

---

---

## *Etat de l'art*

---

---

### **1.1 : Introduction**

Nous présentons dans ce premier chapitre, quelques éléments d'état de l'art couvrant plus au moins, la procédure de détection. Ainsi, nous essayons de toucher aux différents constituants de cette dernière, à savoir la méthode utilisée, le support de calcul considéré et autres.

### **1.2 : Détection**

La détection des régions de changement dans des images d'une même scène ; prises à des instants différents, a un intérêt essentiel dans de nombreuses applications appartenant à des disciplines diverses [23].

Le spectre de son application est large : on peut citer la vidéo-surveillance, l'interaction homme-machine, le guidage des robots par la vision et le diagnostique médical.

La diversité des applications a permis l'élaboration d'un grand nombre d'algorithmes. Ces derniers varient d'une part, selon le support de calcul utilisé : intensité ou couleur, pixel ou région .... D'autre part, ils peuvent être d'origine déterministe ou stochastique. Aussi, on

peut choisir entre différentes méthodes de comparaison : soit entre trames, soit par rapport à une image référence, ou même une double comparaison. Autrement dit, l'environnement à traiter, le but à atteindre et la performance espérée, dictent un choix approprié d'algorithmes et de techniques.

Le problème en question, est comme suit : On dispose d'une séquence d'images acquise avec une caméra stationnaire, et on souhaite identifier l'ensemble des pixels appartenant à l'objet mobile. Ces pixels là font, toujours partie de l'ensemble total de ceux ayant subis un changement entre l'image actuelle et celles qui la précèdent. Ces derniers constituent ce qu'on appelle « *Masque* », ou '*Change Mask*'.

Or ce masque peut être le résultat d'une combinaison de certains facteurs. Citons, à titre d'exemple :

- Mouvement de l'objet par rapport à l'arrière plan '*Background*'.
- Changement de la forme de l'objet.
- Changement d'apparence de l'objet.

Un point essentiel est que ce masque doit être significatif. C'est-à-dire, il ne doit pas contenir des formes de changement nuisantes, telles que :

- ✓ Le mouvement de la caméra.
- ✓ Le bruit du capteur et d'acquisition.
- ✓ Variation d'illumination.

Notons tout de même que l'adjectif « *signifiant* », varie selon l'application et le problème traité. Aussi, rappelons que dans cet état de l'art, on s'intéresse surtout à l'application traitée, mais seulement à la méthode sélectionnée au niveau de la « *Détection* ». Malgré que l'application délimite plus au moins, la méthode de détection choisie, ainsi que la performance attendue.

La détection donc, est une étape préliminaire dans les applications liées à la vision. Dans la littérature, la notion « *Détection* » est parfois exprimée par « *Soustraction de l'arrière plan* » ou '*Background subtraction*'. Cela revient à effectuer une comparaison de l'image d'entrée à une autre référence appelée '*Background*'. Ainsi, une partie d'algorithmes de

détection exploitent cette notion. D'autres se basent sur une différence temporelle. Les premiers sont bénéfiques, lorsque le *Background* est connue d'avance, et ne change pas significativement dans le temps. Si des changements sont prévus, une étape d'adaptation et maintenance est nécessaire. Elle peut être de plus en plus sophistiquée pour répondre aux différents changements (conditions d'illumination, mouvements des branches d'une arbre, ...). Tandis qu'une différence temporelle peut s'adapter rapidement aux changements d'illumination par exemple.

