

Université ABBES LAGHROUR Khenchela

Faculté des Sciences et de la Technologie

Département de Génie Industriel



جامعة عباس لغرور خنشلة

كلية العلوم والتكنولوجيا

قسم الهندسة الصناعية

N° Série :.....

Mémoire de fin d'étude

Présenté pour l'obtention du diplôme de Master

Filière : Automatique

Spécialité : Automatique et informatique
industrielle

THEME

**Détection et poursuite d'un objet coloré dans
une séquence vidéo**

Réalisé par :

MAACHE AYOUB

BELFOUNES SALAH EDDINE

Devant Le Jury :

Président: MALLEM ALI

Examineur: BERKANI ABD ELHAKIM

Encadreur : M^{me} FRIHA Souad

Promotion 2022/2023

REMERCIEMENTS

✓ Nous tenons à remercier tout d'abord ALLAH le tout puissant qui nous a donné la santé, la force et la volonté pour réaliser ce travail.

✓ Nous remercions profondément notre encadreur **Mme FRIHA Souad** qui nous a guidés tout au long de la réalisation de ce mémoire et qui a été toujours présente ainsi que pour le temps qu'elle a consacré pour évaluer notre travail et la confiance qu'elle nous a accordés tout au long de la réalisation de ce projet. Nous la remercions pour son soutien, ses conseils précieux et également ses critiques.

✓ Nos remerciements vont aussi aux membres du jury d'avoir bien voulu évaluer notre travail.

✓ Nos remerciements vont également aux enseignants qui nous ont enseignés durant notre cursus universitaire.

✓ Un grand merci à nos familles, nos amis et à tous ceux qui nous ont aidés des près ou de loin.

DEDICACES

Je dédie ce modeste travail :

- *A mon père **Ammar** et ma mère **Mahdjouza** pour leurs affection et amour.*
- *A mes sœurs : **Sadja, Ahlam***
- *A mes frères : **Thabet, Fares** .*
- *À mon binôme : **Saleh eddine***
- *A tous ceux et celles que j'aime énormément.*
- *Et à tous ceux qui m'ont soutenu de près comme de loin.*

Ayoub

DEDICACES

Je dédie ce modeste travail :

- *A mon père **Ahmed** et ma mère **abida** pour leurs affections et amours.*
- *A ma sœurs : **malia***
- *A mes frères : **Ala eddine, ayoub.djhad***
- *À mon binôme : **Ayoub***
- *A tous ceux et celles que j'aime énormément.*
- *Et à tous ceux qui m'ont soutenu de près comme de loin.*

Salah

Table des matières

Introduction générale	1
Chapitre 1 : Généralité sur les images et les vidéos	
1. Introduction.....	
1. Définition de l'image.....	
2. Taille des données d'une image.....	
3. Image numérique	
4. Caractéristiques des images	
4.1. Pixel.....	
4.2. Dimension.....	
4.3. Résolution	
4.4. Bruit.....	
4.5. Histogramme.....	
4.6. Contours et textures.....	
4.7. Luminance	
4.8. Contraste.....	
4.9. Images binaires	
4.10. Images niveau de gris (Monochromes)	
4.11. Images en couleurs (Polychromes)	
4.11.1. Représentation en couleurs réelles.....	
4.11.2. Représentation en couleurs indexées	
4.11.3. Autres modèles de représentations.....	

5. Types d'images	
5.1. Image vectorielle	
5.2. Image Matricielle (bitmap)	
6. Formats d'images matricielles.....	
6.1. BMP (bitmap)	
6.2. PNG (Portable Network Graphics)	
6.3. TIFF (Tagged Image File Format)	
6.4. GIF (Graphics Interchange Format).....	
6.5. JPEG (Joint Photographic Experts Group).....	
7. Formats d'images vectorielles.....	
7.1. PICT	
8.2. PS	
8.3. DXF.....	
8.4. WPG	
8. Traitements généraux.....	
8.1. Amélioration ou retouche de l'image	
8.2. Compression	
10. La vidéo.....	
10.1. Définition.....	
10.2. Résolution temporelle d'une vidéo.....	
10.3. Séquence d'image.....	
10.4. Types de vidéo	
10.4.1. La vidéo analogique.....	
10.4.2. La vidéo numérique.....	
10.5. Les fondamentaux physiques et techniques.....	

10.5.1. La vision.....
10.5.2. La couleur.....
10.6. Le signal.....
10.7. Les caractéristiques d'un signal vidéo.....
10.8. Objet vidéo.....
11. Suivi d'objets
12. Conclusion.....

Chapitre 2 : Détection de blobs

1. Introduction.....
2. L'analyse de blob.....
2.1. La détection de blob.....
2.2. Extraction de blob.....
2.2.1. L'algorithme GrassFire récursif.....
2.2.2. L'algorithme Grassfire séquentiel.....
2.3. Caractéristiques d'un bon algorithme de suivi.....
2.3.1. L'Aire d'un blob (zone du blob).....
2.3.2. Enveloppes de blob.....
2.3.3. Coefficient de boite enveloppante du blob.....
2.3.4. Compacité d'un blob.....
2.3.5. Centre de masse (centre de gravité ou Centroïde).....
5 Conclusion.....

Chapitre 3 : résultat et commentaire

1. Introduction.....
2. Environnement du travail.....

2.1. Environnement matériel ou le hardware.....	
2.2. Environnement immatériel ou le software.....	
3. Matlab.....	
4. Implémentation de l'algorithme de détection et de suivi des couleursRVB-B...	
4.1. Les seuils de couleurs.....	
4.2. Le cycle de prétraitement.....	
4.2.1. Extraction de la composante rouge	
4.2.2. Obtention de la couleur rouge.....	
4.2.3. Filtrage médian	
4.2.3. Filtrage médian.....	
4.3. Analyse de blobs.....	
5. Résultats pour la détection des couleurs.....	
6. Suivi d' un objet.....	
7. Discuter et interpréter les résultats	
8. conclusion.....	

Liste des figures

CHAPITRE 1

Figure 1-1 : Représentation d'un pixel.....	
Figure 1-2 : Image groupe de pixels.....	
Figure 1-3: Image sans et avec bruit.....	
Figure 1-4: Image originale et image avec contours.....	
Figure 1-5: Image niveau de gris.....	
Figure 1-6: Nombre de couleurs dépend de nombre de bits.....	
Figure 1-7: Synthèse additive.....	
Figure 1-8: Image vectorielle.....	
Figure 1-9: Image bitmap.....	
Figure 1-10 : Exemple d'une séquence d'images.....	
Figure 1-11 : codage de couleur en RVB.....	
Figure 1-12 : Schéma de principe d'un Capture Tri-CCD (charged coupled device).....	

CHAPITRE 2

Figure 2.1 : Représentation de blobs dans une image binaire.....	
Figure 2.2 : L'effet de l'application de deux différents types de connexités 4 et 8....	
Figure 2. 3 : l'algorithme de Grassfire.....	
Figure 2.4 : Blobs extrait (chaque blob avec une couleur unique).....	
Figure 2.5 : différents types d'enveloppes de blob.....	

CHAPITRE 3

Figure. 3.1. Image originale.....	
-----------------------------------	--

Figure. 3.2. Composante rouge de l'image.....	
Figure. 3.3. Image différence entre la composante rouge et l'image niveau de gris....	
Figure.3.4. Image filtrée (Filtre médian de taille 3x3)	
Figure. 3.5. Image binaire (Seuil =0.05).....	
Figure 3.6. Détection d'un 1 ^{er} objet rouge.....	
Figure.3.7. Détection de plusieurs objets rouges.....	
Figure 3.8. Détection d'un objet blanc.....	
Figure.3.9. Détection des plusieurs objets.....	

Liste des abréviations

Pixel	P icture E lément
PPI	P ixel P er I nh
RGB	R eed G reen B lue
RVB	R ouge V ert B leu
YCbCr	Luminance (Y) , Chrominance (Red-Yellow), Chrominance (Blue-Yellow)
3D	Images A Trois D imensions.
CCD	C harged C oupled D evice
SSD	S um of S quare D ifferences
MATLAB	M atrix- L aboratory

Introduction générale

Le domaine de traitement d'images numériques est très vaste et trouve des applications dans de nombreux domaines de notre vie quotidienne. Des domaines tels que la médecine, la météo, les communications, la cartographie, la géologie et bien d'autres utilisent des images numériques pour améliorer la qualité des informations ou extraire des connaissances à partir des images.

En particulier, la détection et le suivi d'objets dans les images et les vidéos sont essentiels pour de nombreuses applications telles que la vidéosurveillance, la compression vidéo, l'imagerie médicale, la robotique, l'interaction homme-machine et l'analyse de séquences mathématiques. La détection de mouvement dans une scène, la détection d'objets en mouvement, la segmentation de scène basée sur le mouvement, l'estimation de mouvement, le suivi de primitives ou de régions, ainsi que la reconnaissance et la modélisation d'activités ou de gestes sont des problèmes clés dans ce domaine.

Ces problèmes sont souvent résolus à l'aide d'algorithmes de vision par ordinateur qui utilisent des techniques d'apprentissage pour produire des résultats utiles. La détection de mouvement consiste à identifier les zones en mouvement dans une image. La segmentation de scène basée sur le mouvement implique la détection et la segmentation de chaque zone de l'image présentant un mouvement distinct.

L'estimation de mouvement consiste à estimer le mouvement apparent des objets dans une scène 3D à partir d'une séquence d'images. Le suivi des primitives ou des régions a pour objectif de déterminer la position de chaque primitive ou région dans l'image à chaque instant. La reconnaissance et la modélisation des activités ou des gestes sont également des problèmes importants dans ce domaine.

La détection d'objets en mouvement est généralement une première étape dans le développement d'outils de vision artificielle par ordinateur. Ces outils peuvent être utilisés pour détecter, identifier ou suivre des objets afin d'analyser leur comportement ou leur trajectoire.

Le travail présenté dans ce mémoire se concentre sur la conception d'un code sous MATLAB qui permet dans un premier temps de détecter des objets colorés dans une image d'une séquence vidéo. La détection est basée essentiellement sur l'analyse de blobs. Dans un deuxième temps, cette détection est répétée successivement pour le reste des images de la séquence vidéo pour permettre le suivi de l'objet coloré.

Le mémoire est structuré en trois chapitres. Le premier chapitre présente des informations générales sur le traitement d'images et une étude approfondie des caractéristiques de base des vidéos. Il introduit aussi les notions de la détection et du suivi d'objets en mouvement. Le deuxième chapitre est consacré à la partie détection et discute principalement de l'analyse des blobs. Enfin, le troisième chapitre présente les résultats expérimentaux obtenus pour la détection et le suivi d'objets colorés en utilisant la méthode choisie [1] .

CHAPITRE 1

Généralités sur les images et les vidéos

1. Introduction

Une image vaut plus que mille mots dit-on. En effet, le signal image est un support d'une grande quantité d'information. Partout où on va travers le monde, l'image est devenue le moyen universel privilégié de communication. Tous les domaines sont pratiquement concernés: la médecine, la météo, la télécommunication, la cartographie, la géologie,etc.

Dans ce chapitre, nous présentons des généralités sur les images et les vidéos .

2. Définition de l'image

L'image est un ensemble structuré d'informations qui, après affichage sur écran, ont une signification pour l'œil humain.

Mathématiquement, l'image est un signal bidimensionnel décrit sous la forme d'une fonction $I(x, y)$, tel que x et y sont les coordonnées spatiales d'un point (pixel) .

3. Taille des données d'une image

La taille des données d'une image est mesurée par le nombre total de pixel que comporte l'image multiplié par le nombre de bit par pixel et cela dépendra des dimensions de l'image (nombre de colonne, nombre de ligne), donc on peut écrire la relation suivante [2]:

$$\text{Taille} = M \times N \times C \quad (2.1)$$

Où:

M : Nombre de colonnes.

N : Nombre de lignes.

C : Nombre de bits par pixel.

4. Image numérique

L'image numérique est une surface divisée en éléments de tailles fixes appelés cellules ou pixels, ayant chacun comme caractéristique un niveau de gris ou de couleurs prélevés à l'emplacement correspondant dans l'image réelle. La numérisation d'une image est la conversion de celle-ci de son état analogique (distribution continue d'intensités lumineuses) en une image numérique représentée par une matrice bidimensionnelle de valeurs numériques. Donc, l'image numérique est une matrice composée d'échantillons élémentaires appelés pixels dont « **M** » le nombre de lignes de l'image, et « **N** » le nombre de colonnes. La position d'un pixel sur la matrice image sera indiquée par son indice de ligne **m** $[0, M [$, et son indice de colonne **n** $[0, N [$.

La qualité de l'image numérique dépend, d'une part, de la qualité des images d'origine et d'autre part, des moyens mis en œuvre pour convertir un signal analogique en signal numérique. Elle dépend aussi de :

- ✓ La qualité des périphériques de numérisation de l'image, du nombre de niveaux de gris ou de couleurs enregistrées, ...etc.
- ✓ La qualité de l'affichage à l'écran : définition de l'écran, nombre de teintes disponibles simultanément, calibrage de l'écran, ...etc [3].

