

Introduction générale

L'œil occupe une place privilégiée chez les êtres vivants. Il n'est pas étonnant que les scientifiques essaient de le reproduire artificiellement à l'aide de caméras, d'ordinateurs et d'algorithmes. La vision par ordinateur est un des axes majeurs utilisés pour la robotique et pour toutes les applications d'analyse et de traitement d'images.

Le rôle de la vision est d'apporter une perception dynamique de l'environnement évolutif ou non. Le robot évolue dans un environnement dont il fait partie et doit sans cesse l'observer et en relater les différents changements. Grâce au progrès de l'électronique et de l'informatique, la vision artificielle s'est considérablement améliorée au point de devenir un capteur intelligent.

Le domaine de la vision artificielle est vaste. Bon nombre de corps de métier s'y intéressent de près comme de loin, cela va des militaires, à la médecine afin d'assister les chirurgiens dans les opérations de microchirurgie ou de la reconstruction anatomique du corps et de la robotique mobile.

Quel que soit le domaine d'application, le principe général est identique. A partir d'une image ou séquence d'images, sont analysées des informations issues des scènes tridimensionnelles, Afin d'informer le système et de l'aider à la prise de décision dans le cas d'une intelligence artificielle.

La vision artificielle est aussi beaucoup étudiée et utilisée pour les systèmes artificiels de navigation, en particulier pour les véhicules dits intelligents, c'est-à-dire présentant un certain degré d'autonomie. La capacité d'un tel véhicule à se localiser et à se mouvoir dépend en grande partie de la connaissance de son environnement. La vision artificielle ou vision par ordinateur s'est avéré un moyen privilégié.

L'étude menée dans ce projet, qui a été conduite par notre promoteur est basée sur la conception et le développement d'un module de reconnaissance des panneaux de signalisation routières.

Les panneaux routiers sont acquis à partir d'une voiture qui se déplace le plus souvent sur des routes chaotiques et à des vitesses variables et autorisées pour un cycle urbain. La principale difficulté rencontrée, est d'essayer de minimiser l'impact des vibrations sur l'image à analyser.

La nature et le type de panneaux routiers, sans occlusion, sont analysés à partir de leur couleur.

La méthode de reconnaissance utilisée est celle du k plus proche voisin (KPPV) appliqué sur les contours des panneaux.

La reconnaissance d'objets est un sujet particulièrement relativement difficile au problème de la détection, en raison notamment de la variabilité d'apparence, et des situations possibles.

Ce mémoire est organisé en trois chapitres : Dans le premier, on présente des généralités sur la vision artificielle, ainsi que quelques outils de traitement d'image utilisés dans la conception de notre application. En deuxième chapitre on expose l'état de l'art des méthodes de reconnaissance de panneau routier, en précisant que ce type de problème est d'actualité. Une étude comparative a été effectuée, afin de pouvoir sélectionner les méthodes répondant au mieux aux contraintes imposées dans ce travail. Le dernier chapitre a été réservé à l'étude puis l'implémentation des algorithmes choisis sous/C++ builder, en utilisant la librairie OpenCV,

Chapitre I Généralités sur la vision artificielle

I.1. Introduction

C'est, entre autres, discerner et reconnaître les formes, les couleurs et les textures du monde qui nous entoure.

On pourrait croire, à tort, que nous avons besoin uniquement de nos yeux pour Cela, mais la réalité est tout autre : nos yeux ne servent qu'à capter le signal contenu dans les rayons lumineux qui viennent frapper nos rétines, pas à en extraire des informations. C'est le travail de la zone de notre cerveau que nous appelons le cortex visuel, cela veut dire que la vision, c'est l'association entre le sens de la vue et un ensemble de processus cognitifs pour lesquels nous avons besoin d'utiliser notre cerveau.

Aujourd'hui, cela fait déjà bien longtemps que nous avons donné le sens de la vue aux ordinateurs (et notamment aux robots) grâce aux caméras numériques, mais si l'on se fie à la définition que nous venons de formuler, cela ne suffit pas à les doter de vision : il leur manque encore la capacité à extraire des informations des images et des flux vidéo, de manière à percevoir, analyser et comprendre le monde qu'ils observent.