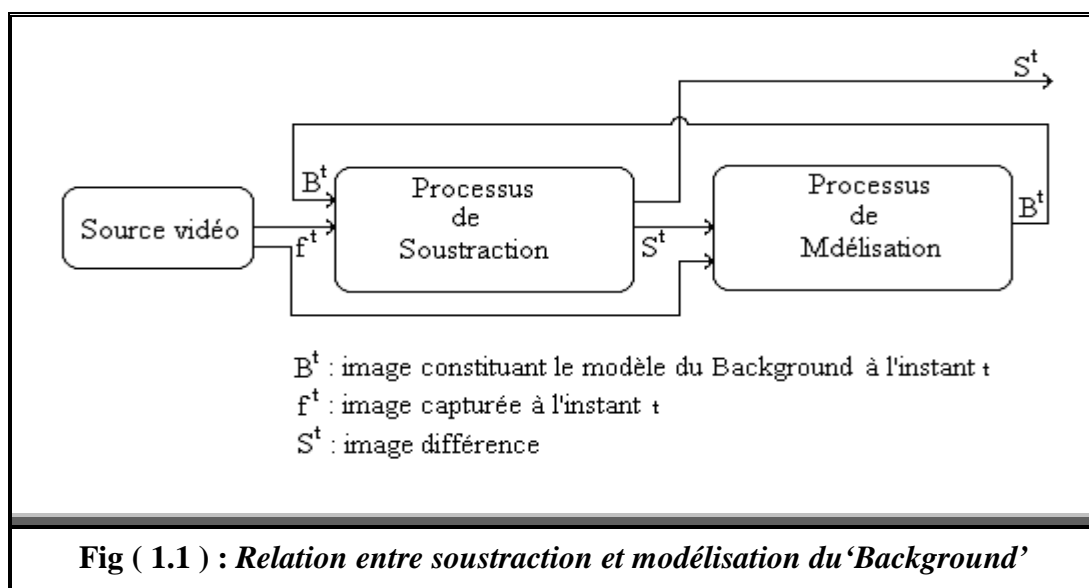
En résumé, trois types de différence peuvent être utilisés :

- Une première catégorie ; celle qui détecte les changements par rapport à un '*Background*', en comparant l'image actuelle à une image référence vide (sans objet mobile), et acquise précédemment. Généralement, ces systèmes demandent une adaptation [3, 14, 16] [21, 46] (surtout dans des séquences naturelles).

En effet, les deux expressions : '*Background subtraction*' et '*Background modeling*', sont interchangeables. Or réellement, ils sont distincts.

Modéliser le *Background* : est le processus de création d'un modèle, dans lequel, on intègre les changements dynamiques qu'il subi, en but de sa maintenance dans le temps. Alors que la soustraction du '*Background*' définit le processus de comparaison de l'image actuelle au modèle du '*Background*', déjà construit.

Les deux expressions sont généralement utilisées en conjoint [4], comme le montre la figure **Fig (1.1)** suivante:



La forme la plus simple de cette image de référence, est la moyenne temporelle [16]. Mais, celle-la a besoin d'une étape d'apprentissage effectuée sur une scène vide. En addition, elle ne répond pas aux changements d'illumination. Une adaptation continue du modèle est conseillée.

Généralement, la modélisation du '*Background*' opère au niveau pixel. Elle utilise parfois des distributions [1,16]. L'intensité du pixel est alors, modélisée par un mélange de distributions Gaussiennes [1]. Leurs paramètres et leurs nombres sont mis à jours, à l'aide d'équations récursives. Tandis que Rowden et ses collègues dans [16], présentent un algorithme du même genre, ayant en plus la possibilité de séparer entre l'objet mobile et son ombre. Ils exploitent l'information chrominance.

Donc, un algorithme appartenant à ce groupe doit être capable de :