5. Caractéristiques des images

Les images sont caractérisées par les éléments suivants : [4]-[5]

5.1. Pixel

Contraction de l'expression anglaise "Picture éléments" : éléments d'image, le pixel est le plus petit point de l'image, c'est une entité calculable qui peut recevoir une structure et une quantification. Si le bit est la plus petite unité d'information que peut traiter un ordinateur, le pixel est le plus petit élément que peuvent manipuler les matériels et logiciels d'affichage ou d'impression (figure 1-1).

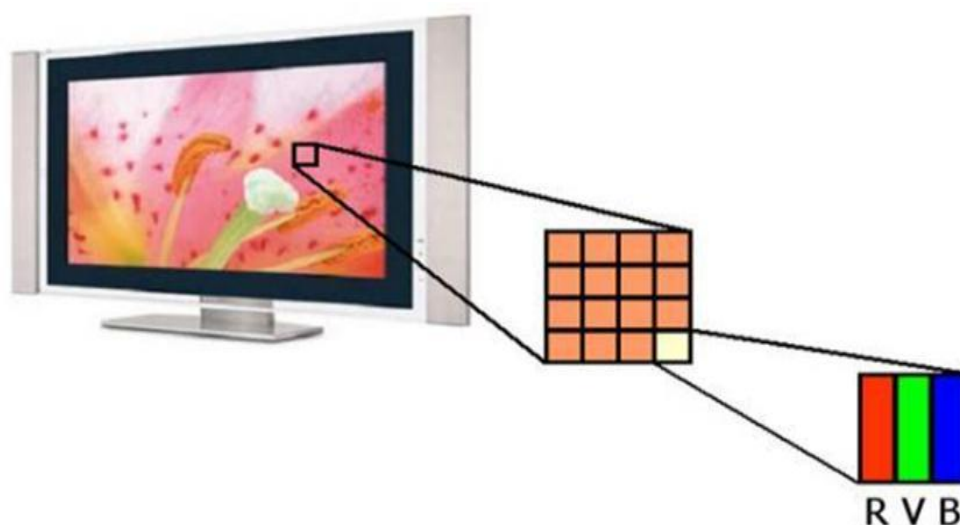


Figure 1.1. Représentation d'un pixel.

La lettre A, par exemple, peut être affichée comme un groupe de pixels (figure (1-2)).

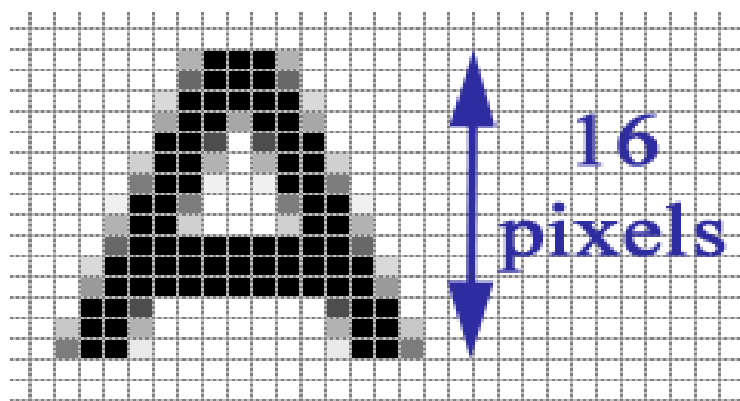


Figure 1.2 . Image groupe de pixels.

La quantité d'information que véhicule chaque pixel donne des nuances entre images monochromes et images couleur. Dans le cas d'une image monochrome, chaque pixel est codé sur un octet, et la taille mémoire nécessaire pour afficher une telle image est directement liée à la taille de l'image. Dans une image couleur (RVB), un pixel peut être représenté sur trois octets : un octet pour chacune des couleurs : Rouge (R), Vert (V) et Bleu (B) [16].

5.2. Dimension

C'est la taille de l'image. Cette dernière se présente sous forme de matrice dont les éléments sont des valeurs numériques représentatives des intensités lumineuses (pixels). Le nombre de lignes de cette matrice multiplié par le nombre de colonnes nous donne le nombre total de pixels dans une image.

5.3. Résolution

C'est la clarté ou la finesse de détails atteinte par un moniteur ou une imprimante dans la production d'images. Sur les moniteurs d'ordinateurs, la résolution est exprimée en nombre de pixels par unité de mesure (pouce ou centimètre).

5.4. Bruit

Un bruit (parasite) dans une image (cf. Fig.1.3) est considéré comme un phénomène de brusque variation de l'intensité d'un pixel par rapport à ses voisins, il provient de l'éclairage des dispositifs optiques et électroniques du capteur.

Le bruit peut provenir de différentes causes :

- ✓ Environnement lors de l'acquisition.
- ✓ Qualité du capteur.
- ✓ Qualité de l'échantillonnage.



(a) Image originale



(b) Image avec bruit(salt & pepper)



(c) Image débruitée (filtre médian).

Figure 1.3. Image sans et avec bruit.

5.5. Histogramme

L'histogramme des niveaux de gris ou des couleurs d'une image est une fonction qu'en pratique on compte le nombre de pixels de l'image correspondant à chaque niveau de gris (couleur). Il permet de donner un grand nombre d'information sur la distribution de ces niveaux. Il peut être utilisé pour améliorer la qualité d'une image (Rehaussement d'image) en introduisant quelques modifications, pour pouvoir extraire les informations utiles de celle-ci donne la fréquence d'apparition de chaque niveau de gris (couleur) dans l'image. On peut distinguer trois types d'histogramme d'image :

a) Histogramme uni modal

Ce type d'histogramme n'a qu'un seul pic, il présente soit un objet soit un fond.

b) Histogramme bimodal

Il est formé de deux modes bien séparés (deux pics séparés par une vallée) et l'on déduit ainsi qu'il existe un objet sur un fond.

c) Histogramme multimodal

Il est formé de plusieurs modes séparés (plusieurs pics séparés par plusieurs vallées). Qui nous renseigne sur la présence de plusieurs objets.

5.6. Contours et textures

Les contours représentent la frontière entre les objets de l'image, ou la limite entre deux pixels dont les niveaux de gris représentent une différence significative. Les textures décrivent la structure de ceux-ci. L'extraction de contour consiste à identifier dans l'image les points qui séparent deux textures différentes (figure 1.4).



(a) Image originale



(b) Image avec contours.

Figure 1.4. Image originale et image avec contours.

5.7. Luminance

C'est le degré de luminosité des points de l'image. Est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface, pour un

observateur lointain, le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet. Une bonne luminance se caractérise par :

- ✓ Des images lumineuses (brillantes);
- ✓ Un bon contraste : il faut éviter les images où la gamme de contraste tend vers le blanc ou le noir; ces images entraînent des pertes de détails dans les zones sombres ou lumineuses.
- ✓ L'absence de parasites.

5.8. Contraste

C'est l'opposition marquée entre deux régions d'une image, plus précisément entre les régions sombres et les régions claires de cette image.

5.9. Images binaires

L'image binaire est une matrice rectangulaire dont les éléments valent 0 ou 1. Donc, il est possible de définir deux couleurs (noir et blanc), en stockant 1 bit par pixel (bpp).

5.10. Images niveau de gris (Monochromes)

Le niveau de gris est la valeur de l'intensité lumineuse en un point. La couleur du pixel peut prendre des valeurs allant du noir au blanc en passant par un nombre fini de niveaux intermédiaires [6] (voir figure (1-5)).

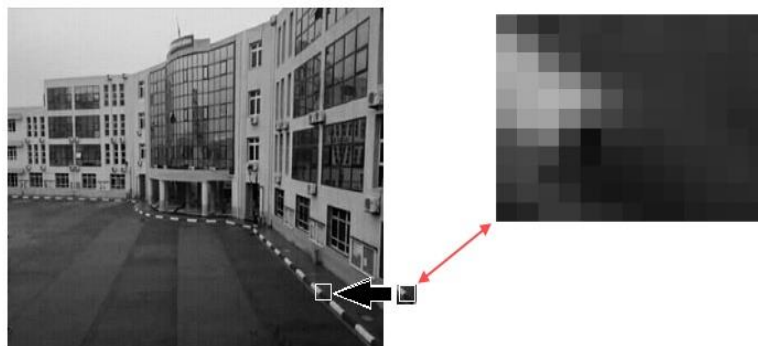


Figure 1.5. Image niveau de gris.

5.11. Images en couleurs (Polychromes)

Il existe plusieurs types de représentation des images couleurs, parmi celle-ci on peut citer :

5.11.1. Représentation en couleurs réelles

Elle consiste à utiliser 24 bits pour chaque point de l'image. Huit bits sont employés pour décrire la composante rouge (R), huit bits pour la composante verte (V) et huit bits pour la composante bleue (B). Il est ainsi possible de représenter environ 16,7 millions de couleurs différentes simultanément (voir figure (1-6)).

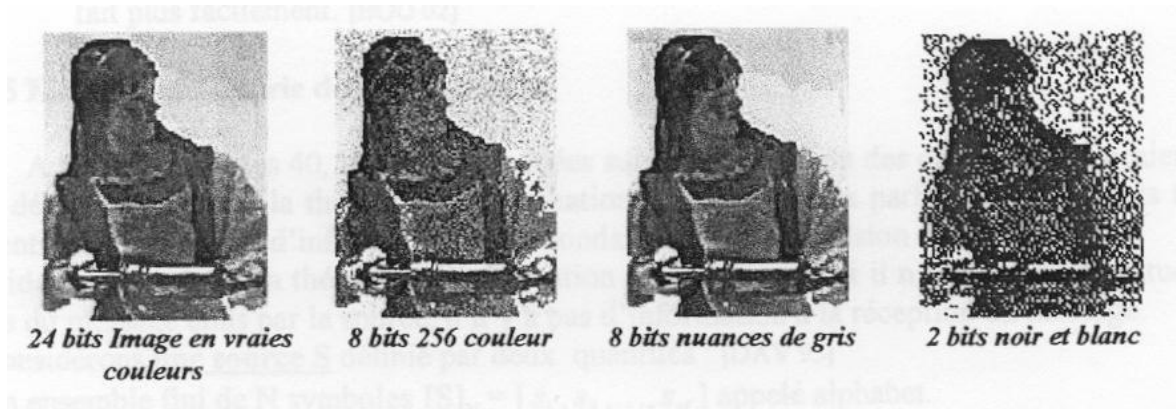


Figure 1.6. Nombre de couleurs dépend de nombre de bits.

5.11.2. Représentation en couleurs indexées

Pour réduire la place occupée par l'information de couleur, on utilise une palette de couleurs attachée à l'image. On parle alors de couleurs indexées : la valeur associée à un pixel ne véhicule plus la couleur effective du pixel, mais renvoie à l'entrée correspondant à cette valeur dans une table (ou palette) de couleurs appelée look-up table ou LUT en anglais, dans laquelle on dispose de la représentation complète de la couleur considérée.

5.11.3. Autres modèles de représentations

Le modèle RVB représentant toutes les couleurs par l'addition de trois composantes fondamentales, n'est pas le seul possible (voir figure 1-7). Il en existe de nombreux autres.

L'un d'eux est particulièrement important. Il consiste à séparer les informations de couleurs (chrominance) et les informations d'intensité lumineuse (luminance). Il s'agit du principe employé pour les enregistrements vidéo. La chrominance est représentée par deux valeurs (selon des modèles divers) et la luminance par une valeur [7].

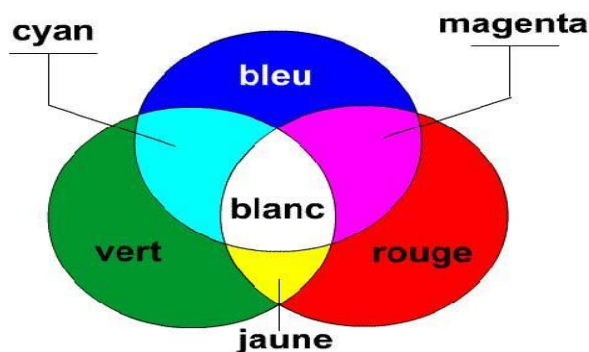


Figure 1.7. Synthèse additive.

6. Types d'images

Les images appartiennent à deux grandes familles : Images vectorielles et images matricielles.

6.1. Image vectorielle

L'image vectorielle (figure 1-8) est une représentation conceptuelle de forme calculée par des formules mathématiques, (exemple : un cercle n'est pas déterminé par des pixels mais par une formule mathématique qui détermine sa forme, sa taille et son emplacement).

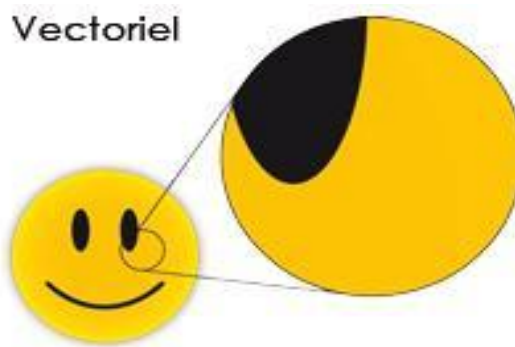


Figure 1.8. Image vectorielle.

6.2. Image Matricielle (bitmap)

L'image bitmap est représentée par une trame de (pixels). Ce ne sont pas des formules mathématiques qui définissent les formes, mais un ensemble de pixels qui agissent comme un tableau pointilliste [8].

