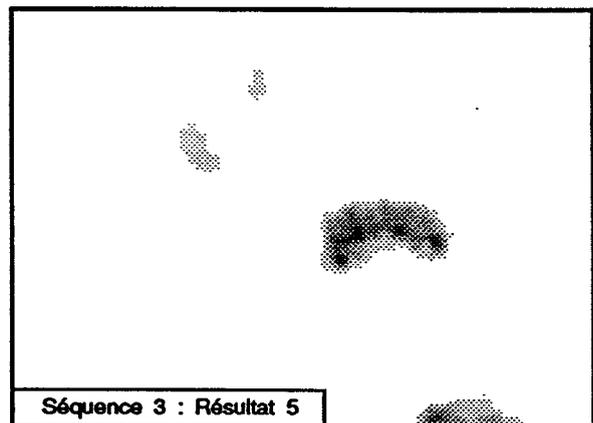
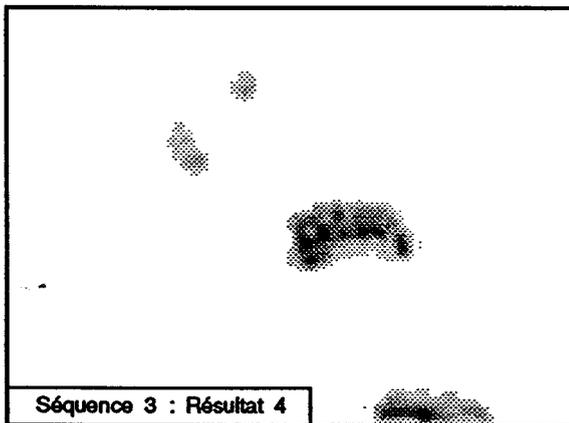
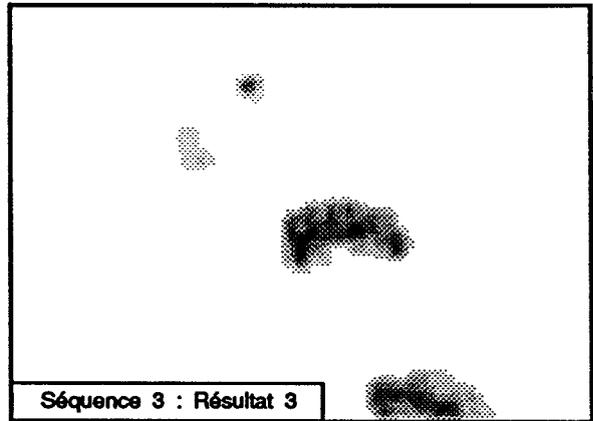
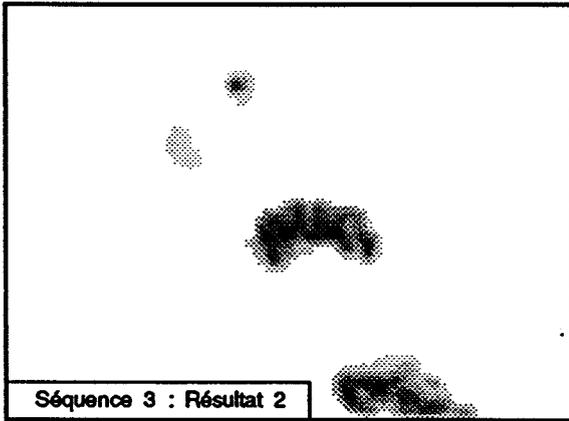
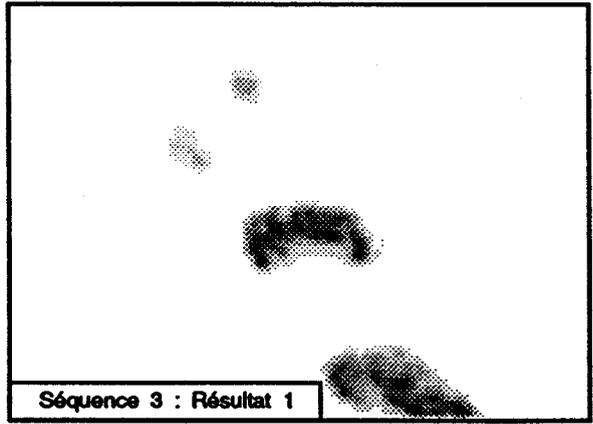
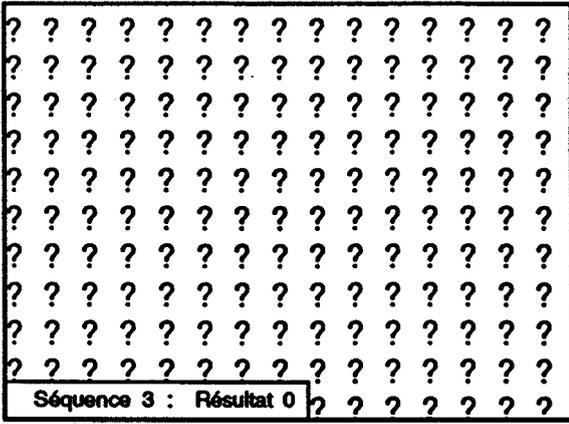


Figure 4.6a : Séquence 3, images résultat de 1 à 7

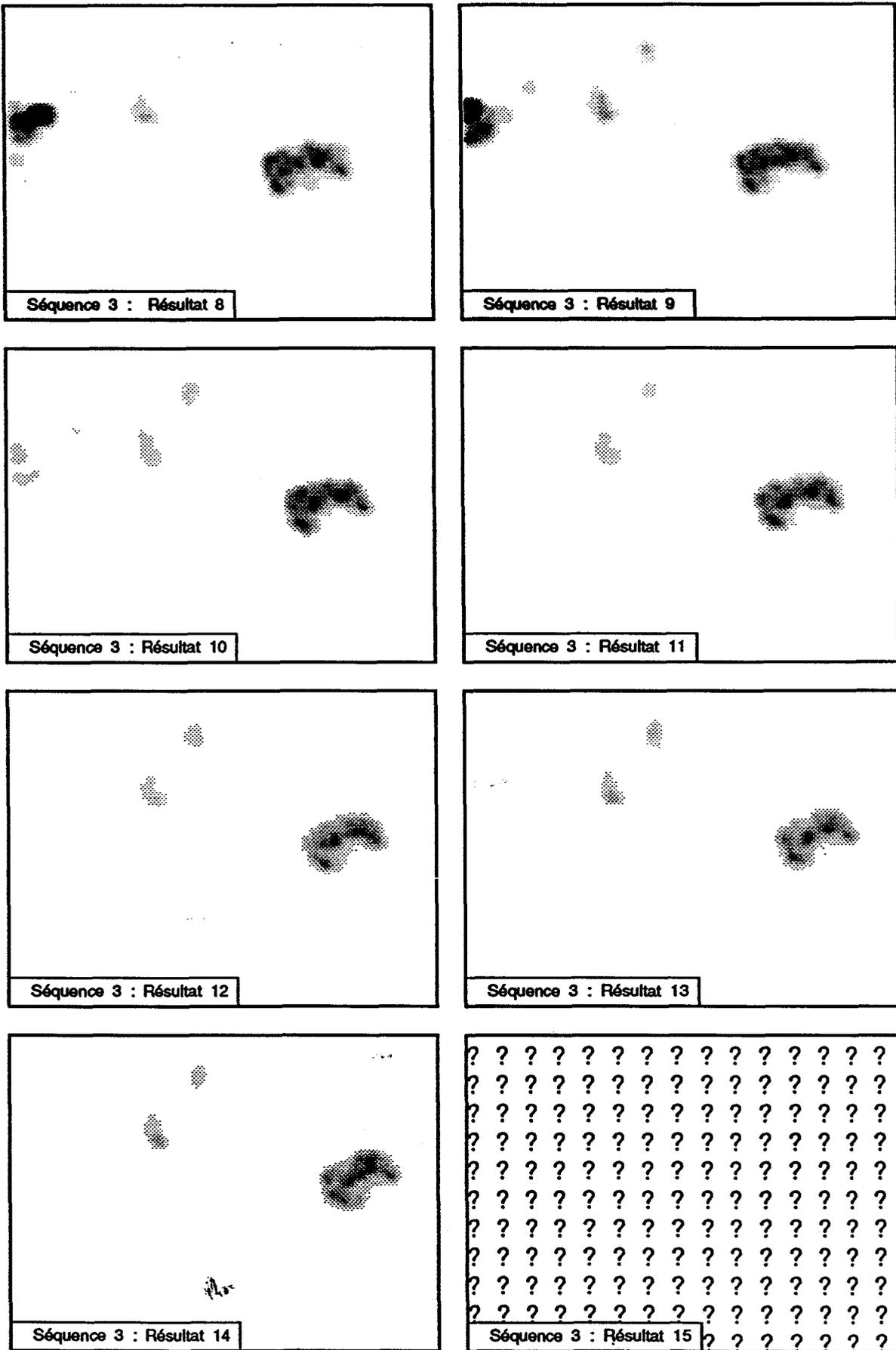


Figure 4.6b : Séquence 3, images résultat de 8 à 14

Sur les images résultat 6 à 10, les contours mobiles détectés sur le coté gauche sont dus à l'arrivée d'un véhicule de couleur blanche en sortie du mini-tunnel qui permet de circuler sous le carrefour. Les véhicules détectés dans cette partie de l'image ne sont pas susceptibles de gêner la progression du tramway sur le carrefour. On pourrait alors envisager de masquer les zones correspondantes des images résultat, en définissant des zones d'intérêt dans l'image en fonction de la configuration de prise de vue.

IV.1.3) CONCLUSION SUR L'AIDE A LA CONDUITE

Le processeur STREAM, permettant la détection des contours en mouvement présents dans une séquence d'image vidéo a été utilisé afin de vérifier l'efficacité de l'algorithme sur des scènes réelles. Les séquences ayant servi aux tests ont été acquises dans des conditions d'éclairage naturel, à la lumière du jour. Les résultats se sont avérés tout à fait satisfaisants. La détection des contours en mouvement est réalisée sans ambiguïté dès lors que le contraste existant entre l'objet mobile et le fond est suffisant.

Dans l'application considérée, visant à améliorer la sécurité de la conduite d'un véhicule de type tramway en site non protégé, la rapidité de calcul est un critère primordial. Les traitements permettant la détection dans l'image des différents mobiles sont réalisés en temps-réel par le processeur spécialisé.

IV.2) DETECTION DE CHUTES DE PASSAGERS AU NIVEAU DE L'INTERFACE QUAI VOIE DANS LE METRO PARISIEN

Un véhicule de transport en commun guidé de type métro évolue dans un environnement très protégé, aucun autre mobile ne pouvant circuler sur son parcours si ce n'est le véhicule de même type qui le précède. Les zones dans lesquelles un métro est susceptible de rencontrer un obstacle sont les stations où il doit s'arrêter afin de laisser monter ou descendre des passagers. C'est au conducteur de s'assurer qu'aucun obstacle n'est présent sur les voies, et en particulier qu'aucune personne n'est tombée sur celles-ci.

Lorsqu'on envisage d'automatiser une installation de métro, le conducteur doit être remplacé par un certain nombre d'automatismes assurant le pilotage du véhicule. Les problèmes tenant à la sécurité des voyageurs se posent alors principalement au niveau des stations, où il est nécessaire de surveiller l'état des voies.

Afin d'éliminer les possibilités de chute de personnes sur les voies, on peut isoler totalement les quais du reste de la station en utilisant des portes palières, ouvertes uniquement en présence des rames de métro. Ce procédé, quoique parfaitement sécuritaire, présente toutefois certains inconvénients. Tout d'abord, le coût global de l'installation est majoré, tant par le coût des portes palières, que par celui des automatismes permettant l'arrêt précis des véhicules en face des portes. Ensuite, lorsque l'automatisation du métro est réalisée sur des installations déjà existantes, il s'avère souvent impossible de modifier la configuration des stations pour les équiper de telles portes palières.

Une autre possibilité consiste à surveiller les voies afin de vérifier qu'aucun obstacle n'est présent lorsque le véhicule automatique entre en station. Une approche souvent utilisée consiste à surveiller l'interface quai-voies, afin de détecter d'éventuelles "intrusions" de passagers sur les voies. Différents procédés permettent d'implanter ce type de surveillance, les plus prometteurs sont basés sur l'utilisation de barrières optiques ou de systèmes de vision plus évolués [Duv91].

Nous avons tenté d'appliquer le procédé de détection de contours en mouvement à la surveillance de l'interface quai-voies dans le métro parisien. Pour cela, des essais ont été réalisés avec le concours du Service des Etudes de la RATP [Cab90a, Cab90b], les résultats sont présentés dans la suite de ce chapitre.

IV.2.1) MISE EN PLACE DE L'EXPERIMENTATION

IV.2.1.a) Conditions d'expérimentation

Les essais ont été réalisés dans la station expérimentale de la RATP, située Porte des Lilas à Paris. Lors de l'expérimentation, la caméra était positionnée sur la voûte du tunnel de métro, au milieu de l'entrevoie. Depuis cette position, il est possible d'utiliser divers angles de prises de vue afin de surveiller l'interface quai-voies. L'axe optique de la caméra peut être orienté parallèlement aux voies (Cf. figure 4.7), en oblique (Cf. figure 4.8) ou perpendiculairement aux voies (Cf. figure 4.9).

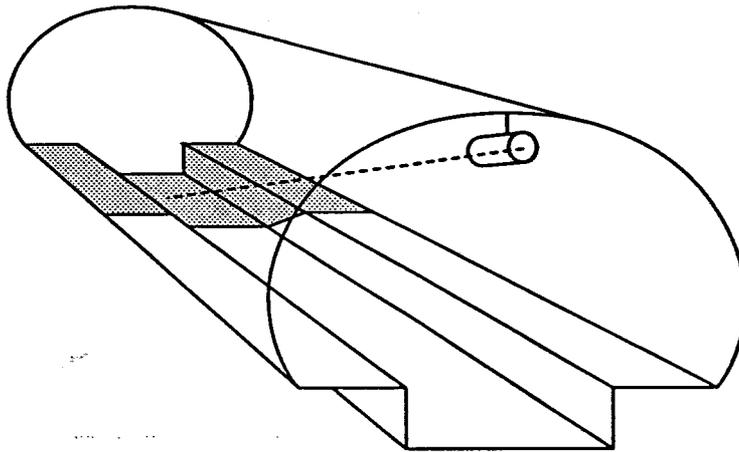


Figure 4.7 : Caméra orientée parallèlement aux voies

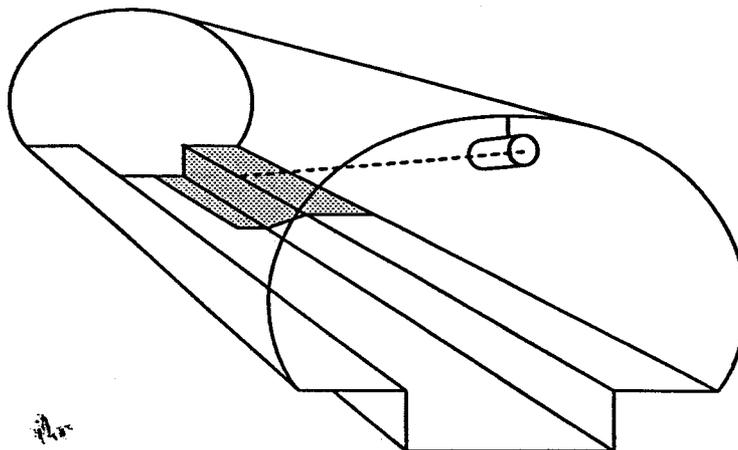


Figure 4.8 : Caméra oblique

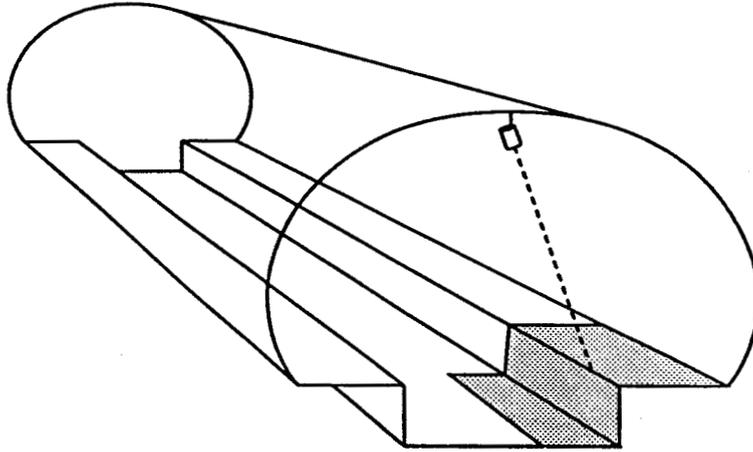


Figure 4.9 : Caméra orientée perpendiculairement aux voies

La première configuration permet de surveiller simultanément les deux cotés de la station de métro avec une seule caméra. Cependant, la résolution de l'image s'avère insuffisante dans les zones à risque, à savoir le nez de quai et les voies. Nous ne présentons pas de séquences fournies par la caméra placée dans cette configuration.

Les séquences retenues afin de vérifier l'efficacité du procédé de surveillance, illustrant différentes situations caractéristiques, ont été enregistrées sur magnéto-scope puis traitées en temps différé en laboratoire. Lors de l'acquisition des séquences sur le site surveillé, le STREAM a été utilisé comme moyen de contrôle.

IV.2.1.b) Post-traitement des contours détectés

Les images résultant du traitement des séquences acquises dans la station de métro par le STREAM ont été soumises à différentes procédures de bas niveau afin de mettre en évidence la précision de la détection. Nous avons pour cela utilisé divers post-traitements permettant de fermer les contours, puis de segmenter les images en zones homogènes.

Les traitements utilisés afin de fermer les contours fournis par le STREAM font partie de la classe des opérations morphologiques en niveaux de gris [Har87]. L'image des contours issue du processeur est tout d'abord dilatée par un élément structurant de type chapeau haut de forme. Cette opération de dilatation en niveaux de gris permet d'éliminer les faibles discontinuités apparaissant dans les contours.

Après cette phase de dilatation, les contours subissent une procédure d'amincissement préservant leurs propriétés de connexité. Une phase d'analyse de connexité permet ensuite de déterminer plus précisément les caractéristiques des objets détectés. Diverses mesures concernant la surface, la position et la forme des mobiles sont alors effectuées.

IV.2.2) RESULTATS DE L'EXPERIMENTATION

Les traitements ont été réalisés sur diverses séquences d'images, présentant des situations particulières : personne tombant sur les voies, rame de métro entrant dans la station, etc... Afin d'illustrer les résultats obtenus, nous présentons plusieurs séries de trois images successives extraites de séquences vidéo, accompagnées de l'image résultant de la détection des objets en mouvement.

Les cinq premières séries d'images sont extraites de séquences acquises par une caméra visualisant la totalité d'un quai sur toute la longueur de la station (caméra oblique). Dans cette configuration, il est possible de surveiller une longueur de quai d'environ trente mètres avec un seul système d'acquisition d'images.

Les trois séquences suivantes, ont été filmées par une caméra visant le bord du quai (position perpendiculaire). Cette position présente l'avantage de ne pas déformer les objets par effet de perspective : la taille réelle d'un objet détecté ne dépend pas de sa position dans l'image.

IV.2.2.a) Caméra oblique

Sur la séquence numéro 1, les trois piétons évoluant sur le quai ont été parfaitement détectés, ainsi que la rame de métro entrant en station. Toutefois, les piétons situés en premier plan ne sont pas détectés dans leur intégralité. Une observation attentive de la série de trois images montre que les jambes de ces deux piétons sont restées immobiles, et que seuls leurs bras et leurs tronc se sont déplacés. Par contre, le piéton situé en arrière plan s'est complètement déplacé entre les prises de vue, et de ce fait, il est intégralement détecté.

La rame de métro apparaît sous forme de trois régions en mouvement. On reconnaît l'ombre portée sur les rails qui avance avec la rame. L'avant de celle-ci, qui se déplace presque parallèlement à l'axe optique de la caméra, a très peu bougé entre les images successives, ce qui explique qu'il soit morcelée en deux zones distinctes. Cependant, dans le cadre de cette détection, le morcellement de la rame en plusieurs zones distinctes n'est pas préjudiciable à la fiabilité du procédé car sa présence peut être confirmée par d'autres moyens.

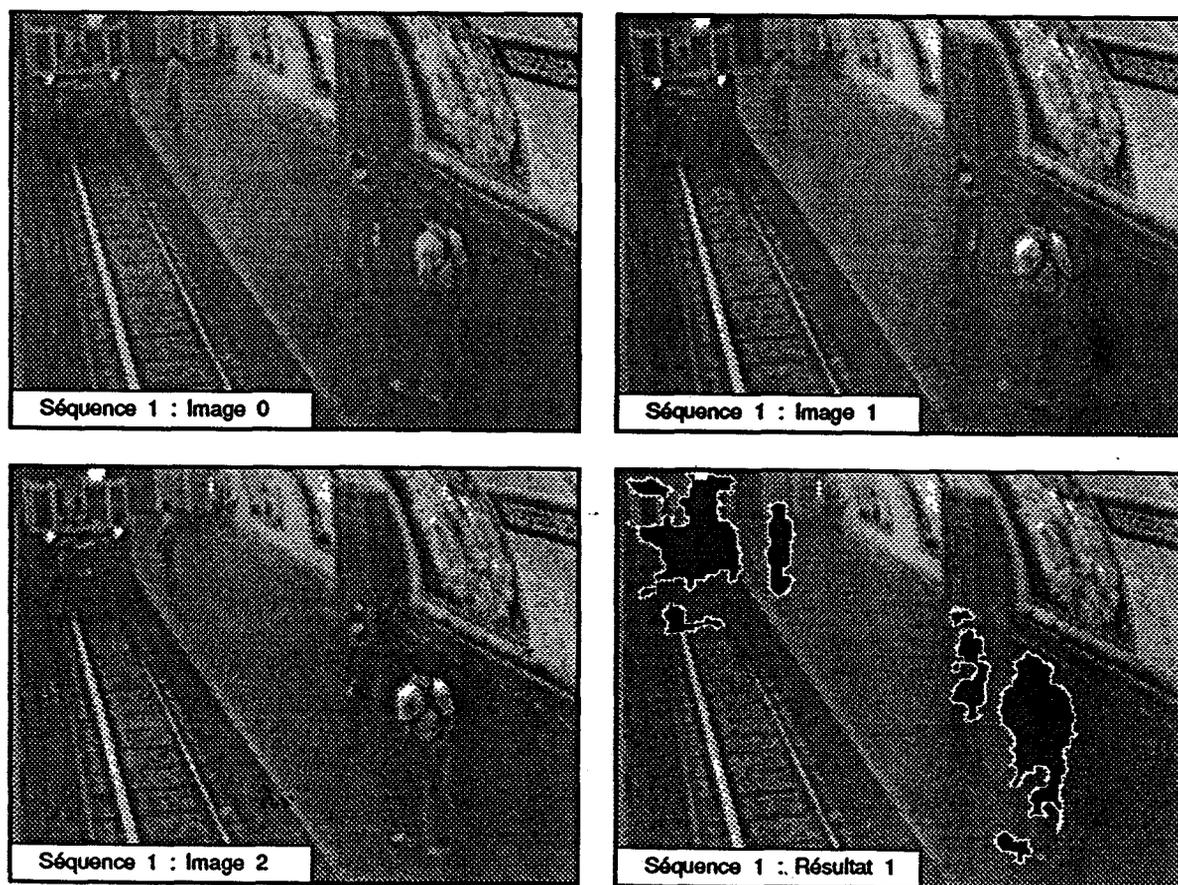


Figure 4.10 : Séquence n° 1

Dans la séquence numéro 2, un passager tombe sur les rails. Celui-ci est parfaitement détecté dans l'image courante.

Cette détection se retrouvera sur chacune des images analysées de la séquence, ce qui permettra de suivre le mouvement de ce piéton dans la zone interdite. Cette dernière peut être déterminée de manière interactive lors de l'installation du système de surveillance dans la station, afin de couvrir en totalité les parties de l'image visualisant le bas du nez de quai ainsi que les rails.

La précision sur la taille et les caractéristiques dynamiques des objets détectés dépend de leur position dans l'image : un objet mobile situé loin de la caméra sera détecté avec une moins bonne précision qu'un objet approché. Il est cependant possible de corriger les mesures si l'on connaît les caractéristiques optiques du système d'acquisition d'images.

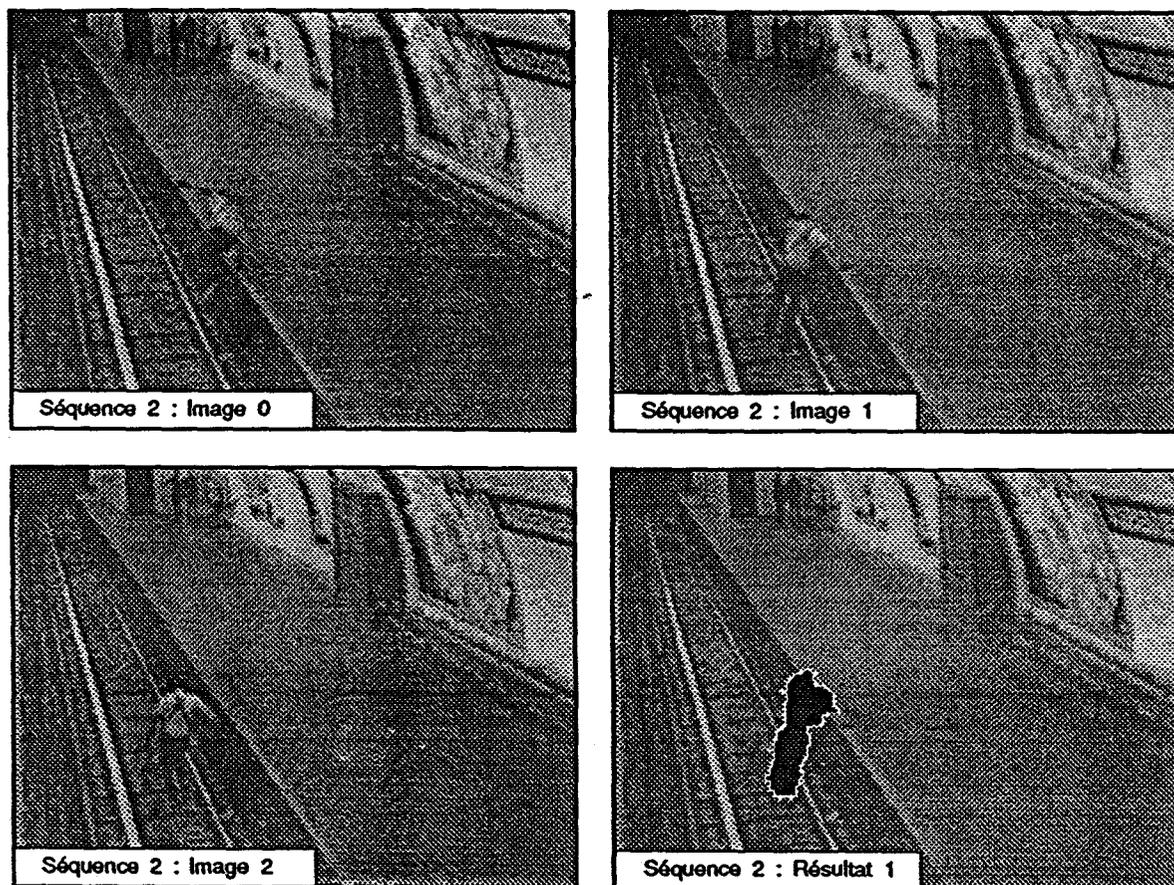


Figure 4.11 : Séquence n° 2

Les trois séquences suivantes, présentées sur les figures 4.12, 4.13 et 4.14, mettent en évidence les possibilités de discrimination entre différents types d'intrusion. Le piéton marchant sur les rails dans la séquence numéro 3 est détecté sous la forme d'une région beaucoup plus étendue que le journal jeté par le passager de la séquence numéro 4. Enfin, le ticket de métro lâché par le piéton sur la séquence 5 est également détecté malgré sa taille très réduite.