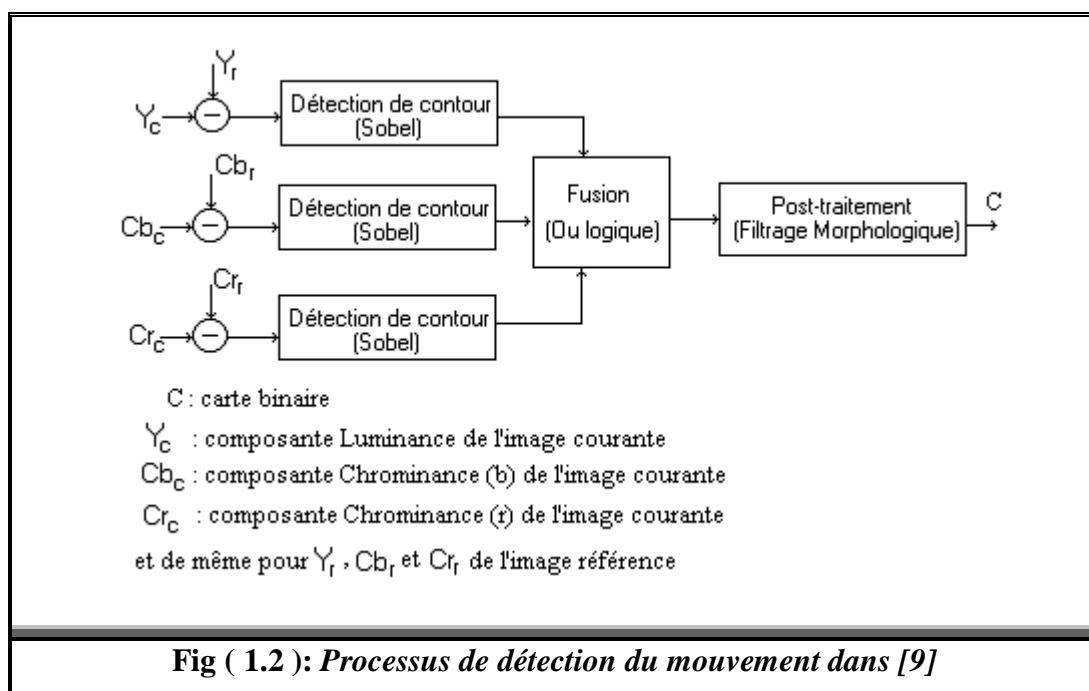
- ✓ Déterminer le nombre de distributions.
  - ✓ Estimer les paramètres de chacune.
  - ✓ Mettre à jours le '*Background*'.
- Une deuxième catégorie : celle de la différence temporelle, nécessitant au moins deux images [14, 20, 24], ou parfois trois [7], et donc nommée par « différence double ». Remarquons dans ce cas, qu'un changement temporel détecté ne détermine pas exactement l'objet mobile. La présence de ce dernier cause l'apparition de trois régions :
- Région du '*Background*' découvert.
  - Région du '*Background*' couvert.
  - Région de glissement de l'objet sur lui-même. Celle-ci est difficile à distinguer si l'objet est de nature uniforme. Cela exige une étape supplémentaire nommée localisation de cet objet mobile sur l'image plane. Une des solutions disponibles est l'approche markovienne ; avec les champs aléatoires de Markov '*MRF*<sup>1</sup>'. Cette méthode est décrite dans [25]. Des distributions gaussiennes sont utilisées au niveau de la détection, et de la localisation. A chaque niveau, une fonction de coût est construite, puis minimisée. La classification est faite par le critère *MAP* « *Maximum A Posteriori* ». Il est de même dans [13], sauf qu'au premier stade, les gaussiennes sont remplacées par des distributions de Laplace.

---

<sup>1</sup> : Markov Random Feilds

Les méthodes se différencient non seulement dans la comparaison, mais par le support de calcul. L'un considère le pixel [11, 8]. L'autre opère sur des régions [47]. Un dernier s'intéresse aux contours [9]. Duric et ses collègues [6], intègrent la notion de « *contour* » à côté de la « *couleur* » pour modéliser le '*Background*'. Le modèle global est divisé en sous-modèles appropriés aux trois canaux couleurs. Le changement dans l'un des canaux représente un indice de mouvement. Le calcul du contour est élaboré via l'opérateur de *Sobel*. Cette stratégie est appliquée pour détecter des individus dans des scènes de l'extérieur.