C'est ici qu'intervient ce vaste domaine de recherche qu'est la vision par ordinateur, à la croisée des chemins entre les mathématiques, le traitement du signal et l'intelligence artificielle.

La vision par ordinateur se base essentiellement sur le traitement d'image pour effectuer des tâches comme la segmentation d'objets, la reconnaissance de formes, la

détection du mouvement, le suivi de cibles et autres techniques. Pour pouvoir effectuer ces tâches, on utilise des outils qu'on appelle : outils de traitement d'images, donc on parlera de quelques notions de base de traitement d'image, plus précisément les outils qu'on utilise au cours de notre projet (reconnaissance de panneaux routier).

I.2 .Vision Assisté par Ordinateur « VAO »

La vision nous permet de percevoir et d'interpréter le monde qui nous entoure. La VAO est un ensemble d'outils qui permet à l'ordinateur d'imiter la perception humaine afin d'extraire des informations d'une image brute pour pouvoir prendre des décisions. C'est un problème difficile en raison du fait que l'information disponible : des images 2D fournies par des capteurs (CCD, CMOS etc.), correspondent à une projection du monde 3D. La projection 3D-2D entraîne une perte d'informations importante, de plus, l'information disponible n'est pas parfaite (numérisation des capteurs, déformation des objectifs, bruitages). Nous pouvons alors dire que la VAO constitue une chaîne de traitements allant de l'acquisition de l'image brute jusqu'à son interprétation par la machine [1]

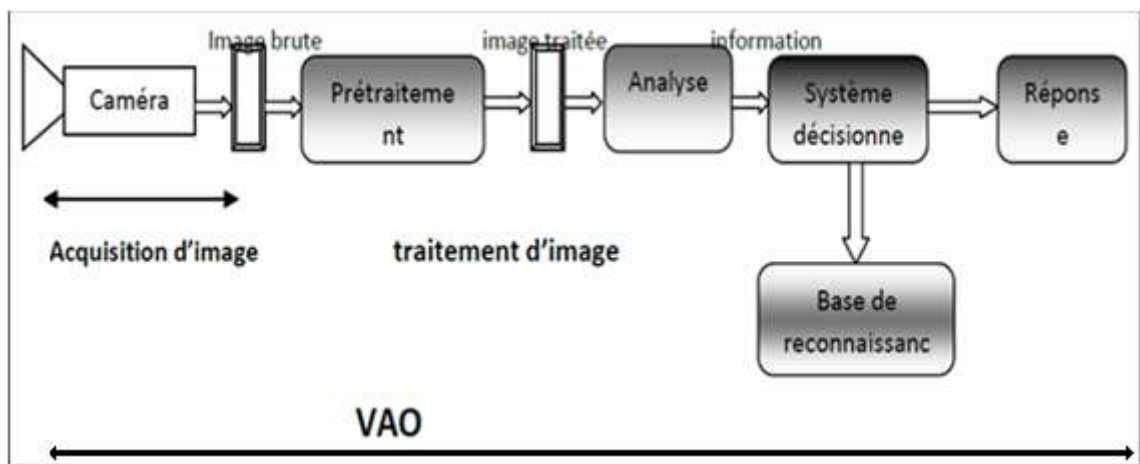


Figure I.1 : la chaîne de la VAO.

I.2.1. Domaines d'applications VAO

Contrôle de présence/absence : Sur des chaînes de production, on vérifie en bout de chaîne avec une caméra vidéo la présence d'une pièce dans un ensemble plus Complexe.

- Aide à la conduite de voiture. (Voir **figure I.2**).
- Construction et correction de cartes géographiques d'après des images satellites ou des images aériennes. (Voir **figure I.3**).
- Surveillance et évaluation de la production agricole. Il est possible de déterminer le degré de maturation des cultures, la quantité d'eau nécessaire pour l'irrigation, le rendement moyen etc.
- Reconnaissance de l'écriture.
- Reconnaissance des matricules. (Voir **figure I.4**).
- Recherche d'image par le contenu.
- Vérification et identification d'empreinte digitale. (Voir **figure I.5**).
- Analyse de la vidéo.
- Segmentation et suivi de cellules vivantes en microscopie.
- En robotique : segmentation de l'environnement, détection d'obstacles, détection, reconnaissance et suivi de cibles etc.