Exemple

Une image (comme un cercle par exemple) de petite taille, que l'on agrandit dix fois. Cette opération peut entraîner une perte dans la qualité des couleurs ou la netteté. Alors l'image devient déformée (figure 1-9).

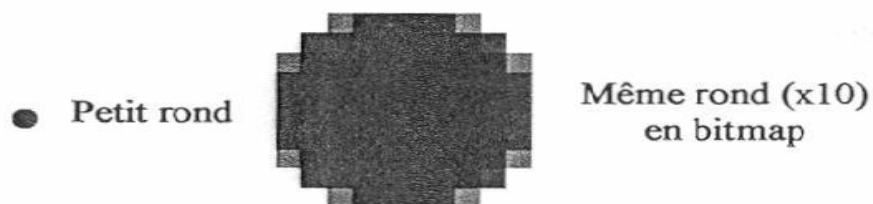


Figure 1.9. Image bitmap.

7. Formats d'images matricielles

7.1. BMP (bitmap)

Un fichier bitmap affiche un petit point dans un motif qui, lorsqu'il est vu de loin, crée une image globale.

7.2. PNG (Portable Network Graphics)

Le format PNG fournit une norme portable, légalement inutilisée, bien compressée et bien définie pour les fichiers d'images bitmapiques sans perte.

Bien que la motivation initiale pour le développement de PNG soit de remplacer GIF, la conception offre de nouvelles fonctionnalités utiles non disponibles dans GIF, avec un coût minimal pour les développeurs [9].

7.3. TIFF (Tagged Image File Format)

Le format de fichier d'image marqué (TIFF) est un autre format de fichier d'image populaire. Développé par Aldus Corporation dans les années 1980, il a ensuite été soutenu par Microsoft. Son support pour l'ajout d'informations complémentaires (dénommé "tags") offre une grande flexibilité physique [10].

7.4. GIF (Graphics Interchange Format)

Essentiellement, chaque image GIF contient une «boîte de crayons» prédéfinie et il n'y a aucun moyen de mélanger ces couleurs pour créer de nouvelles couleurs. Alors que GIF est généralement, un choix médiocre pour les images avec une grande variation de couleur, la limite de 256 couleurs peut aider à réduire les dimensions des fichiers, ce qui est idéal même pour les vitesses d'Internet les plus lentes. Pendant de nombreuses années, GIF a fourni la seule option de transparence du Web - bien que PNG et SVG offrent maintenant cela aussi [11].

7.5. JPEG (Joint Photographic Experts Group)

C'est un format de fichier graphique permettant des taux de compression impressionnants mais au détriment de la qualité de l'image : la compression se fait avec perte d'information. L'extension de fichier correspondante est JPG. Ce format ne supporte pas la transparence contrairement au GIF. La norme JPEG utilise la DCT (Transformée en Cosinus Discrète) [12].

8. Formats d'images vectorielles

8.1. PICT

PICT pour Picture de Apple est obsolète comparé aux autres formats disponibles. Le format PICT est le format standard d'images du monde Macintosh, toutes les applications de dessin sous cet environnement sont généralement capables d'exporter des images dans ce format. Les fichiers PICT peuvent provenir directement du Macintosh, ou encore être générés par des applications de dessin Windows comme Photoshop ou CorelDraw. L'utilisation du format PICT à l'intérieur de la base de données permet de visualiser ces images à la fois sur Macintosh et sur PC. L'extension des fichiers PICT sous Windows peut être soit PIC, soit PCT, suivant le logiciel ayant généré l'image. Les fichiers PICT sont compressés ou non par QuickTime.

8.2. PS

PS pour PostScript utilisé avec la majorité des applications d'aujourd'hui, autant les logiciels de mise en pages, de traitement de textes et autres, il est possible d'exporter un document en format PS (PostScript) lequel pourra être acheminé vers un périphérique d'impression. Ce format est également une façon sûre de rendre disponible un document seulement pour impression sans droit de modification. Il s'agit toutefois d'un format très lourd à éviter lorsqu'il doit être transféré par Internet sur des liens à basse vitesse.

8.3. DXF

Le format DXF est un format créé par la compagnie AutoDesk pour son logiciel de CAO AUTOCAD. Bien qu'étant un format très répandu dans le monde de la conception et du dessin assisté par ordinateur, le format DXF est très peu répandu en d'autres domaines.

8.4. WPG

Le format WPG est un format utilisé par les logiciels de la gamme de WordPerfect (WordPerfect, DrawPerfect, WP Présentation et autres) sous DOS, Windows ou Macintosh. Ce format donne un résultat acceptable lors de l'impression, mais qui doit surtout être utilisé en tant que format de travail. D'autant plus que ce n'est pas un format qui est reconnu par tous les logiciels [13] [14].

9. Traitements généraux

Les traitements généralement appliqués aux images sont de deux types :

9.1. Amélioration ou retouche de l'image

Le traitement de l'image informatique correspond à toutes les possibilités de travail, retouches ou modifications que l'on peut réaliser avec un ordinateur sur une image (enregistrée) quand elle est affichée à l'écran [15].

Les logiciels de traitement d'image en pixels permettent de réaliser des modifications sur ces images pour obtenir des effets précis à savoir :

- ✓ Recadrer une image
- ✓ Modifier la taille (à l'écran ou à l'impression)
- ✓ Retoucher des couleurs
- ✓ Intégrer un texte
- ✓ Changer un fond
- ✓ Effacer une partie de l'image.

9.2. Compression

La qualité d'une image dépend de sa résolution (nombre de pixels) et du nombre de couleurs utilisées pour coder chaque pixel [15].

- En général IMAGE =GROS FICHER.
- Le volume d'une image numérisée est égal à la quantité d'octets utilisés pour la coder.

10. La vidéo

10.1. Définition

La vidéo est une succession d'images animées défilant à une certaine cadence afin de créer une illusion de mouvement pour l'œil humain.

Elle peut être analogique (signal continu d'intensité de luminance) ou numérique (suite de trames ou images) [17].

10.2. Résolution temporelle d'une vidéo

La résolution temporelle d'une vidéo est définie par le nombre d'images défilant par seconde. Afin d'éviter les désagréments dus aux papillotements, et prenant en considération les spécificités de l'œil humain qui garde une image environ 10 micro secondes au niveau de sa rétine ; les images doivent défiler à une certaine cadence:

- environ 24 images/s pour un film de cinéma.
- 25 images/s pour la télévision européenne.
- environ 30 images/s pour la télévision nord-américaine et japonaise.

10.3. Séquence d'image

Une séquence d'images (voir figure (1-10)) est une succession d'images bidimensionnelles qui montre l'évolution temporelle d'une scène. La cadence est de 25 images par seconde, ce qui correspond au seuil à partir duquel l'œil humain perçoit la séquence comme un stimulus continu, grâce à la persistance rétinienne. Par la suite, nous appellerons «trame» ou «plan» chaque image bidimensionnelle correspondant à un instant donné de la séquence.

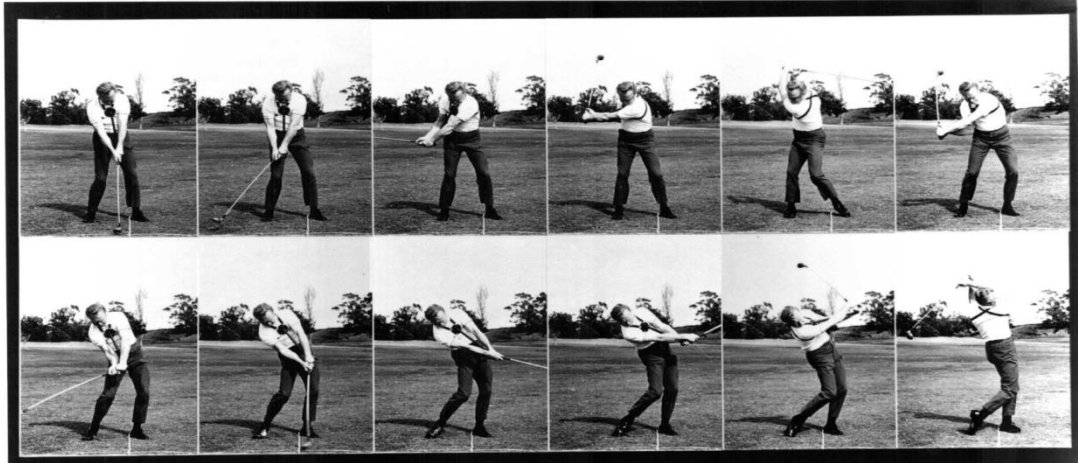


Figure 1.10. Exemple d'une séquence d'images.

10.4. Types de vidéo

On distingue deux grandes familles de systèmes vidéo : les systèmes vidéo analogiques et les systèmes vidéo numériques.

10.4.1. La vidéo analogique

La caméra balaye l'image bidimensionnelle qu'elle a devant elle par un faisceau d'électrons qui se déplace très rapidement de gauche à droite et plus lentement de haut en bas et produit une tension en fonction du temps. Elle enregistre ainsi l'intensité lumineuse, et à la fin du balayage, on a alors une trame. Le faisceau revient à l'origine pour recommencer. Le récepteur va recevoir cette intensité en fonction du temps, et pour reconstruire l'image, va répéter le processus de balayage [18].

10.4.2. La vidéo numérique

Le processus de captation de l'image vidéo en mode numérique est essentiellement le même que pour la vidéo analogique. Un système optique sépare la lumière en trois composants. À la sortie toutefois, plus d'enregistrement d'un signal électrique mais plutôt l'enregistrement d'une valeur numérique définie pour chacune des trois couleurs de chacun des pixels (**RVB**). Le processus de transformation d'un signal électrique et valeurs numériques s'appelle la numérisation ou l'échantillonnage. Deux paramètres servent à

échantillonner un signal électrique : la fréquence d'échantillonnage et la quantification de l'échantillonnage ou le nombre de bits utilisés pour le codage.

10.5. Les fondamentaux physiques et techniques :

10.5.1. La vision

Dans le phénomène de la vision humaine, un flux lumineux composé de photons frappe l'œil. Dans l'œil, les bâtonnets vont réagir à l'intensité de la lumière (la luminance ou Y) et les cônes vont réagir à la couleur (la chrominance ou C).

Le cerveau effectue la synthèse de ces informations pour composer une image. L'œil humain est davantage sensible à l'intensité lumineuse (Y) qu'à la couleur (C).

10.5.2. La couleur

La vidéo utilise ce qui est identifié comme la synthèse additive de la couleur. Dans le système colorimétrique additif, les trois couleurs primaires sont le rouge, le vert et le bleu (RVB ou RGB dans la terminologie anglophone). En combinant ces trois couleurs primaires, il est possible de reproduire tout le spectre visible par l'œil humain. L'écran de visualisation vidéo sera donc composé d'une série de triades rouge-vert-bleu. L'activation de l'ensemble de ces triades formera l'image .

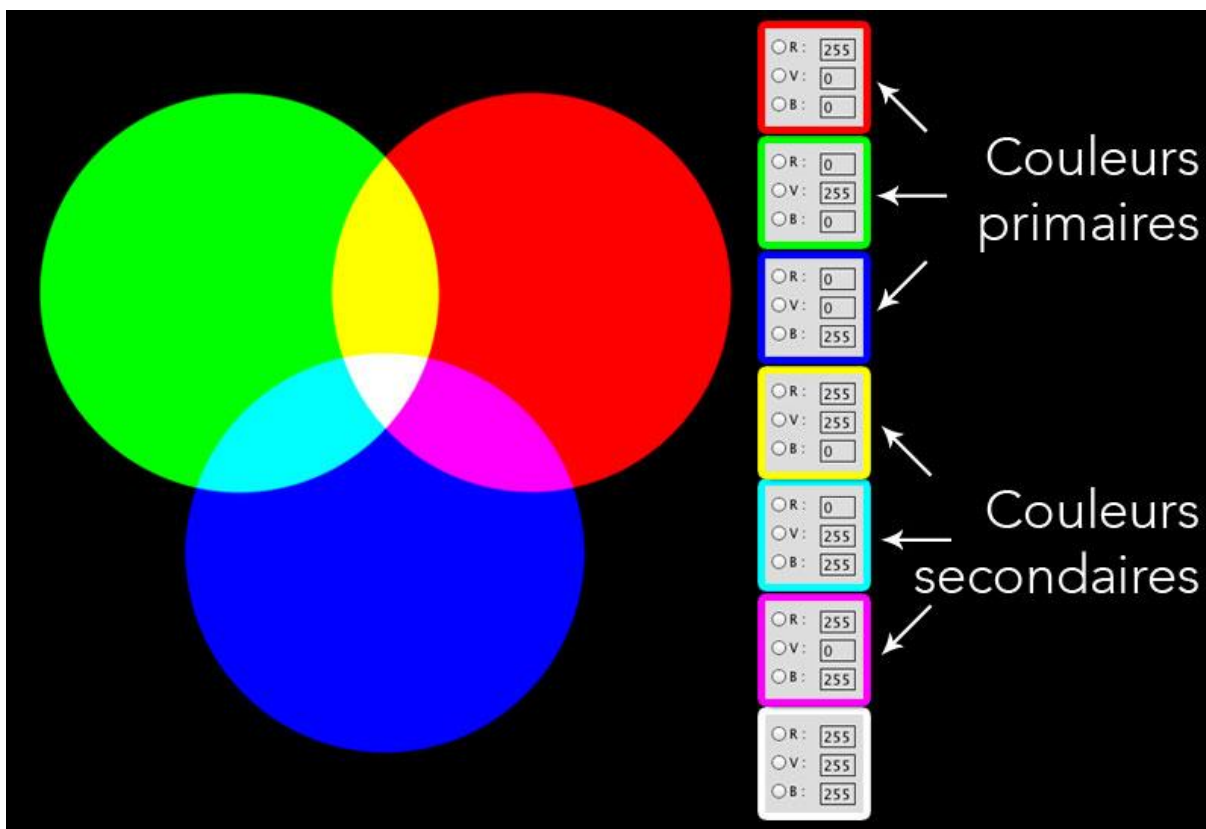


Figure 1.11. Codage de couleur en RVB

10.6. Le signal

En vidéo, c'est la caméra qui transforme l'information lumineuse (photons) en signal électrique (électrons), en vidéo analogique, l'intensité de ce signal électrique varie de façon continue. Le processus de transformation de l'information lumineuse en signal électrique est le suivant :

- L'élément optique de la caméra, l'objectif, sépare la lumière en trois composants le rouge, le vert, le bleu. Cette opération est réalisée en faisant passer le flux lumineux par une succession de filtres dichroïques réfléchissant certaines couleurs et en laissant passer d'autres.
- En vidéo professionnelle, les trois images sont projetée sur trois capteurs photo sensibles distincts formés chacun de centaines de milliers de points généralement entre 400 000 et 700 000. Ces capteurs sont nommés CCD (charged coupled device) ou

dispositifs à transfert de charge. Les caméras domestiques ne sont généralement équipées que d'un seul capteur CCD.