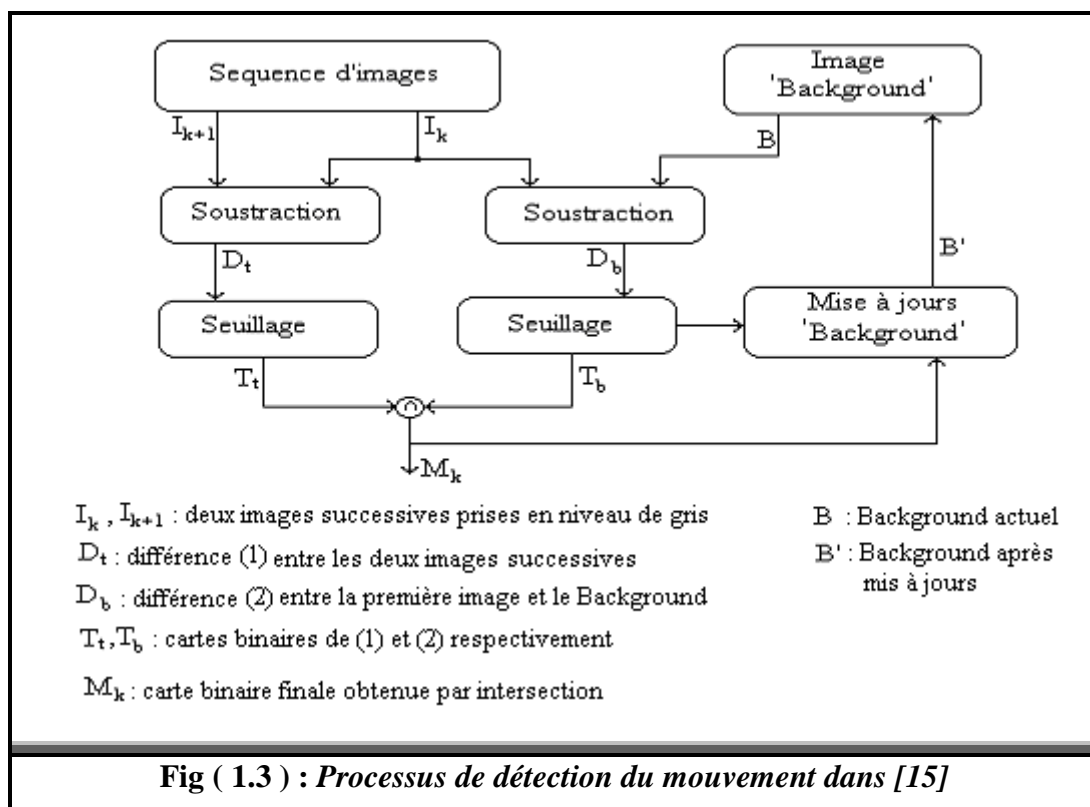
Il faut signaler que la couleur est fréquemment utilisée pour distinguer entre l'objet et son ombre. La détection fautive (comme étant objet) de ce dernier, dans certaines applications, tel que la reconnaissance, amène à un traitement supplémentaire. Une technique connue est la décomposition de la couleur en deux composantes : luminance et chrominance. Une autre association couleur-contour se trouve dans [9].



Les premiers détecteurs à base de contours, effectuent un calcul sur l'image référence et un autre sur l'image considérée. Puis dressent une comparaison entre les deux cartes contour par opérateurs logiques (un exemple : le ou exclusif). Une contrainte majeure est sa sensibilité envers les changements d'illumination. Une amélioration est de faire une soustraction en premier, suivie de la détection de contours. Or, les contours à base d'intensité sont de nature ouverte ; d'où un '*post-processing*', en vue de leur fermeture. Ainsi dans [9], les

contours sont définis par l'opérateur de *Sobel* au niveau de chaque canal (Y, Cb, Cr), suivi d'une fusion et d'un post-traitement. Cette finalisation est constituée d'un filtrage morphologique, qui a pour objectif de remplir les contours, donnant ainsi la carte binaire de la détection. Cette idée est illustrée par la figure précédente **Fig ( 1.2 )**.

Une troisième catégorie, parfois utilisée, est celle qualifiée par hybride. Ici, on combine entre les deux types de différence. L'idée est exploitée dans [15], dans l'objectif de déterminer un seuil automatique. Cela est montrée par la figure suivante **Fig ( 1.3 )**.



L'étape de classification des pixels, comme mobiles ou 'Background' s'effectue, elle aussi selon différentes techniques. La plus ancienne ; celle d'origine intuitif est le seuillage ; parfois appelé « *binarisation* ». La valeur de ce seuil est critique. Elle dépend évidemment des bruits, des changements d'illumination, aussi bien de la différence naturelle à détecter (objet mobile). Des changements non importants seront détectés si la barre est trop basse. Ceux primordiales seront négligés si le seuil est trop élevé. Donc, sa sélection appropriée pour chaque pixel est indispensable.

Dans les méthodes stochastiques basées sur les champs de Markov ; dont lesquelles l'information voisinage est introduite, ou celles basées sur les champs de Gibbs, on cherche à minimiser une fonction de coût, par l'intermédiaire d'une fonction d'énergie maximisée [4,13, 24]. Généralement, on choisit entre les algorithmes itératifs de relaxation déterministe, tel que l'algorithme *HCF* pour '*Highest Confidence First*', et celui nommé *ICM* pour '*Iterated Conditional Mode*', ou ceux de relaxation stochastiques tel que le « *recuit simulé* ». Ce dernier est coûteux en temps de calcul.