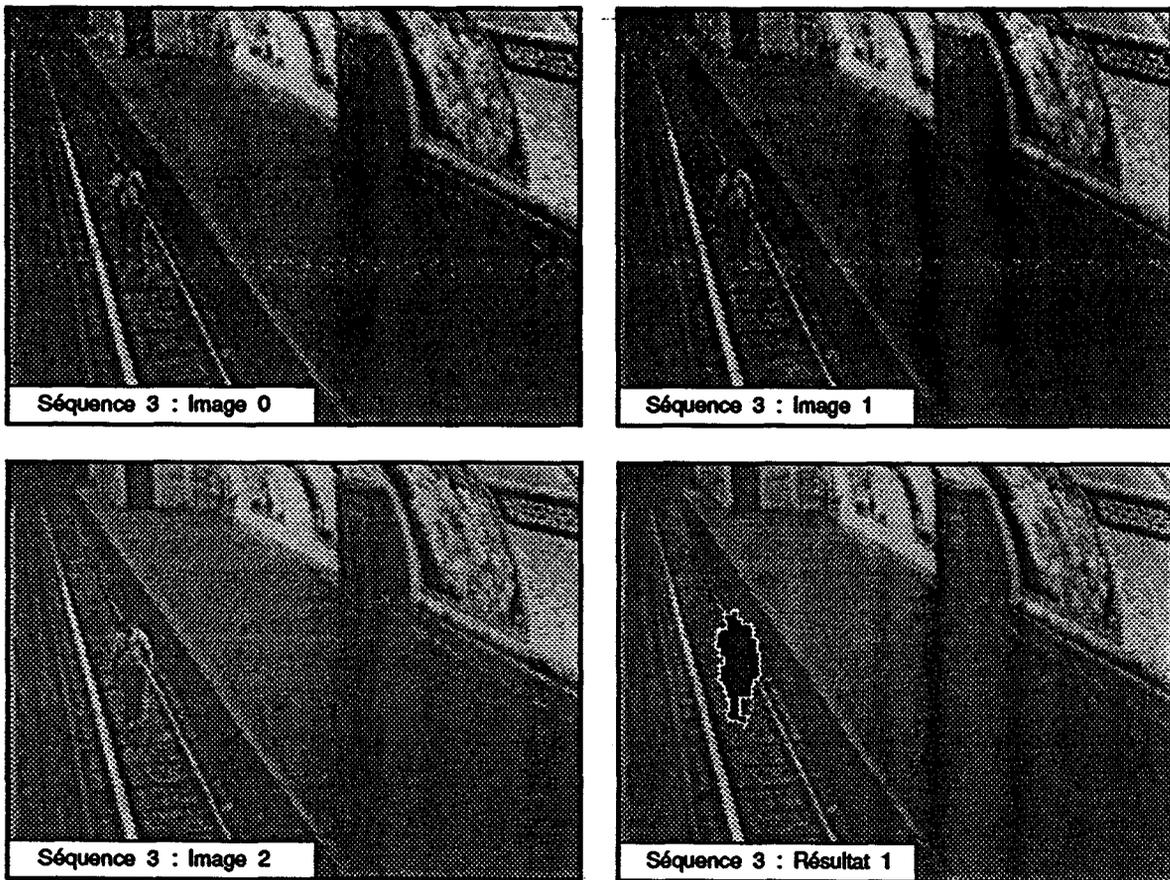


Figure 4.12 : Séquence n° 3

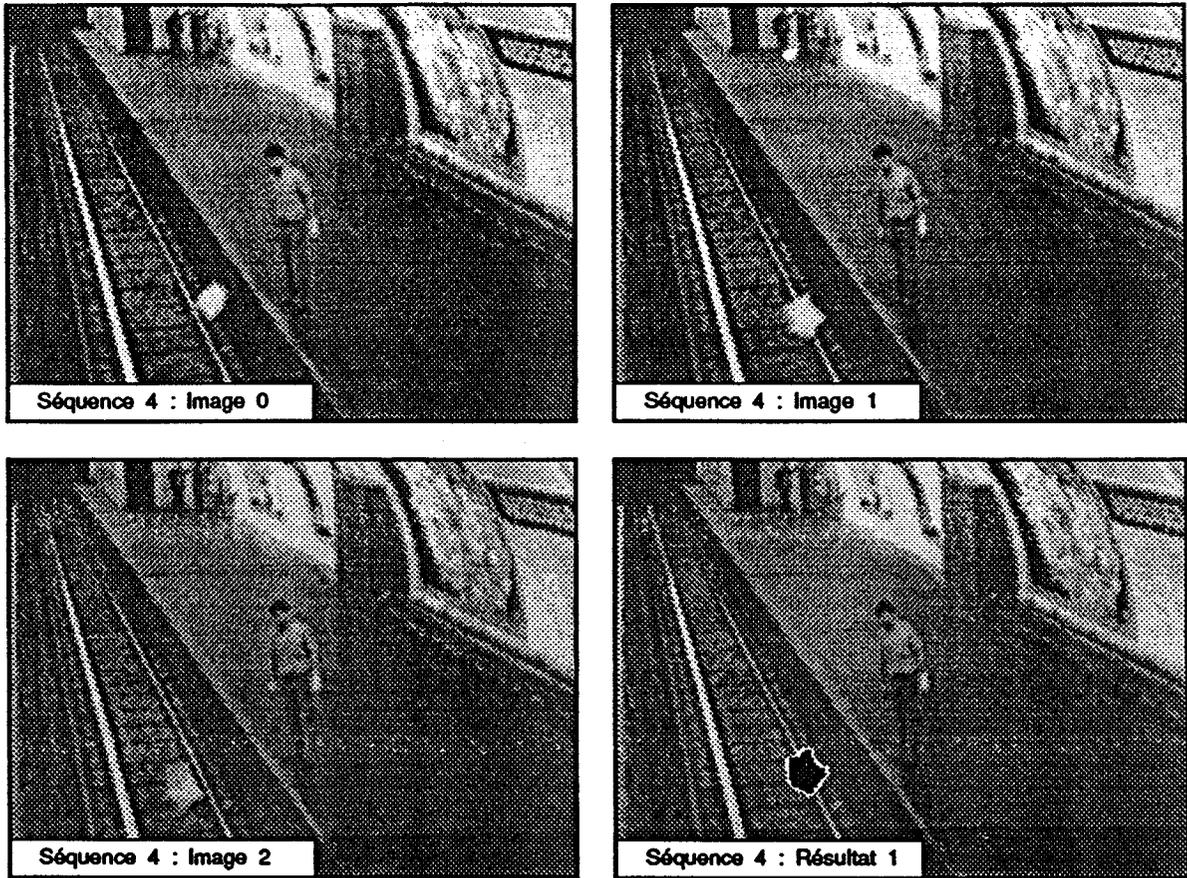


Figure 4.13 : Séquence n° 4

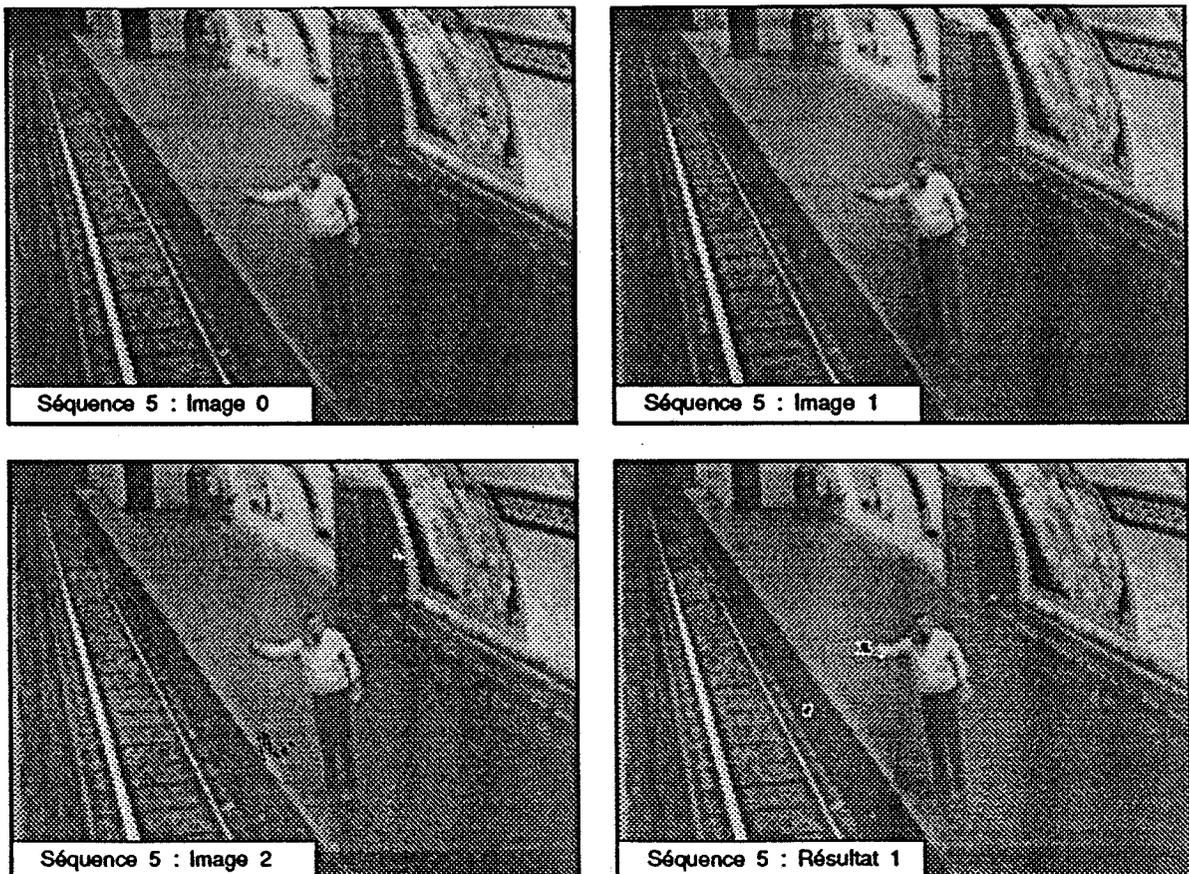


Figure 4.14 : Séquence n° 5

IV.2.2.b) Caméra orthogonale aux voies

Pour les séquences suivantes, la caméra était orientée perpendiculairement aux voies. Sur la séquence numéro 6, la détection des piétons est réalisée de façon satisfaisante. Certaines parties des personnes n'ont pas bougé entre deux images successives, ce qui explique qu'elles ne soient pas détectées.

Le mouvement des jambes du piéton de gauche, n'a pas été détecté. Cela est dû au manque de contraste existant entre le pantalon de cette personne et le quai dans les conditions d'éclairage faible de la station. Afin de résoudre ce problème, il devrait être possible d'utiliser un revêtement de quai de texture suffisamment marquée qui garantirait une détection fiable des objets, quelle que soit leur couleur.

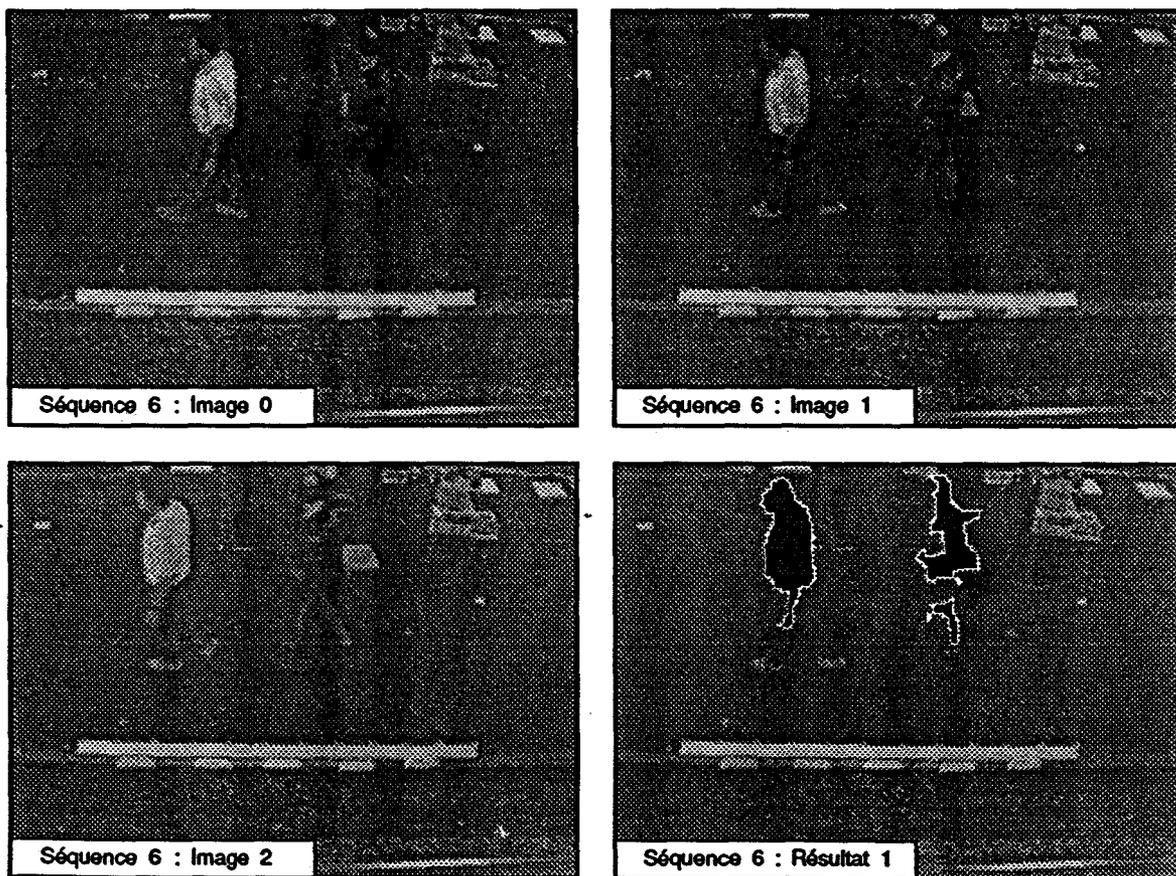


Figure 4.15 : Séquence n° 6

La séquence numéro 7 illustre le départ d'une rame de métro arrêtée devant la caméra. La forme de la région mobile détectée est caractéristique dans ce sens que sa hauteur est très importante. La position du métro devant la caméra peut en outre être confirmée par d'autres capteurs.

La séquence numéro 8 montre le résultat de la détection d'une chute de passager sur les rails. Partant de cette détection, il est relativement simple de déclencher une alarme lorsqu'un objet de taille suffisante pénètre dans la zone interdite. Cette dernière est définie comme étant la partie de l'image située sous le nez de quai.

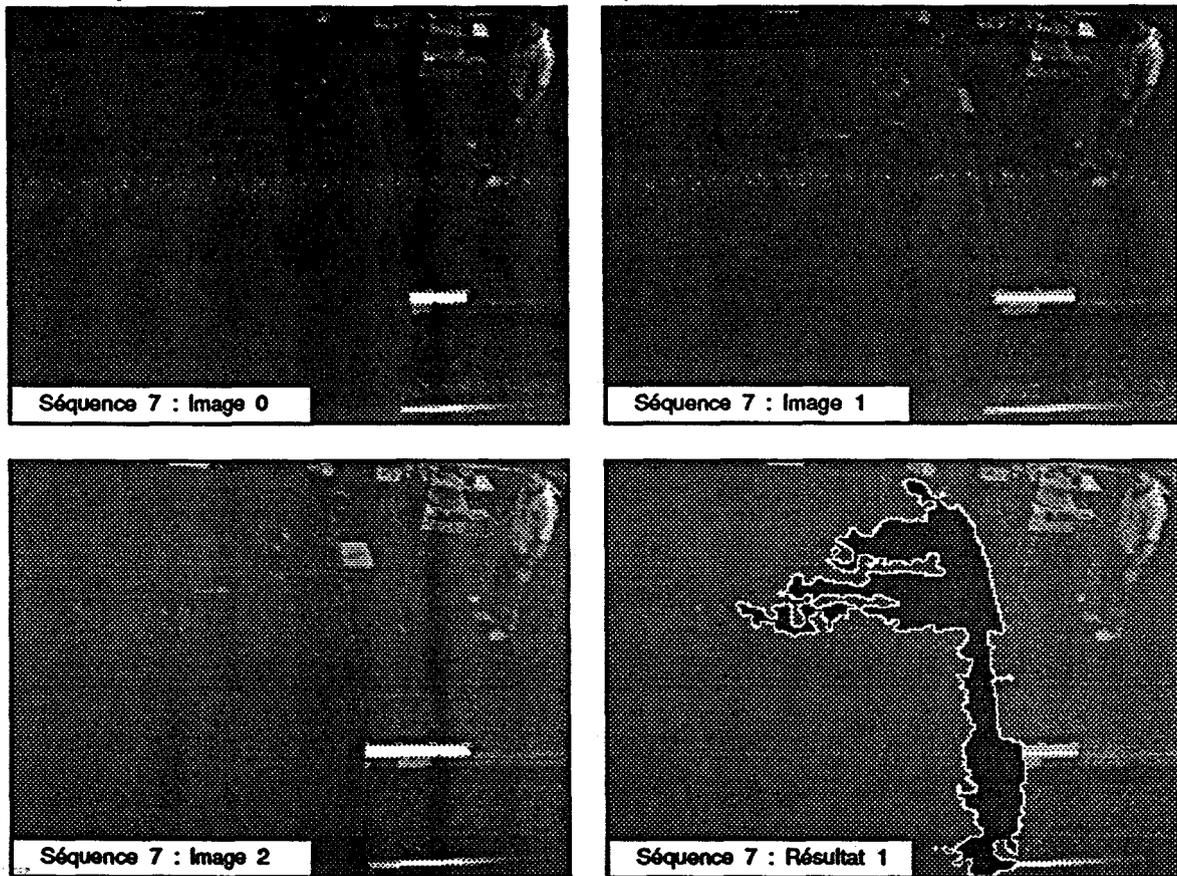


Figure 4.16 : Séquence n° 7

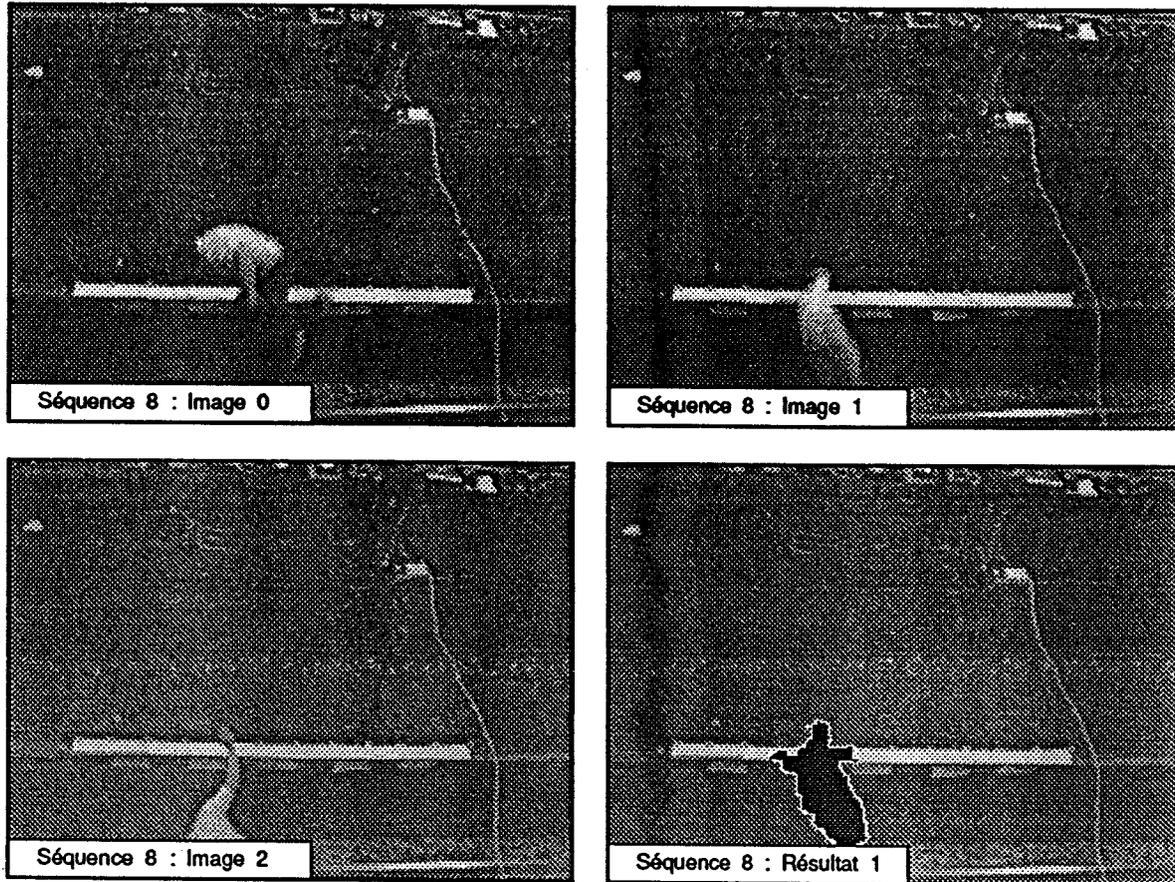


Figure 4.17 : Séquence n° 8

IV.2.3) CONCLUSION SUR LA DETECTION DE CHUTES

Les essais menés dans la station expérimentale de la RATP ont permis de vérifier l'intérêt du détecteur d'objets mobiles STREAM pour la surveillance des interfaces quai-voies. Dès lors que l'on dispose d'une image suffisamment contrastée, on extrait parfaitement de la scène les objets mobiles quelles que soient leur taille, puisque même un ticket de métro a été détecté, leur position dans l'image et leur vitesse.

Une implantation future de ce procédé de surveillance devrait tenir compte du problème de l'éclairage des stations. Bien que l'algorithme de détection de contours en mouvement soit relativement insensible aux variations de la luminosité ambiante, la qualité de la détection est améliorée lorsque les images sont bien contrastées. Afin de garantir des conditions optimales de fonctionnement, la caméra servant à la prise

de vue doit être de sensibilité importante, et si possible dotée d'un système de contrôle automatique de gain.

Des deux configurations de prise de vue qui ont été testées, celle dans laquelle la caméra est fixée orthogonalement aux voies est a priori la plus sûre. Les images n'étant pas déformées par un effet de perspective, les dimensions réelles d'un objet mobile sont aisément calculées d'après son image.

La configuration avec caméra oblique risque de poser des problèmes au niveau de l'identification des véritables chutes de passagers sur les voies. La taille apparente des objets détectés dépend de leur position dans l'image, et la faible résolution obtenue aux alentours du point de fuite des voies risque de nuire à la fiabilité du système.

CONCLUSION

Dans cette étude, nous avons présenté la conception et la réalisation d'un processeur spécifique dédié à l'analyse de scènes dynamiques : le STREAM, Système Temps Réel d'Extraction et d'Analyse du Mouvement. Ce processeur permet la détection en temps-réel des contours de tous les objets se déplaçant dans une scène observée par une caméra fixe.

Dans un premier temps, nous avons évalué l'efficacité de plusieurs algorithmes existants, qui permettent d'extraire d'une séquence d'images les contours des objets en mouvement. Les méthodes testées utilisent les différences d'images successives de la séquence afin de mettre en évidence le mouvement, et extraient les contours des objets par estimation du gradient des images. La combinaison de ces traitements permet d'obtenir les contours des objets en mouvement.

La comparaison des performances de ces algorithmes a été réalisée en adaptant diverses mesures, utilisées habituellement pour l'évaluation des opérateurs de détection des contours statiques, à l'évaluation des opérateurs de détection de contours dynamiques. Ces différents critères de comparaison nous ont permis de montrer que la méthode de détection des contours en mouvement proposée par M^r Vieren [Vie88] est la mieux adaptée au traitement de scènes réelles.

Nous avons ensuite conçu un processeur spécifique permettant la détection des objets mobiles en temps-réel selon cette méthode. L'algorithme a été partitionné en opérateurs simples organisés en réseau pipeline, qui ont été ensuite implantés sur une série de cartes électroniques communicant par l'intermédiaire de liens vidéo numériques.

L'efficacité et la fiabilité du processeur STREAM ont été vérifiées dans le cadre de deux expérimentations liées à l'amélioration des conditions de sécurité dans les systèmes de transport guidés. La première étude avait pour but de surveiller un

carrefour urbain traversé par une ligne de tramway, afin de détecter d'éventuels risques de collision entre la rame et d'autres véhicules. La seconde expérimentation a montré qu'il est possible d'utiliser ce processeur comme système de détection de chutes de passagers sur les voies du métro parisien.

On peut envisager d'utiliser le STREAM dans d'autres domaines d'application nécessitant un traitement rapide des informations visuelles pour satisfaire des contraintes de sécurité : surveillance de zones à risque dans le voisinage immédiat des robots industriels, détection d'obstacles sur le trajet emprunté par les chariots filoguidés dans les ateliers de production, systèmes de surveillance de zones interdites.

Le STREAM a été développé afin de résoudre de façon très spécifique le problème de l'implantation en temps-réel de l'algorithme de détection des contours des objets en mouvement présenté par M^r Vieren. La structure pipeline obtenue s'avère cependant très évolutive. Des nouvelles fonctionnalités peuvent être implantées en insérant des opérateurs pipelines dans le réseau de connexions vidéo numérique. Actuellement, des modules qui permettent de réaliser des traitements relevant de la morphologie mathématique à niveaux de gris sont en cours de développement au Centre d'Automatique de Lille, ces opérateurs ayant pour but d'améliorer la connexité des contours mobiles détectés par le processeur.

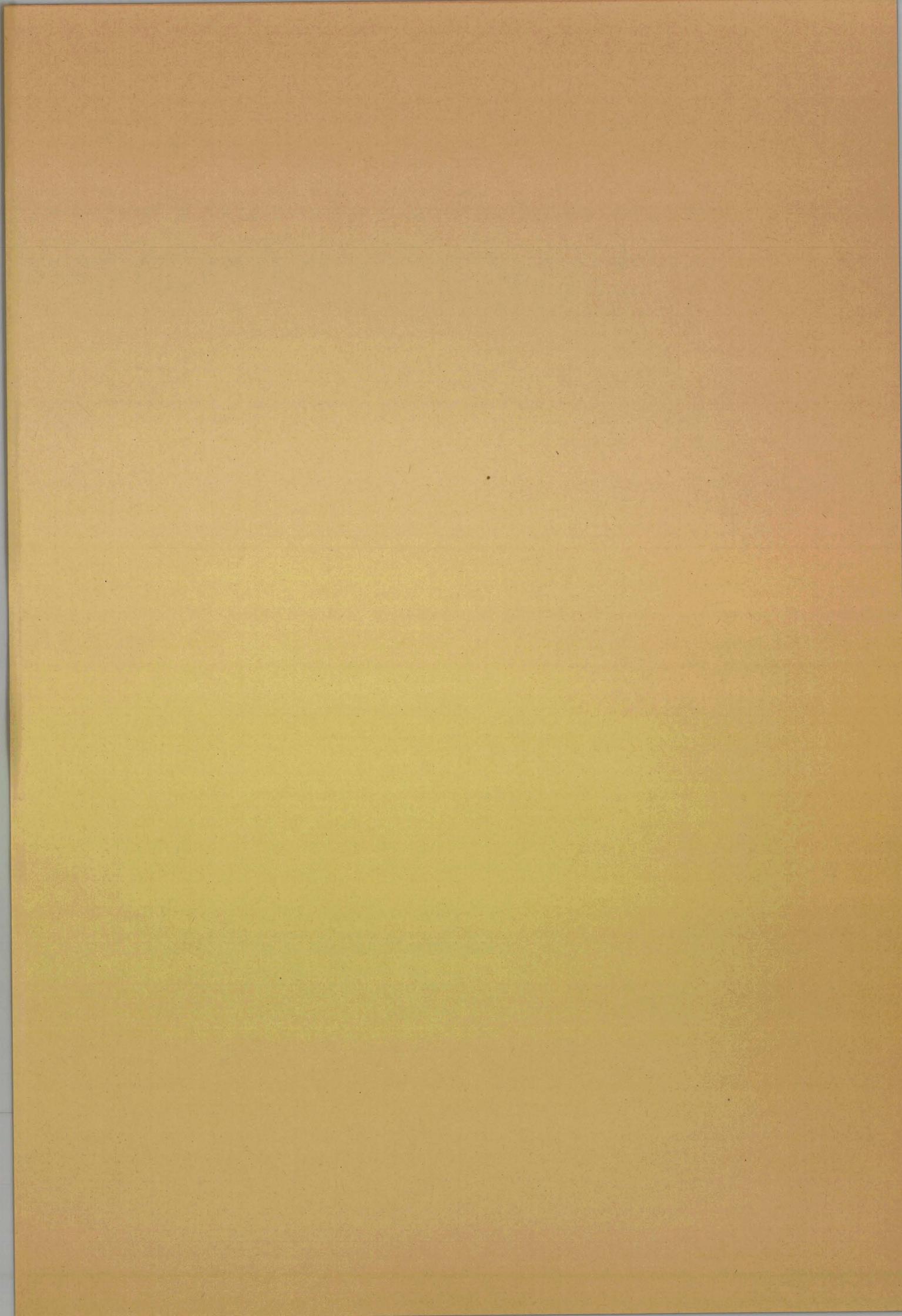
Pour mettre en évidence de façon plus précise le mouvement des différents mobiles détectés, il s'avère nécessaire de poursuivre le traitement afin d'obtenir un modèle d'objet à partir des contours en mouvement. Le suivi des mobiles dans la séquence d'images se ramène ensuite à un problème de mise en correspondance de modèles entre les images successives.

Actuellement, la phase de modélisation est réalisée par un système de traitement d'image standard, exploitant les informations issues du STREAM. La méthode utilisée consiste à remplacer chaque groupe de points connexes de l'image des contours par une forme polygonale exinscrite décrite par les coordonnées de ses sommets. Le modèle polygonal choisi ne nécessite qu'un faible nombre d'opérations élémentaires pour estimer les paramètres caractérisant un objet et permet donc de limiter le temps nécessaire à la modélisation. Cependant, le modèle obtenu est très approximatif, ce qui entraîne souvent des erreurs de mise en correspondance lors de la phase de suivi des objets dans la séquence d'images. Nous travaillons actuellement sur une nouvelle méthode plus adaptative de modélisation des mobiles, basée sur

l'utilisation de contours actifs.

D'un point de vue plus fondamental, une évolution envisageable est celle de l'utilisation du procédé d'extraction des contours en mouvement dans un contexte tridimensionnel. Actuellement, les traitements sont réalisés sur une seule image, la position des objets ne peut alors être définie que par deux coordonnées dans un plan perpendiculaire à l'axe optique de la caméra. Il s'avère donc impossible de gérer les situations pour lesquelles le mouvement des objets ne s'effectue pas dans ce plan. En combinant les résultats fournis par deux systèmes STREAM identiques, assurant la détection des objets en mouvement présents dans une scène observée par deux caméras selon des angles de vue différents, il doit être possible de développer des procédures de stéréovision dynamique visant à localiser les objets détectés dans l'espace à trois dimensions.

Les différentes voies de recherche présentées auparavant semblent toutes très prometteuses. Certaines sont en cours d'étude au sein de notre laboratoire.



REFERENCES BIBLIOGRAPHIQUES

PRESENTATION DE L'ETUDE

[Dep84]

DEPARIS J.P., KUHN F.

"Perspectives de Développement des Automatismes dans les Transports Guidés de Surface. 1ère Partie : Résultats d'une étude sur les conditions de circulation du Mongy"

Rapport Interne INRETS/CRESTA

Juin 1984.

[Bou88]

BOUTRY F.

"Analyse de scènes dynamiques par représentation espace-temps des éléments mobiles. Application à la conduite des véhicules guidés".