Figure I.2 : détection de ligne de route



Figure I.3 schéma d'une carte d'une ville



Figure 1.4 : *Reconnaissance des matricules*

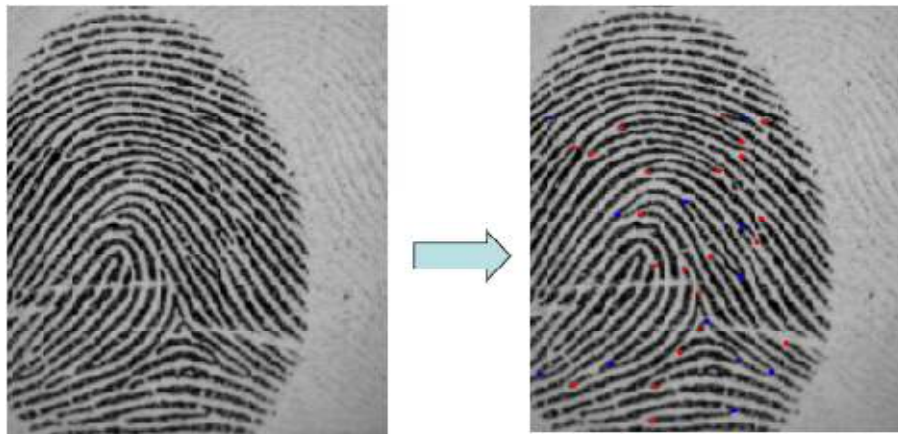


Figure 1.5 : *Vérification et identification d’empreinte digitale*

1.2.2. Image numérique

Le terme image numérique désigne, dans son sens le plus général, toute image qui a été acquise, traitée et sauvegardée sous une forme codée représentable par des nombres (valeurs numériques).

La numérisation est le processus qui permet de passer de l’état d’image physique (tel que l’image optique) qui est caractérisée par l’aspect continu du signal qu’elle représente (une infinité de valeurs dans l’intensité lumineuse par exemple), à l’état d’image numérique qui est caractérisée par l’aspect discret (l’intensité lumineuse ne peut prendre que des valeurs quantifiées en un nombre fini de points distincts). C’est cette forme numérique qui permet une exploitation ultérieure par des outils logiciels sur ordinateur [2].

L’image en pixels se présente sous forme d’une grille constituée d’une multitude de points, comparable à une mosaïque : c’est ce qu’on appelle les pixels.

1.2 .3 Les plans de l'image couleur

- **le modèle RGB:** Dans le modèle RGB, une couleur est définie par la part qu'elle contient dans chacune des couleurs fondamentales Rouge, Green (vert) et Bleu . Chaque couleur a donc une valeur pour le rouge, une valeur pour le vert et une valeur pour le bleu. Chacune des trois valeurs est définie par des nombres entre 0 et 255. La valeur 0 signifie: aucune part de la couleur fondamentale concernée, la valeur 255 signifie: la part maximale de la couleur fondamentale concernée. Un bleu foncé a, d'après ce schéma par exemple, la valeur de couleur 0, 0,153. ((0 rouge, 0 vert, 153 bleu). D'après ce schéma, jusqu'à 16,7 million de couleurs différentes peuvent être définies. **Figure 1.6.**