Un point à ne pas oublier est que, avant l'étape de classification, les modèles construits nécessitent toujours une estimation de paramètres ; paramètres des gaussiennes par exemple ou de régularisation pour les *MRFs*. Le critère de maximum de vraisemblance *MV*, est utilisé dans [13]. Tandis que Bowden dans [16] utilise en ligne, l'algorithme *EM*, à dire '*expectation maximisation*'.

Notons qu'un algorithme *EM* en ligne représente une amélioration d'un autre hors-ligne ; dans ce dernier, on a besoin d'acquiescer d'avance, une séquence dite d'apprentissage ; '*training*'. Les méthodes récentes utilisent parfois l'algorithme *ICE* ; '*Iterative Conditional Estimation*'. Il est basé sur une moyenne conditionnelle et à nature statistique.

Un dernier modèle, récemment introduit (2000), modélisant les régions '*Background*' et '*Foreground*' utilise les Modèles de Markov cachés *MMCs* de premier ordre et à une seule dimension. Ces modèles sont basés sur les Chaînes de Markov Cachées.

Son principe est d'estimer les états cachés à travers les valeurs observées, dans un cadre probabiliste [2, 11, 26]. Cette méthode est essentiellement composée de deux étapes. Une pour l'apprentissage dans laquelle, l'algorithme *EM* est généralement utilisé. Une autre pour la classification, où on maximise la probabilité choisie, en vue d'un choix d'états optimal, selon un critère choisi. Un critère de « Maximum A Posteriori *MAP* », ou *MPM*, à dire « Mode de la Marginale à Posteriori » peut être utilisé. Un algorithme associé aux *MMCs* est celui de '*Viterbi*', utilisé dans [13].

Un modèle *MMC* est capable de s'adapter aux changements globaux du '*Background*' [4], tel que les changements d'illumination. Il est aussi extensible. En ajoutant un troisième état, il devient capable de détecter les ombres [11, 26].

En fin, les *MMCs* offrent au modèle, la possibilité de changer de structure ; cas décrit dans [11], pour s'adapter à la nature non stationnaire des phénomènes réels. Des algorithmes

dites de fusion d'états, existent et répondent aux problèmes liés à la modification de cette topologie.

Signalons à la fin, que tous les algorithmes précédents tentent à s'exécuter en temps réel, dans le but de répondre aux exigences naturelles de traitement.

### 1.3 : Problématique

Une séquence d'images est un support riche d'informations. Son traitement, dans le but d'extraire l'information utile, nécessite énormément de calculs complexes.

Elle peut être traitée en tant que fonctions déterministes. Comme, elle peut être considérée comme une réalisation de processus stochastiques. Les outils mathématiques utilisés reposent sur la théorie des systèmes linéaires, des mathématiques discrètes et des processus stochastiques.

L'extraction d'objet mobile à partir de scènes dynamiques est l'étape préliminaire nécessaire dans de nombreuses applications, telles que la compression, la surveillance, ..., etc. Un taux de compression élevé est le résultat d'une bonne détection. Différentes approches ont été proposées au niveau de la détection. Citons à titre d'exemple, le seuillage, la modélisation par différentes densités, l'approche floue, ..., etc.

Notre objectif est de développer un algorithme permettant la détection en temps réel d'un objet mobile dans une séquence d'images, dans le but d'améliorer les procédés de détection et de compression d'images. Comme support théorique, on s'appuie sur les modèles basés sur les « *Chaînes de Markov Cachées CMCs* », appelés aussi « *Modèles de Markov cachés MMCs* ». Ces derniers permettent une utilisation souple et extensible.

Le traitement de l'information en temps réel, est devenu d'une exigence extrême. Pour cela des circuits performants spécifiques à l'imagerie sont introduits tel que : les *FPGAs*<sup>2</sup>, et les *DSPs*<sup>3</sup>, en vue d'une implantation de ces algorithmes souple et en temps réel.

L'algorithme de détection ainsi choisi, si les conditions matérielles et logicielles sont favorables, sera par la suite implanté dans une chaîne de traitement d'images piloté par un *DSP* de la famille TMS320C6000 de TEXAS INSTRUMENTS.

---

<sup>2</sup> : Feild Programmable Gate Array

<sup>3</sup> : Digital Signal Processor