Thèse de l'université

Université des Sciences et Techniques de Lille Flandres-Artois. Mai 1988.

[Ste85]

STELMASZYK P.

"Analyse de scènes dynamiques par recherche des contours en mouvement. Application à la conduite automatique d'un tramway"

Thèse de doctorat

Université des Sciences et Techniques de Lille Flandres-Artois. Novembre 1985.

[Vie88]

VIEREN C.

"Segmentation de scènes dynamiques en temps-réel. Application au traitement de séquences d'images pour la surveillance de carrefours routiers"

Thèse de doctorat

Université des Sciences et Techniques de Lille Flandres-Artois. Avril 1988.

PREMIER CHAPITRE**[Adi83]**

ADIV G.

"Recovering Motion Parameters in Scenes Containing Multiple Moving Objects"*IEEE, Computer and Information Science Dept.*

Université Massachusetts, 1983

[Adi85]

ADIV G.

"Determining 3-D motion and structure from optical flow generated by several moving objects"*IEEE Trans. on Pattern Analysis and Machine Intelligence*

Vol. 7, N° 4, Juin 1985, pp 384-401

[Agg75]

AGGARWAL J.K.

"Computer analysis of moving polygonal images"*IEEE Computer*

Vol C-24, N° 10, Octobre 1975, pp 966-976

[Agg81]

AGGARWAL J.K.

"Correspondence processes in dynamic scene analysis"*IEEE Trans. on Pattern Analysis and Machine Intelligence*

Vol. 69, N° 5, Mai 1981

[Ais89]

AISBETT J.

"Optical Flow with an Intensity-Weighted Smoothing"*IEEE Trans. on Pattern Analysis and Machine Intelligence*

Vol. 11, N° 5, Mai 1989, pp 512-522

[Aok84]

AOKI M.

"Detection of Moving Objects using Line Image Sequences"*7th Int. Conf. on Pattern Recognition*

1984

[Asa80]

ASADA M., YICHIDA M., TSUJI S.

"Three dimensionnal motion interpretation of the sequence of line drawings"*Proc. of the Int. Conf. on Pattern Recognition*

1980, pp 1266-1273

[Bal83]

BALLARD D.H., KIMBALL O.A.

"Rigid body motion from depth and optical flow"*Computer Vision Graphics and Image Processing*

Vol. 22, N° 1, Avril 1983, pp 95-115

[Bar80]

BARNARD S.T., THOMPSON W.B.

"Disparity Analysis of Images"*IEEE Trans. on Pattern Analysis and Machine Intelligence*

Vol. 2, N° 4, Juillet 1980, pp 333-340

[Bar86]

BARBA D.

"Automatic Region Construction by Edge Detection and Contour Following in Image Segmentation"

8th Int. Conf. on Pattern Recognition
1986**[Ber80]**

BERS K.H.

"Object detection in image sequences"

IEEE 5th Int. Conf. on Pattern Recognition
1980**[Blo87]**

BLOSSEVILLE J.M. et Coll.

"TITAN : Système de mesure de trafic par analyse d'image"

Rapport interne INRETS
Arcueil, 1987**[Bou86a]**

BOUTHEMY P.

"A Method of Integrating Motion Information Along Contours Including Segmentation"

8th Conf. on Pattern Recognition
1986**[Bou86b]**

BOUTHEMY P.

"Un nouveau schéma d'estimation du champ de vitesses sur les contours dans une séquence d'images"

Sem. Int. Image and Electronics
Avril 1986**[Bou86c]**

BOUTHEMY P.

"Un nouveau schéma d'estimation du champ de vitesses sur les contours dans une séquence d'images"

2^{ème} Colloque Image CESTA
Nice, Avril 1986, pp 237-245**[Bou89]**

BOUTHEMY P.

"A Maximum Likelihood Framework for Determining Moving Edges"

IEEE Trans. on Pattern Analysis and Machine Intelligence
Vol. 11, N° 5, Mai 1989, pp 499-511**[Clo80]**

CLOCKSIN W.F.

"Perception of surface slant and edge label from optical flow : a computational approach"

Perception
Vol. 9, Novembre 1980, pp 253-269**[Dav79]**

DAVIS L.S., WU Z., SUN H.

"Contour based motion estimation"

Computer Graphics Image Processing
Vol. 9, 1979

[Did82]

DIDAY E. et Coll
 "Eléments d'analyse de données"
Dunod, Bordas
 1982

[Dre78]

DRESCHER L.
 "Using "Affinity" for extracting Images of Moving Objects from TV-Frame Sequences"
Institut für Informatik
 Hamburg, Février 1978

[Dre83]

DRESCHLER L., NAGEL H.H.
 "On the selection of critical points and local curvature. Extrema of region boundary for interframe matching"
Image Sequence Processing
 Ed. T.S.Huang, Berlin, 1983, pp 457-470

[Fac87]

FACON J.
 "EDOSI : Un système d'étude du déplacement d'objets à partir de séquences d'images"
Thèse de doctorat
 Université de Compiègne, Novembre 1987

[Fan83]

FANG J.Q., HUANG T.S.
 "Solving 3-D small-rotation motion equations"
Proc. IEEE Conf. on Computer Vision and Pattern Recognition
 Washington DC, 1983, pp 253-258

[Fan84a]

FANG J.Q., HUANG T.S.
 "Solving 3-D small-rotation motion equations : uniqueness algorithms and numerical results"
Computer Graphics and Image Processing
 Vol. 26, 1984, pp 183-206

[Fan84b]

FANG J.Q., HUANG T.S.
 "Some experiments on estimating the 3-D motion parameters of a rigid body from two consecutive image frames"
IEEE Trans. on Pattern Analysis and Machine Intelligence
 Vol. PAMI-6, Septembre 1984, pp 545-554

[Fen79]

FENNEMA C.L., THOMPSON W.B.
 "Velocity determination in scenes containing several objects"
Computer Graphics and Image Processing
 Vol. 9, Avril 1979, pp 301-315

[Gen77]

GÉNNERY D.B.
 "A stereo system for an autonomous vehicle"
Proc. of the 5th Int. Conf. on Artificial Intelligence
 Août 1977, pp 576-582

[Gen79]

GENNERY D.B.

"Object detection and measurement using stereo-vision"

Proc. of the 6th Joint Conf. on Artificial Intelligence
Août 1979**[Has74]**

HASKELL B.G.

"Frame to frame coding of television pictures using two-dimensional Fourier transform"

IEEE Trans. on Information Theory
IT-20, 1974, pp 119-120**[Hwa89]**

HWANG K., KUMAR V.K.P., KIM D., ALNUWEIRI H.M.

"An hybrid SIMD/MIMD Architecture for Image Understanding"

Proc. SPIE Int. Soc. Opt. Eng.
Vol. 1058 High Speed Computing II, 17-18 Janvier 1989, pp 15-26**[Hor81]**

HORN B.K.P., SCHUNK B.G.

"Determining optical flow"

Artificial Intelligence
Vol. 17, 1981, pp 185-203**[Hua81a]**

HUANG T.S.

"Image Sequence Analysis : what can we learn from applications ?"

Image Sequence Analysis
Ed. T.S.Huang, 1981, pp 104-116**[Hua81b]**

HUANG T.S., TSAI R.Y.

"Image sequence analysis : Motion estimation"

Image Sequence Analysis
Ed. T.S.Huang, 1981, Chap. 1**[Hun87]**

HUNTSBERGER T.L., JAYARAMAMURTHY S.N.

"Determination of the optic flow field using the spatio temporal deformation of region properties"

Pattern Recognition Letters
Vol. 6, Août 1987, pp 169-177**[Jac80]**

JACOBIOUS C.J., CHIEN R.T., SELANDER J.M.

"Motion detection and analysis of matching graphs of intermediate level primitive"

IEEE Trans. on Pattern Analysis and Machine Intelligence
Vol. 2, N°6, Novembre 1980, pp 495-510**[Jai79a]**

JAIN R.

"On the Analysis of Difference Pictures from Images Sequences of Real World Scenes"

IEEE Trans. on Pattern Analysis and Machine Intelligence
Avril 1979

- [Jai79b]**
 JAIN R.
 "Extraction of moving objects images through change detection"
Proc. of the 6th Int. Conf. on Artificial Intelligence
 1979, pp 425-428
- [Jai79c]**
 JAIN R.
 "Segmentation through the detection of change due to motion"
Computer Graphics and Image Processing
 Vol. 11, 1979
- [Jai81a]**
 JAIN R.
 "Dynamic Scene Analysis Using Pixel-Based Processes"
Computer
 Août 1981
- [Jai84]**
 JAIN R.
 "Difference and Accumulative Difference Pictures in Dynamic Scene Analysis"
Image & Vision Computer
 Mai 1984
- [Kan85]**
 KANATANY K.
 "Detecting the motion of a planar surface by line and surface integrals"
Computer Vision Graphics and Image Processing
 Vol. 29, 1985, pp 13-22
- [Kap83]**
 KAPPAGANTULA S.
 "Motion Compensated Interframe Image Prediction"
IEEE Trans. on Communications
 Septembre 1985
- [Kor80]**
 KORN A.
 "Motion Analysis in Natural Scenes Picked up by a Moving Optical Sensor"
5th Int. Conf. on Pattern Recognition
 Miami, 1980
- [Kor84]**
 KORIES R., ZIMMERMANN G.
 "Motion detection in image sequences : an evaluation of feature detectors"
7th Int. Conf. on Pattern Recognition
 1984
- [Law83]**
 LAWTON D.T.
 "Processing translational motion sequences"
Computer Vision Graphics and Image Processing
 Avril 1983, pp 116-144
- [Lee70]**
 LEESE J.A.
 "The detection of cloud pattern motions from geosynchronous satellite image data"
Pattern Recognition
 Vol. 2, 1970

- [Lee71]**
 LÉESE J.A.
 "Analysis automatic technique for obtaining cloud motion from geosynchronous satellite data using cross-correlation"
J. Appl. Meteorol.
 Vol. 10, Février 1971, pp 118-132
- [Lee80]**
 LEE D.N.
 "The optic flow field : the foundation of vision"
Philosophical Transactions of Royal Society of London
 Vol. B, N° 290, 1980, pp 169-179
- [Len83]**
 LÉNZ R., GERHARD A.
 "Image sequence coding using scene analysis and spatio-temporal interpolation"
Image Sequence Processing
 Eds T.S. HUANG, 1983, pp 663-709
- [Lev83]**
 LEVINE M.D.
 "A rule-based system for characterizing blood cell motion"
Image Sequence Processing
 Ed. T.S.Huang, 1983, pp 663-709
- [Ma87]**
 MA J., LU X., WU C.
 "A motion constraint equation under space varying or time varying illumination"
Pattern Recognition Letters
 Vol. 5, 1987, pp 301-315
- [Mai84]**
 MAITRE H., LOPEZ K.J.
 "Transformation de Hough : théorie et application à la detection de symétries circulaires"
4^{ème} congrès de Reconnaissance des Formes
 Paris, Janvier 1984
- [Man87]**
 MANTAS J.
 "Methodologies in pattern recognition and image analysis - a brief survey"
Pattern Recognition
 Vol. 20, N° 1, 1987, pp 1-6
- [Mar77]**
 MARR D., POGGIO T.
 "A theory of human stereovision"
Artificial Intelligence Memo 451
 M.I.T., Novembre 1977
- [Med85]**
 MEDIONI G., YASUMOTO Y.
 "Robust estimation of 3-D motion parameters of sequence of image frames using regularization"
Proc. of Image Understanding Workshop
 DARPA, Miami Beach, Décembre 1985, pp 117-128
- [Mit84]**
 MITICHE A.
 "Computation of optical flow and rigid motion"
Proc. Workshop on Computer Vision : Representation and Control
 Annapolis, MD, 1984, pp 63-71

[Mit86]

MITICHE A.

"On kineopsis and computation of structure and motion"

IEEE Trans. on Pattern Analysis and Machine Intelligence

Vol. PAMI-8, N° 1, Janvier 1986

[Mit87]

MITICHE A.

"Experiments in computing optical flow with the gradient-based multiconstraint method"

Pattern Recognition

Vol. 20, N° 2, 1987, pp 173-179

[Mor77]

MORAVEC H.P.

"Towards Automatic Visual Obstacle Avoidance"

Proc. of the 5th Int. Conf. on Artificial Intelligence

Août 1977, p 584

[Mor83]

MORAVEC H.P.

"The Stanford Cart and the CMU Rover"

Proc. of the IEEE

Vol. 71, N° 7, Juillet 1983

[Nag79]

NAGEL H.H.

"Evaluation of several dissimilarity grading approaches for the analysis of image sequences"

IEEE Workshop on Computer Analysis of Time-Varying Imagery

Philadelphia, PA., Avril 1979

[Nag81a]

NAGEL H.H.

"On the derivation of 3-D rigid point configuration from image sequences"

Proc. of the IEEE Conf. on Pattern Recognition and Image Processing

Dallas (Texas), Août 1981

[Nag81b]

NAGEL H.H.

"On 3-D reconstruction from two perspective views"

Int. Joint Conf. on Artificial Intelligence

Vancouver, Août 1981

[Oro81]

O'ROURKE J.

"Motion detection using Hough techniques"

Proc. of the IEEE Conf. on Pattern Recognition and Image Processing

1981, pp 82-87

[Pra81a]

PRAZDNY K.

"Determining the instantaneous of motion from optical flow generated by a curvilinear moving observer"

Computer Graphics and Image Processing

Vol. 17, 1981, pp 238-248

[Pra81b]

PRAZDNY K.

"A simple method for recovering relative depth map in the case of a translating sensor"

Proc. of the Int. Joint Conf. on Artificial Intelligence
Vancouver, Août 1981**[Pri75]**

PRICE K.E., REDDY R.

"Change detection in multi-sensor images"

10th Int. Symposium on Remote Sensing of Environment
A. Arbor, Michigan, Octobre 1975**[Pri77]**

PRICE K.E., REDDY R.

"Change detection and analysis in multispectral images"

Proc. of the Int. Joint Conf. on Artificial Intelligence
1977, pp 619-625**[Pog87]**

POGGIO T.

"Computer Vision"

SPIE, Image Pattern Recognition
Vol. 755, 1987, pp 54-62**[Raj83]**

RAJALA S.A.

"Application of the One-Dimensional Fourier Transform for Tracking Moving Objects in Noisy Environment"

Computer Vision Graphics and Image Processing
Vol. 21, 1983**[Roa79]**

ROACH J.W.

"Computer tracking of objects moving in space"

IEEE Trans. on Pattern Analysis and Machine Intelligence
Vol. 1, N° 2, Avril 1979**[Riv86]**

RIVES G.

"Planar Partially Occluded Objects Scene Analysis"

8th Conf. on Pattern Recognition
1986**[Ros75]**

ROSENFELD A., WESZKA J.S.

"An improved method of angle detection on digital curves"

IEEE Trans. on Computers
Septembre 1975**[Ros76]**

ROSENFELD A.

"Scene labelling by relaxation operations"

IEEE Trans. on Systems, Man and Cybernetics
Vol. SMC-6, 1976, pp 420-433**[Sha86]**

SHARIAT H.

"The motion problem : How to use more than two frames"

Institute of Robotics and Intelligent Systems
Los Angeles (California), Octobre 1986

- [Sha90]**
 SHARIAT H.
 "Motion Estimation with More Than Two Frames"
IEEE Trans. on Pattern Analysis and Machine Intelligence
 Vol. 12, N° 5, Mai 1990, pp 417-434
- [Sch86a]**
 SHUNCK B.G.
 "Image flow continuity equations for motion and density"
Proc. IEEE Workshop Motion : Representation and Analysis
 Kiawah Island, SC, 1986, pp 89-94
- [Sch86b]**
 SHUNCK B.G.
 "The image flow constraint equation"
Computer Vision Graphics and Image Processing
 Vol. 35, 1986, pp 20-46
- [Smi72]**
 SMITH S.P.
 "Automatic cloud tracking using precisely aligned digital ATS pictures"
IEEE Trans. on Computers
 Vol. 21, N° 7, Juin 1972, pp 715-729
- [Sny80]**
 SNYDER W.E.
 "Image modeling the continuity assumption and tracking"
Int. Conf. on Pattern Recognition
 Vol. 1, Décembre 1980
- [Sny81]**
 SNYDER W.E.
 "Computer analysis of time varying images"
Computer
 Vol. 14, N° 8, Août 1981, pp 7-9
- [Sny83]**
 SNYDER W.E., RAJALA S.A.
 "Track Acquisition of Sub-Pixel Targets"
Image Sequence Analysis
 Ed. T.S.Huang, 1983, p 503
- [Spo83]**
 SPOER P.
 "Displacement estimation for objects on moving background"
Image Sequence Processing
 Ed. T.S.Huang, 1983
- [Ste85]**
 STELMASZYK P.
 "Analyse de scènes dynamiques par recherche des contours en mouvement.
 Application à la conduite automatique d'un tramway"
Thèse de doctorat
 Université des Sciences et Techniques de Lille Flandres-Artois. Novembre 1985.
- [Tan78]**
 TANIMOTO S.L.
 "A comparison of some image searching methods"
Proc. of the IEEE Int. Conf. on Pattern Recognition and Image Processing
 1978, pp 280-286

[Tan82]

TANG I.S.

"Extraction of Moving Objects in Textured Dynamic Scenes"

IEEE Conf. Publ.

Las Vegas, Juin 1982

[Tho80]

THOMPSON W.B.

"Combining motion and contrast for segmentation"

IEEE Trans. on Pattern Analysis and Machine Intelligence

Vol. 2, N° 6, Janvier 1980, pp 543-549

[Tsa82]

TSAI R.Y., HUANG T.S., ZHU W.L.

"Estimating 3 dimensional motion parameters of a rigid planar patch, II :

Singular value decomposition"

IEEE Trans. on Acoustic, Speech and Signal Processing

Vol. ASSP-30, Août 1982, pp 525-534

[Ull78]

ULLMAN S.

"Two dimensionality of the correspondance process in apparent motion"

Perception

Vol. 7, 1978, pp 683-693

[Ull79]

ULLMAN S.

"The interpretation of visual motion"

M.I.T. Press

Cambridge, 1979

[Ull81]

ULLMAN S.

"The effect of similarity between line segments on the correspondance strength in apparent motion"

Perception

Vol. 9, 1981, pp 617-622

[Vie88]

VIEREN C.

"Segmentation de scènes dynamiques en temps-réel. Application au traitement de séquences d'images pour la surveillance de carrefours routiers"

Thèse de doctorat

Université des Sciences et Techniques de Lille Flandres-Artois. Avril 1988.

[Web81]

WEBB J.A., AGGARWAL J.K.

"Structure from motion and jointed bodies"

Proc. of the 7th Int. Joint Conf. on Artificial Intelligence

Août 1981

[Web82]

WEBB J.A., AGGARWAL J.K.

"Structure from motion of rigid and jointed objects"

Artificial Intelligence

Vol. 19, Septembre 1982, pp 107-130

[Wee89]

WEEMS C.C., LEVITAN S.P., HANSON A.R., RISEMAN M.R.

"The Image Understanding Architecture"

Int. Journal of Computer Vision

Vol. 2, 1989, pp 251-282

[Wil80]

WILLIAMS T.D.

"Depth from camera motion in a real world scene"

IEEE Trans. on Pattern Analysis and Machine Intelligence

Vol. PAMI-2, Novembre 1980

[Woh83]

WOHN K.

"Motion estimation based on multiple local constraints and nonlinear smoothing"

Pattern Recognition

Vol. 16, N° 6, 1983

[Yac78]

YACHIDA M.

"Automatic motion analysis system of moving objects from the record of natural processes"

Proc. of the Int. Joint Conf. on Pattern Recognition

Kyoto (Japan), 1978, pp 726-730

[Yac83]

YACHIDA M.

"Determining velocity maps by spatio-temporal neighbourhoods from image sequences"

Computer Vision Graphics and Image Processing

Vol. 21, Février 1983

[Yal80]

YALAMANCHI S.

"Difference Operations for the Segmentation of Moving Objects in Dynamic Scenes"

5th Int. Conference on Pattern Recognition

Miami 1980

[Yal82]

YALAMANCHI S.

"Extraction of Moving Object Descriptions via Differencing"

Computer Graphics and Image Processing

Vol. 18, 1982

DEUXIEME CHAPITRE**[Abd79]**

ABDOU I.E., PRATT W.K.

"Quantitative design and evaluation of enhancement/thresholding edge detectors"

Proc. of the IEEE

Vol. 67, N° 5, Mai 1979, pp 753-763

[Bry79]

BRYANT D.J., BOULDIN D.W.

"Evaluation of edge detectors using relative and absolute grading"

Proc. IEEE Comp. Soc. Conf. on Pattern Recognition and Image Processing

Chicago, Août 1979, pp 138-145

[Can83]

CANNY J.

"Finding Edges and Lines"

M.I.T. Cambridge, Technical Report 720

1983

[Can86]

CANNY J.

"A computationnal approach to edge detection"

IEEE Trans. on Pattern Analysis and Machine Intelligence

Vol. PAMI-8, N° 6, Novembre 1986

[Dav75]

DAVIS L.S.

"A Survey of Edge Detection Techniques"

Computer Graphics and Image Processing

Vol. 4, 1975, pp 248-270

[Der87]

DÉRICHE R.

"Using Canny's criteria to derive a recursively implemented optimal edge detector"

Int. Journal on Computer Vision

Vol. 1, N° 2, Mai 1987

[Hay83]

HAYNES S.M.

"Detection of moving edges"

Computer Graphics and Image Processing

Vol. 21, 1983

[Har81]

HARALICK R., WATSON L.

"A facet model for image data"

Computer Graphics and Image Processing

Vol. 15, 1981, pp 113-129

[Har84]

HARALICK R.

"Digital step edges from zero crossing of second directionnal derivatives"

IEEE Trans. on Pattern Analysis and Machine Intelligence

Vol. PAMI-6, N° 1, Janvier 1984, pp 58-68

- [Hue73]**
 HUECKEL M.
 "A local operator which which recognizes edges and lines"
J. Ass. Computer Mach.
 Vol. 20, Octobre 1973, pp 634-647
- [Kir71]**
 KIRSCH R.
 "Computer determination of the constituent structure of biological images"
Computer Biomedical Research
 Vol. 4, N° 3, 1971, pp 315-328
- [Kit81]**
 KITCHEN L., ROSENFELD A.
 "Edge Evaluation using Local Edge Coherence"
IEEE Trans. on Systems, Man, and Cybernetics
 Vol. SMC-11, N° 9, Septembre 1981, pp 597-605
- [Kit87]**
 KITCHEN L, EGGLETON J., ILLINGWORTH J., PALER K.
 "An averaging edge detector"
Pattern Recognition Letters 6
 Juin 1987, pp 27-32
- [Mar79]**
 MARR D., HILDRETH E.
 "Theory of Edge Detection"
MIT Artificial Intelligence Memo 518
 Avril 1979
- [Mor81]**
 MORGENTHALER D.G., ROSENFELD A.
 "Multidimensionnal Edge Detection by Hypersurface fitting"
IEEE Trans. on Pattern Analysis and Machine Intelligence
 Vol. PAMI-3, N° 4, Juillet 1981, pp 482-486
- [Nal86]**
 NALVA V.S., BINFORD T.O.
 "On detecting Edges"
IEEE Trans. on Pattern Analysis and Machine Intelligence
 Vol. PAMI-8, N° 6, Novembre 1986, pp 699-714
- [Neu86]**
 NEUVO Y. HEINONEN P., DEFEE I.
 "New types of edge detectors based on linear and median operators"
8th Int. Conf. on Pattern Recognition
 1986
- [Pel82]**
 PELI T., MALAH D.
 "A Study of Edge Detection Algorithms"
IEEE Trans. on Computer Graphics and Image Processing
 Vol. 20, 1982, pp 1-21
- [Pet91]**
 PETROU M., KITTLER J.
 "Optimal Edge Detectors for Ramp Edges"
IEEE Trans. on Pattern Analysis and Machine Intelligence
 Vol. 13, N° 5, Mai 1991, pp 483-491

[Pos86]

POSTAIRE J.G., STELMASZYK P., BONNET P.
 "A visual surveillance system for traffic collision avoidance"
5th IFAC/IFIP/IFORS International Conference on Control in Transportation Systems
 Vienne, 1986

[Pra78]

PRATT W.K.
 "Digital Image Processing"
Wiley, Interscience
 N.Y. 1978

[Pre70]

PREWITT J.M.S.
 "Object enhancement and extraction"
Picture Processing and Psychopictorics
 New York Academic press, 1970, pp 75-149

[Rob77]

ROBINSON G.S.
 "Edge Detection by Compass Gradient Mask"
Computer Graphics and Image Processing
 Vol. 6, 1977, pp 492-501

[Ste85a]

STELMASZYK P.
 "Analyse de scènes dynamiques par recherche des contours en mouvement.
 Application à la conduite automatique d'un tramway"
Thèse de doctorat
 Université des Sciences et Techniques de Lille Flandres-Artois. Novembre 1985.

[Ste85b]

STELMASZYK P., BONNET P., POSTAIRE J.G.
 "Analyse de scènes dynamiques par recherche des contours en mouvement"
5^{ème} congrès AFCET/INRIA/ADI sur la Reconnaissance des Formes et l'Intelligence Artificielle
 Grenoble, 1985

[Suc82]

SUCYU R.E., REEVES A.P.
 "A comparison of differential and moment based edge detectors"
IEEE ?
 1982, pp 97-102

[Vie87]

VIEREN C., STELMASZYK P., BONNET P., POSTAIRE J.G.
 "Image processing techniques for security improvement in L.R.T. systems"
Int. Conf. on Local Public Transport
 Liverpool, England, Novembre 1987

[Vie88]

VIEREN C.
 "Segmentation de scènes dynamiques en temps-réel. Application au traitement
 de séquences d'images pour la surveillance de carrefours routiers"
Thèse de doctorat
 Université des Sciences et Techniques de Lille Flandres-Artois. Avril 1988.

[Vie89]

VIEREN C., DEPARIS J.P., BONNET P., POSTAIRE J.G.
 "Dynamic scene modelling for automatic traffic data extraction"
Int. Conf. on Applications for Advanced Technologies in Transportation Engineering
 ASCE, San Diego, 6-8 Février 1989

TROISIEME CHAPITRE**[Bas85]**

BASILE J.L., CASTAN S.
 "Multilevel architecture for image processing"
SPIE, Architectures and Algorithms for Image Processing
 Vol. 596, 1985, pp 46-52

[Bat80]

BATCHER K.E.
 "Design of a massively parallel processor"
IEEE Transactions on Computers
 Vol C-29, 1980, pp 836-840

[Ber85]

BÉRGER R.W.
 "VLSI Structures for Real-Time Image Convolution"
Proc. IEEE Int. Conf. on System Man and Cybernetics
 Tucson, AZ, Novembre 1985, pp 676-679

[Can85]

CANTONI V., and al.
 "The PAPIA image analysis system"
Proc. 2nd Int. Symposium on Optical and Electro-optical Applied Science and Engineering
 Cannes, France, Decembre 1985

[Cas85]

CASTAN S.
 "Architectures adaptées au traitement d'images"
TSI, Vol. 4, N° 5
 1985, pp 413-445

[Cha88]

CHASSERY J.M.
 "Architectures et Images", Cours
Equipe de Reconnaissance des Formes et Microscopie Quantitative
 Laboratoire TIM3-IMAG, Grenoble

[Com83]

Special Issue on Computer Architectures for Image Processing
IEEE, Computer
 Janvier 1983

[Dou86]

DOURS D.
 "Conception d'un système multiprocesseur traitant un flot continu de données
 en temps réel"
Thèse de doctorat d'état - Sciences Informatiques
 Université Paul Sabatier de Toulouse, Février 1986

[Duf86]

DUFF M.J.B., FOUNTAIN T.J.
 "Cellular Logic Image Processing"
Academic Press, New-York
 1986

[Err91]

ERRAMI A.

"Conception et réalisation d'un automate de vision temps-réel pour la robotique"

Thèse de l'université

Université Blaise Pascal de Clermont-Ferrand II. 1991.

[Gue86]

GUERRA C., LEVIALDI S.

"Computer Vision : Algorithms and Architectures"

Advances in Machine Vision, Springer Verlag

Chapitre 8, pp 283-296

[Hei89]

HEIDAR R.

"Architectures pour le traitement d'images"

Thèse de l'Institut National Polytechnique de Grenoble

Grenoble, Juillet 1989

[Hoo87]

HOOSE N., WILLUMSEN L.G.

"Automatically extracting traffic data from video-tape using the CLIP4 parallel image processor"

Pattern Recognition Letters

Vol. 6, Août 1987, pp 199-213

[Hwa85]

HWANG K.

"Computer Architecture and Parallel Processing"

Eds Mc Graw Hill

1985

[Hwa89]

HWANG K., KUMAR V.K.P., KIM D., ALNUWEIRI H.M.

"A hybrid SIMD/MIMD architecture for image Understanding"

Proc. SPIE, Int. Society of Optical Engineers

Vol. 1058, High Speed Computing II, Janvier 1989, pp 15-26

[Lee85]

LEE S.Y., AGGARWAL J.K.

"A problem-driven approach to parallel image processing : system design and scheduling"

Proc. IEEE Int. Conf. on System Man and Cybernetics

Tucson, AZ, Novembre 1985, pp 680-686

[Lee90]

LEE S.Y., AGGARWAL J.K.

"A system Design / Scheduling strategy for parallel image processing"

IEEE Trans. on Pattern Analysis and Machine Intelligence

Vol. PAMI-12, N° 2, Février 1990, pp 194-204

[Lou80]

LOUGHEED R.M., McCUBBREY D.L.

"The Cytocomputer : A practical pipelined image processor"

Proc. of the 7th Int. Symposium on Computer Architectures,

La Baule, France, Mai 1980, pp 271-277

[Lue80]

LUETJEN K., GEMMER P.