Figure 1.12 . Schéma de principe d'un Capture Tri-CCD (charged coupled device)

- Pour chacun des points de chacun des capteurs, l'énergie lumineuse sera transformée en énergie électrique.

Ainsi, à la sortie des capteurs, trois signaux électriques d'intensité variable, un signal pour chacune des trois composantes. La lumière blanche est formée par la somme des trois composantes **RVB**.

Toutefois, elle n'est pas composée des trois couleurs primaires en quantité égale. Les propositions sont les suivantes :

$$100\% Y = 29,9\%(R) + 58,7\%(V) + 11,4\%(B) \quad (2.2)$$

- Une information supplémentaire devra être ajoutée à ces signaux. Comme nous l'avons dit, une valeur doit être établie pour chacun des points des capteurs, c'est cette valeur pour chacun des points de chacun des pixels qui permettra de recomposer l'image sur un écran de visualisation, chacun des points de la surface d'affichage pouvant alors prendre la même valeur qu'au moment de la capture par la caméra.

La restitution de l'information pour chacun des pixels se fait par balayage ligne par ligne de la surface de l'écran de reproduction, les valeurs enregistrées pour chacun des pixels au moment de la capture étant transférées aux pixels de la surface de reproduction. Pour synchroniser ces balayages, un signal de fin de ligne est ajouté pour chacune des lignes. Un signal de fin d'image est également inséré.

10.7. Les caractéristiques d'un signal vidéo

Chaque fichier vidéo a des attributs qui décrivent ce qui constitue le signal vidéo. Ces caractéristiques comprennent :

- **Châssis de taille** : Il s'agit de la dimension de pixel de l'image
- **Le ratio d'aspect** : C'est le rapport entre la largeur et la hauteur
- **Vitesse de défilement**: C'est la vitesse à laquelle les images sont capturées et destinées à la lecture.
- **Débit**: Le taux de débit ou de données est la quantité de données utilisées pour décrire la partie audio ou vidéo du fichier. Il est généralement mesuré en unités par seconde et peut être en kilo-octets, méga-octets ou giga-octets par seconde. En général, plus la vitesse de transmission, meilleure est la qualité.
- **Le taux d'échantillonnage audio**: C'est à quelle fréquence le signal audio est échantillonné lors de la conversion d'une source analogique à un fichier numérique.

10.8. Objet vidéo :

10.8.1. Définition :

Définir un objet vidéo est une chose difficile. Intuitivement, un objet vidéo est défini par une forme, une texture et un mouvement (rigide ou non rigide). Cependant, la notion d'objet vidéo est beaucoup plus descriptive qu'une simple région, un objet vidéo peut par exemple être un modèle **3D**.

La notion d'objet ne fait pas forcément référence à un objet du monde réel. En effet, dans le domaine de la vidéo, un objet n'est pas nécessairement un objet d'une scène **3D** mais plutôt le résultat de l'analyse de la projection d'un monde **3D** sur un plan. Ainsi, un objet vidéo est défini comme une région de la vidéo conforme à un modèle. On peut par exemple avoir pour modèle :

- un modèle de mouvement.
- un modèle d'objet physique.

La notion d'objets est alors subjectivement définie par rapport aux modèles utilisés, On constate que pour pouvoir trouver des objets vidéo, il est nécessaire de proposer des modèles auxquels ces objets répondront [19].

11. Suivi d'objets

Le suivi d'objets quelconques dans une séquence vidéo réelle enregistrée ou en live est une tâche très délicate. Surtout quand ces objets sont non-rigides, le fond de la scène n'est pas fixe et dans le cas de plusieurs objets mobiles dans la même scène. Lorsqu'il s'agit de suivre des objets particuliers comme le visage, la main, le bras. La tâche est plus simple car le modèle et les contraintes de variation de ces objets sont connus a priori. D'autres critères de type géométrique et/ ou statique peuvent aussi être introduits dans l'identification des ces objets.

Une variante du problème de suivi d'objets est le suivi d'objets en mouvement [23]. Dans ce cas on divise le problème en deux parties :

- La détection d'objets en mouvement dans chaque image,
- Le suivi des objets en mouvement d'une image à l'autre.

La détection d'objets en mouvement utilise un algorithme de soustraction de fond basé sur des modèles de mélange gaussien. Les opérations morphologiques sont appliquées au masque du premier plan résultant pour éliminer le bruit. Enfin, l'analyse blob détecte des groupes de pixels connectés, susceptibles de correspondre à des objets en mouvement.

Le suivi proprement dit utilise des approches statistiques en général et le filtre de Kalman est préconisé dans la plupart du temps.

L'analyse blob citée ci-haut est une phase importante de l'étape détection dans toute opération de suivi d'objet. Dans notre travail nous mettons l'accent sur l'analyse de blob dans le contexte particulier de détection de couleurs. Cette application est simple étant donné qu'elle permet de faire le suivi par une simple détection de couleurs au travers des images successives d'une vidéo.

12. Conclusion

Dans ce chapitre, nous avons fait un tour d’horizon sur les notions élémentaires des images numériques ainsi que celles des vidéos. Nous avons aussi introduit les notions d’objet, de suivi d’objet en mouvement et de l’analyse blob, objet de la présente étude et que nous détaillons au prochain chapitre.

CHAPITRE 2

Détection de blobs

1. Introduction

Dans le traitement d'image, la détection de blob fait référence à des modules qui visent à détecter des points et/ou des régions de l'image qui diffèrent par des propriétés telles que la luminosité ou la couleur par rapport à l'environnement. Il existe plusieurs motivations pour étudier et développer des détecteurs de blob. L'une des principales raisons est de fournir des informations complémentaires sur les régions, qui ne sont pas obtenues à partir de détecteurs de bord ou de détecteurs de coin. Il est utilisé pour obtenir des régions d'intérêt pour un traitement ultérieur. Ces régions pourraient signaler la présence d'objets ou de parties d'objets dans le domaine de l'image avec une application à la reconnaissance d'objets et/ou au suivi d'objets.

La détection des taches est généralement effectuée après la détection des couleurs et la réduction du bruit pour finalement trouver l'objet requis à partir de l'image. De nombreuses gouttes sans importance de la couleur requise peuvent être présentes dans l'image et nous avons également besoin des détails de la goutte comme le centre de la goutte, les coins de la goutte et tout pour lequel il est nécessaire que nous connaissions les coordonnées exactes de tous les pixels dans lesquels le blob requis est présent dans. [18]

2. L'analyse de blob

2.1. La détection de blob

Dans la vision par ordinateur, les méthodes de détection de blob visent à détecter

Les régions dans une image numérique qui diffèrent par leurs propriétés, telles que la luminosité ou la couleur, par rapport aux régions environnantes. Officieusement, un blob est une région d'une image dans laquelle certaines propriétés sont constantes ou à peu près

constantes; tous les points dans un blob peuvent être considérés en quelque sorte comme semblables les uns aux autres.

Avec la terminologie plus récente utilisée dans le domaine, ces détecteurs peuvent également être désignés comme opérateurs de point d'intérêt, ou encore les opérateurs de région d'intérêt. Il y a plusieurs motivations pour l'étude et le développement de détecteurs de blob. Une raison principale est de fournir des informations complémentaires sur les régions, qui ne sont pas obtenus à partir de détecteurs de bord ou de détecteurs d'angle.

Pour les premiers travaux dans le domaine, la détection de blob a été utilisée pour obtenir des régions d'intérêt pour un traitement ultérieur. Ces régions pourraient signaler la présence d'objets ou des parties d'objets dans le domaine de l'image avec l'application de reconnaissance d'objets et/ou de suivi l'objet.

Dans d'autres domaines, tels que l'analyse d'histogramme, les descripteurs de blob peuvent également être utilisés pour la détection de crête avec application à la segmentation. Une autre utilisation commune des descripteurs de blob en tant que primitives principales pour la l'analyse de texture et reconnaissance de texture.

Dans des travaux plus récents, les descripteurs de blob ont trouvé une utilisation de plus en plus populaire en tant que points d'intérêt pour une large base de techniques stéréo matching et pour signaler la présence de fonctionnalités d'images informatives pour la reconnaissance d'objets à base d'apparence s'appuyant sur les statistiques d'image locales. Il y a aussi la notion connexe de détection de crête pour signaler la présence d'objets allongés (voir figure 2.1) [19]

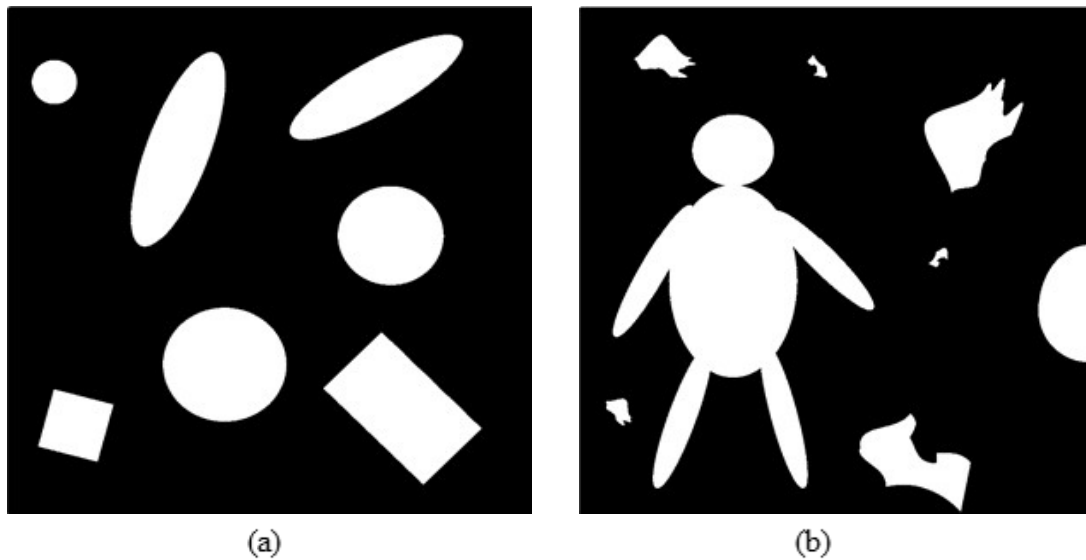


Figure 2.1 : Représentation de blobs dans une image binaire.

- (a) Une image binaire contenant différentes formes
- (b) Une image binaire contenant une silhouette humaine et du bruit

2 2.Extraction de blob

Le but de l'extraction de blob est d'isoler les blob (objets) dans une image binaire. Un blob se compose d'un groupe de pixels connexes. Les deux types les plus souvent appliquées de connexité sont illustrés sur la (figure (2-2)).

La 8-connexité est plus précise que la 4-connexité, mais la 4-connexité est la plus souvent utilisée car elle nécessite moins de calculs, ce qui permet une rapidité de traitement de l'image. L'effet des deux différents types de connexité est illustré sur la Figure 2.2 où les images binaires contiennent un ou deux blobs en fonction de la connexité.