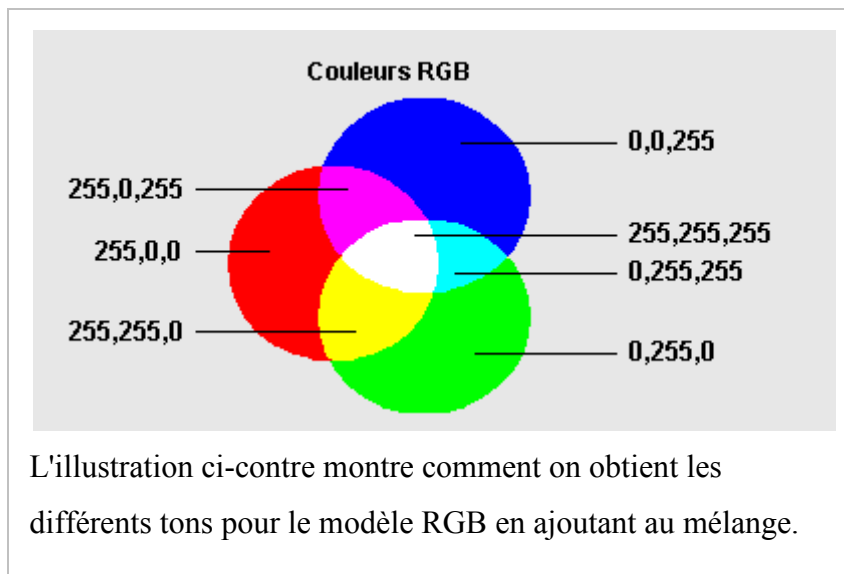


Figure 1.6. Les plans RGB

- **Le modèle HSB:** Même pour ce modèle, il y a trois valeurs numériques à définir. **H** figure pour *Hue (tonalité)*, **S** pour *Saturation (saturation)*, et **B** pour *Brightness (luminosité)*. Les coloris peuvent accepter des valeurs entre 0 et 359, la saturation et la luminosité des valeurs entre 0 et 255. Il est ainsi possible de définir environ 23,6 millions de couleurs. **Figure 1.7.**

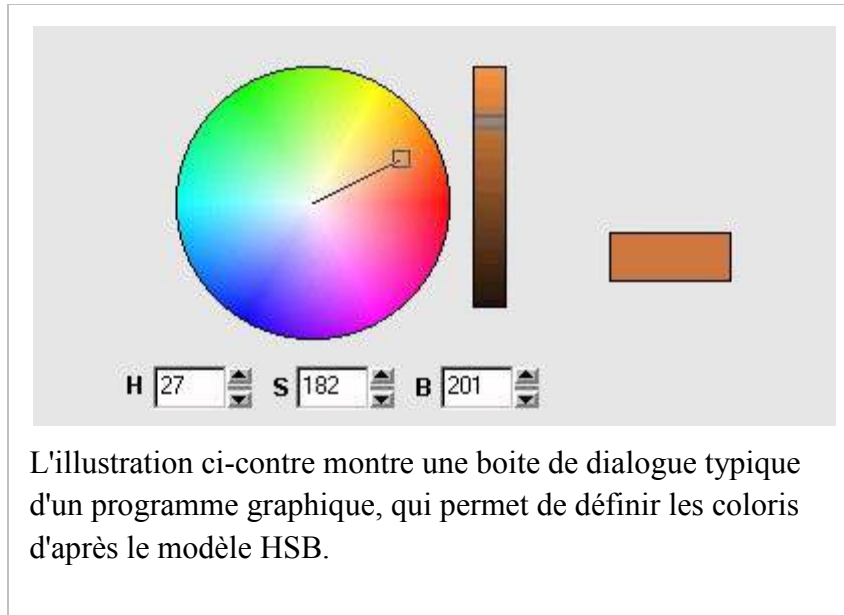


Figure1.7. les plans HSV.

I.2.4.Types d'image

a- Image à niveau de gris

Le niveau de gris est la valeur de l'intensité lumineuse en un point, celle-ci varie de la lumière vers l'absence totale de la lumière. Une image en niveau de gris est une image composée de points gris plus ou moins foncés. Pour chaque point, l'ordinateur enregistre une valeur de gris entre le noir et le blanc. Par convention la valeur '0' représente le noir (intensité lumineuse nulle) et la valeur 255 le blanc (l'intensité lumineuse maximale), chaque pixel étant représenté par un octet, (en général on sauvegarde les images à 256 teintes de gris)[3]

Couleur	BLANC	GRIS 12.5%	GRIS 25%	GRIS 37.5%	GRIS 50%	GRIS 62.5%	GRIS 75%	GRIS 87.5%	NOIR
Niveau de gris	255	224	192	160	128	96	64	32	0