"FLIP : a flexible multiprocessor system for image processing"

Proc. of the 5th Int. Conf. on Pattern Recognition

Miami, Dec 1980

- [Mer89]**
 MÉRIGOT A., and al.
 "SPHINX, un processeur pyramidal massivement parallèle pour la vision artificielle"
Proc. 7^{ème} Congrès AFCET-INRIA sur la Reconnaissance de Formes et l'Intelligence Artificielle
 Décembre 1989, pp 185-196
- [Oza89]**
 OZAKI T., OHZORA M., KURAHASHI K.
 "An image processing system for autonomous vehicle"
SPIE, Mobile Robots IV
 Vol. 1195, 1989, pp 256-266
- [Pot83]**
 POTTER J.L.
 "Image processing on the massively parallel processor"
IEEE Computer
 Vol. C-13, N° 1, Janvier 1983
- [Pra86]**
 PRATT W.K.
 "Algorithmic-Based Machine Vision Computing"
Digital Design
 Octobre 1986, pp 56-58
- [Pra87]**
 PRATT W.K.
 "Review of Machine Vision Architectures"
Proc. SPIE, Image Pattern Recognition : Algorithm Implementations, Techniques and Technology
 Vol. 755, 1987, pp 2-12
- [Ree84]**
 REEVES A.P.
 "Survey parallel computer architectures for image processing"
Computer Vision Graphics and Image Processing
 N° 25, 1984, pp 68-88
- [Ros87]**
 ROSENFELD A.
 "A note on Shrinking and Expanding operations in Pyramids"
Pattern Recognition Letters
 Vol. 6, Septembre 1987, pp 241-244
- [Sas85]**
 SASAKI S., and al.
 "High speed pipeline image processor with a modifiable network"
Proc. of the 1st Int. Conf. on Supercomputing Systems
 IEEE, Decembre 1985
- [Sie81]**
 SIEGEL H.J.
 "PASM : a partitionnable SIMD / MIMD system for image processing and pattern recognition"
IEEE Transactions on Computers
 Vol. C-30, 1981, pp 934-946
- [Shu86]**
 SHU D.B., NASH G., WEEMS C.
 "Image Understanding Architecture and Applications"
Advances in Machine Vision, Springer Verlag
 Chapitre 9, pp 297-355

[Swa82]

SWARTZLANDER E.E.

"Computer networking in the context of very high scale integration"

Proceedings of the SPIE

Vol. 341, 1982, pp 278-285

[Ull81]

ULLMANN J.R.

"Video-Rate Digital Image Analysis Equipment"

Pattern Recognition

Vol. 14, N° 6, 1981, pp 305-318

[Wee89]

WEEMS C.C., LEVITAN S.P., HANSON R.S., RISEMAN E.M.

"The image understanding architecture"

International Journal of Computer Vision

N° 2, 1989, pp 251-282

[Yal85]

YALAMANCHI S., PALEM K.V., DAVIS L.S., WELCH A.J., AGGARWAL J.K.

"Image Processing Architectures : A Taxonomy and Survey"

Progress in Pattern Recognition, II

North-Holland, 1985, pp 1-37

CHAPITRE IV**[Cab90]**

CABESTAING F., POSTAIRE J.G., DEPARIS J.P.
 "Détection de chutes au niveau de l'interface quai-voie par traitement d'images"
Rapport Interne INRETS/CRESTA, Ref. Mai90-61
 Mai 1990

[Cab90]

CABESTAING F., POSTAIRE J.G., DEPARIS J.P.
 "Détection d'intrusions par analyse de séquences d'images en temps-réel"
Rapport de fin de contrat INRETS/CRESTA, Ref. Sep90-116
 Septembre 1990

[Duv91]

DUVIEUBOURG L.
 "Analyse de séquences d'images linéaires stéréoscopiques - Application à la réalisation d'un système de détection d'intrusions intelligent pour les transports guidés"
Thèse de l'Université des Sciences et Techniques de Lille Flandres-Artois
 Janvier 1991

[Har87]

HARALICK R.M., STERNBERG S.T., ZHUANG X.
 "Image Analysis using mathematical morphology"
IEEE Trans. on Pattern Analysis and Machine Intelligence
 Vol. 9, N° 4, 1987, pp 532-550

[Ste85]

STELMASZYK P.
 "Analyse de scènes dynamiques par recherche des contours en mouvement. Application à la conduite automatique d'un tramway"
Thèse de doctorat
 Université des Sciences et Techniques de Lille Flandres-Artois. Novembre 1985.

[Vie88]

VIEREN C.
 "Segmentation de scènes dynamiques en temps-réel. Application au traitement de séquences d'images pour la surveillance de carrefours routiers"
Thèse de doctorat
 Université des Sciences et Techniques de Lille Flandres-Artois. Avril 1988.

TABLE DES MATIERES

PRESENTATION DU CONTEXTE DE L'ETUDE	3
LE PROJET DE CONDUITE AUTOMATIQUE	3
LA VISION ARTIFICIELLE	4
PLAN DU MEMOIRE	5
I) DETECTION DE MOUVEMENT DANS UNE SEQUENCE D'IMAGES	7
I.1) LE MOUVEMENT DANS UNE SEQUENCE D'IMAGES	7
I.2) METHODES STRUCTURELLES (HAUT NIVEAU)	9
I.2.1) Extraction des éléments structurels	9
I.2.2) Mise en correspondance	10
I.2.3) Avantages et inconvénients	12
I.3) ANALYSE DU FLUX OPTIQUE (NIVEAU INTERMEDIAIRE)	13
I.3.1) Détermination du flux optique	14
I.3.1.a) Corrélation	14
I.3.1.b) Gradient spatio-temporel	14
I.3.1.c) Transformation de Fourier	16
I.3.2) Segmentation	16
I.3.2.a) Analyse directe du flux optique	16
I.3.2.b) Analyse de la transformée de Hough	17
I.3.3) Estimation de paramètres	18
I.3.3.a) Estimation du mouvement	18
I.3.3.b) Reconstruction de formes 3D	19
I.3.4) Avantages et inconvénients	19
I.4) METHODES ORIENTEES MOUVEMENT (BAS NIVEAU)	21
I.4.1) Extraction des régions mobiles	21
I.4.1.a) Utilisation d'une image de référence	22
I.4.1.b) Utilisation d'images successives	24
I.4.2) Avantages et inconvénients	25
I.5) CONCLUSION ET CHOIX D'UNE METHODE	26

II) EVALUATION DES METHODES DE DETECTION DE CONTOURS EN MOUVEMENT	27
II.1) DETECTION DES CONTOURS EN MOUVEMENT	28
II.1.1) Généralités	28
II.1.2) Opérateur de Haynes et Jain	28
II.1.3) Opérateur de Stelmaszyk	30
II.1.4) Opérateur de Vieren	31
II.1.5) Spécificités des opérateurs de détection de contours en mouvement	33
II.2) EVALUATION DES OPERATEURS DE DETECTION DE CONTOURS STATIQUES	36
II.2.1) Généralités	36
II.2.2) Evaluation d'un opérateur de détection de contours	36
II.2.3) Mesures de la qualité d'un opérateur de détection de contours	37
II.2.3.a) Facteur de mérite selon Abdou et Pratt	37
II.2.3.b) Distance moyenne	38
II.2.3.c) Facteur de bruit	39
II.3) APPLICATION AUX OPERATEURS DE DETECTION DE CONTOURS EN MOUVEMENT	40
II.3.1) Mode opératoire	40
II.3.2) Test des opérateurs de détection de mouvement	42
II.4) INTERPRETATION DES RESULTATS	46
II.4.1) Précision de la position du contour	46
II.4.2) Influence du détecteur de contours statiques	49
II.4.2.a) Précision en position	49
II.4.2.b) Facteur de bruit	51
II.5) CONCLUSION	54
III) CONCEPTION ET REALISATION D'UN PROCESSEUR CABLE POUR L'EXTRACTION DES CONTOURS MOBILES EN TEMPS REEL	55
III.1) LES ARCHITECTURES D'ORDINATEUR ADAPTEES AU TRAITEMENT D'IMAGES	57
III.1.1) Classification des machines parallèles	57
III.1.1.a) Machines SISD	57
III.1.1.b) Machines MISD	58
III.1.1.c) Machines SIMD	59
III.1.1.d) Machines MIMD	60

III.1.2) Utilisation en traitement d'images	61
III.1.2.a) Traitements de bas niveau	61
Machines SIMD	62
Machines MISD	62
III.1.2.b) Traitements de moyen et de haut niveau	63
Machines MIMD	64
Machines pyramidales de type MSIMD	64
III.1.2.c) Conclusion	65
III.2) CHOIX D'UNE ARCHITECTURE SPECIFIQUE	66
III.2.1) Cahier des charges	66
III.2.1.a) Interfaçage	66
III.2.1.b) Aspect temps-réel	66
III.2.1.c) Fonctionnement autonome	67
III.2.2) Partitionnement d'un algorithme en opérateurs simples	67
III.2.2.a) Généralités	67
III.2.2.b) Parallélisme de fonction	68
III.2.2.c) Parallélisme spatial	68
III.2.3) Application à l'opérateur de Vieren	69
III.3) IMPLANTATION MATERIELLE	74
III.3.1) Module d'acquisition et de restitution vidéo	74
III.3.2) Modules de retard d'image	75
III.3.3) Modules arithmétiques	76
III.3.4) Module de traitement de voisinage	76
III.4) CONCLUSION	79
IV) APPORT DU PROCESSEUR STREAM POUR L'AMELIORATION DE LA SECURITE DANS LES SYSTEMES DE TRANSPORT GUIDES	80
IV.1) DETECTION ET LOCALISATION DES OBJETS MOBILES EVOLUANT DANS UN CARREFOUR URBAIN	81
IV.1.1) Conditions d'expérimentation	81
IV.1.2) Résultats de l'expérimentation	81
IV.1.2.a) Séquence n° 1	82
IV.1.2.b) Séquence n° 2	87
IV.1.2.c) Séquence n° 3	87
IV.1.3) Conclusion sur l'aide à la conduite	96

IV.2) DETECTION DE CHUTES DE PASSAGERS AU NIVEAU DE L'INTERFACE QUAI VOIE DANS LE METRO PARISIEN	97
IV.2.1) Mise en place de l'expérimentation	98
IV.2.1.a) Conditions d'expérimentation	98
IV.2.1.b) Post-traitement des contours détectés	99
IV.2.2) Résultats de l'expérimentation	100
IV.2.2.a) Caméra oblique	101
IV.2.2.b) Caméra orthogonale aux voies	105
IV.2.3) Conclusion sur la détection de chutes	107
REFERENCES BIBLIOGRAPHIQUES	113
PRESENTATION DE L'ETUDE	113
PREMIER CHAPITRE	114
DEUXIEME CHAPITRE	125
TROISIEME CHAPITRE	128
QUATRIEME CHAPITRE	132

S ystème
T emps -
R éel d'
E xtraction et d'
A nalyse du
M ouvement

Fiches techniques

USTL
FLANDRES ARTOIS



ANNEXE 1

CARTE D'ACQUISITION ET DE RESTITUTION VIDEO

CARTE D'ACQUISITION ET DE RESTITUTION VIDEO

Elle est constituée de deux modules spécifiques :

- **Partie ANALOGIQUE** : qui réalise l'interface entre les signaux vidéo analogiques et le système.
- **Partie NUMERIQUE** : qui assure la génération des différents signaux de séquençement du système en synchronisme avec la vidéo d'entrée.

USTL
FLANDRES ARTOIS



I) PARTIE ANALOGIQUE

1) DESCRIPTION

Elle regroupe six fonctions principales :

- Amplification et mise en forme de la vidéo analogique d'entrée (Adaptation d'impédance et Gain de 3).
- Extraction des signaux de synchronisation contenus dans ce signal (Syn. Verticale, Horizontale et composite, Parité).
- Conversion Analogique/Numérique de la vidéo d'entrée (sur 8 bits - 256 niveaux de gris).
- Oscillateur principal à 20MHz commandé par une boucle à verrouillage de phase (en synchronisme avec la vidéo).
- Conversion Numérique/Analogique de l'image résultat (8 bits).
- Amplification de ce dernier signal (afin de respecter les normes RS170).

2) FONCTIONNEMENT

2.1) MISE EN FORME DU SIGNAL D'ENTREE

L'entrée prise en compte par la carte est un signal vidéo normalisé (RS170) comprenant les informations luminance et synchro (toute information couleur doit être éliminée au préalable si nécessaire). La valeur efficace du signal doit être de 1V sur 75ohms. Le format reconnu par la carte est de 25 images de 625 lignes par seconde (50 trames de 312 lignes entrelacées).

La carte élimine tout d'abord la composante continue pouvant être contenue dans ce signal (Tension résiduelle fournie par l'étage de sortie de la caméra ou du magnétoscope). Elle assure ensuite l'amplification afin d'obtenir un signal interne de 3 Volts efficaces (Amplificateur vidéo intégré SE592).

2.2) EXTRACTION DES SIGNAUX DE SYNCHRONISATION

Un circuit intégré LM1881 réalise ensuite l'extraction des différents signaux de synchronisation contenus dans la vidéo :

- Syn. verticale et horizontale
- Syn. Composite.
- Signal de parité de trame.

L'utilisation d'un signal supplémentaire fourni par ce circuit (Clamping) permet après intégration de détecter la présence de vidéo valide en entrée de la carte

2.3) CONVERSIONS ANALOGIQUE/NUMERIQUE

Un circuit intégré unique (UVC3101 de ITT) assure les deux conversions A/N et N/A.

Le signal d'entrée amplifié par les étages précédents est pris en compte par ce composant qui va lui faire subir certains prétraitements :

- Remise à niveau de la composante continue (Clamping) grâce à un signal fourni par la carte numérique.
- Elimination des parties du signal négatives ou supérieures à 2V (Référence interne).

Il réalise ensuite la conversion proprement dite en signaux numériques à une fréquence imposée par la carte digitale (environ 5MHz : 256 points par ligne).

Les signaux vidéo numériques arrivant de la partie numérique sont envoyés à ce circuit pour y être reconvertis en analogique. Il fournit en sortie uniquement la partie luminance de la vidéo. Les deux fréquences de conversion (A/N et N/A) sont identiques.

2.4) MISE EN FORME DU SIGNAL DE SORTIE

Le signal de luminance provenant du convertisseur est appliqué à un étage dont la fonction est triple :

- Ajout des informations de synchronisation.
- Amplification.
- Adaptation d'impédance.

Afin d'éviter la prise en compte du bruit contenu dans les alimentations continues, un étage à amplificateur opérationnel a été choisi pour réaliser ces différentes fonctions.

Le signal restitué en sortie est conforme à la norme : 1 volt sur 75 ohms.

2.5) OSCILLATEUR 20MHz

Il fournit l'horloge principale du système en synchronisme avec la vidéo d'entrée (1280 périodes d'horloge par ligne vidéo).

Il est du type oscillateur LC dont la capacité est réalisée par une diode Varicap. La tension de commande est fournie par filtrage du signal d'erreur d'une boucle à verrouillage de phase contenue sur la carte numérique.

2.6) LISTE DES COMPOSANTS

Circuits Intégrés :

C11	SE592
C12	LM1881
C13	UVC3101
C14	LM318

Connecteurs :

CO1,CO2	Cable plat 16 pts
COV1,COV2	Embase BNC

Condensateurs :

C2	10pF
C4	Ajustable 10/40pF
C10	47pF
C13	500pF
C3	1nF
C6,C7,C8,C9	100nF
C12	0.35 μ F
C1,C14,C15,C16	47 μ F

Résistances :

R1,R2,R7	56
R21	100
R14	180
R16	470
R11	680
R19	2.2k
R15	10k

R8	15k
R4,R20	18k
R5	33k
R10	47k
R9	68k
R6	100k
R17,R18	180k
R3	680k

Selfs :

L1,L2	Selfs de couplage
L3	2.2 μ H sur pot ferrite
L5,L6,L7	CHOC

Condensateurs de découplage :

DEC1,...,DEC10	Découplage 100nF
----------------	------------------

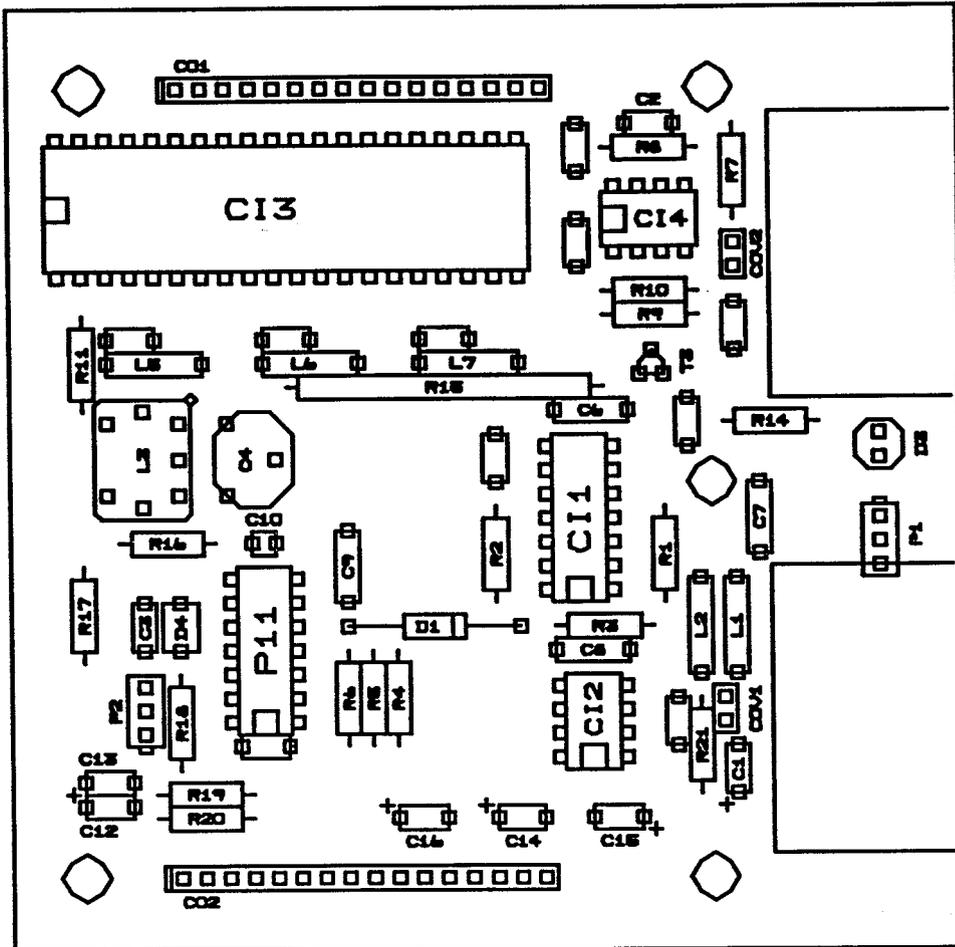
Potentiomètres :

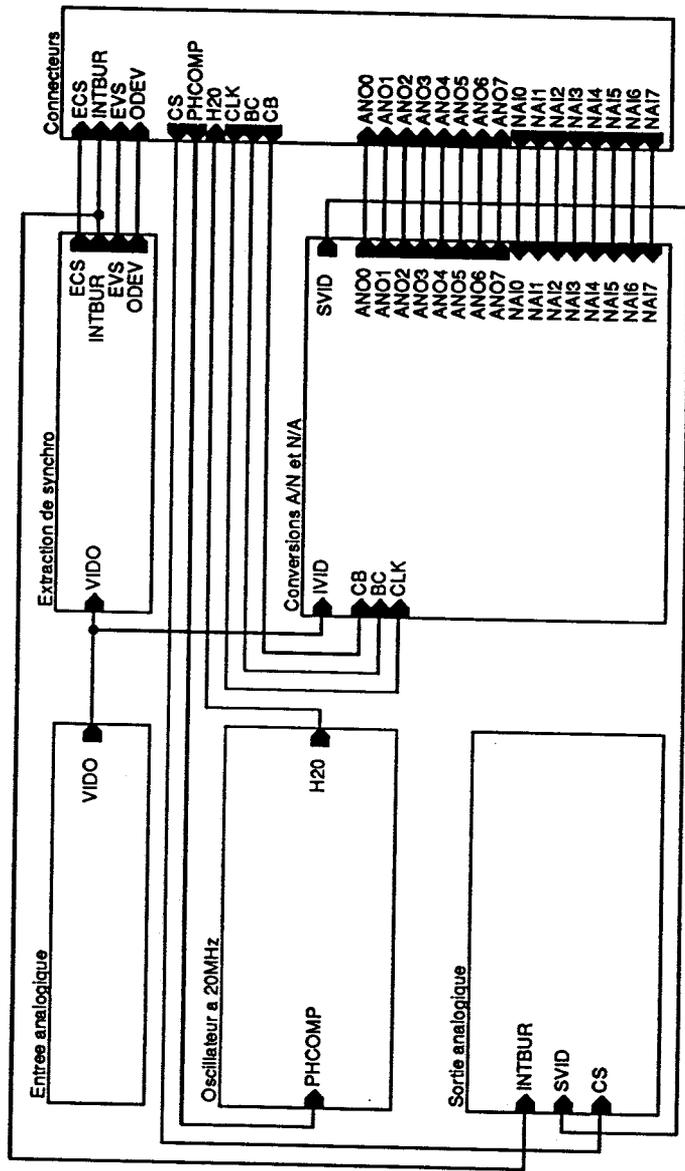
P1	4.7k
P2	22k

Divers

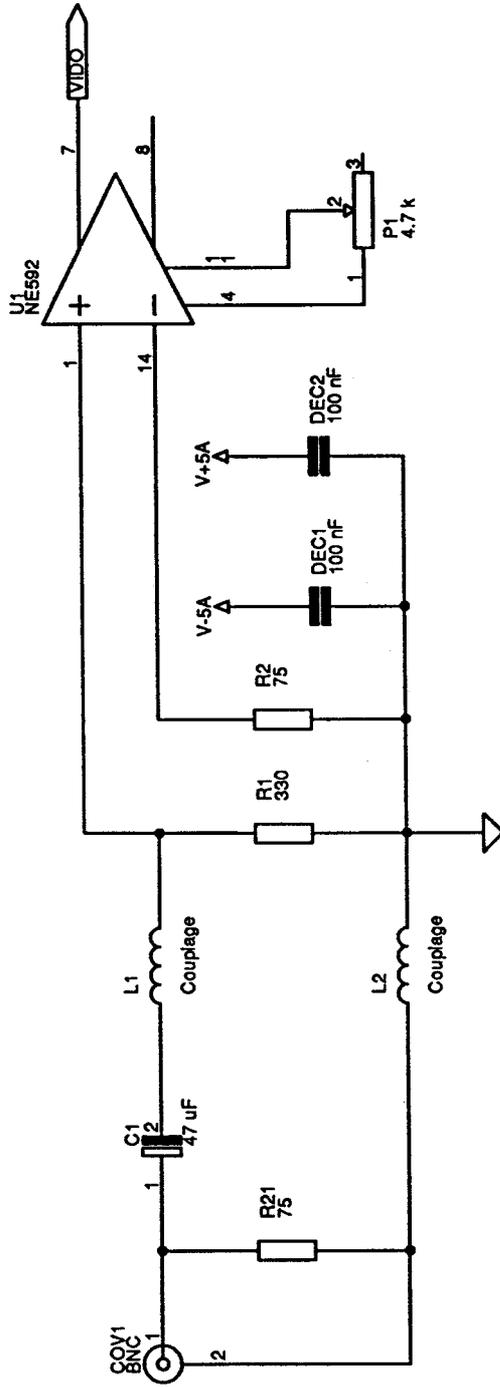
D4	VaricapBB109
D1	1N4148
D3	LED
T3	BC548B

3) IMPLANTATION DES COMPOSANTS





F. Cabestaling	
Carte d'acquisition analogique	
Title	
Size	Synoptique
A	Document Number
	1
Date:	REV
Novembre 7, 1990	1
	Sheet
	1
	of
	7