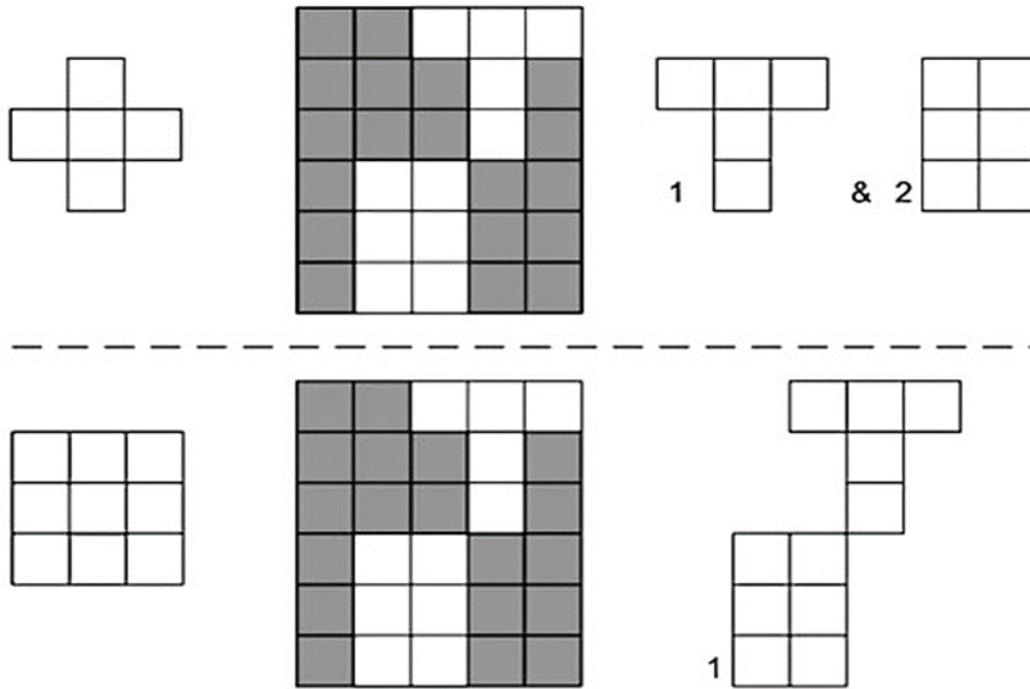


Figure 2.2 : L'effet de l'application de deux différents types de connexités 4 et 8.

Un certain nombre d'algorithmes existent pour la recherche des blobs, de tels algorithmes sont généralement référés à l'analyse ou l'étiquetage de composantes connexes. Dans ce qui suit, nous décrivons l'un de ces algorithmes connus sous le nom de l'algorithme Grassfire. Nous utilisons une 4-connectivité pour sa simplicité .

2.2.1.L'algorithme GrassFire récursif

L'algorithme commence dans le coin supérieur gauche de l'image binaire. Il analyse ensuite l'image entière de gauche à droite et de haut en bas .

Tout d'abord, dans l'image résultante, il donne à ce pixel une étiquette d'objet (essentiellement un nombre) et deuxièmement, il "brûle" le pixel dans l'image initiale par sa mise à zéro (noir). La mise à zéro du pixel indique qu'il a été brûlé et ne fera donc pas partie d'un autre processus d'élimination. Cependant, nous ne pouvons effectuer qu'une seule action à la fois et l'algorithme Grassfire est donc exécuté comme suit :

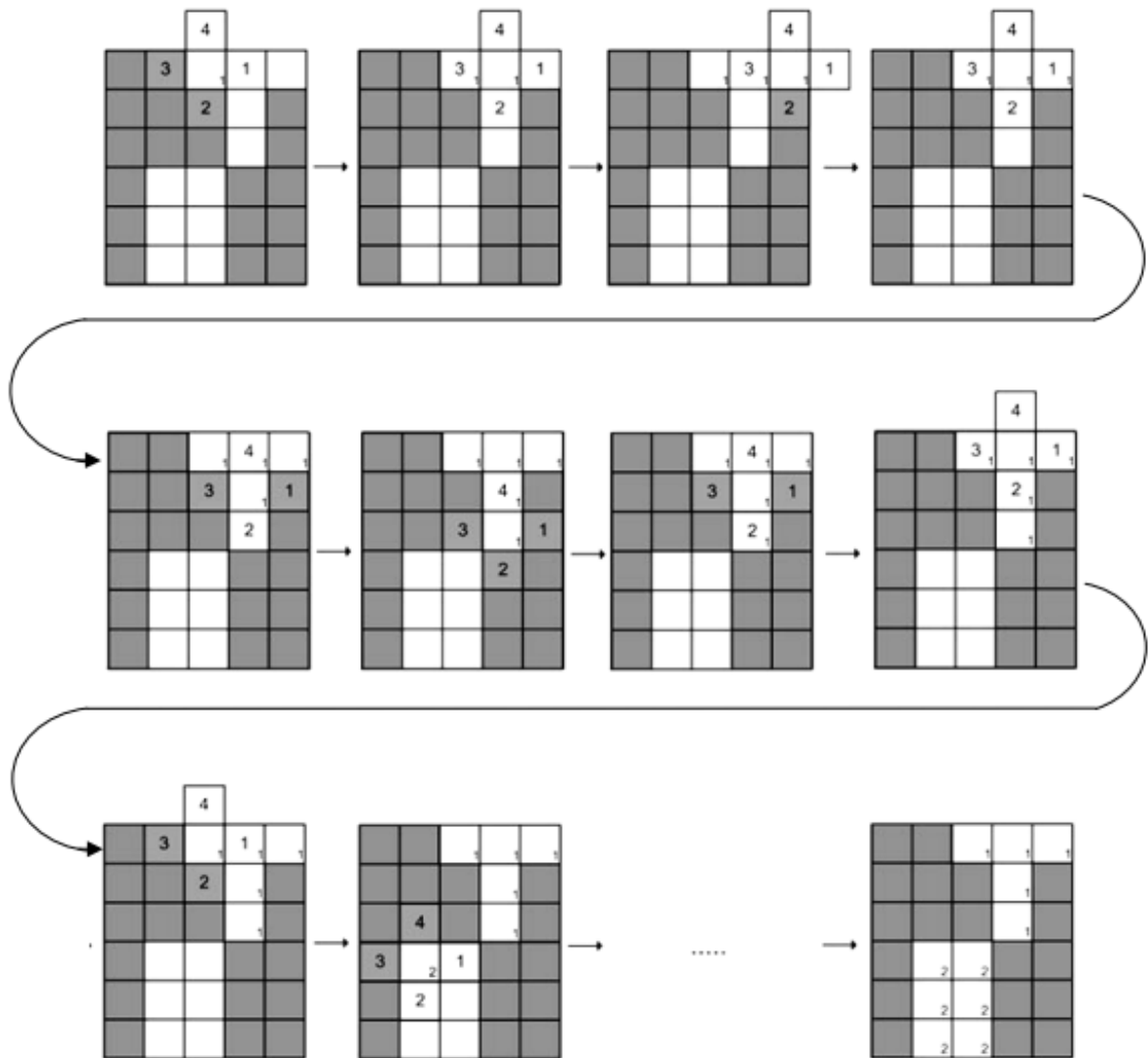


Figure 2. 3 : l'algorithme de Grassfire.

Le pixel se trouvant à la position (1, 3) est étiqueté, dès lors qu'il est le premier blob et puis sera brûlé (marqué par un 1 dans le coin inférieur droit). Après cela l'algorithme essaie de déclencher un feu au premier voisinage de (1, 4), en vérifiant si c'est un pixel objet ou non. Comme c'est un pixel objet (même objet) il sera donc étiqueté par 1 et sera "brûlé."

Du fait que le pixel (1, 4) est un pixel objet, il devient maintenant le centre d'attention, et son premier voisinage examiné sera le pixel (1, 5). Encore une fois, comme c'est un pixel objet il sera donc étiqueté par 1, et sera "brûlé" et devient le centre d'attention. Le premier

voisinage de (1, 5) est en dehors de l'image, et donc, par définition, n'est pas un pixel d'objet : l'algorithme examine donc son deuxième voisinage (2, 5). Comme Celui ci n'est pas un pixel objet donc il ne sera pas pris en compte, le troisième voisinage du pixel (1, 5) est donc examiné c'est le (1,4). Etant donné que celui ci a été déjà brûlé et donc n'est plus considéré comme un pixel objet. Ca sera ensuite autour du dernier voisinage du (1, 5) d'être examiné et qui est le pixel se trouvant a la position (-1, 5), ce dernier est en dehors de l'image et donc n'est pas un pixel d'objet.

Tous les voisinages du pixel se trouvant a la position (1, 5) ont maintenant été examinés, ensuit l'algorithme retourne et contrôle le second voisinage de (1, 4), à savoir le (2, 4). Il s'agit d'un pixel objet et est donc étiqueté 1, brûlé et devient le nouveau centre d'attention. De cette façon, l'algorithme trouve aussi (3, 4) qu'il fait partie de l'objet 1 et finit par étudier le quatrième voisinage de (1, 3).

Tous les pixels qui font partie de l'objet supérieur sont maintenant étiquetés avec la même étiquette 1, ce qui signifie que ce blob est segmenté. Ensuite l'algorithme se déplace en suivant le parcours de balayage jusqu'à ce qu'il rencontre le pixel objet se trouvant a la position (2, 4), qui est alors étiqueté par 2, et déclenche un nouveau feu. Le résultat est représenté sur la Figure II.10, où chaque blob a une étiquette unique. Dans la Figure 2.4, les blobs de la Figur2.1 ont été extraites et codés par des couleurs selon leur étiquette. [19]

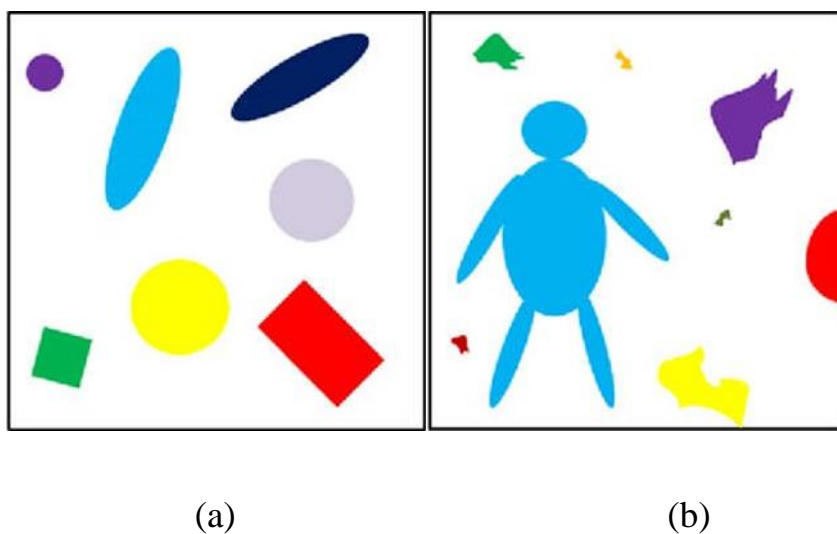


Figure 2.4 : Blobs extrait (chaque blob avec une couleur unique).

L'algorithme peut être mis en œuvre de manière très efficace par une fonction récursive. De tel algorithme requièrent des soins particuliers pour s'assurer que le programme se termine correctement car les algorithmes récursifs n'ont pas de stratégie de terminaison intégrée. Un autre inconvénient est celui de disposer d'un espace mémoire limitée allouée aux appels de fonctions. Car l'algorithme grassfire peut avoir plusieurs milliers d'appels de fonction mis en file d'attente .

2.2.2. L'algorithme Grassfire séquentiel

L'algorithme balaye l'image du haut gauche jusqu'en bas à droite. Quand un pixel d'objet est trouvé, il fait deux choses :

En premier lieu, dans l'image résultante il attribue à ce pixel objet une étiquette 1. En second lieu il "brûle" le pixel dans l'image initiale en la mettant à zéro (noir).

L'étape suivante consiste à vérifier les voisinages (4 ou 8 pixels en fonction de la connexité) et voir si l'un d'eux est un pixel d'objet. Jusqu'à présent, c'est exactement le même que l'algorithme Grassfire récursif. Sauf que, si l'un des voisinages est un pixel objet, il est étiqueté par 1 dans l'image résultante et mis à zéro (brûlé) dans l'image initiale. En outre, il est placé dans une liste.

La prochaine étape est de prendre le premier pixel dans la liste et de vérifier ses voisinages. Si parmi eux il existe des pixels d'objet alors ils seront étiquetés en sortie, mis à zéro à l'entrée et placé dans la liste. Ceci se poursuit jusqu'à ce que tous les pixels de la liste aient été examinés. L'algorithme poursuit le chemin de balayage jusqu'à ce qu'il rencontre le prochain pixel objet, qui est ensuite marqué par 2, et commence une nouvelle étape de grassfire. [20]

2.3. Caractéristiques d'un bon algorithme de suivi

Il ressort des techniques de suivi présentées précédemment des points important permettant de définir la qualité d'un algorithme de suivi.

Les caractéristiques d'un bon algorithme de suivi sont les suivantes :

- La méthode doit être capable d'initialiser automatiquement les cibles et doit gérer les arrêts et les sorties du champ de la caméra.
- Elle doit de plus être robuste aux changements d'illumination et aux éventuels changements de topologie. La topologie n'est importante que dans le cas où une segmentation de la cible est exigée.
- Enfin, la méthode doit permettre de continuer à suivre la cible même en cas d'occultations partielles ou totales par un autre objet ou par le fond.

La première caractéristique ne peut être obtenue qu'avec l'ajout d'une méthode de détection des objets, des observations étant apportées à chaque instant ou à des instants espacés par un faible pas de temps.

Les changements d'illumination ou de photométrie ne sont bien pris en compte que par les méthodes utilisant des histogrammes ou des mélanges de gaussiennes sur l'intensité ou la couleur. Ces distributions doivent régulièrement être mises à jour.