Tableau I.1 les niveaux de gris

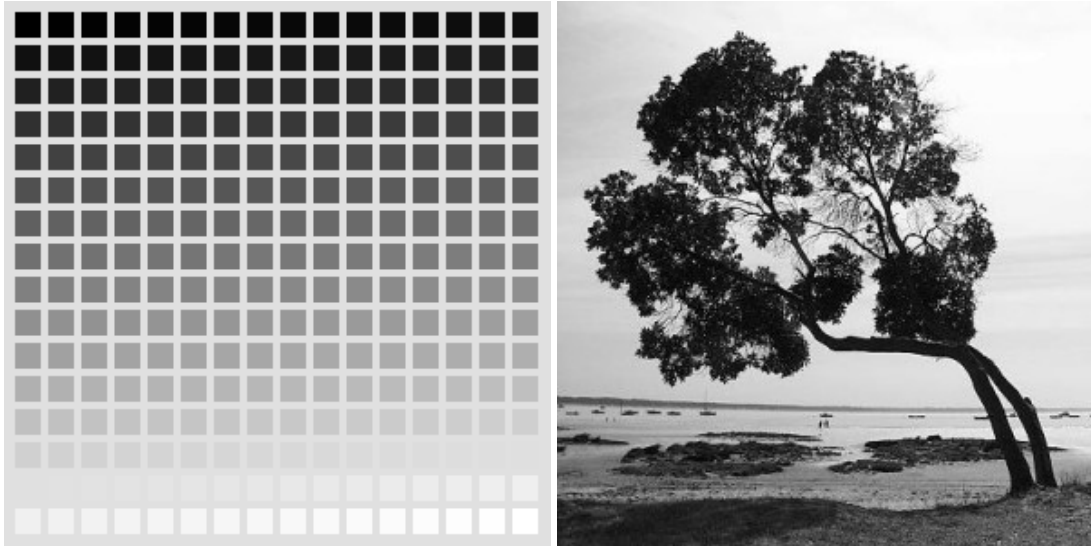


Figure 1.8. Image niveau de gris.

b- Image binaire

Parmi les images numériques, les images binaires sont les plus simples. Bichromes (la plupart du temps noire et blanche) elles sont ontologiquement numériques c'est-à-dire que leur codage et leur décodage peuvent être faits directement vers la base 2.

c- Image Truecolor

Image Truecolor, ou image RGB, représente directement les valeurs de couleurs, au lieu d'être exprimée à travers une palette de couleurs ou colormap. Une image truecolor est un tableau de dimension $m*n*3$, avec $m*n$ les dimensions de l'image et 3 représente les composantes de couleurs R, V et B. Donc pour chaque pixel (x, y) de l'image, la couleur est représentée par le triplet $(x, y, 1:3)$.

R	V	B	Couleur
0	0	0	Noir
255	0	0	Rouge
0	255	0	Vert
0	0	255	Bleu
128	128	128	Gris
255	255	255	Blanc

Tableau I.2 : Les couleurs RVB.

I.3.Traitement d'image

Le traitement d'images désigne une discipline de l'informatique et des mathématiques appliquées qui étudie les images numériques et leurs transformations, dans le but d'améliorer leur qualité ou d'en extraire de l'information. Il n'y a pas création d'informations, mais mise en évidence de l'information pertinente déjà présente.

I.3.1-Histogramme

Un histogramme est un graphique statistique permettant de représenter la distribution des intensités des pixels d'une image, c'est-à-dire le nombre de pixels pour chaque intensité lumineuse, il représente le niveau d'intensité en abscisse en allant du plus foncé au plus clair.

Ainsi, l'histogramme d'une image en 256 niveaux de gris sera représenté par un graphique possédant 256 valeurs en abscisses, et le nombre de pixels de l'image en ordonnées.

L'histogramme est le premier outil permettant de se rendre compte de la qualité intrinsèque d'une image, qui permet de voir la qualité d'information en fonction du niveau de densité et de vérifier si l'image présente des détails dans les ombres et dans les hautes lumières [4].