F. Cabestaing
 Carte d'acquisition analogique

Title

Entree video analogique

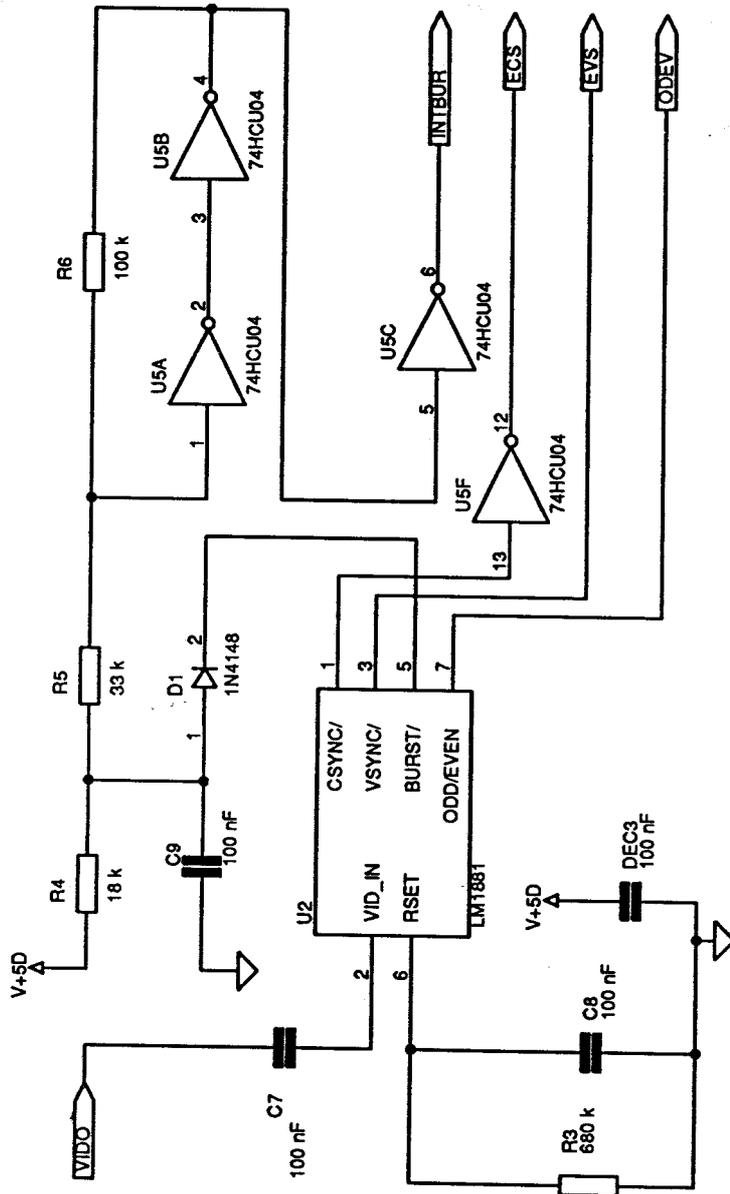
Size A

Document Number 1

REV 1

Date: Novembre 7, 1990

Sheet 2 of 7



F. Cabestaing

Carte d'acquisition analogique

Title

Extraction des signaux de synchronisation

Size

Document Number

A

1

REV

1

7

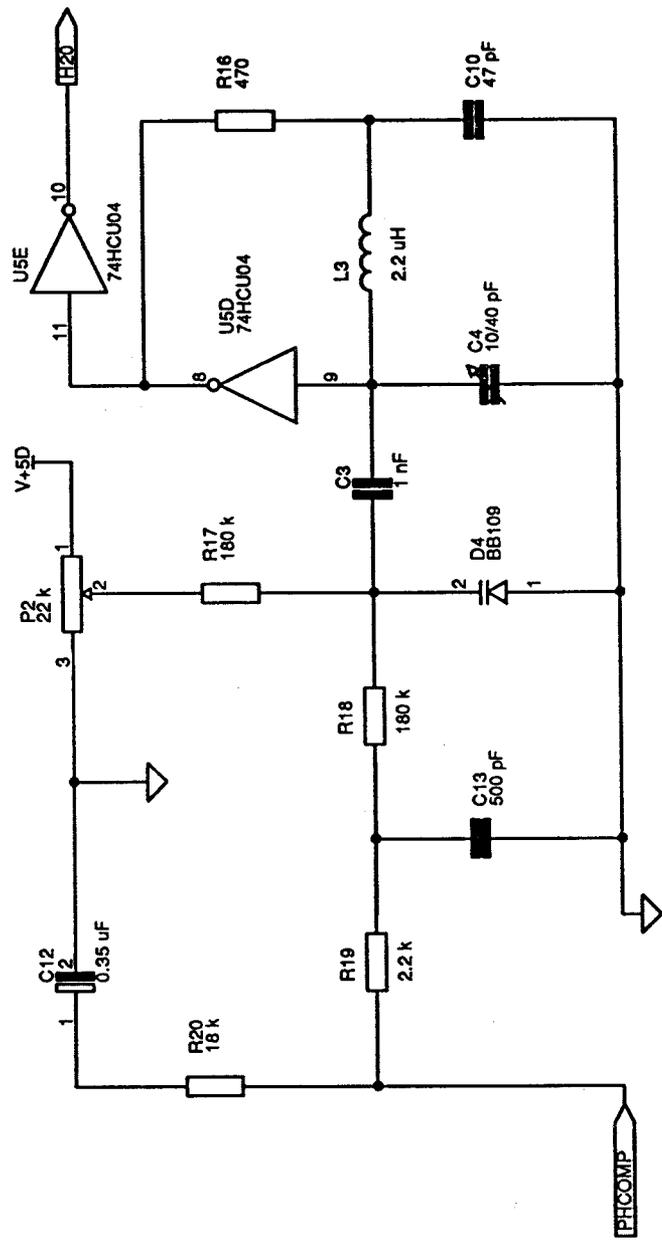
Date: Novembre 7, 1990

Sheet

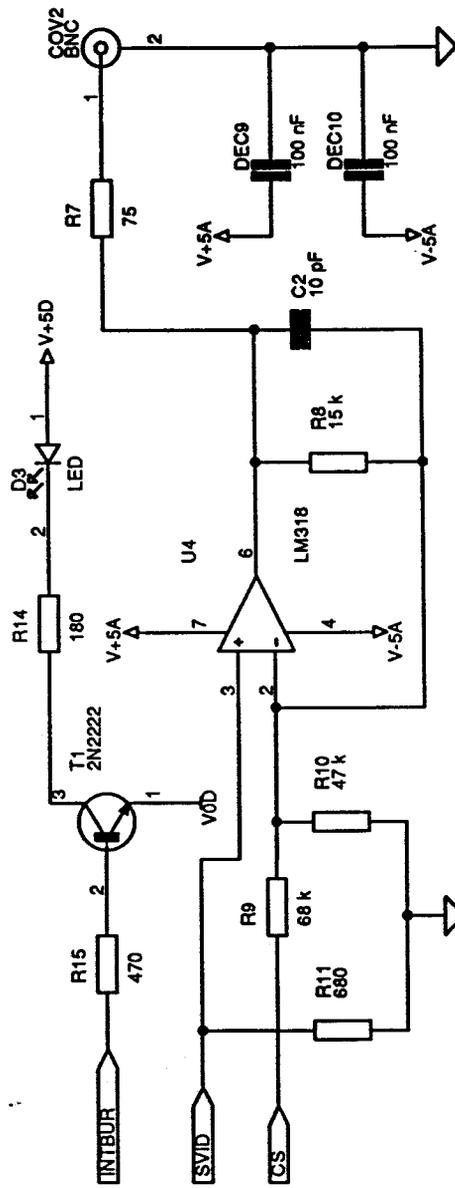
3

of

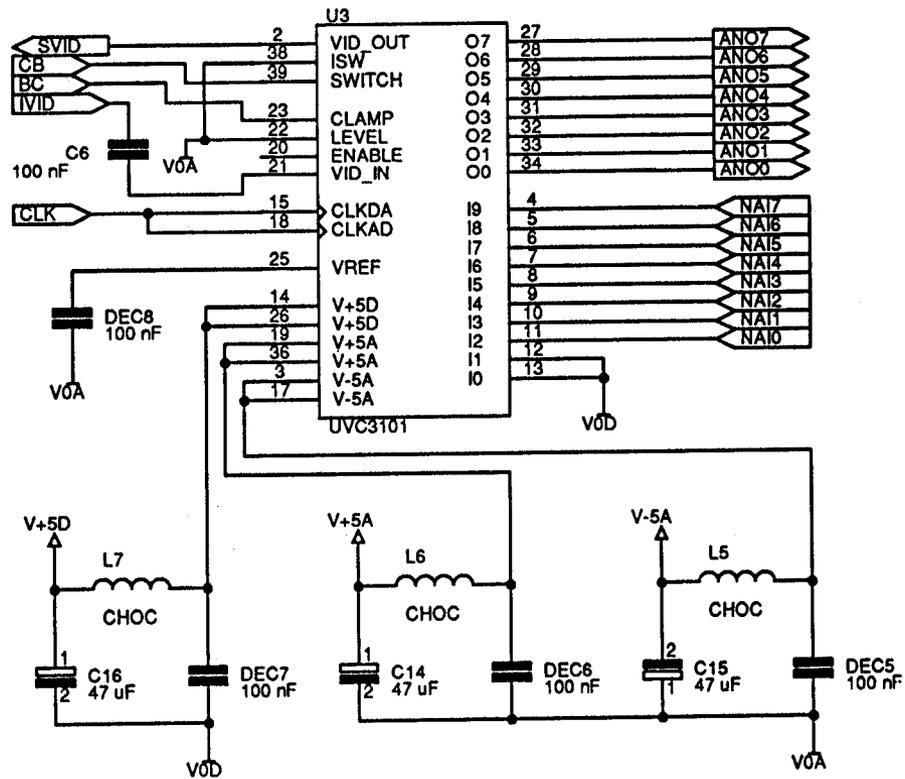
7



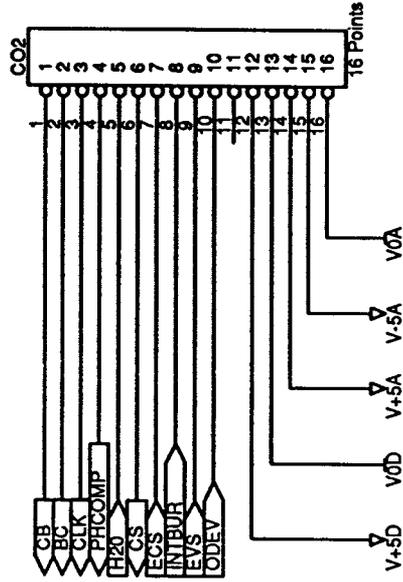
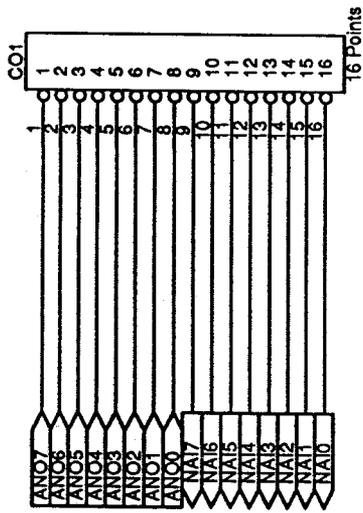
F. Cabestaing	
Carte d'acquisition analogique	
Title	
Oscillateur 20 MHz	
Size	Document Number
A	1
Date:	REV
Novembre 7, 1990	1
Sheet	4 of 7



F. Cabestaing	
Carte d'acquisition analogique	
Title	
Sortie video analogique	
Size	Document Number
A	1
Date:	November 7, 1990
Sheet	5 of 7
REV	1



F. Cabestaing		
Carte d'acquisition analogique		
Title		
Conversions A/N et N/A		
Size	Document Number	REV
A	1	1
Date:	Novembre 7, 1990	Sheet 6 of 7



F. Cabestaing	
Carte d'acquisition analogique	
Title	
Connecteurs inter-cartes	
Size	Document Number
A	1
Date:	REV
Novembre 7, 1990	1
Sheet	of
7	7

II) PARTIE NUMERIQUE

1) DESCRIPTION

Elle réalise deux fonctions distinctes :

- Génération des horloges et différents signaux nécessaires au fonctionnement des autres cartes contenues dans le système.
- Mémorisation du dernier résultat provenant d'une autre carte pour affichage durant toutes les trames inactives suivantes.

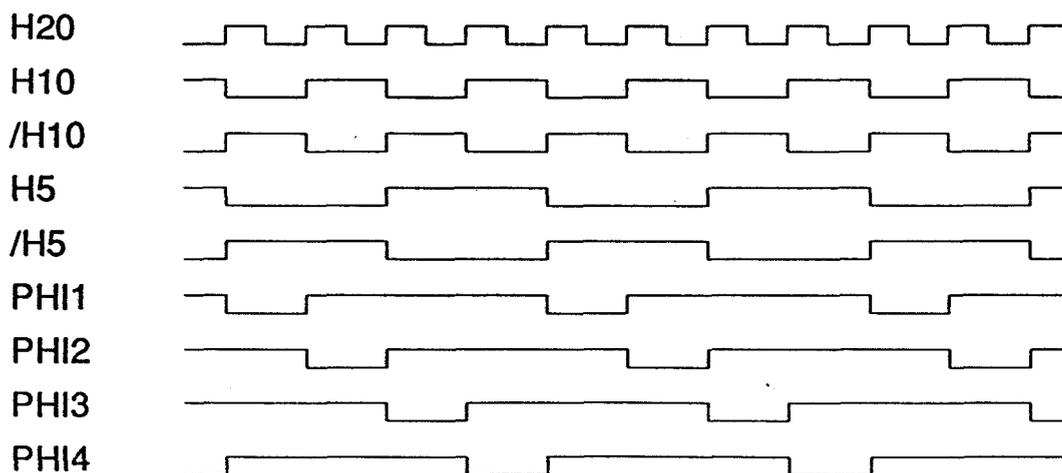
2) FONCTIONNEMENT

2.1) GENERATION DES HORLOGES

L'horloge à 20MHz provenant de la partie analogique est traitée par une PAL (PAL003) qui fournit par division et combinaison différents signaux :

- Deux horloges à 10MHz en opposition de phase (H10 et $\overline{H10}$).
- Deux horloges à 5 MHz en opposition de phase (H5 et $\overline{H5}$).
- Quatre signaux de phase de durée active 50nS et inactive 150nS (PHI1, PHI2, PHI3 et PHI4).

Les positions temporelles relatives des différents signaux sont représentées sur le graphe suivant :



Le signal H5 est ensuite appliqué à un circuit générateur de synchro TV (SAA1043 de RTC) qui va donner par divisions successives tous les signaux de gestion vidéo. Il contient en outre un comparateur de phase à fenêtre qui permet d'asservir les signaux générés à un signal de synchro composite extérieur. Ce dernier est extrait de la vidéo d'entrée par la carte analogique puis appliqué sur la broche de référence du SAA1043. Le signal d'erreur fourni par ce circuit sert ensuite à piloter le VCO.

Tous les signaux résultant du 20MHz sont donc asservis en phase sur la vidéo d'entrée. Six d'entre eux sont envoyés sur le bus de fond de panier :

- Synchro composite (CS).
- Synchros verticales (2 décalées : V1 et V2).
- Synchros horizontales (2 décalées : H1 et H2).
- Signal de blanking (Effacement ligne et trame : CB).

Deux signaux supplémentaires sont envoyés vers le bus :

- Un signal de présence de vidéo valide en entrée : INTBUR (fourni par la partie analogique).
- Un signal de validation de trame de calcul (TRAMEN).

Ce dernier est actif pendant une trame paire sur n , n étant programmable de 1 à 16 (cf annexe). Il est généré par comptage du nombre d'images (circuit 74LS163) puis est combiné au signal de validité vidéo INTBUR. Ce signal permettra en fait de dévalider les calculs effectués par les autres cartes du système selon le "Numéro" d'image.

2.2) MEMORISATION D'IMAGE

Etant donné que certaines trames peuvent être considérées comme inactives vis à vis du traitement, Une restitution pure et simple des seules images valides entrainerait une discontinuité du signal vidéo de sortie. L'adjonction d'une mémoire de trame dans la circuiterie assurant la restitution du signal vidéo permet d'éviter ce problème. Le dernier résultat valide est donc mémorisé en vue d'être répété en sortie durant les phases inactives.

La partie mémoire de trame peut se décomposer en deux blocs :

- Le séquençement et la gestion des adresses.
- Les circuits de mémorisation.

Les différents signaux de séquençement sont fournis par une PAL combinatoire (PAL 001) à partir des signaux d'horloge (PHI ...), de synchronisation (CS, H1 ...) et du signal de validité vidéo INTBUR. La fenêtre de mémorisation (ligne et colonne de début de fenêtre) est programmable par straps (cf annexe)

La partie mémoire est constituée des deux circuits RAM 32k x 8 bits de temps d'accès maximum 100nS.

Durant les trames actives, la partie séquençement autorise l'écriture dans les RAMs des pixels provenant du bus numérique d'entrée ainsi que leur conversion en vidéo par la partie analogique. Durant la période d'inactivité du traitement, le contenu des mémoires est envoyé au convertisseur pour assurer la continuité de la restitution.

2.3) Liste des composants

Circuits Intégrés :

U1,U2	55257-10
U3,U4,U7,U8	74541
U5,U6	74867
U9,U10	DEFFEN
U11	74163
U12	HORLOG(16R8)
U13	SAA1043
U14	COMBIN(16L8)

Connecteurs :

CO1,CO2	Extension 16 pts
CO3,CO4	Cable plat 16 pts
CO5	Connecteur Europe G96

Condensateurs :

C1	47 μ F
C2	47 μ F
C3	47 μ F

Résistances :

R1	2.2M
R2	3.9k
RR1,RR2,RR3	4.7k

Selfs :

L1,L2	Self de choc
-------	--------------

Condensateurs de découplage :

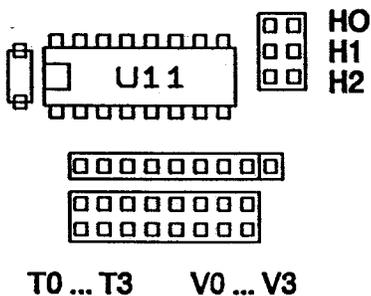
DEC1,...,DEC14	Découplage 100nF
----------------	------------------

Divers

ST1	STRAP8
ST2	STR3

2.4) Programmation des straps

Des straps de configuration permettent de définir le comportement de la carte d'acquisition. Ils représentent les valeurs binaires de trois paramètres. Le nombre de trames inactives à insérer entre deux trames actives. Le nombre de lignes inactives à insérer entre la synchro verticale et la première ligne prise en compte. Le nombre de pixels inactifs en début de ligne vidéo.



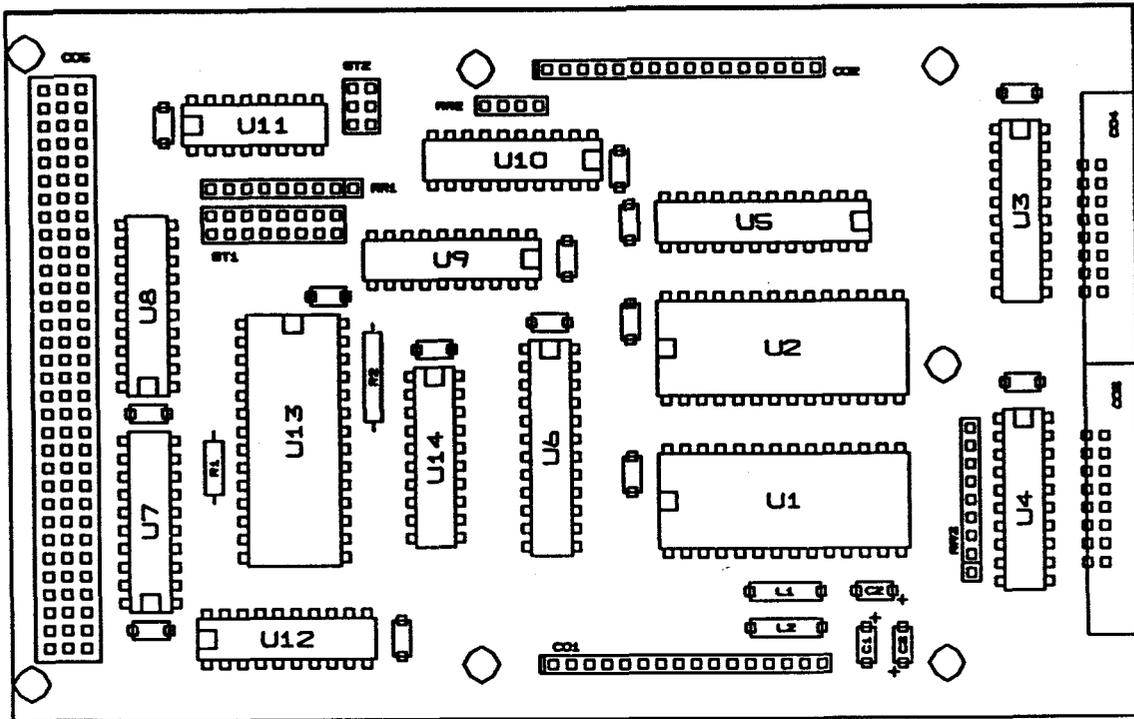
La figure ci-contre précise la position de ces straps sur la carte numérique. Un strap placé représente un bit à zéro. Les rangées de straps représentent successivement :

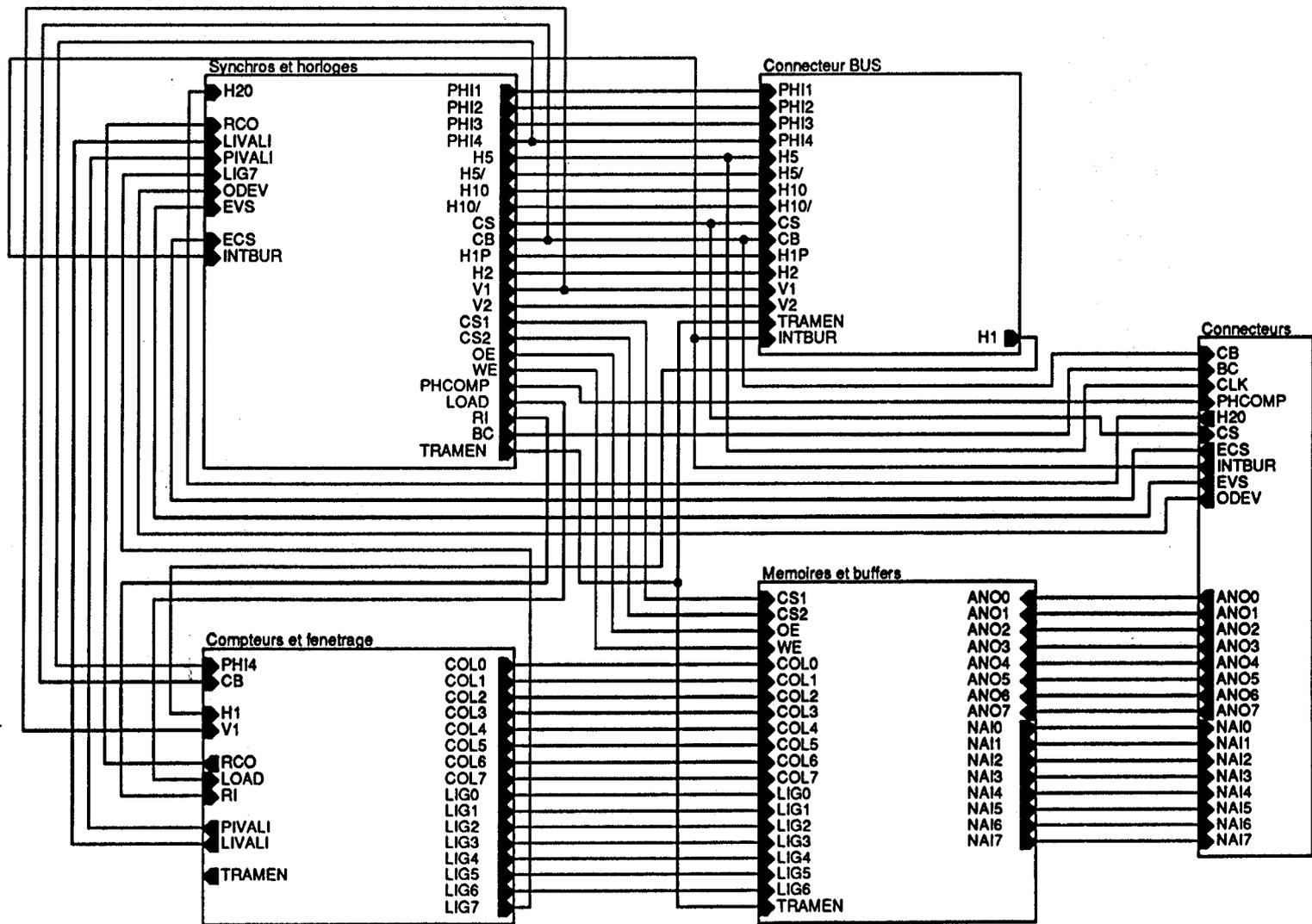
H2 ... H0 : Pixels inactifs

V3 ... V0 : Lignes inactives

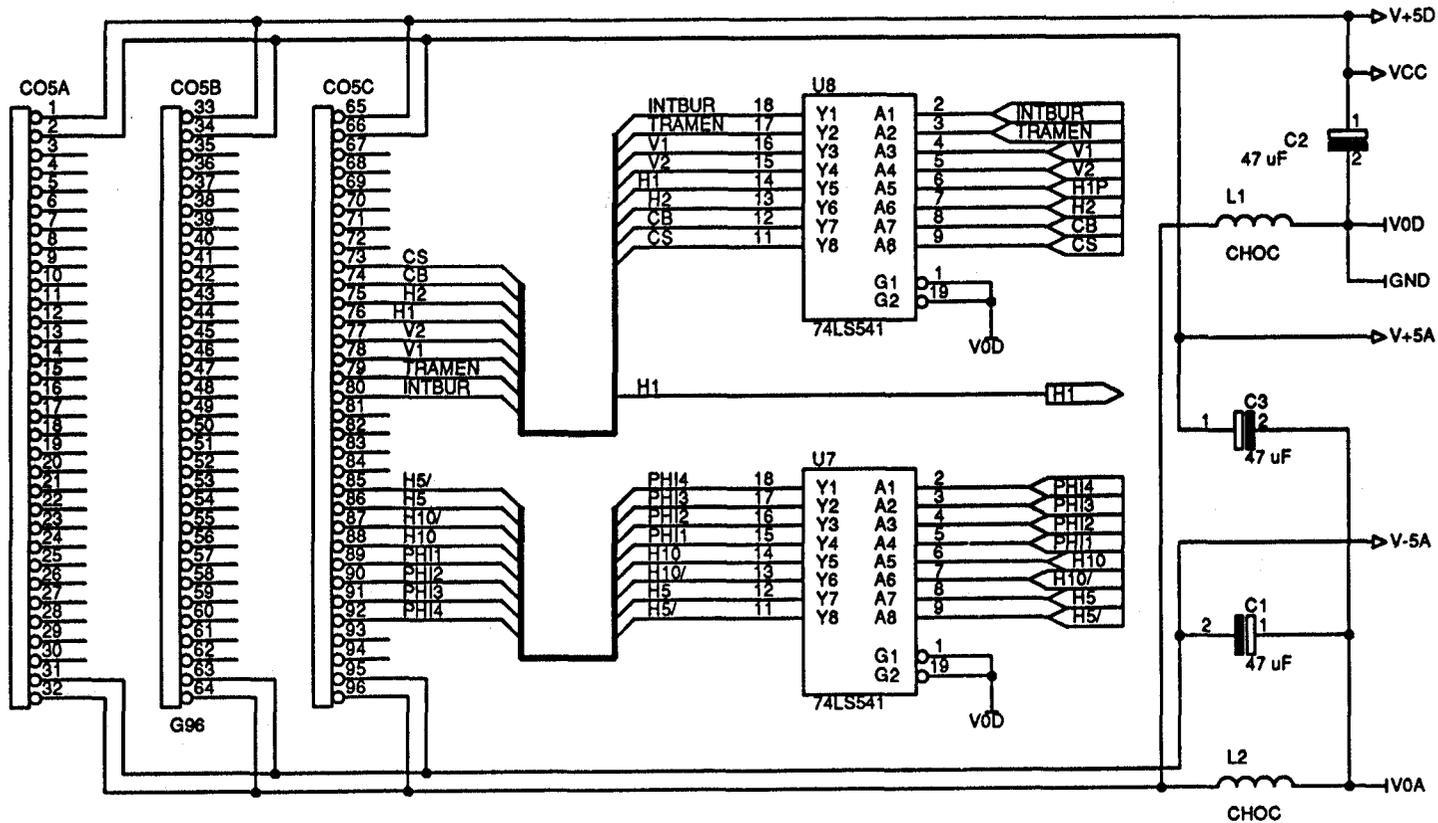
T3 ... T0 : Nombre de trames vidéo inactives

3) IMPLANTATION DES COMPOSANTS

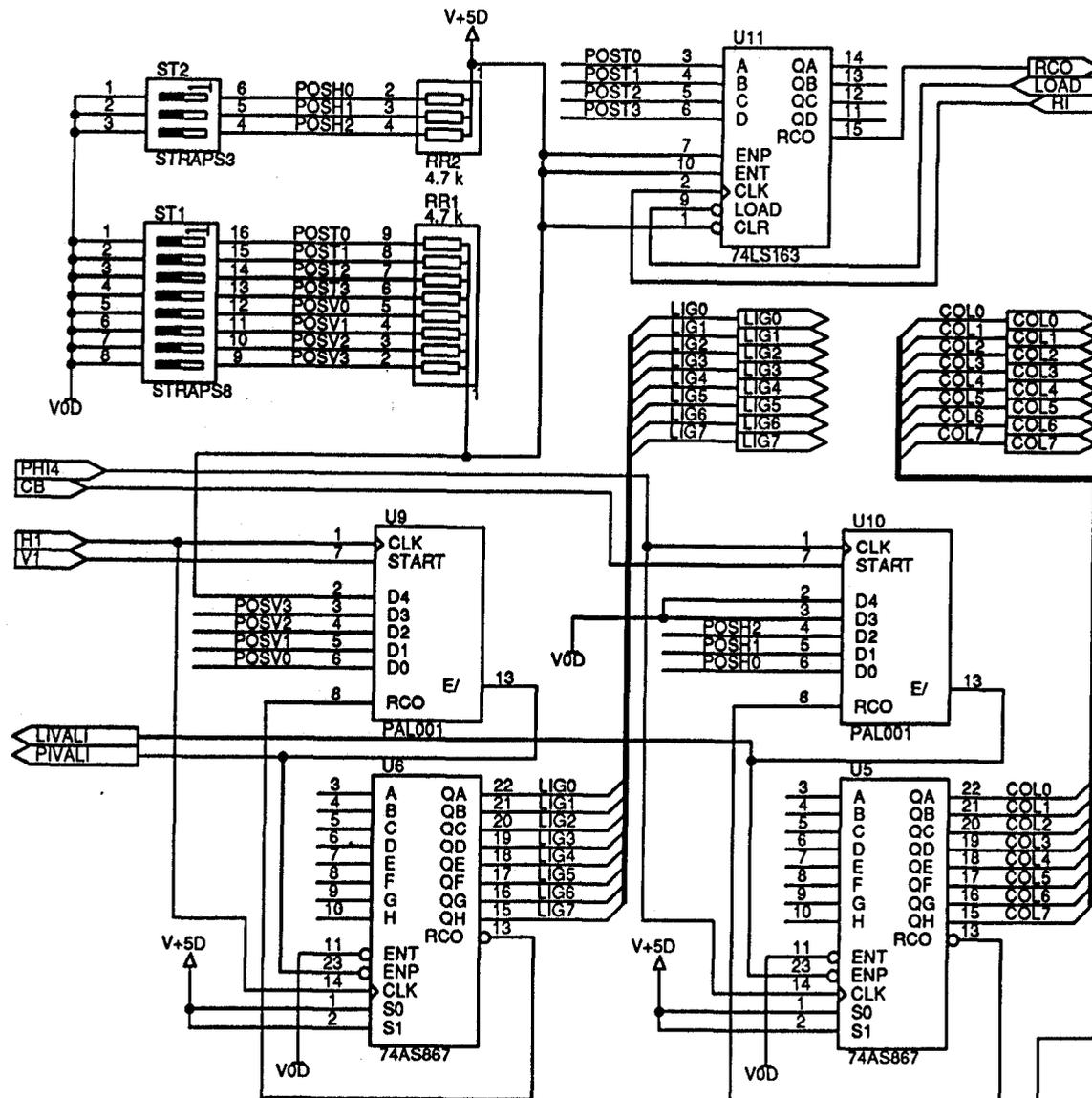




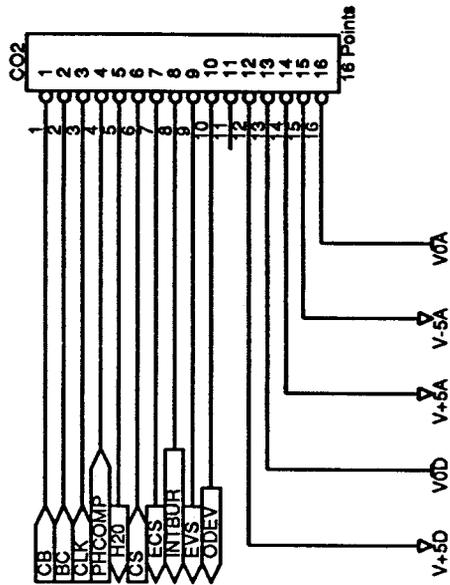
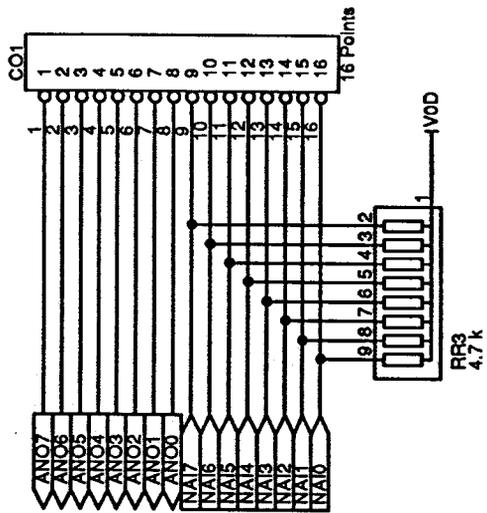
F. Cabestain			
Carte d'acquisition numerique			
Title			
Synoptique			
Size	Document Number		REV
A	1		1
Date:	September 11, 1990	Sheet	1 of 6



F. Cabestaing		
Carte d'acquisition numerique		
Title		
Connecteur BUS		
Size	Document Number	REV
A	1	1
Date:	Novembre 7, 1990	Sheet 2 of 6



F. Cabestaing		
Carte d'acquisition numerique		
Title		
Compteurs et fenetrage		
Size	Document Number	REV
A	1	1
Date:	Novembre 7, 1990	Sheet 3 of 6



F. Cabestaing
Carte d'acquisition numerique

Title

Connecteurs inter-cartes

Size

A

Document Number

1

REV

1

Date:

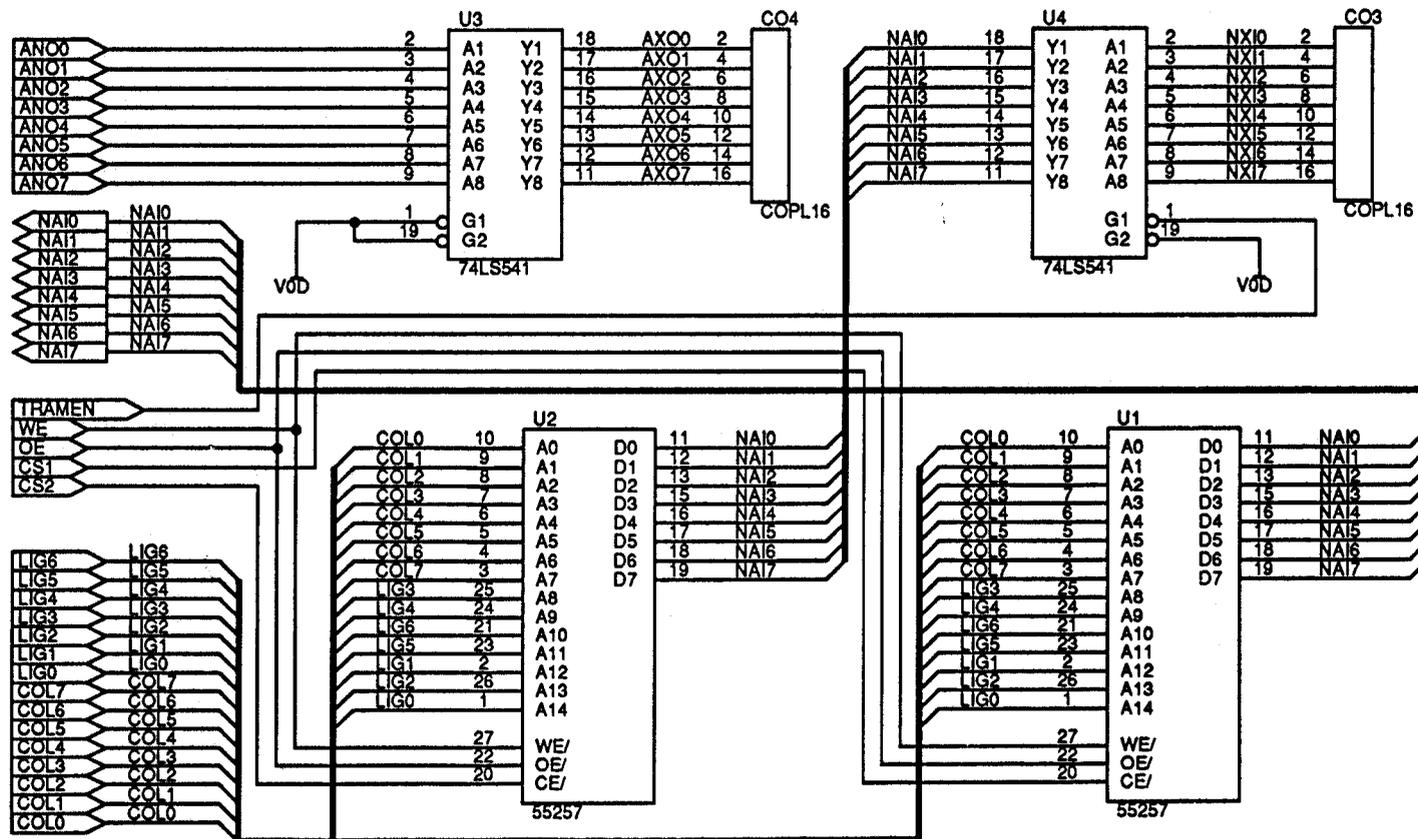
Novembre 7, 1990

Sheet

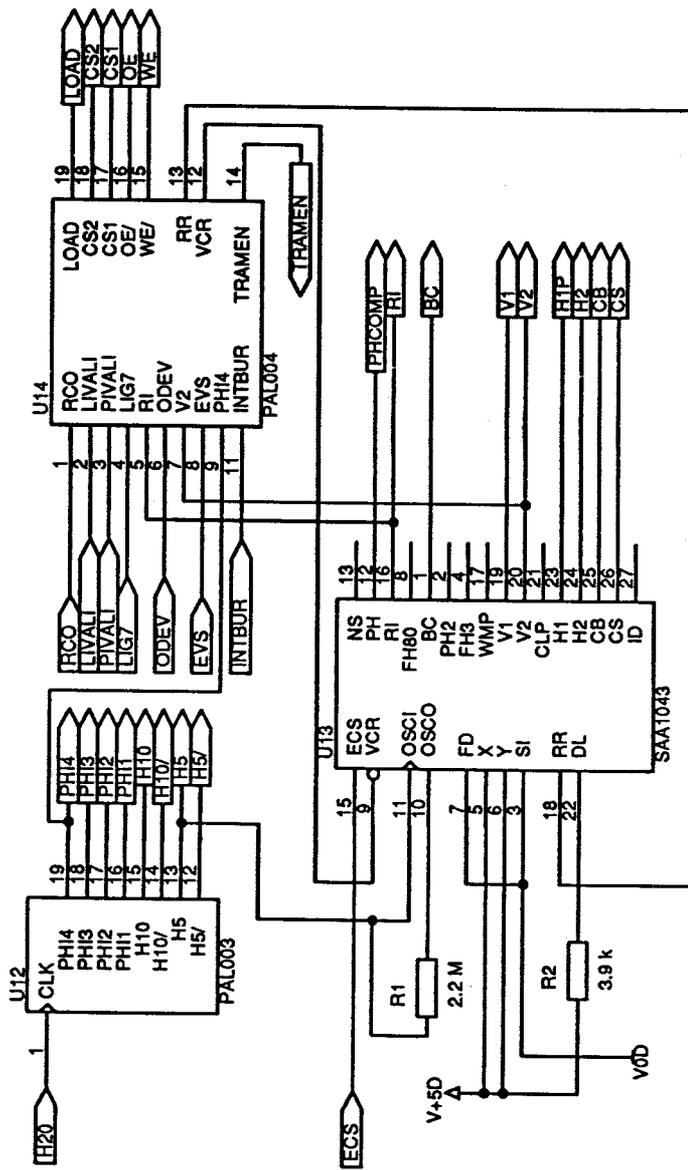
6

of

6



F. Cabestaing		
Carte d'acquisition numerique		
Title		
Memorisation de trame		
Size	Document Number	REV
A	1	1
Date:	Novembre 7, 1990	Sheet 5 of 6



F. Cabestain	
Carte d'acquisition numerique	
Title	
Generation des horloges	
Size	Document Number
A	1
Date:	September 11, 1990
Sheet	4 of 6
REV	1

ANNEXE 2

CARTE DE RETARD D'IMAGE

CARTE DE RETARD D'IMAGE

Elle permet d'ajouter un retard d'une image active sur le flot vidéo numérique. Elle conserve en mémoire les différentes images successives utilisées dans l'algorithme de détection de mouvement. Ces images sont restituées en synchronisme avec l'image courante en cours d'acquisition.

USTL
FLANDRES ARTOIS



1) DESCRIPTION

Les données constituant le flot vidéo numérique sont stockées dans une mémoire de 256 lignes sur 256 colonnes (64k x 8 bits) organisée en FIFO : L'écriture d'un pixel sur la carte entraîne la lecture et la restitution simultanée du pixel qui occupait sa place dans la mémoire.

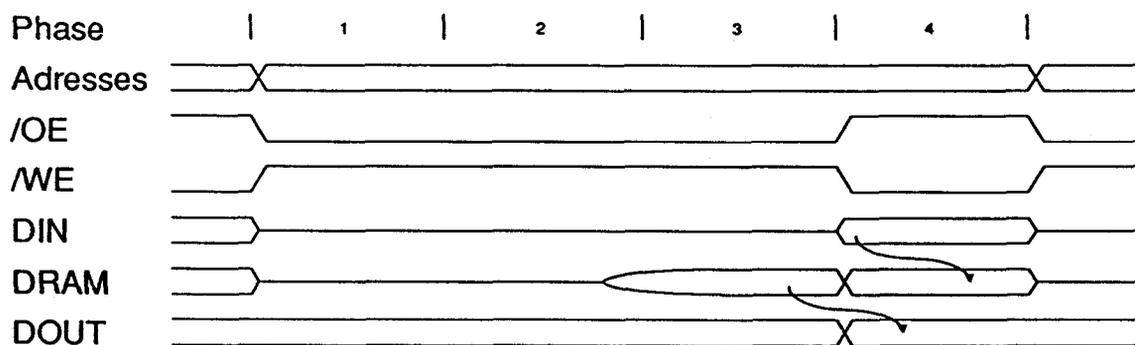
La fenêtre de mémorisation de trame (ligne et colonne de début) est programmable par straps. La carte n'ajoute pas de décalage spatial au niveau du pixel : Le point image restitué est situé au même endroit que celui qui est mémorisé.

L'adjonction d'une carte fille comportant uniquement une mémoire de trame supplémentaire permet de disposer de deux retards trame dans un même emplacement du bus (mémorisation de deux images actives indépendantes).

2) FONCTIONNEMENT

Les signaux permettant la synchronisation ainsi que les horloges de cadencement sont récupérés sur le bus de fond de panier et amplifiés. Une PAL combinatoire (PAL005) fournit les informations de gestion des RAMs et de validation de fenêtre temporelle.

Les adresses mémoire sont générées par deux compteurs 8 bits synchrones eux même pilotés par deux PALs PAL001 (gestion par straps du début de fenêtre). Le cycle d'accès mémoire élémentaire est explicité sur le synoptique suivant :



Les lignes d'adresse mémoire sont incrémentées au début de la première phase (PHI1 active). Le temps d'accès des RAMs étant de 100ns, les données deviennent valides durant la phase PHI3. Elles sont lachées par le circuit de sortie en fin d'activité de PHI3.

Les données d'entrée (pixel courant) devenant valides en début de phase PHI4 sont alors présentées en écriture sur la mémoire (validation de l'écriture en fin de phase PHI4). Ce timing permet de disposer en sortie du pixel mémorisé durant la trame précédente sans lui imposer de cycle de décalage temporel (restitution en début de phase PHI4).

2.1) LISTE DES COMPOSANTS

Circuits Intégrés :

U1,U2	55257
U3	74273
U4,U7,U8	74541
U5,U6	74867
U9,U10	PAL001
U14	PAL005

Connecteurs :

CO1,CO3	Cable plat 16 pts
CO2,CO4	Extensions 16 pts
CO5	Connecteur Europe G96

Condensateurs :

C2	47μF
----	------

Résistances :

RR1	4.7k
-----	------

Selfs :

L1	Self de choc
----	--------------

Condensateurs de découplage :

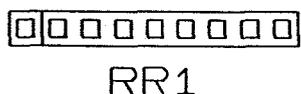
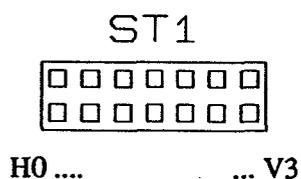
DEC1,...,DEC14	Découplage 100nF
----------------	------------------

Divers

ST1	7 Cavaliers pas 2,54
-----	----------------------

2.2) CHOIX DE LA FENETRE DE CALCUL (STRAPS)

Des straps disposés sur la carte mère permettent de sélectionner les positions horizontale et verticale de la fenêtre de calcul. 16 choix sont possibles quant à la position verticale (ligne de début : 4 straps), 8 pour la position horizontale (pixel de début : 3 straps).



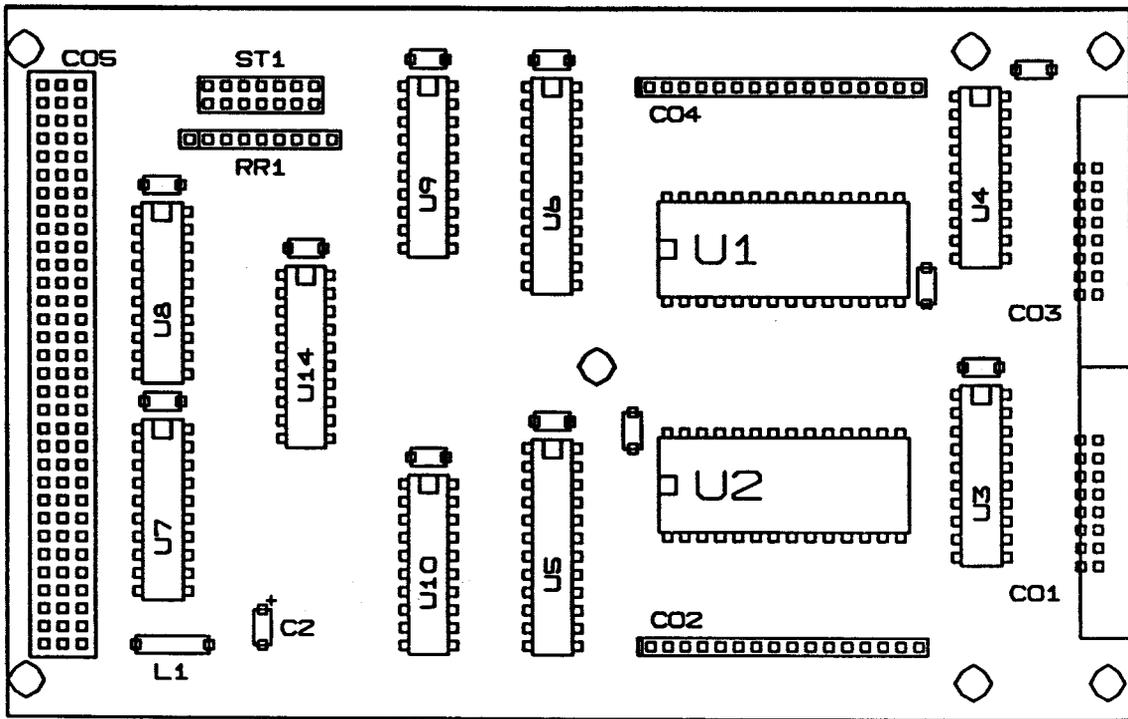
Le nombre de lignes (ou pixels) à inhiber après le top de synchro trame (ou ligne) est fixé par sa représentation binaire. Un strap placé équivaut à un bit à 0.

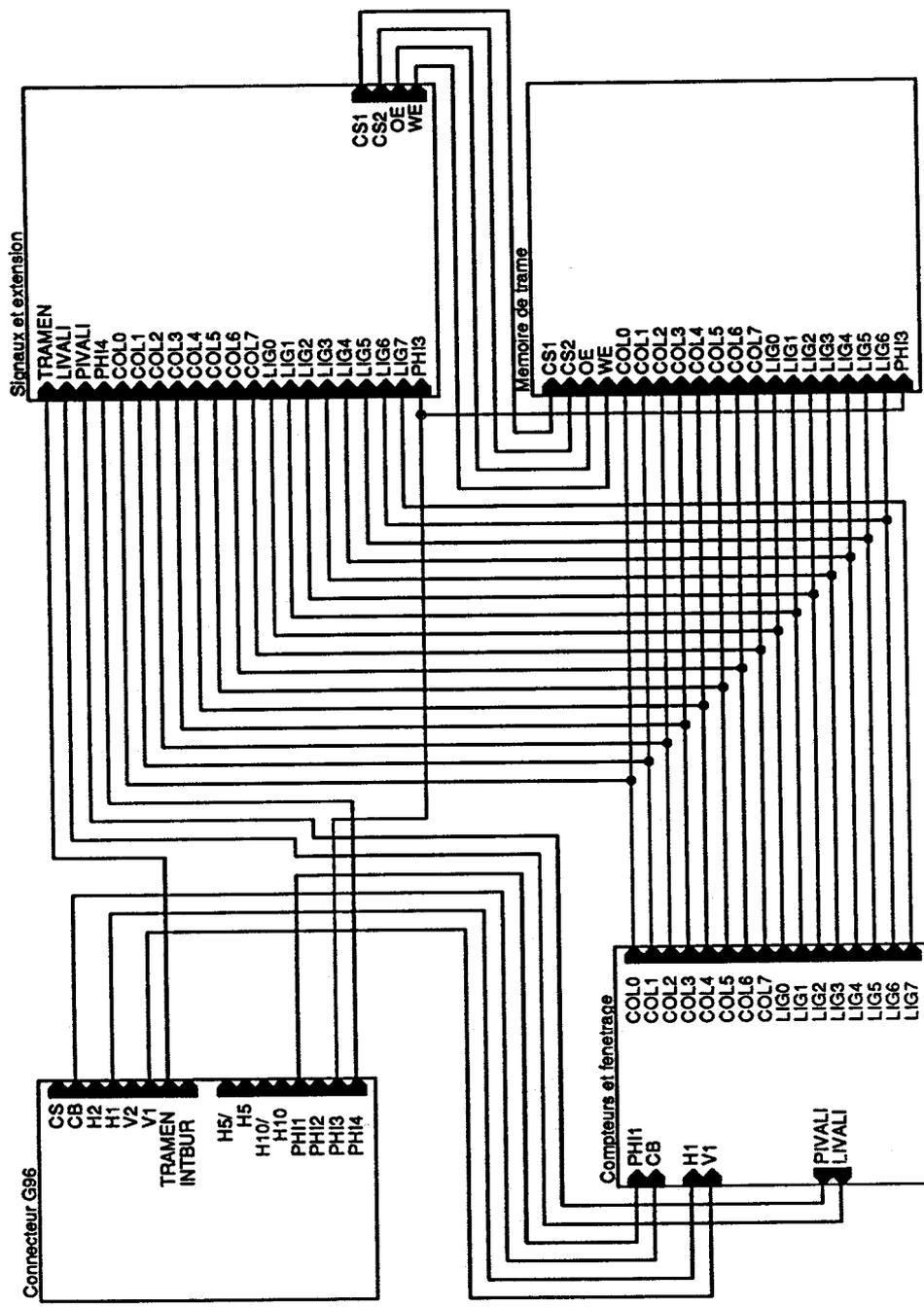
H0 .. H2 : Position horizontale

V0 .. V3 : Position verticale

H0 H1 H2 V0 V1 V2 V3

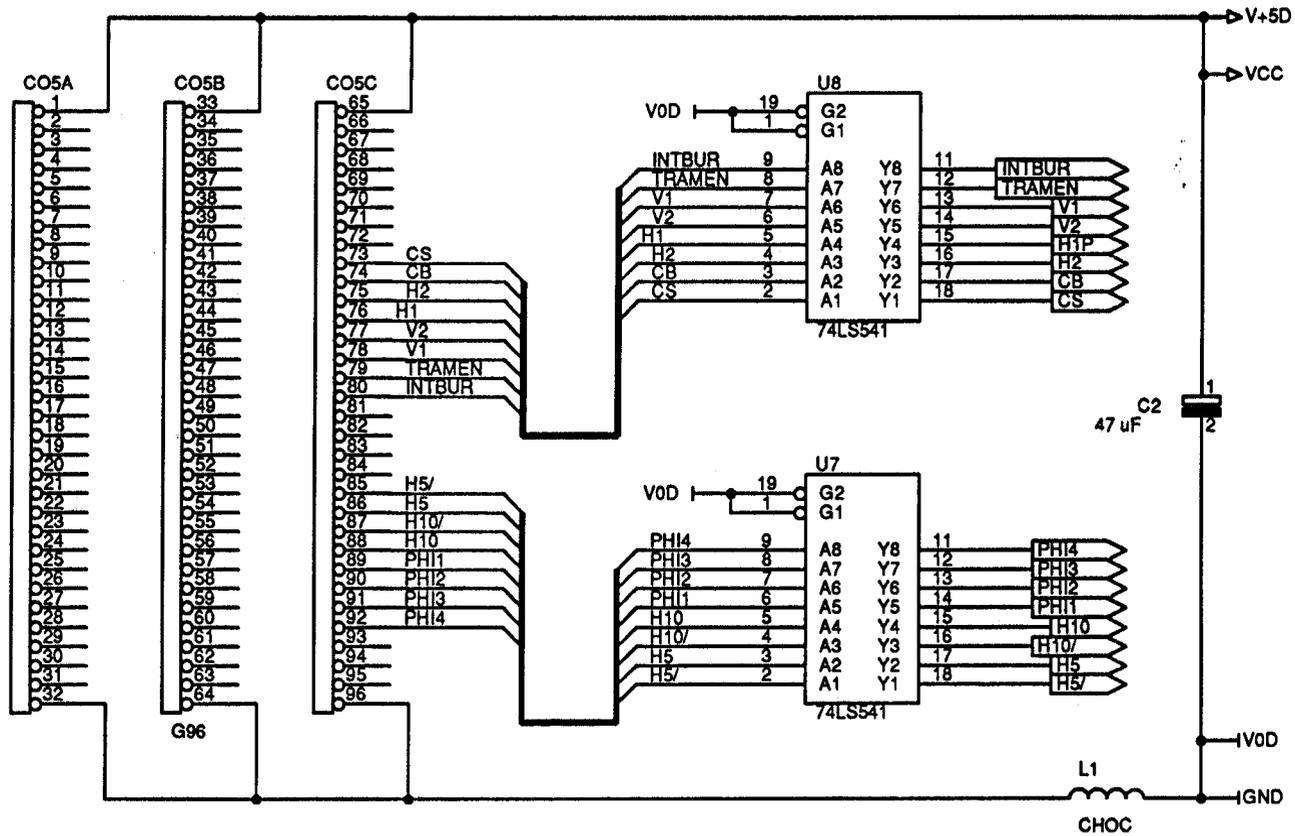
3) IMPLANTATION DES COMPOSANTS



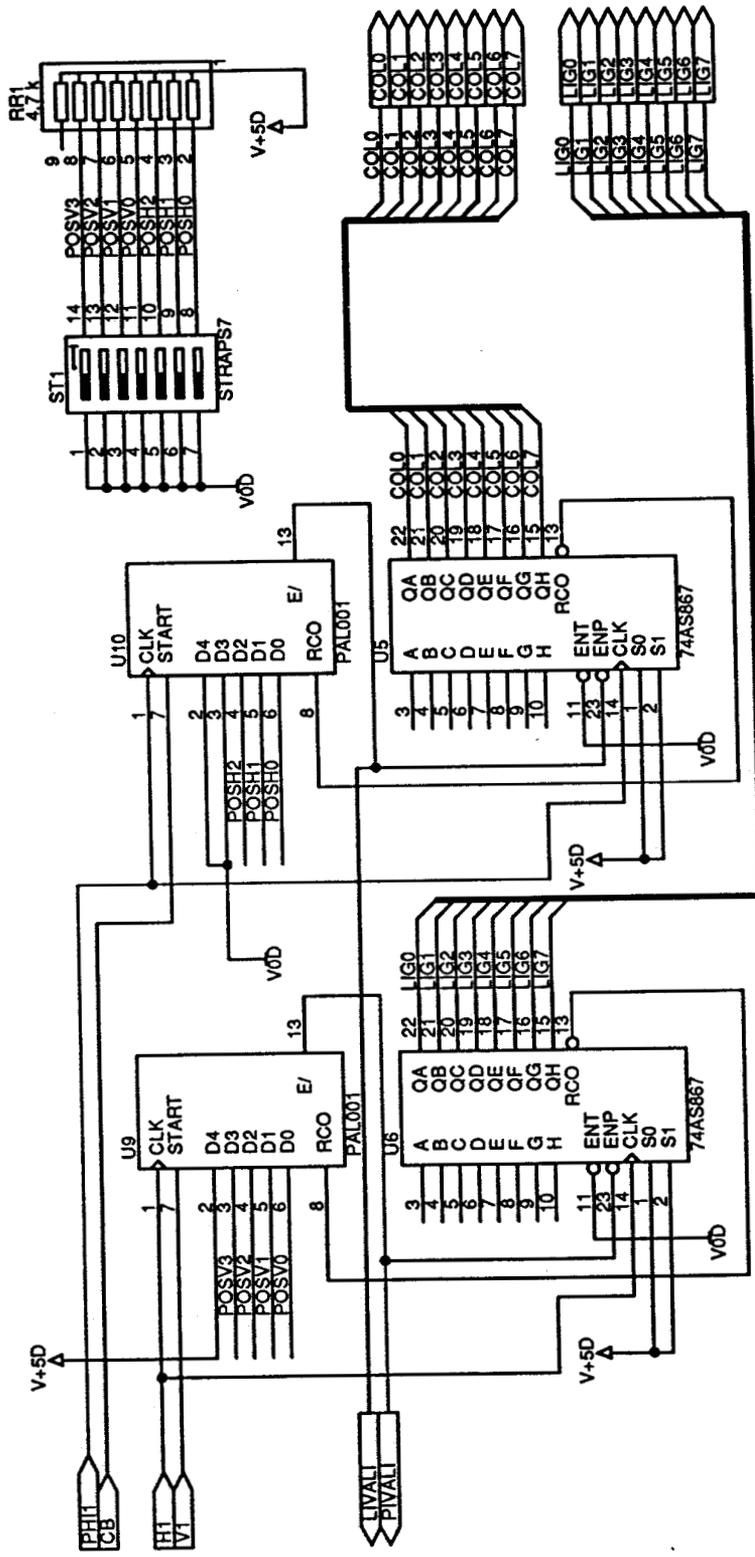


F. Cabestaing
Carte de retard d'image

Title	
Size	REV
A	1
Date: Novembre 7, 1990	
Sheet	of
1	5



F. Cabestaing		
Carte de retard d'image		
Title		
Connecteur G96		
Size	Document Number	REV
A	1	1
Date:	Novembre 7, 1990	Sheet 2 of 5



F. Cabestaing
 Carte de retard d'image

Title
 Compteurs et fenêtrage

Size
 A

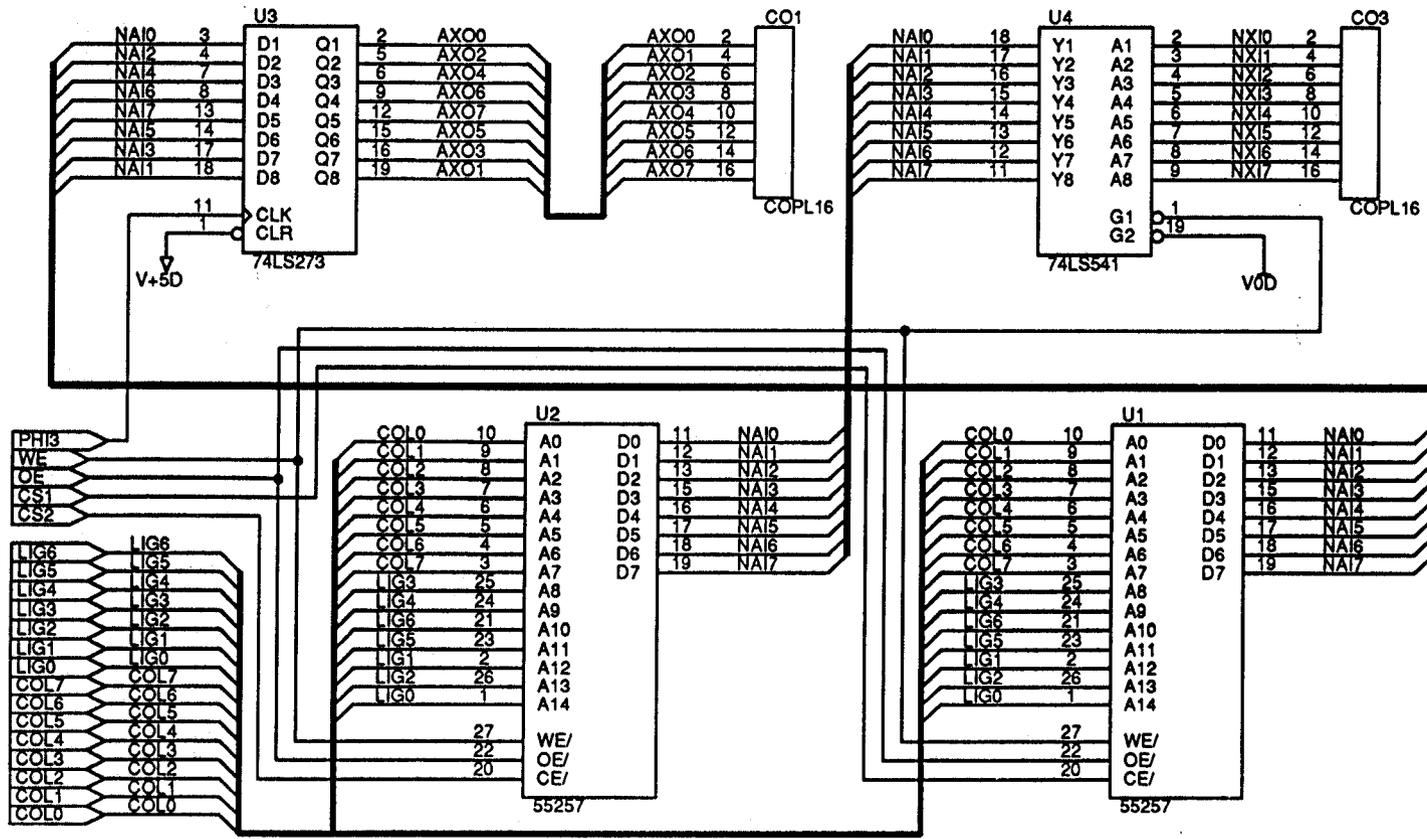
Document Number
 1

Date:
 Novembre 7, 1990

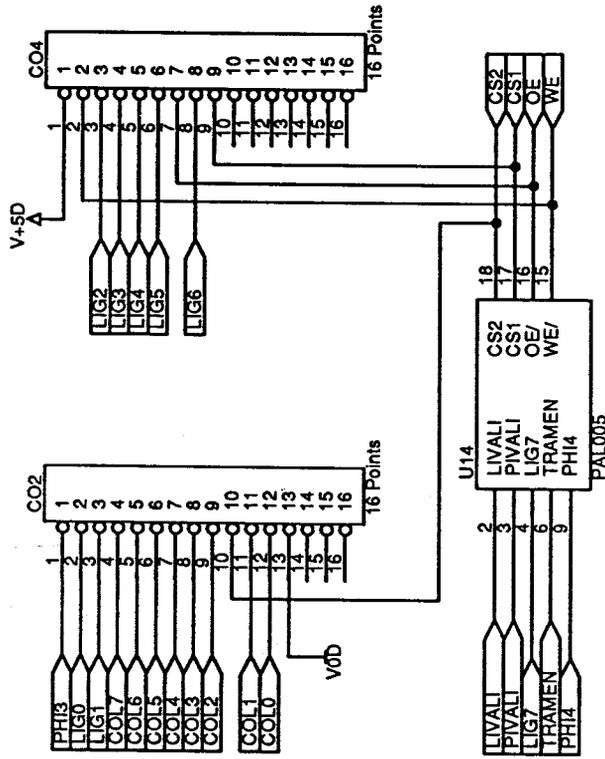
Sheet
 3

of
 5

REV
 1



F. Cabestaing		
Carte de retard d'image		
Title		
Memoire de trame		
Size	Document Number	REV
A	1	1
Date:	Novembre 7, 1990	Sheet 4 of 5



F. Cabestaing
Carte de retard d'image

Signaux et extension

Size Document Number

A 1

Date: Novembre 7, 1990

Sheet 5 of 5

REV 1

MODULE DE RETARD D'IMAGE

Ce module, enfichable sur une carte de retard d'image principale, permet de disposer d'un plan mémoire supplémentaire pour le retard d'image.