Enfin, la gestion des occultations n'est bien considérée qu'avec l'utilisation d'observations ou en ajoutant une loi dynamique au processus de suivi, la difficulté étant alors de déterminer cette loi. La plupart du temps les objets peuvent avoir un mouvement quelconque qu'aucune loi ne peut caractériser. [21]

2.3.1. L'Aire d'un blob (zone du blob)

C'est le nombre de pixels dont le blob se compose. Cette caractéristique est souvent utilisée pour éliminer les blobs qui sont trop petit ou trop grand de l'image. Par exemple, sur la Figure 2.4 (à droite) l'être humain peut être segmenté en disant simplement que tous les blobs avec une superficie inférieure à une certaine valeur sont ignorés.

2.3.2 Enveloppes de blob

a) Boîte enveloppante d'un blob

C'est le rectangle minimum qui contient le blob, voir Figure 2.5 (a). Il est définie en traitant tous les pixels pour un blob afin de trouver les quatre pixels ayant les valeurs minimales et maximales de x et de y . De ces valeurs, la largeur de la zone de délimitation est donnée par $X_{\max}-X_{\min}$ et par $Y_{\max}-Y_{\min}$.

b) Cercle enveloppant d'un blob

C'est le cercle minimum qui contient le blob Figure 2.5 (b). On localise d'abord le centre du blob avec l'une des méthodes décrites ci-dessous. On détermine ensuite le rayon en parcourant le blob de son centre jusqu'au point où le blob se termine. Nous faisons cela pour toutes les directions possibles (par exemple avec une résolution angulaire de 10°) et le plus grand rayon définit ainsi le rayon du cercle minimum qui enveloppe le blob.

c) Enveloppe convexe d'un blob

C'est le polygone convexe minimum qui contient le blob Figure 2.5 (c). C'est comme si on place un élastique autour du blob. Il peut être trouvé de la façon suivante. Du plus haut pixel de l'image on fait une recherche de pixel de blob vers la droite le long d'une ligne horizontale. S'il n'y a aucun pixel de blob trouvé on augmente l'angle de la ligne de recherche et on procède à une nouvelle recherche sur la ligne.

Quand un pixel de blob est trouvé la première ligne du polygone est définie et une nouvelle recherche est lancée sur la base de l'angle de la ligne de recherche précédente. Lorsque la recherche réapparaît au niveau du pixel le plus haute, l'enveloppe convexe est ainsi achevée. Notez que la morphologie peut également être appliquée pour trouver l'enveloppe convexe d'un blob. [22]

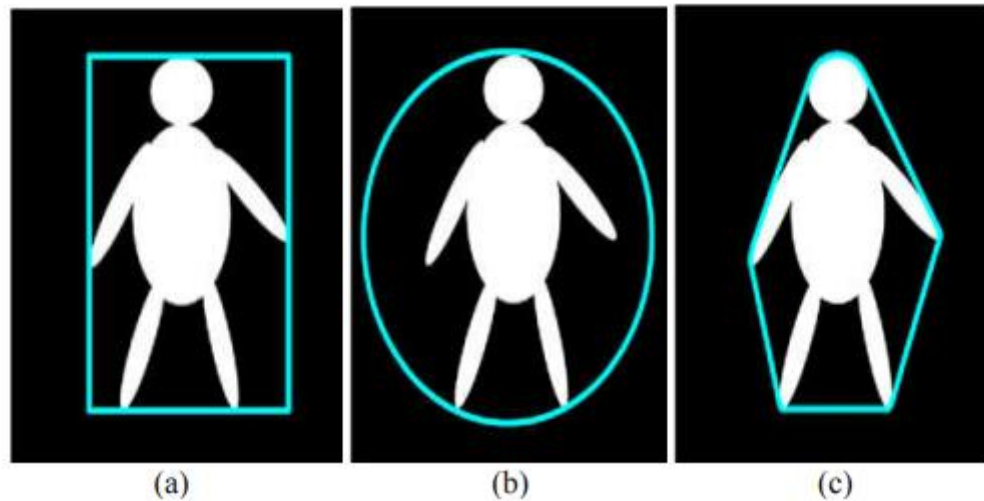


Figure 2.5 : différents types d' enveloppes de blob

- (a) boîte enveloppante
- (b) cercle enveloppant
- (c) enveloppe convexe

2.3.3. Coefficient de boîte enveloppante du blob

Il est défini comme étant la hauteur de la zone de délimitation divisée par sa largeur. Cette caractéristique indique l'allongement du blob, à savoir, si le blob est long, large, ou rectangulaire.

2.3.4. Compacité d'un blob

C'est défini comme étant le rapport de l'aire du blob sur la zone de la boîte enveloppante. Cela peut être utilisé pour distinguer les blobs compacts de ceux qui ne le sont pas. Par exemple, le poing d'une main et une main tendue

$$\text{Compacité} = \frac{\text{Zone de blob}}{\text{Largeur} \cdot \text{Hauteur}} \quad (1)$$

2.3.5. Centre de masse (centre de gravité ou Centroïde)

C'est la moyenne des positions x et y de l'objet binaire. Il est défini comme un point, dont la valeur x est calculée en additionnant les coordonnées x de tous les pixels dans le blob puis en divisant par le nombre total de pixels; de même que pour la valeur y . En terme mathématique, le centre de masse, (x_c, y_c) est calculé comme suit :

$$x_c, y_c = \frac{1}{N} \sum_{i=1}^N x_i, y_c = \frac{1}{N} \sum_{i=1}^N y_i \quad (2)$$

Où N est le nombre des pixels dans le blob et x et y sont les coordonnées des N pixels.

Dans les situations où le blob contient des «pièces jointes» la médiane peut remplacer l'équation 2.3 Un exemple pourrait être illustré dans la recherche de centre du torse d'une personne. Les configurations des bras affecteront le résultat de l'équation II.8, mais la médiane est moins affectée. La N médiane est plus exigeante en matière de calcul que le centre de masse. Une alternative à la médiane est d'éroder le blob avec un grand élément structurant, puis calculer le centre de masse.

Le centre de la boite enveloppante est une approximation plus rapide que celle du centre de masse. En termes mathématiques, le centre de la boite enveloppante, (x_{bb}, y_{bb}) est calculé comme suit

$$\begin{aligned} x_{bb} &= x_{min} + \frac{x_{max} - x_{min}}{2} = x_{min} + \frac{x_{max}}{2} - \frac{x_{min}}{2} = \frac{x_{max} + x_{min}}{2} \\ y_{bb} &= y_{min} + \frac{y_{max} - y_{min}}{2} = y_{min} + \frac{y_{max}}{2} - \frac{y_{min}}{2} = \frac{y_{max} + y_{min}}{2} \end{aligned} \quad (3)$$

Le périmètre d'un blob est la longueur du contour du blob. Cela peut être trouvé par un balayage le long du rebord (contour) d'un objet et en additionnant le nombre de pixels rencontrés. Une simple approximation du périmètre est de trouver d'abord la limite extérieure en utilisant un algorithme de détection de bord. Par la suite on compte simplement le nombre de pixels blancs dans l'image.

5 Conclusion

Dans ce chapitre nous avons détaillé la partie détection des blobs. Nous avons expliqué les différentes étapes de l'algorithme Grassfire utilisé communément pour l'analyse des blobs. Puis nous avons relaté quelques variantes de cet algorithme. Nous avons exposé par la suite quelques statistiques résultantes de l'analyse de blobs tels que les centres de gravités des blobs, leurs surfaces, ...etc.

Le prochain chapitre sera consacré à la mise en œuvre des notions théoriques des chapitres 1 et 2.

CHAPITRE 3

Tests et résultats

1. Introduction

Après avoir couvert les bases théoriques du traitement d'images et de vidéos ainsi que diverses techniques de détection et de suivi d'objets en mouvement dans les chapitres précédents, ce chapitre se concentre sur l'application pratique d'une de ces techniques : la détection et le suivi d'objets en temps réel basés sur leur couleur, en utilisant les algorithmes du programme MATLAB. MATLAB est choisi en raison de sa facilité de programmation grâce à ses commandes conviviales et de ses boîtes à outils spécialisées qui facilitent le traitement des images..

Dans la première partie de ce chapitre, nous présenterons un aperçu du matériel et des logiciels utilisés pour mener à bien cette application. Ensuite, nous aborderons les résultats de la détection et du suivi d'objets basés sur la couleur en temps réel. Enfin, nous conclurons en résumant les points clés et les contributions de cette étude.

Le choix de la détection et du suivi d'objets basés sur la couleur en temps réel repose sur la pertinence de cette technique dans de nombreux domaines tels que la surveillance vidéo, la robotique, la réalité augmentée, etc. En exploitant les propriétés distinctives de la couleur des objets, il est possible d'effectuer une détection et un suivi précis en temps réel.

En utilisant MATLAB, nous bénéficions d'un large éventail d'outils et de fonctions qui simplifient le processus de traitement d'images et de vidéos. Nous avons utilisé en particulier la boîte à outil « vision ».

Les résultats obtenus démontreront l'efficacité de la détection et du suivi d'objets basés sur la couleur en temps réel. Nous évaluerons les performances de l'algorithme en termes de robustesse face à diverses conditions d'éclairage et de fond.

En conclusion, ce chapitre fournira une mise en œuvre pratique des techniques de détection et de suivi d'objets basés sur la couleur en temps réel à l'aide de MATLAB.

2. Environnement du travail

2.1. Environnement matériel ou le hardware

- ✓ UN laptop :

Manufacturer :lenovo.

Model : AMD A4

Processor : AMD A4-9125 RADEON R3,4 COMPUTE CORES 2C+2G 2.30GHz

RAM : 4096MB

DisqueDur : 900 GO.

- ✓ Webcam intégrée: REDMI 9 PHONE CAMERA

Modèle :lenovo t61

Résolution : 1600 x 900 (HD+)

- ✓ Quelques objets colorés en Rouge, Vert, Bleu et Blanc (jouet, cartes, boites...etc.)

2.2. Environnement immatériel ou le software :

- ✓ System d'exploitation : Windows 8 professional N 64-bit
- ✓ Fabricant du system : LENOVO ELECTRONICS CO.,LTD.
- ✓ Carte graphique :intel® HD Graphics 3000.
- ✓ Logiciel : MATLAB R2013a

MATLAB est utilisé en raison de notre expérience avec son utilisation dans de nombreuses applications au cours de notre formation universitaire.

3. MATLAB

Est un logiciel développé par MathWorks, qui vise à faciliter les applications d'ingénierie et de mathématiques. Il permet à ses utilisateurs de manipuler des matrices de manière mathématique, de représenter graphiquement des fonctions mathématiques, d'implémenter divers algorithmes, de créer des interfaces graphiques conviviales, ainsi que de communiquer avec des programmes écrits dans d'autres langages comme C++ et Java. C'est

un outil polyvalent qui offre une large gamme de fonctionnalités pour répondre aux besoins de calculs et de visualisations dans divers domaines.

4. Implémentation de l'algorithme de détection et de suivi des couleursRVB-B

4.1. Les seuils de couleurs

- Rouge [100] = 0.25
- Vert [020] =0.05
- Bleu [003] =0.15
- Blanc [111] =0.75

La lecture du flux vidéo d'entrée RGB est réalisée par : `imaq.VideoDevice`.

4.2. Le cycle de prétraitement

Avant l'analyse de blob proprement dite, une étape de prétraitement est nécessaire. Chaque image du flux vidéo passe par les étapes de prétraitement suivantes. Fig.3.1. montre un exemple d'image couleur originale à traiter. L'image lue est corrigée par la commande `flipdim`.