Pour les images en couleur plusieurs histogrammes sont nécessaires. Par exemple pour une image codée en RGB, on a un histogramme représentant la distribution de la luminance et trois histogrammes représentant respectivement la distribution des valeurs respectives des composantes rouges, bleues et vertes. [5]

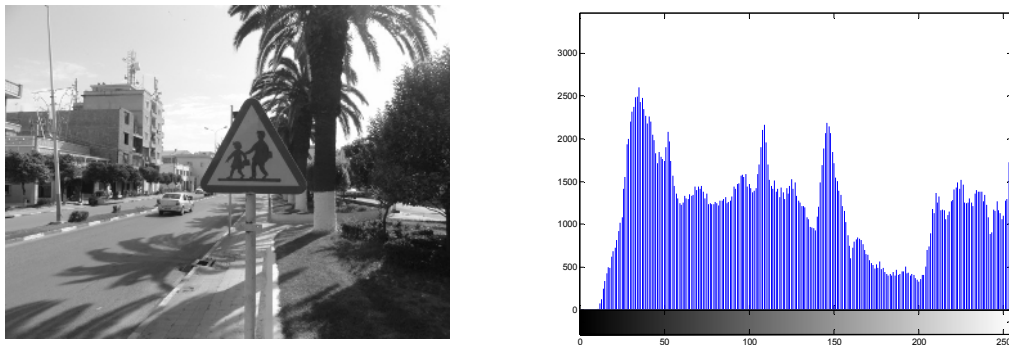


Figure I.9.Image d'un panneau et son Histogramme

a-Egalisation d'histogramme

Elle a pour but d'harmoniser la répartition des niveaux de luminosité de l'image, de telle manière à tendre vers un même nombre de pixel pour chacun des niveaux de l'histogramme. Cette opération vise à augmenter les nuances dans l'image [5] voir **Figure I.10.**

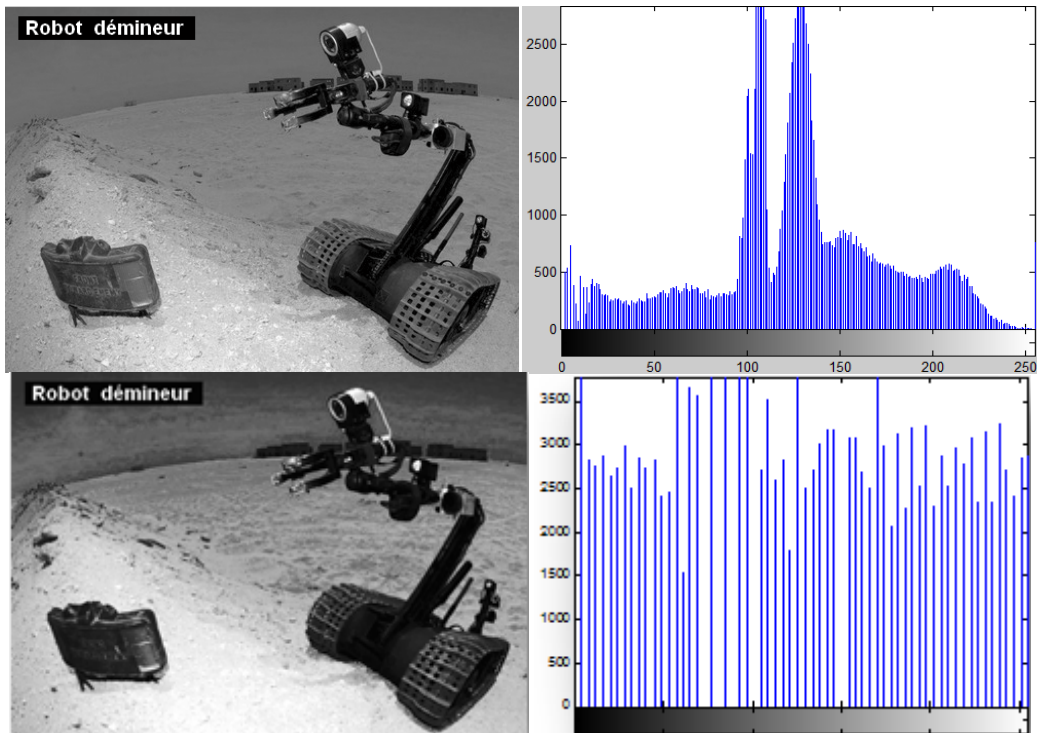


Figure I.10. Image d'un robot a niveau de gris avant et après égalisation

b-Rehaussement de contraste

Ou étirement de l'histogramme, on l'appelle aussi expansion de la dynamique, il consiste à recadrer l'image en étalant les fréquences d'apparition des pixels sur la largeur de l'histogramme, afin que la plus faible valeur soit à 0 et la plus haute soit à 255 (maximum).