USTL
FLANDRES ARTOIS



1) DESCRIPTION

Ce module permet de retarder le flot vidéo numérique d'une trame de traitement active. Il fonctionne dans les mêmes conditions que la carte de retard sur laquelle il est enfiché (fenêtre spatiale identique). Une carte principale ne peut recevoir qu'un seul module additionnel. L'image mémorisée par ce module est totalement indépendante de celle envoyée sur la carte mère.

2) FONCTIONNEMENT

Le fonctionnement est le même que celui de la carte principale au niveau des cycles d'accès mémoire. Seuls les circuits de mémoire et de gestion du bus vidéo sont présents sur ce module, les adresses RAM et différents signaux de synchronisation sont fournis par la carte mère.

2.1) LISTE DES COMPOSANTS

Circuits Intégrés :

U1,U2	55257
U3	74273
U4	74541

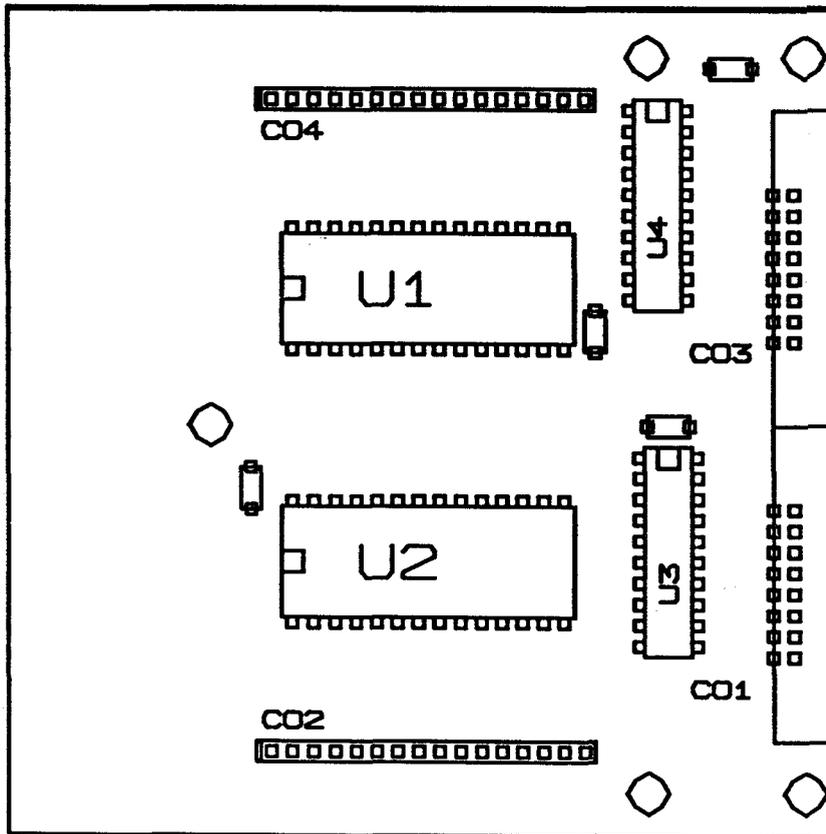
Connecteurs :

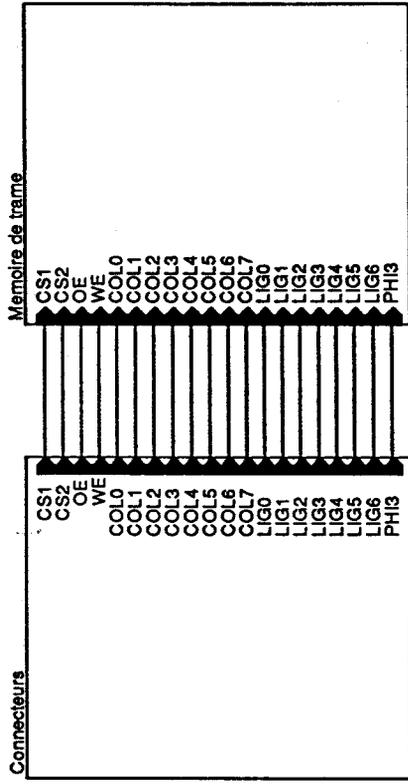
CO1,CO3	Cable plat 16 pts
CO2,CO4	Extensions 16 pts

Condensateurs de découplage :

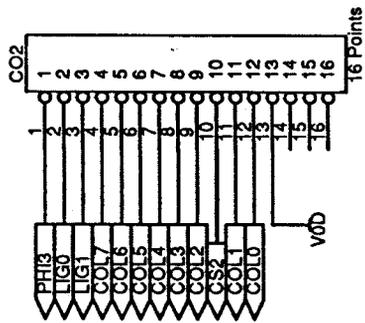
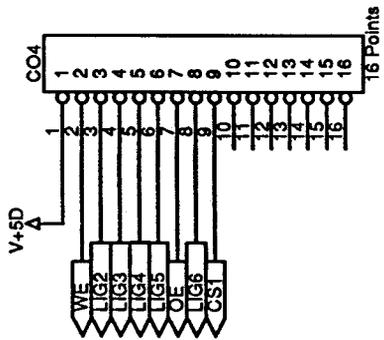
DEC3,....,DEC6	Découplage 100nF
----------------	------------------

3) IMPLANTATION DES COMPOSANTS

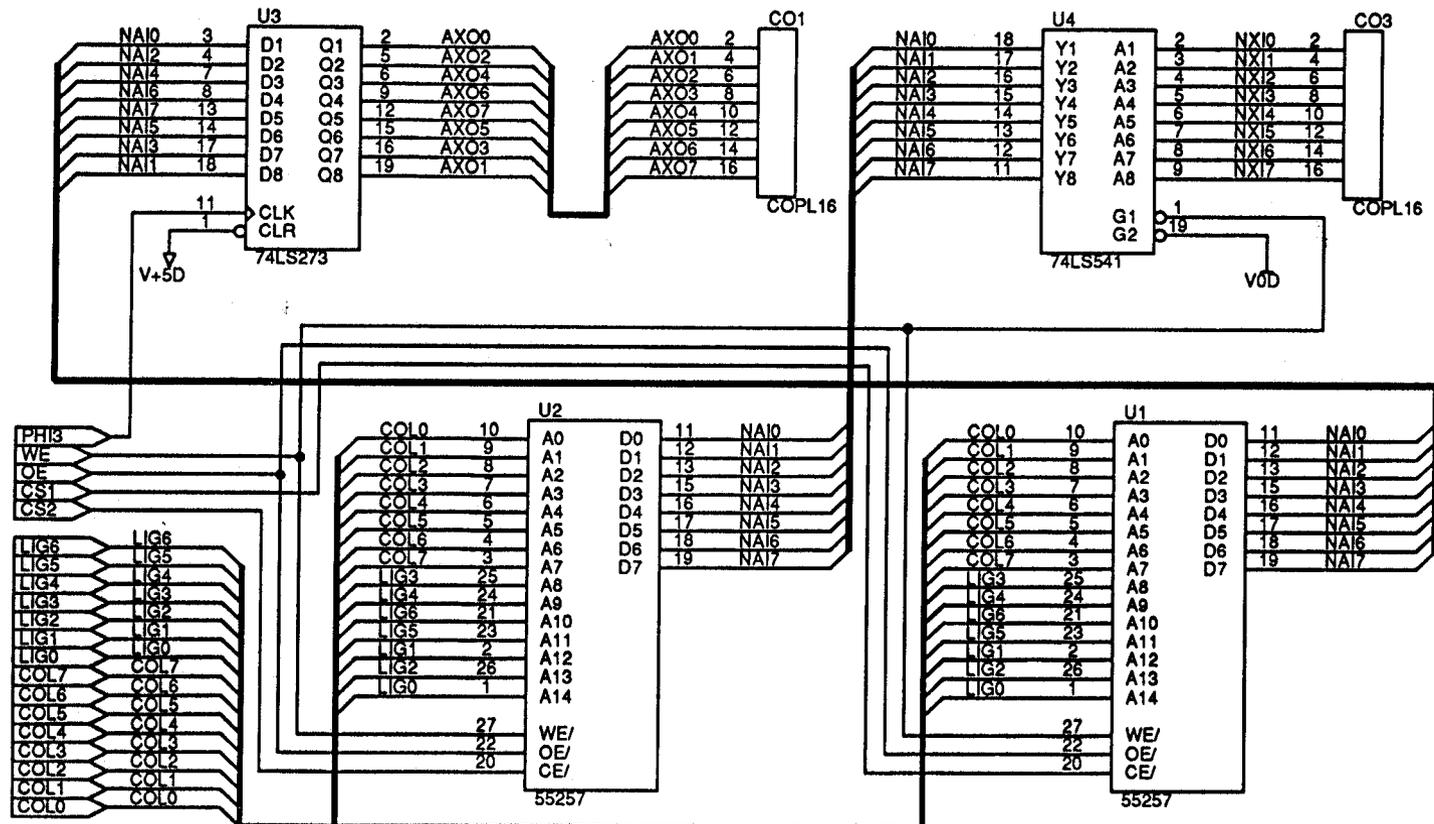




F. Cabestaing	
Module de retard d'image	
Title	
Synoptique	
Size	REV
A	1
Date:	11/08/1990
Sheet	1 of 3



F. Cabestain	
Module de retard d'image	
Connecteur d'extension	
Title	
Size	
Document Number	1
REV	1
Date:	Novembre 8, 1990
Sheet	2 of 3



F. Cabestaing		
Module de retard d'image		
Title		
Memoire de trame		
Size	Document Number	REV
A	1	1
Date:	Novembre 8, 1990	Sheet 3 of 3

ANNEXE 3

CARTE CALCUL

CARTE CALCUL

Elle permet de combiner deux images numériques (sous forme de flot vidéo) pixel à pixel.

USTL
FLANDRES ARTOIS



1) DESCRIPTION

L'opération de combinaison des deux pixels se réalise sous la forme :

Résultat = fonction (entrée1, entrée2).

les entrées et le résultat sont constituées d'une succession de mots de 8 bits. La fonction de combinaison des deux opérandes n'est pas forcément définie sous forme analytique, mais de façon ponctuelle.

2) FONCTIONNEMENT

Les deux bus vidéo numériques constituant les flots de données d'entrée sont récupérés en face avant de la carte puis envoyés sur les lignes d'adressage d'une EPROM rapide du type 27C512 (100 nS). Le résultat est pris sur les 8 bits de données de ce dernier circuit. Cet octet est ensuite mémorisé pour être présent en sortie de carte durant un cycle complet (200nS). La phase utilisée comme horloge pour cette mémorisation est sélectable par strap sur la carte.

Cette configuration particulière permet d'envisager tout type de fonction de transfert arithmétique 8x8 donnant 8 bits. Par exemple, seul le contenu de l'EPROM diffère entre une carte soustraction et une carte multiplication.

Les entrées n'étant pas latchées par un signal de phase, elles doivent avoir une période d'activité commune pour entraîner un résultat valide (prise en compte du temps d'accès de 100nS de l'EPROM). Si la période d'activité du résultat est la même que celle des opérandes (ex: PHI4), la carte introduit un retard de un pixel (200nS) sur le flot de sortie par rapport à ceux d'entrée.

3) LISTE DES COMPOSANTS

Circuits Intégrés :

U1,U7	74541
U2	74540
U3	27512
U4	74574

Connecteurs :

CO1,CO2	Cable plat 16 pts
CO3	Extension 10 points
CO5	Connecteur Europe G96

Condensateurs :

C2 47 μ F

Selfs :

L1 Self de choc

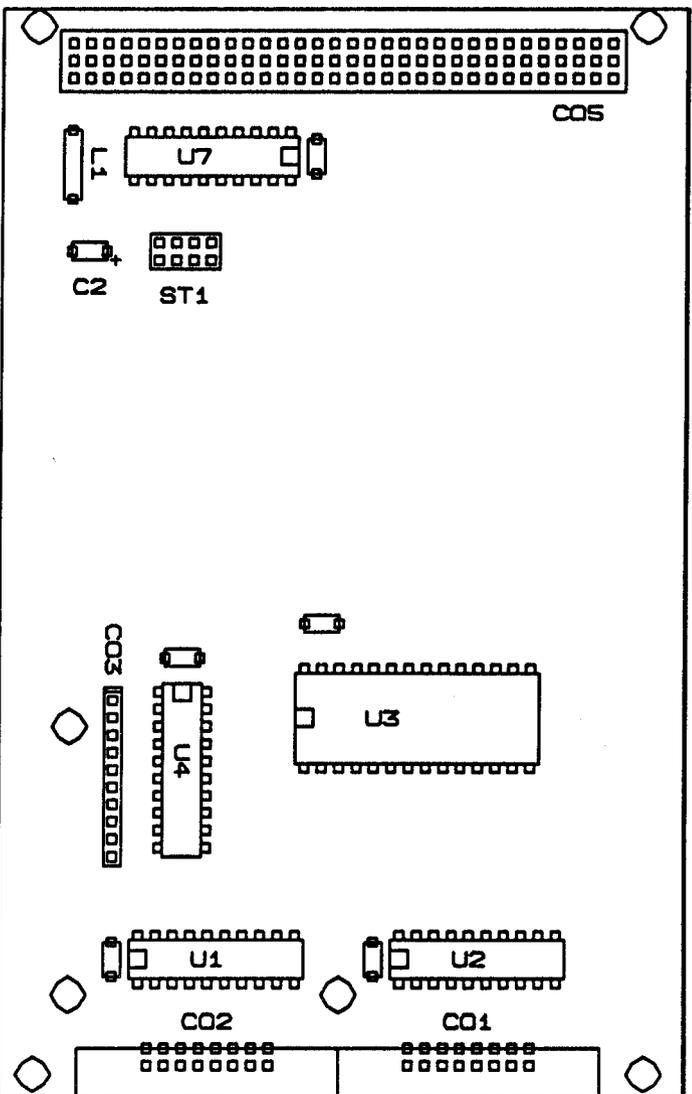
Condensateurs de découplage :

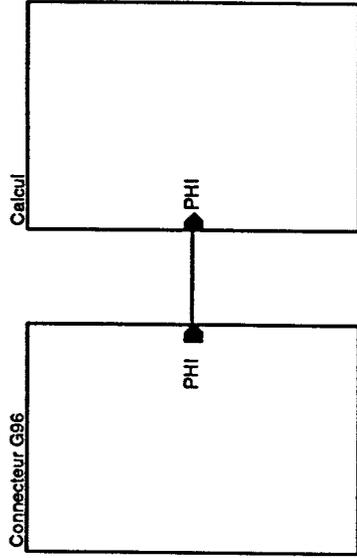
DEC1,DEC7 Découplage 100nF

Divers

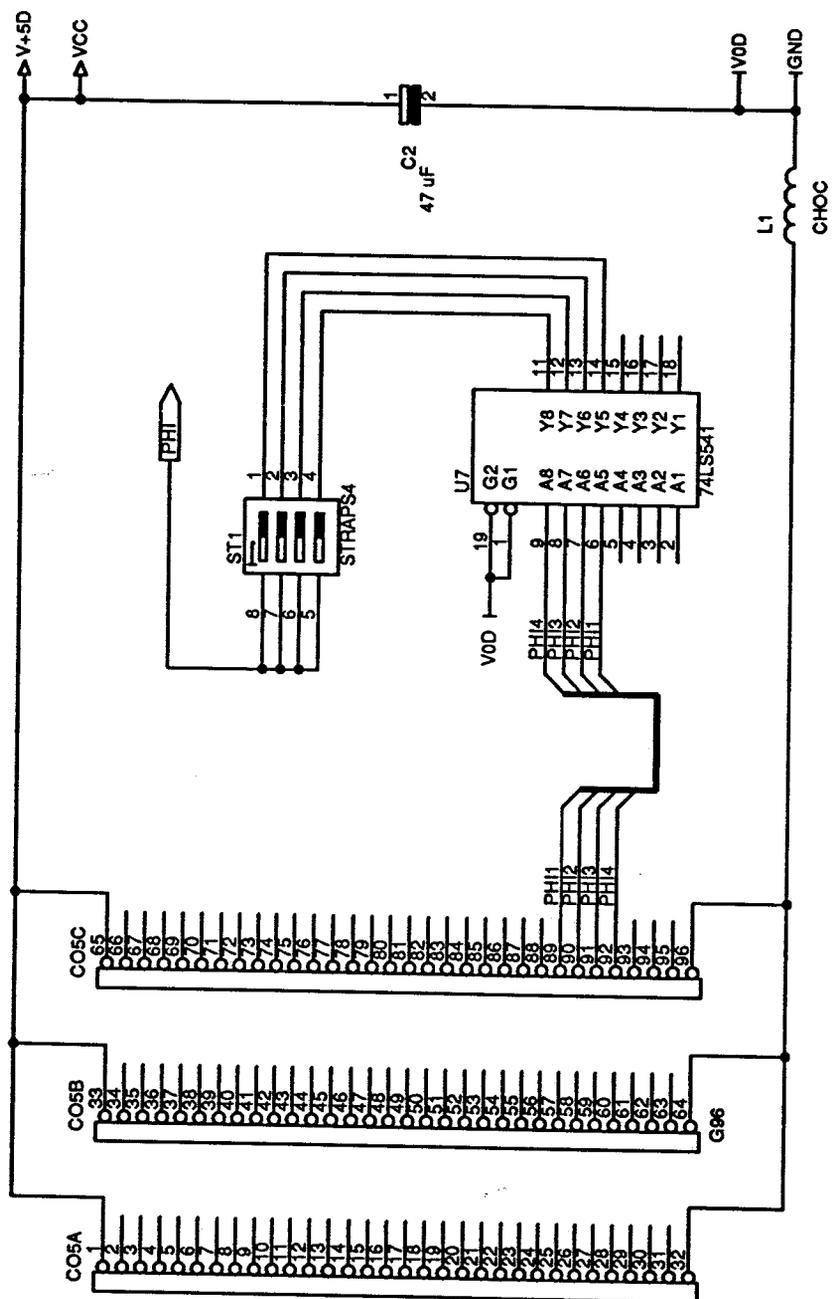
ST1 4 straps

4) IMPLANTATION DES COMPOSANTS

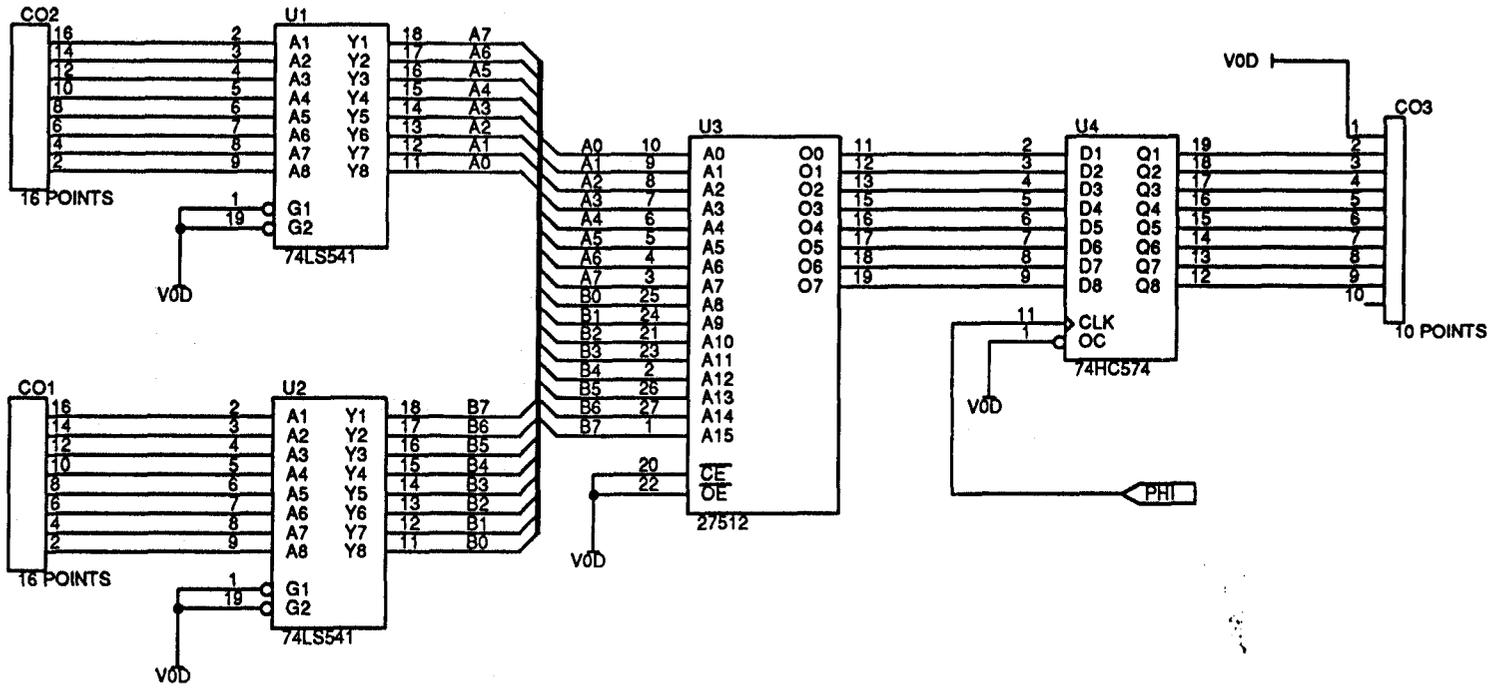




F. Cabestang	
Carte calcul	
Synoptique	
Title	
Size	Document Number
A	1
Date:	Novembre 13, 1990
	Sheet 1 of 3



F. Cabestaing Carte calcul	
Connecteur G96	
Title	
Size A	Document Number 1
Date: Novembre 13, 1990	Sheet 2
	of 3
	REV 1



F. Cabestaing		
Carte calcul		
Title		
Calcul		
Size	Document Number	REV
A	1	1
Date:	Novembre 13, 1990	Sheet 3 of 3

ANNEXE 4

CARTE GRADIENT ROBERTS CENTRE

CARTE DE VOISINAGE 3x3

Elle permet de disposer à un instant donné d'un pixel du flux vidéo ainsi que de ses 8 voisins (zone de 3x3 pixels). Cela permet ensuite de combiner ces valeurs pour une opération de voisinage (convolution, morphologie ...)

USTL
FLANDRES ARTOIS



1) DESCRIPTION

Les données constituant le flot vidéo numérique sont recues sur la carte, bufferisées et acheminées vers deux mémoires de ligne (256 pixels) organisées selon une structure de type FIFO.

Le principe de fonctionnement est semblable à celui des mémoires situées sur la carte de retard d'image. Le numéro du premier pixel devant être mémorisé sur une ligne est programmable par straps. Cette méthode de fonctionnement permet de disposer du pixel courant ainsi que des deux pixels situés à la même abscisse sur les deux lignes précédentes.

Ces trois pixels sont ensuite envoyés vers une série de bascules qui réalisent la fonction retard pixel. Une seule horloge de commande pilote toutes les bascules, on dispose alors de 9 pixels simultanément :

Une carte fille comportant une logique de combinaison de ces pixels est insérée dans les connecteurs présents sur la carte de voisinage 3x3. Le résultat de cette combinaison est ensuite récupéré sur la carte de voisinage puis bufferisé avant d'être replacé sur une sortie vidéo numérique.

2) FONCTIONNEMENT

Les signaux permettant la synchronisation ainsi que les horloges de cadencement sont récupérés sur le bus de fond de panier et amplifiés. Un circuit 74LS00 combine les différentes horloges afin de fournir les signaux de lecture (OE) et d'écriture (WE) pilotant les mémoires. Les adresses sont générées par un compteur 8 bits synchrone lui-même piloté par une PAL001 (gestion par straps du début de fenêtre).

Les lignes d'adresse mémoire sont incrémentées au début de la première phase (PHI1 active). Les données en lecture deviennent valides durant la phase PHI3. Elles sont latchées par les bascules de retard pixel en fin d'activité de PHI3. Le pixel à écrire à la même adresse doit être valide durant la phase PHI4.

Le pixel présent en entrée est alors en compte (écrit dans la première mémoire). Celui qui occupait sa place dans la RAM (situé sur la ligne vidéo précédente) avait été mémorisé dans un latch à la fin de la phase PHI3. Ce dernier est alors présenté en entrée de la deuxième mémoire durant la phase PHI4.

Le pixel présent dans la case mémoire de la deuxième RAM avait lui aussi été mémorisé dans un latch en fin de phase PHI3. En résumé, au début de la phase PHI4, le pixel courant ainsi que les deux pixels situés à la même abscisse sur les deux lignes précédentes sont disponibles.

Tous trois sont envoyés sur une suite de circuits 74HCT574 (octuple bascules D actives sur un front descendant d'horloge) afin de subir un retard de 1 pixel. L'horloge de tous ces circuits est identique : PHI3. Toutes les sorties de bascules sont alors envoyées vers des connecteurs inter-cartes pour ensuite être récupérés par une carte fille réalisant leur combinaison.

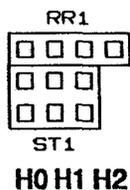
Le résultat de la combinaison est récupéré et latché par un circuit 74HCT574 sur le front descendant de PHI2. Ce pixel est alors présenté en sortie de la carte de voisinage

Si un pixel doit subir une inversion (coefficient négatif lors de la combinaison), le circuit 74HCT574 fournissant ce pixel, ainsi que le circuit 74HCT574 suivant, peuvent être remplacés par des 74HCT564.

2.1) LISTE DES COMPOSANTS

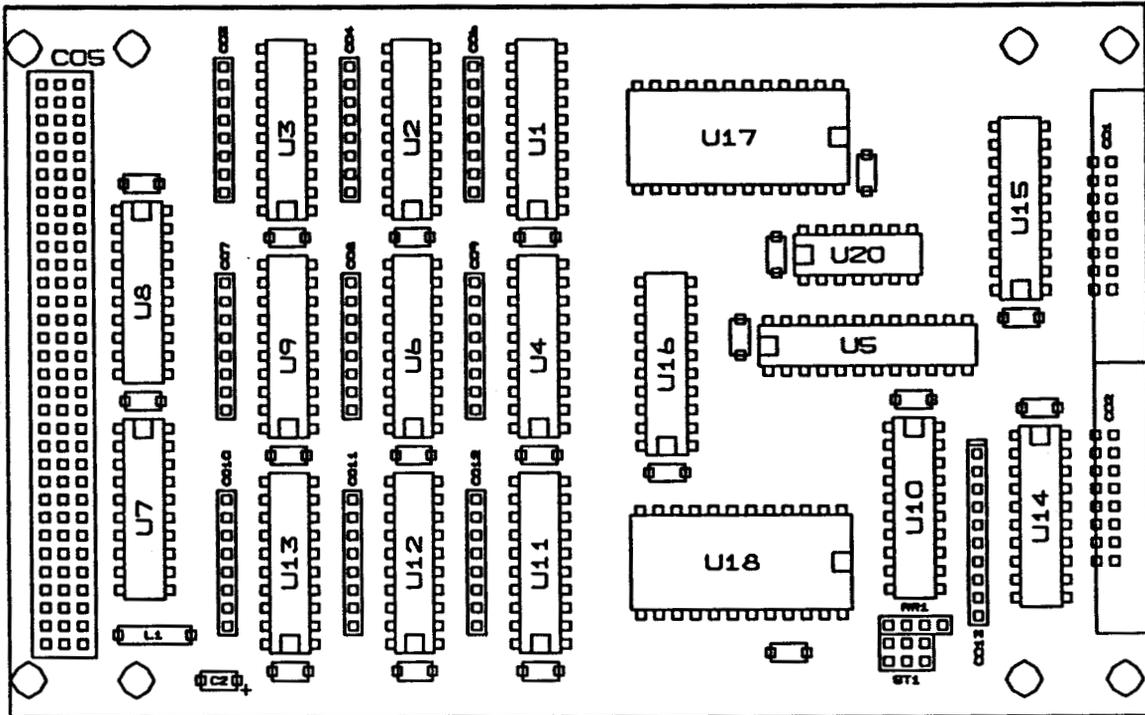
Circuits Intégrés :	CO13	Extension 10 points
U1,U7,U8,U15,U16 74LS541	Condensateurs :	
U17,U18 2015	C2	47µF
U2,U3,U4,U6,U9, 74HCT574	Selfs :	
U11,U12,U13,U14	L1	Self de Choc
U5 74AS867	Divers :	
U10 PAL001	ST1	3 Straps
U20 74LS00	RR1	SIL 4.7 k
Connecteurs :		
CO1,CO2 Cable plat 16 Pts		
CO5 Connecteur Europe G96		
CO3,CO4,CO6, Extension 8 points		
CO7,CO8,CO9,		
CO10,CO11,CO12		

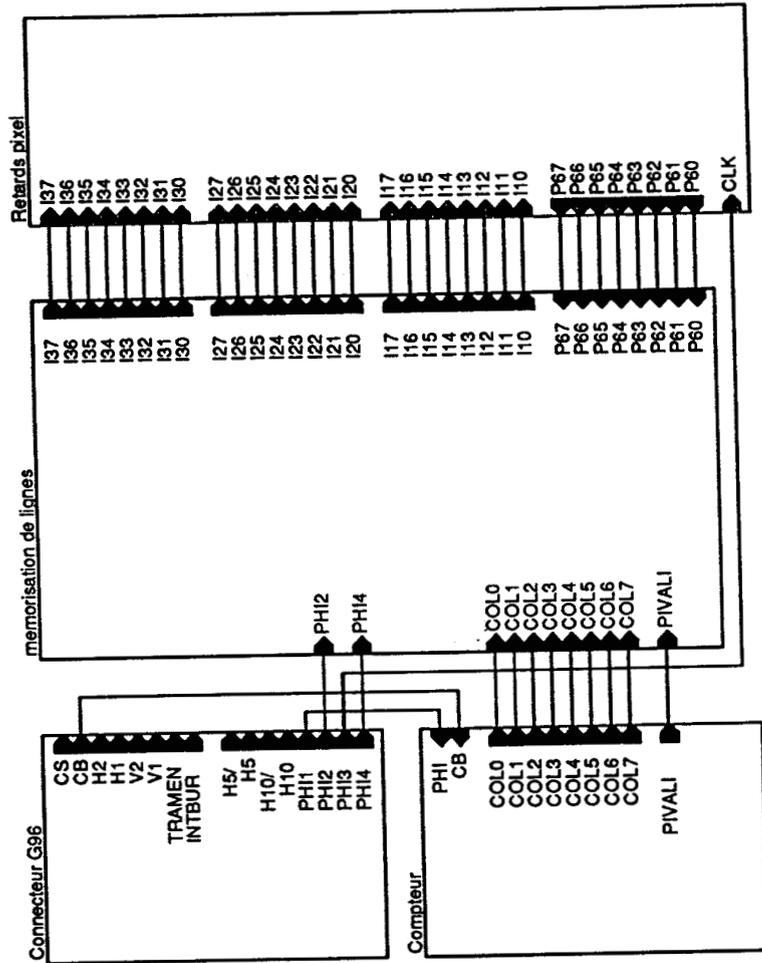
2.2) CHOIX DU PIXEL DE DEBUT DE LIGNE



Des straps disposés sur la carte permettent de sélectionner la position du premier pixel devant être mémorisé sur une ligne vidéo numérique. Le nombre de pixels à inhiber en début de ligne est programmé par sa représentation binaire. Un strap placé équivaut à un bit à 0.