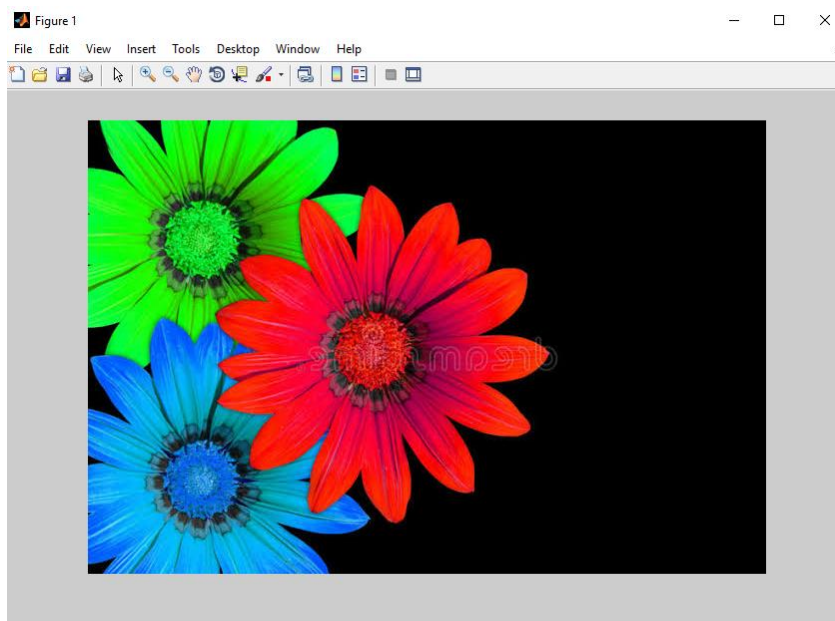


Figure. 3.1. Image originale

4.2.1. Extraction de la composante rouge

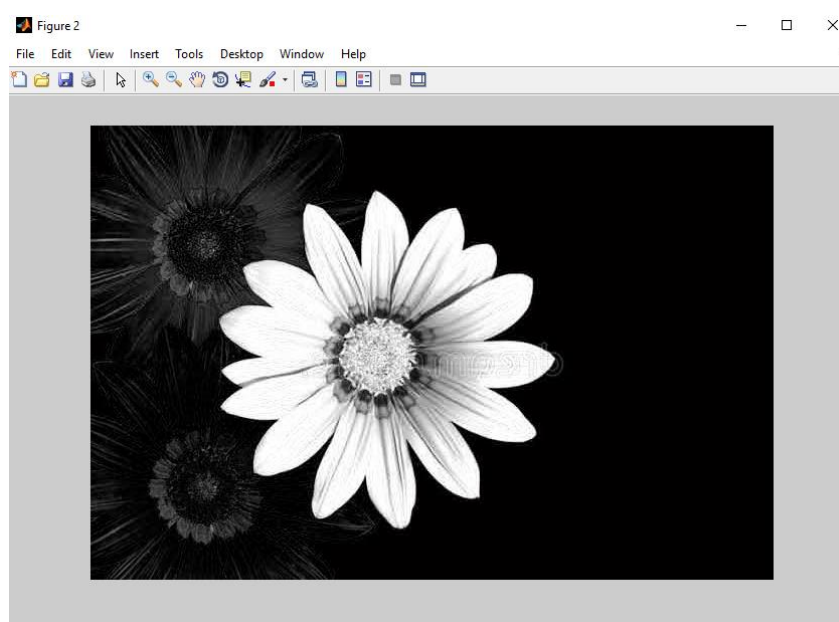


Figure. 3.2. Composante rouge de l'image

4.2.2. Obtention de la couleur rouge

Nous obtenons la composante de couleur à détecter à partir de l'image à l'aide de deux commandes, **rgb2gray** et **imsubtract**, où cette dernière calcule la différence entre l'image principale et l'image niveaux de gris (cf. Fig 3.3.). Le niveau de gris est obtenu par la fonction **rgb2gray**.

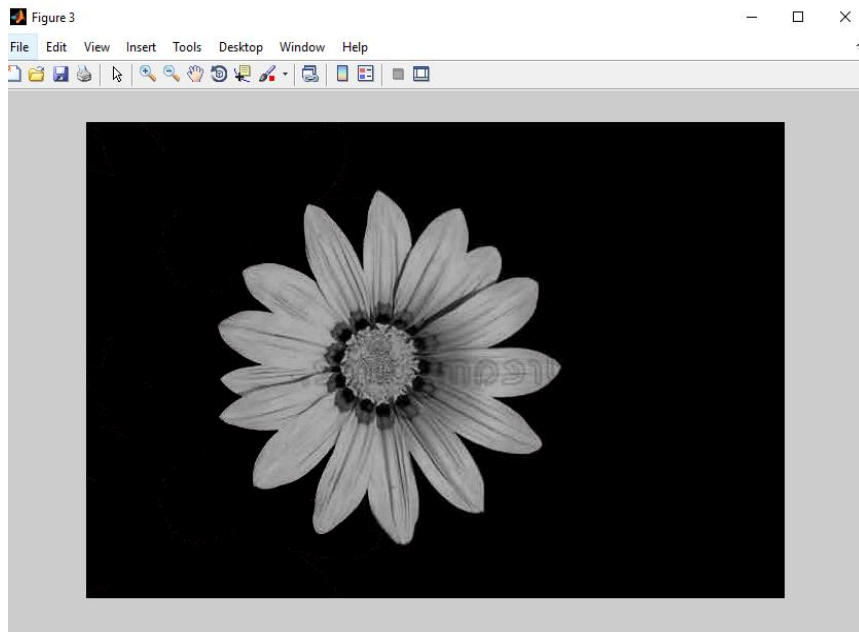


Figure. 3.3. Image différence entre la composante rouge et l'image niveau de gris

4.2.3. Filtrage médian

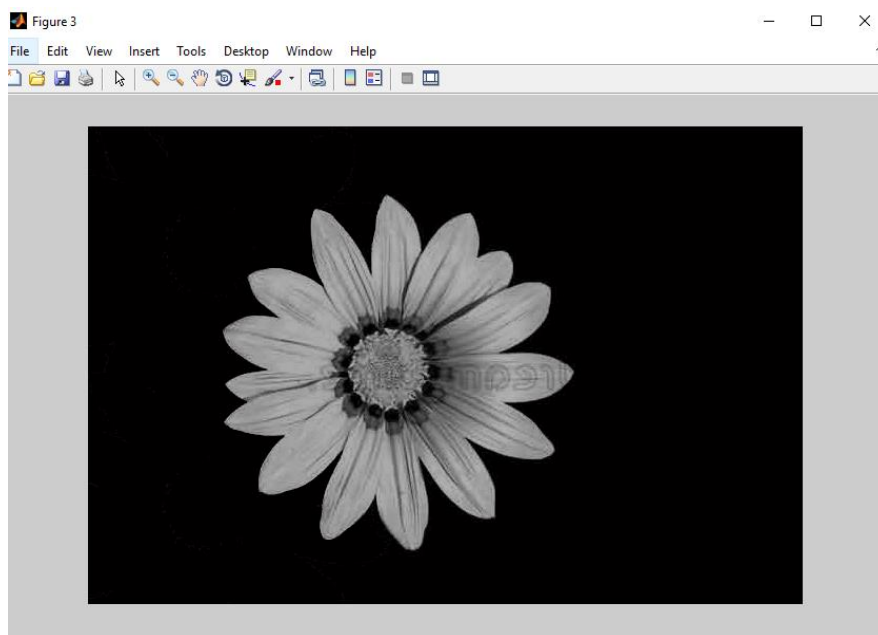


Figure.3.4. Image filtrée (Filtre médian de taille 3x3)

4.2.4. Binarisation

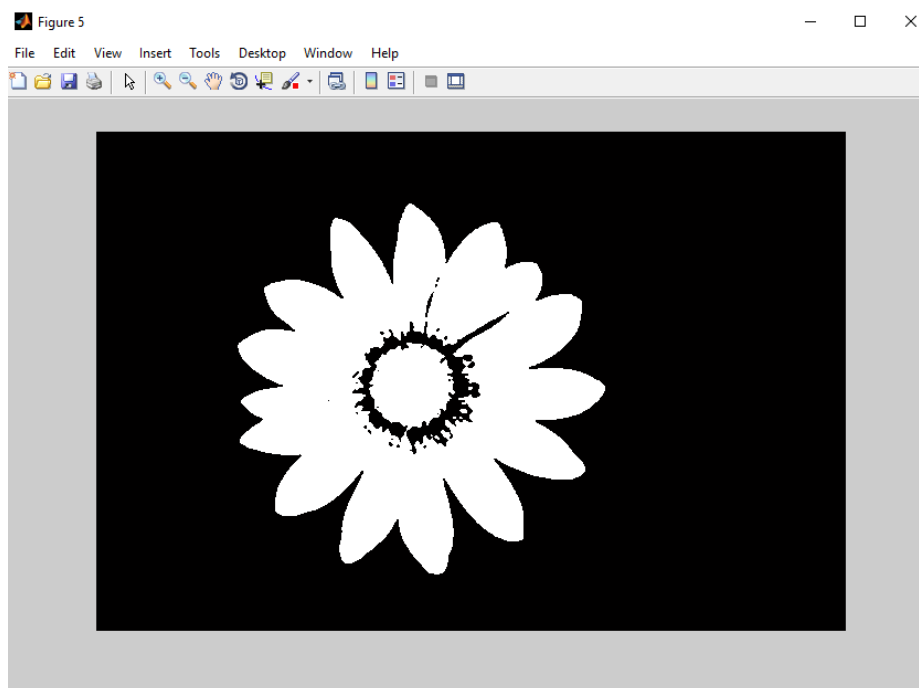


Figure. 3.5. Image binaire (Seuil =0.05)

4.3. Analyse de blobs

Pour la détection de blobs nous utilisons la fonction `vision.blobanalysis` de Matlab . Elle se fait selon le principe expliqué au chapitre 2 précédent.

L'utilisation de `vision.blobanalysis` nous permet de choisir entre autres le nombre de pixels voisins par la propriété `connectivity` de choisir la taille maximale et minimale du blob respectivement par les propriétés `MaximumBlobArea` et `MinimumBlobArea`.

5. Résultats pour la détection des couleurs

Dans cette partie, nous présentons les résultats de la détection d'objets colorés (Rouge, Vert, bleu et blanc), que nous avons obtenus à l'aide des algorithmes de détection décrits précédemment, en utilisant l'interface graphique conçue à cet effet.

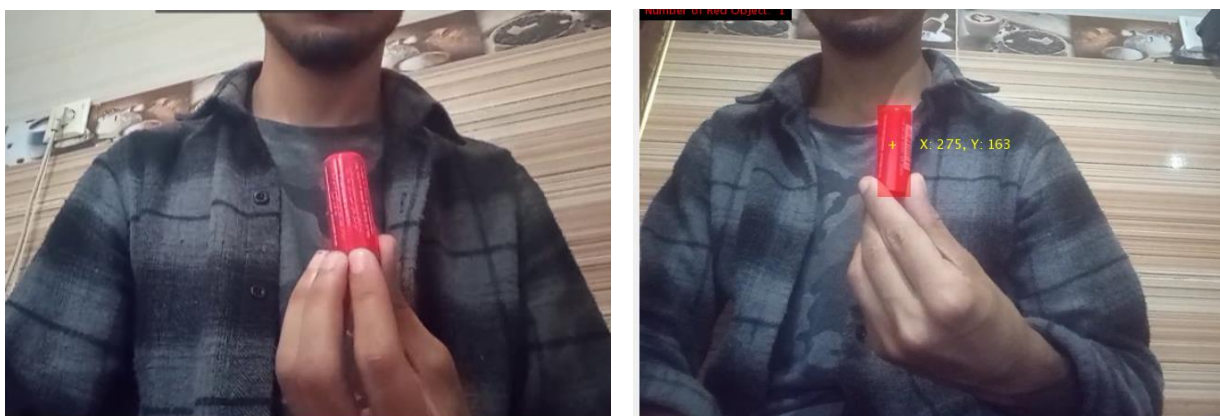


Figure.3.6. Détection d'un 1^{er} objet rouge



Figure.3.7. Détection de plusieurs objets rouges

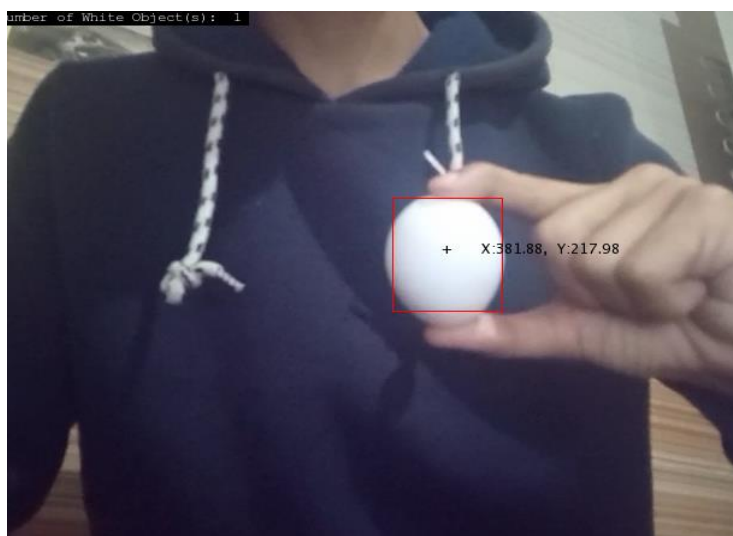


Figure 3.8. Détection d'un objet blanc

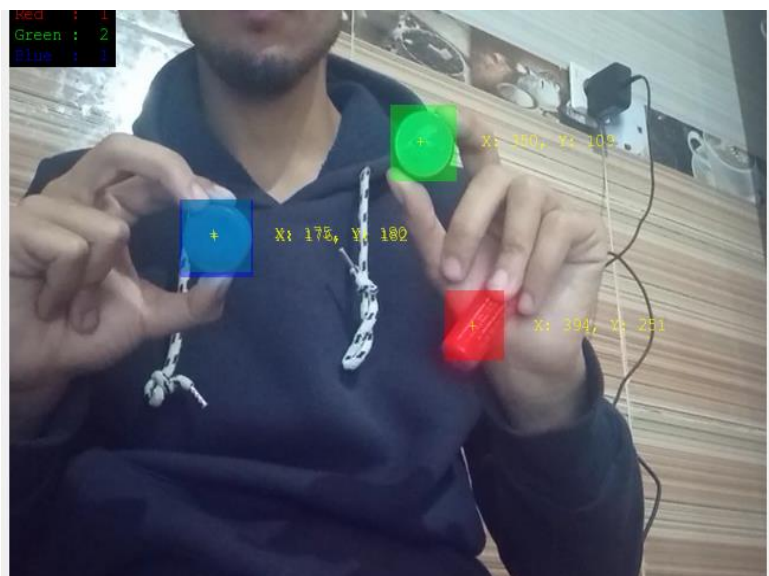


Figure.3.9. Détection des plusieurs objets de couleurs différentes

5.1. Effet du nombre des pixels connexes

Ainsi qu'il a été mentionné au chapitre 2, le nombre des pixels voisins pris en considération lors de la détection des blobs peut avoir un impact sur le résultat final de la détection. Les images de la figure 3.10 permettent de comparer les résultats de la détection faite avec 8 pixels voisins à ceux de la détection faite avec 4 pixels voisins.

Les images correspondent à deux carrés rouges faits au stylo rouge sur un papier blanc. Nous pouvons remarquer que les deux carrés sont considérés comme étant deux blobs séparés (étiquetés 1 et 2) dans le cas de 4 pixels connexes mais qu'ils sont assimilés à un seul blob dans le cas de pixels voisins.

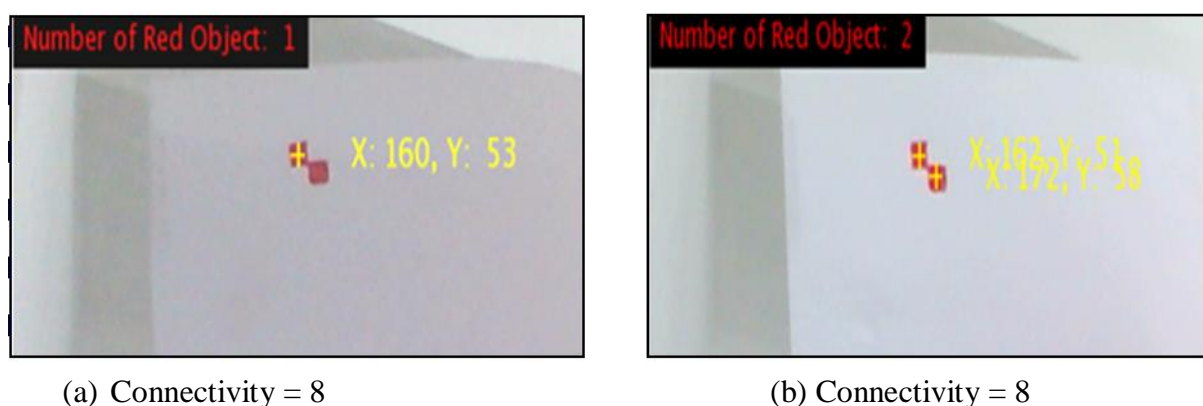


Figure3.10. Effet du voisinage sur le nombre de blobs détectés

5.2. Effet de la surface du blob

Sur un fond blanc (feuille blanche), des cercles rouges de tailles différentes sont dessinés. La feuille est mise respectivement à des distances croissantes de la caméra et le nombre des blobs détectés est mentionné (cf. Figure 3.11). Nous remarquons qu'au fur et à mesure que l'on s'éloigne de la caméra, les tailles deviennent plus petites et de moins en moins les taches sont repérées. En effet la fonction **vision Blob Analysis** permet de décider de la taille des blobs détectables moyennant ses paramètres : **Minimum Blob**

Area et **Maximum Blob Area**. Ces derniers déterminent respectivement la surface minimale et maximale en pixels du blob détectable.

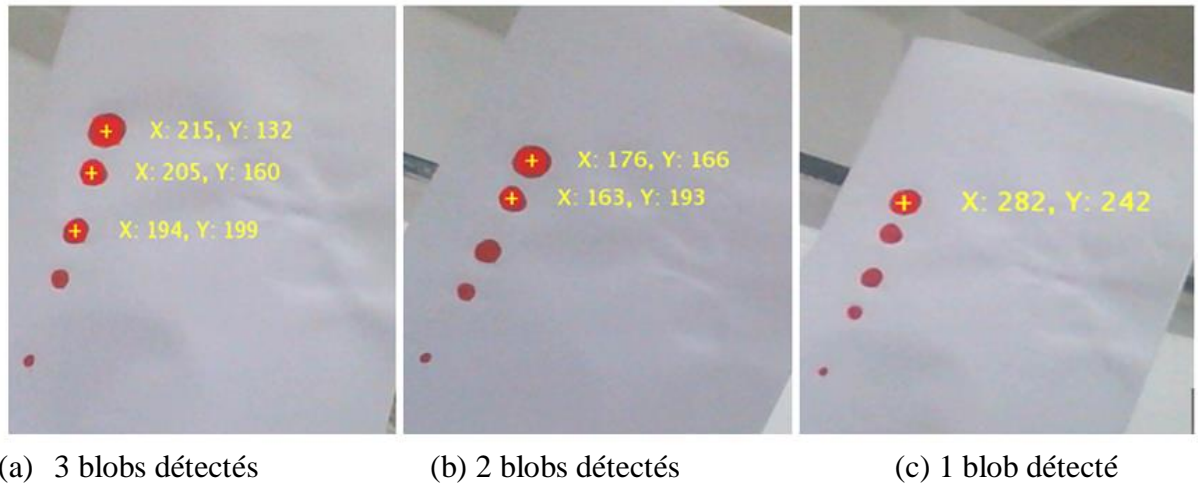


Figure3.11. Effet de la taille du blob détectable

6. Suivi d'objet

Des séquences vidéo live sont créées en lisant un nombre égal à **nframe** images successives. La spécification de **nframe** permet de décider de la durée de fonctionnement de la caméra. Dans notre cas nous avons utilisé **nframe =400**.

L'opération de suivi n'est par conséquent qu'une suite de détection de la couleur au niveau des images successives de la vidéo (cf. Figure III. 12).



Figure III. 12. Suivi d'un objet rouge sur une séquence vidéo

7. Discussion et interprétation des résultats

Les résultats obtenus le long de ce travail montrent que la méthode utilisée est efficace pour détecter les objets colorés dans les couleurs primaires (rouge, vert, bleu et blanc) individuellement et en temps réel. Les tests réalisés ont démontré une bonne précision en ce qui concerne le suivi, que ce soit pour un seul objet ou plusieurs objets simultanément.

Les seuils appropriés ont été choisis pour chaque couleur d'après la littérature, mais malgré cela, des erreurs de détection ont parfois été remarquées, notamment avec la couleur bleue. Ces erreurs peuvent être attribuées à plusieurs facteurs, tels qu'une mauvaise qualité de l'appareil photo, qui peut entraîner un léger flou et une erreur de calcul des vraies couleurs des pixels. Pour atténuer ce problème, un filtre moyen peut être utilisé. Néanmoins, il est toujours possible de rectifier ces seuils pour un ensemble bien déterminé d'images tests quand on connaît à priori les couleurs à traiter, en choisissant manuellement les zones colorées d'intérêt et en calculant le seuil correspondant.

La qualité de l'éclairage joue également un rôle crucial dans le bon fonctionnement de cette méthode. Un éclairage excessif peut provoquer des réflexions blanches sur les surfaces lisses, tandis qu'un éclairage insuffisant peut générer plus de bruit dans la scène et des zones sombres ou noires.

En résumé, les résultats obtenus démontrent l'efficacité de la méthode pour détecter les objets colorés dans les couleurs primaires, mais des erreurs de détection peuvent survenir quand on ne connaît pas à priori la nature et le spectre de couleurs des images à traiter.

8. Conclusion

En conclusion, ce chapitre a présenté les différentes étapes et commandes des algorithmes utilisés dans l'application de détection et de suivi des objets colorés (RVB.B).

Le point fort de la méthode utilisée réside dans le fait qu'il nous a suffi d'utiliser l'analyse des blobs pour la détection et aussi pour le suivi sans devoir utiliser un autre algorithme tel que par exemple le filtre de Kalman, communément utilisé pour le suivi.

Les résultats obtenus ont illustré l'efficacité de l'approche adoptée, ce qui démontre la puissance de la couleur comme primitive et porteur d'information.

Conclusion générale

En conclusion, ce mémoire a appliqué l'analyse des blobs à la détection et le suivi d'objets mobiles dans des scènes vidéo en temps réel, en se concentrant sur des caméras fixes avec un arrière-plan fixe. Dès lors que l'arrière plan est éliminé et que l'image est binarisée, l'analyse des blobs est appliquée. Elle exploite la connexité des pixels à traiter pour détecter et étiqueter les différents objets en mouvement.

La couleur est utilisée pour caractériser les objets à détecter et à suivre. Les algorithmes de reconnaissance des couleurs ont été choisis pour ce travail en raison de leur simplicité et de leur praticité. Cependant, ils ont une limitation majeure, qui est de reconnaître uniquement les objets d'une seule couleur. Malgré cette limitation, l'utilisation des couleurs primaires rouge, vert et bleu a permis de détecter correctement la région désirée et de réaliser un suivi précis des séquences en temps réel.

MATLAB a été utilisé comme langage de programmation pour développer les codes de détection et de suivi. Il offre une interface conviviale et une bibliothèque riche en fonctions utiles pour le traitement d'images et de vidéos. Nous avons pu tirer profit de sa puissante boîte à outil 'vision' et sa fonction 'BlobAnalysis'.

Cette étude a confirmé que la couleur est une méthode efficace pour la détection et le suivi des objets en mouvement. Les résultats obtenus ont démontré l'efficacité de la méthode choisie.

Les principaux points forts de cette méthode résident dans sa simplicité par rapport aux autres méthodes et sa capacité à être appliquée à tout type d'image et de vidéo, indépendamment de la nature de l'objet suivi.

Il serait intéressant de mener une étude comparative entre différentes méthodes de détection d'objets en mouvement afin de comprendre leurs performances réelles dans le domaine de la vision par ordinateur. Cela permettrait d'évaluer les avantages et les inconvénients de chaque méthode et d'identifier les situations où elles peuvent être les plus appropriées.

Bibliographie

- [1] M. Sandeli, “Traitement d’images Par des approches bio-inspirées application à la segmentation d’images”, Université Constantine 2, 2014.
- [2] Z. Said, A. Mokrani et M. Zeghdoud, “Compression d’image hyperspectrales par ondelettes”, Doctoral dissertation, Université Mouloud Mammeri, 2010.
- [3] Encyclopédie Microsoft® Encarta® 2002. © 1993-2001 Microsoft Corporation, Cd-Rom.
- [4] M. Hadallah, “Codage des images fixes par une méthode hybride basée sur la QV et les approximations fractales” , Pfe Usthb, 1997.
- [5] K. Chakib et A. B. Salim, “Compression d’images fixes par fractales basée sur la triangulation Delaunay et la quantification vectorielle”, Mémoire de fin d’étude, Institut National d’Informatique (INI), Alger, 1999.
- [6] N. Morceau, “ Techniques de compression des signaux ”, Edition Masson, 1995.
- [7] A.C. Houassine, “segmentation d’images par Une approche biomimétique hybride”, Université Mohamed Bougara- Boumerdes, 2012.
- [8] [http://edutechwiki.unige.ch/fr/Image vectorielle](http://edutechwiki.unige.ch/fr/Image_vectorielle)
- [9] C. W. Paeth, “Image File Compression Made Easy”, in Graphics Gems II, James Arvo, editor. Academic Press, San Diego, 1991.
- [10] Z. N. Li and M. S. Drew, “Fundamentals of Multimedia”, Pearson Education, United States of America, 2006.
- [11] https://fr.wikipedia.org/wiki/Graphics_Interchange_Format.
- [12] <http://www.futura-sciences.com/tech/definitions/informatique-jpeg-1232>.
- [13] D. Lingrand , “Introduction aux traitement d’images” , Vuibert, 2008.
- [14] L. Khaled, “Tatouage numérique des images dans le domaine des ondelettes

-
- basé sur la décomposition en valeurs singulières et l'optimisation multi-objective”, Mémoire de Doctorat, Université Laval Québec, 2010.
- [15] W. Bendjeride et W. Boureba, “Compression d’image couleur par ondelettes à base de la structure Lifting : Application aux images satellitaires ”, Mémoire de Master, Univ Belhadj Bouchaib d’Ain-Temouchent, 2016.
- [16] P. Vincent, L. Franck and H. Vincent, “Compression des Données”, Exposé, 2004.
- [17] MEDJAHED Fatiha, " Détection et Suivi d’Objets en Mouvement Dans Une Séquence d’Images ", Université Mohamed Boudiaf Oran, 2011.
- [18] Ahmed Ben Atitallah, "Etude et Implantation d’Algorithmes de Compression d’Images dans un Environnement Mixte Matériel et Logiciel", Université Bordeaux 1, Juillet 2007.
- [19] Introduction to Video and Image Processing (what-when-how.com)Blob detection - Wikipedia
- [20]
- [19] What-when-how.com “Introduction to video and image processing” part1 blob analysis.
- [20] https://en.wikipedia.org/wiki/Blob_detection
- [21] Maradj Mohammed, « Extraction d’Objets en Mouvement dans les Séquences Vidéo par Mélange de Gaussiennes », Mémoire de Magister, Dpt. Génie Electrique, Université des Sciences et de la Technologie d’Oran Mohamed Boudiaf, 2018.
- [22] Yedjour Hayat, « Détection de contours et suivi d’objet dans une séquence d’images par les réseaux de neurones impulsionnels »,Mémoire de Magister, Dpt. Informatique, Université Mohamed Boudiaf (USTO) ORAN 2010.
- [23] <https://medium.com/image-processing-in-robotics/blob-detection-309226a3ea5b>