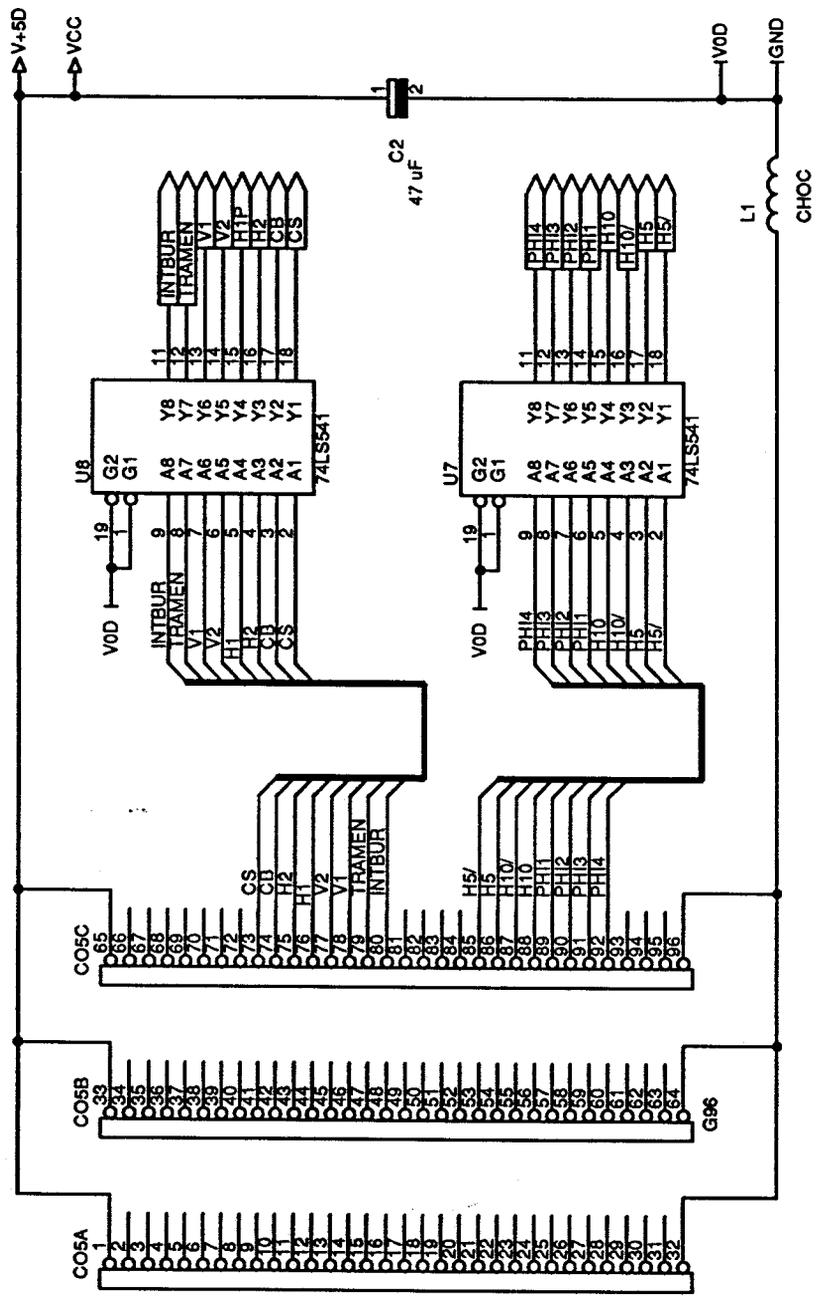
3) IMPLANTATION DES COMPOSANTS



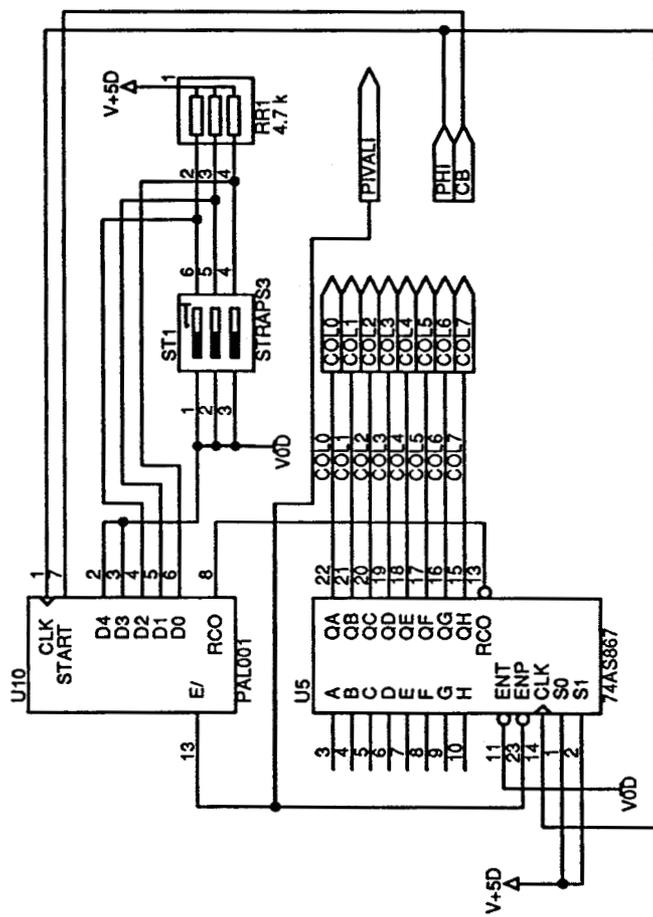


F. Cabestaing
 Carte de voisinage 3x3

Title		Synoptique	
Size	Document Number	1	1
A			
Date:	September 11, 1990	Sheet	1 of 5
REV			1



F. Cabestaing	
Carte de voisinage 3x3	
Title	
Interface bus G96	
Size	Document Number
A	1
Date:	November 7, 1990
Sheet	2 of 5
REV	1



F. Cabestaling
Carte de voisinage 3x3

Title

Compteur colonne pixel

Size A

Document Number

1

REV

1

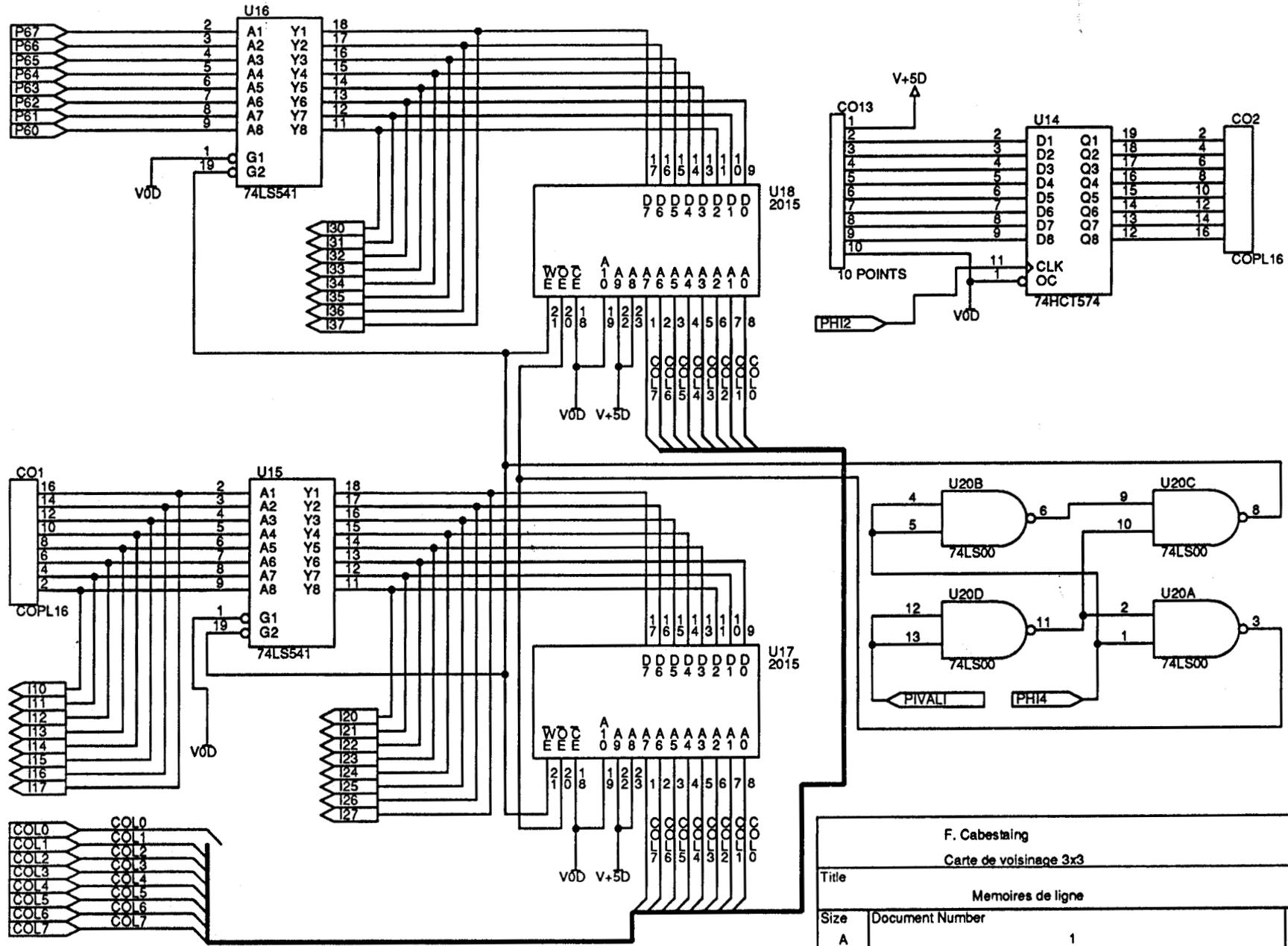
Sheet

3

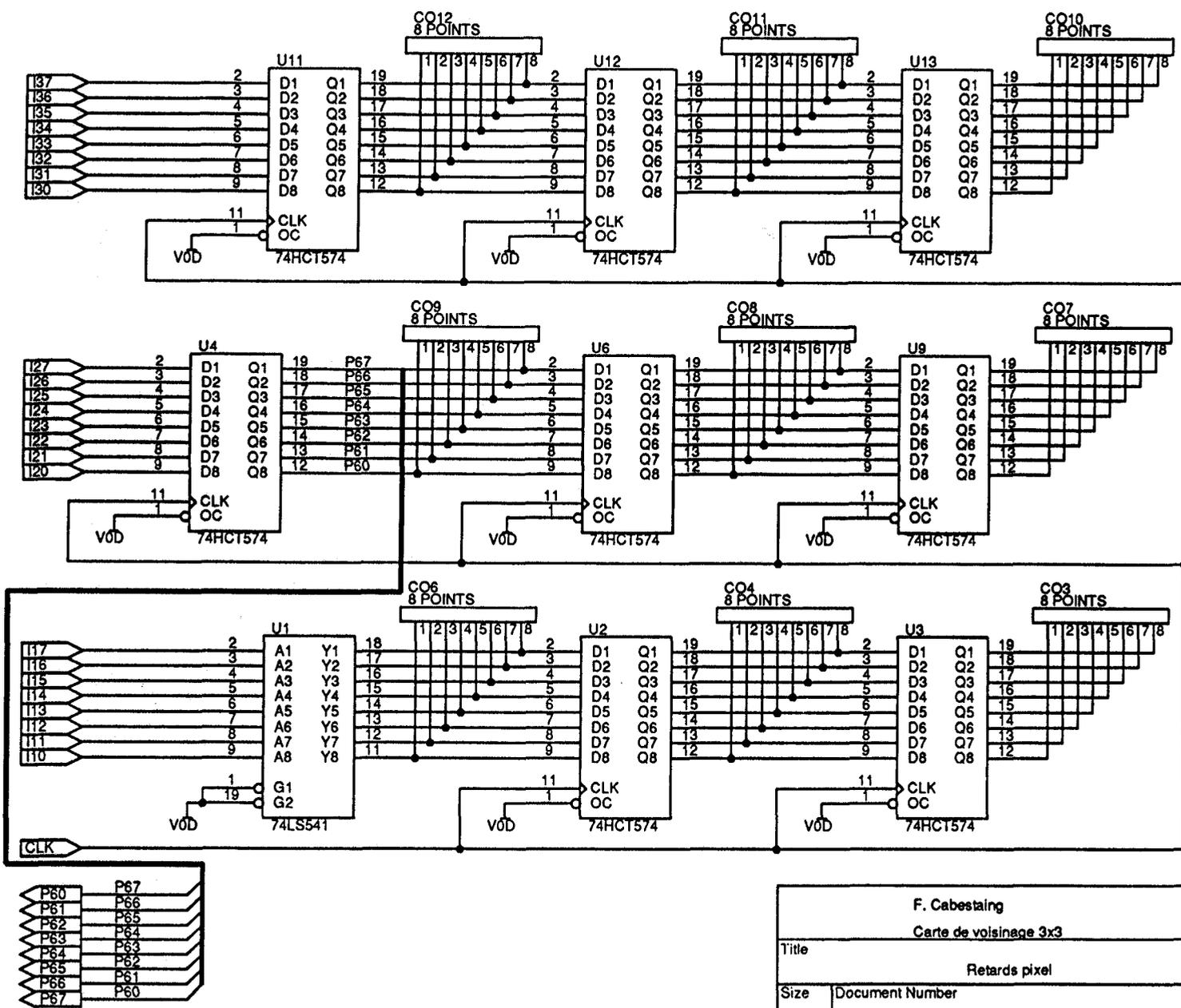
Date: Novembre 7, 1990

of

5



F. Cabestaing		
Carte de voisinage 3x3		
Title		
Memoires de ligne		
Size	Document Number	REV
A	1	1
Date:	September 11, 1990	Sheet 4 of 5



F. Cabestaing		
Carte de voisinage 3x3		
Title		
Retards pixel		
Size	Document Number	REV
A	1	1
Date:	September 11, 1990	Sheet 5 of 5

MODULE GRADIENT

Il réalise la combinaison arithmétique de quatre des neuf pixels issus de la carte de voisinage 3x3 afin de fournir une approximation du module du gradient de l'image au point central de la fenêtre locale.

USTL
FLANDRES ARTOIS



1) DESCRIPTION

Les quatre pixels situés aux coins de la fenêtre locale 3x3 sont récupérés grâce aux connecteurs inter-cartes de la carte de voisinage. ils sont alors combinés selon la formule suivante :

$$\text{Résultat} = | P(-1,-1) - P(1,1) | / 2 + | P(-1,1) - P(1,-1) | / 2$$

Le résultat correspond alors à la réponse à un estimateur de gradient local de type Roberts 3x3 selon les masques suivants :

$$G_{xy} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix}, G_{yx} = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, G = \frac{|G_{xy}| + |G_{yx}|}{2}$$

2) FONCTIONNEMENT

Les deux pixels affectés d'un coefficient négatif doivent être issus d'une combinaison de buffers de la carte de voisinage assurant leur inversion par rapport au pixel fourni en entrée de carte. Cela permet de s'affranchir d'une série d'inverseurs au niveau du module gradient.

Les deux couples de pixels (P(-1,-1), P(1,1)) et (P(-1,1), P(1,-1)) pourvus de leurs signes respectifs sont additionnés par l'intermédiaire de circuits 74LS283 (additionneur 4 bits). La retenue d'entrée de ces groupes d'additionneurs est forcée à 1 afin de compléter le changement de signe du pixel affecté du coefficient -1; En effet :

$$-x = (\text{complément à 2 de } x) = (\text{complément à 1 de } x) + 1$$

Les deux résultats intermédiaires sont des nombres signés sur 9 bits. Il sont traités par deux circuits PAL002, dont la fonction est de générer un nombre non signé sur 8 bits correspondant à la valeur absolue d'un nombre signé de 9 bits.

Les deux valeurs résultantes (correspondant aux composantes directionnelles du gradient local) sont alors additionnées par deux autres circuits 74LS283. La sortie de ces derniers devant être divisée par deux, le résultat est prélevé en négligeant le bit de poids faible, et en prenant en compte la retenue de sortie (décalage d'une position).

Ce résultat final est envoyé sur un connecteur inter-cartes vers la carte de voisinage 3x3.

3) LISTE DES COMPOSANTS

Circuits Intégrés :

U1,U2,U3, 74LS283
 U4,U7,U8
 U5,U6 PAL002

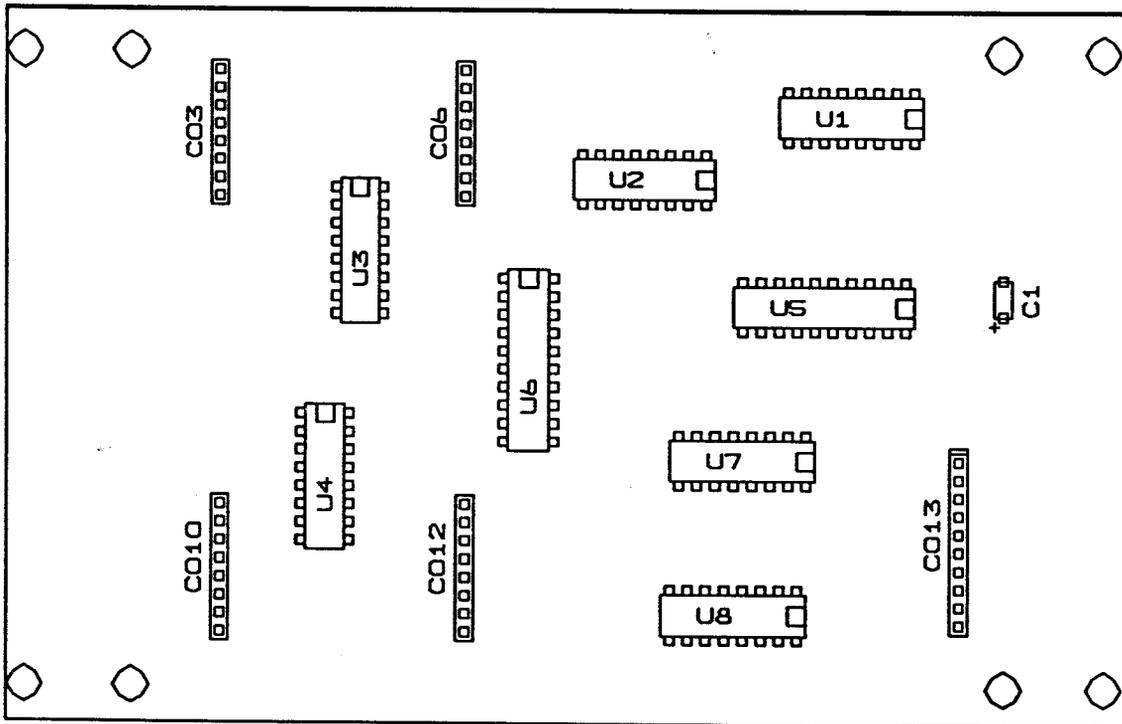
CO10,CO3, Extension 8 Points
 CO6,CO12
 CO13 Extension 10 Points

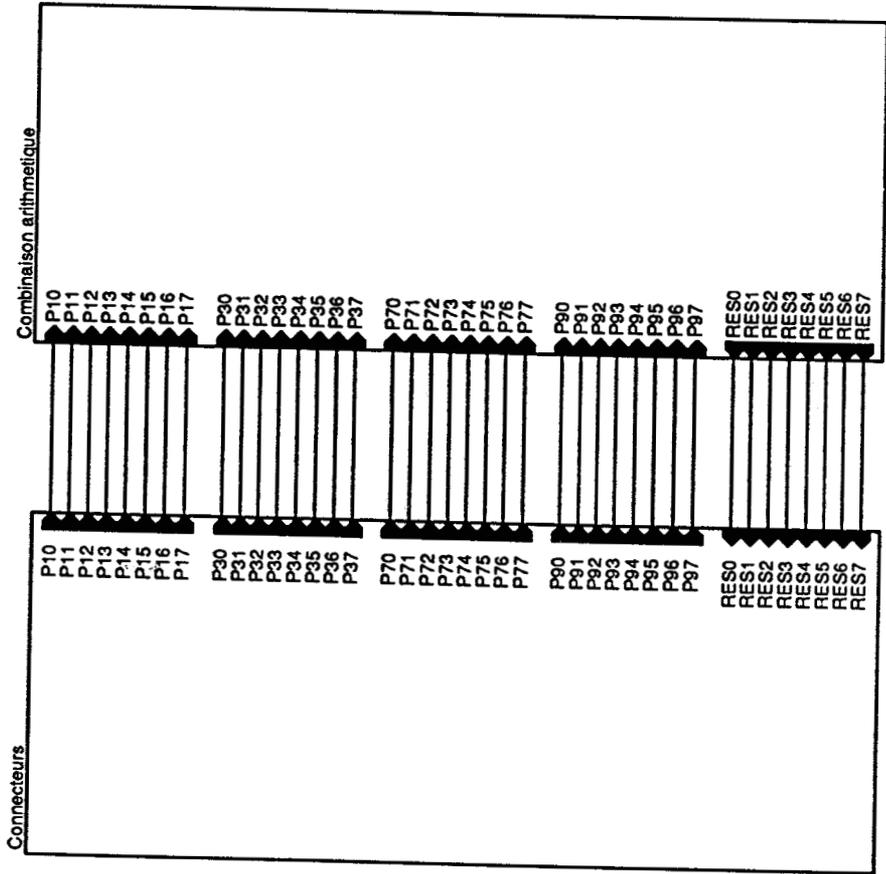
Condensateurs :

Connecteurs :

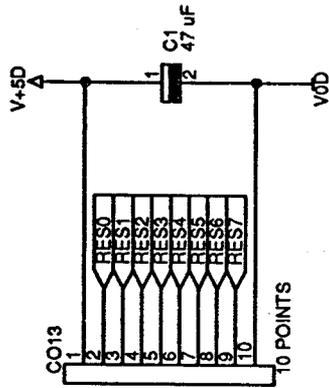
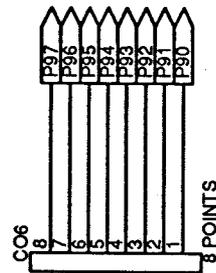
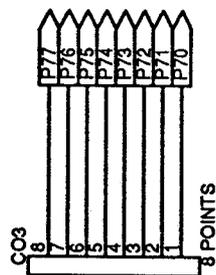
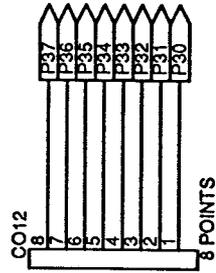
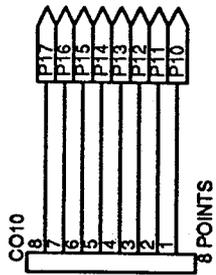
C1 47 μ F

4) IMPLANTATION DES COMPOSANTS





F. Cabestaing	
Module gradient	
Title	
Synoptique	
Size	Document Number
A	1
Date:	REV
Novembre 8, 1990	1
Sheet	of
1	3



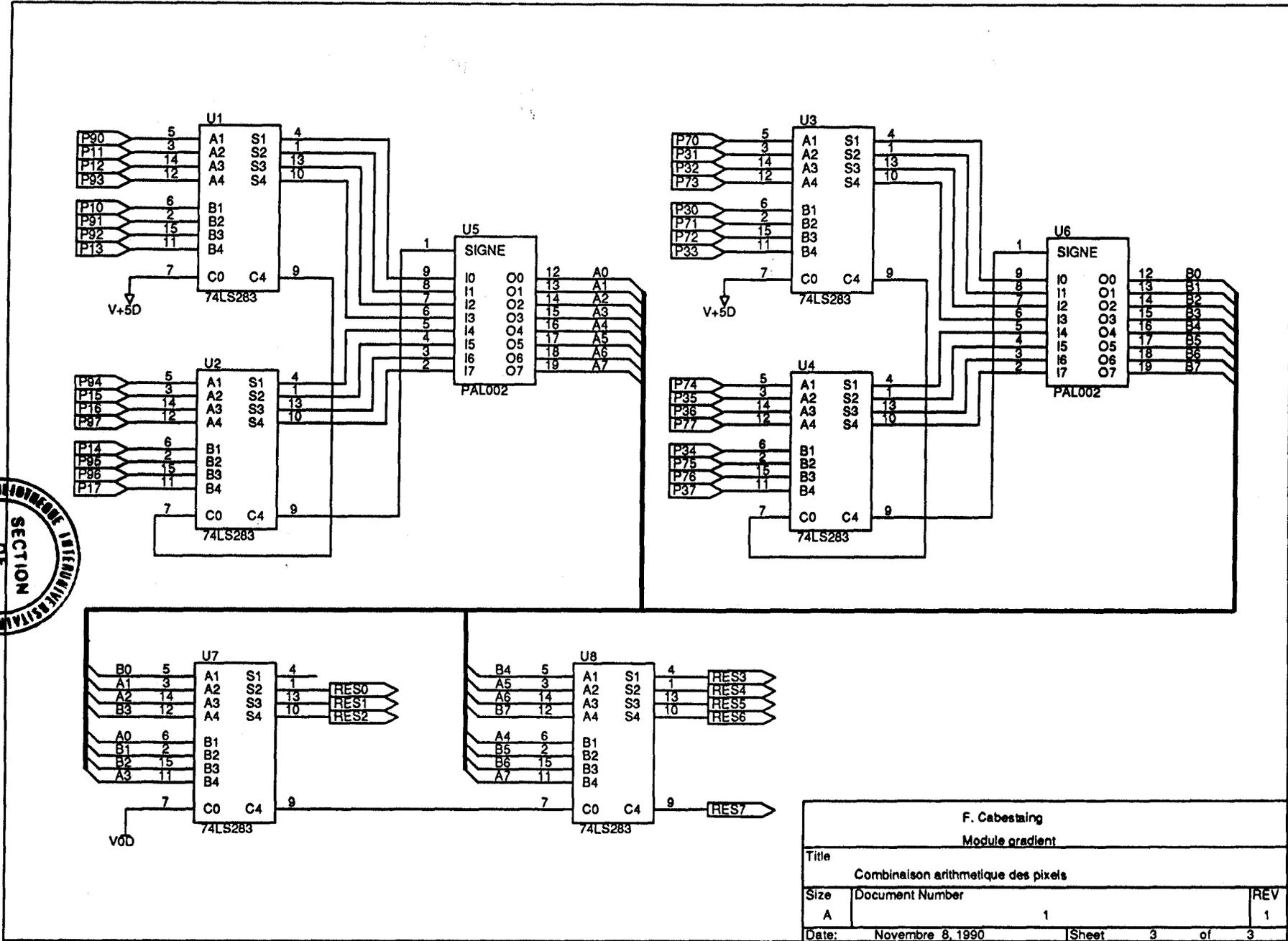
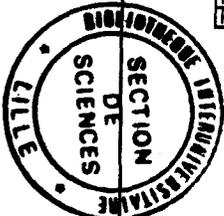
F. Cabestaling
Module.gradient

Title

Connecteurs inter-cartes
Size A Document Number 1

REV
1

Date: Novembre 8, 1990 Sheet 2 of 3



F. Cabestaing Module gradient			
Title Combinaison arithmétique des pixels			
Size A	Document Number 1		REV 1
Date:	Novembre 8, 1990	Sheet 3	of 3

PPN 036110337

Dans ce travail, nous présentons la conception et la réalisation d'un processeur spécifique dédié à l'analyse de scènes dynamiques : le STREAM, Système Temps Réel d'Extraction et d'Analyse du Mouvement. Ce processeur permet la détection en temps-réel des contours de tous les objets se déplaçant dans une scène observée par une caméra fixe.

Nous avons évalué l'efficacité de plusieurs algorithmes permettant d'extraire d'une séquence d'images les contours des objets en mouvement, en adaptant au cas des contours mobiles divers critères de comparaison utilisés en traitement d'images statiques. Cette étude nous a permis de sélectionner la méthode de détection la plus adaptée au traitement de scènes réelles.

L'algorithme retenu a été partitionné en opérateurs simples qui ont été implantés sur une série de cartes électroniques communiquant par l'intermédiaire de liens vidéo numériques. L'efficacité du processeur STREAM a été vérifiée dans le cadre de deux expérimentations liées à l'amélioration de la sécurité dans les systèmes de transport en commun de voyageurs.