Figure I.11. Image de panneau en couleur avant et après rehaussement de contraste

I.3.2- Seuillage

Une image numérique, pour pouvoir être exploitée a généralement besoin d'être simplifiée, c'est le but du seuillage. Ce dernier consiste à transformer l'image codée sur 6, 8 ou 16 bits, en une image binaire ou les pixels à 1 correspondent aux objets et les pixels à 0 au fond de l'image. A la différence des autres traitements de l'image numérique, le seuillage est un passage obligatoire pour toute analyse morphologique ultérieure. Le seuillage permet de sélectionner les parties de l'image qui intéressent l'opérateur, par exemple 2 types de grains (blancs et sombres) dans un mélange. On peut donc, par exemple, attribuer à tous les pixels de l'image numérique qui ont un niveau de gris compris entre deux valeurs i_1 et i_2 , choisies par l'opérateur, la valeur 1; à tous les autres pixels est attribuée la valeur 0. Après seuillage, les parties de l'image sélectionnées seront traduites en noir et blanc. L'image, digitalisée par l'ordinateur (0 et 1), est appelée image binaire. Cette dernière, tout comme l'image numérique contient des informations superfétatoires, qu'il convient d'éviter, ou masquées qu'il faut révéler [6]



Figure I.12 : Image d'un panneau binarisée

I.3.3- Segmentation

La segmentation, consiste à Identifier et définir des objets dans l'image en divisant l'image en zones homogènes afin de séparer les divers composants visibles et de les

identifier. Deux approches sont nécessaires pour la réalisation de la segmentation, regroupement des pixels présentant une caractéristique commune, dérivée par exemple de l'intensité des pixels (région) et mise en place des frontières aux positions qui rendent localement maximale la variation d'un critère (contour). Ces deux conceptions sont duales : une région définit son contour, un contour définit une région [7]

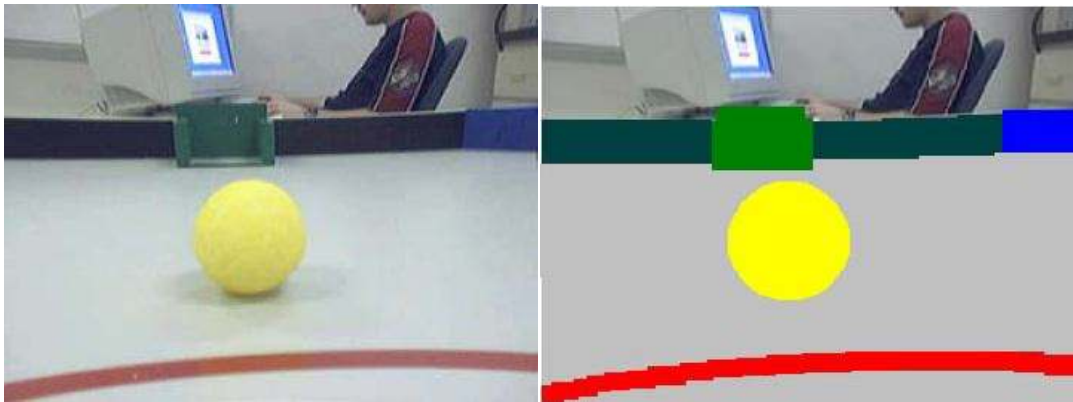


Figure I.13.Image prise d'une caméra embarquée sur un robot segmentée

➤ **Détection de contours**

La détection est basée sur la dérivation selon les deux coordonnées. Si on considère classiquement les signaux comme des sommes de sinusoïdes, la dérivation apparaît comme un filtre passe-haut qui introduit donc du bruit à l'origine de faux contours. Pour l'amateur il est recommandé, avant d'utiliser un filtre simple, d'atténuer ce bruit par passage dans un filtre flou. Des méthodes plus élaborées ont été systématisées pour les professionnels.

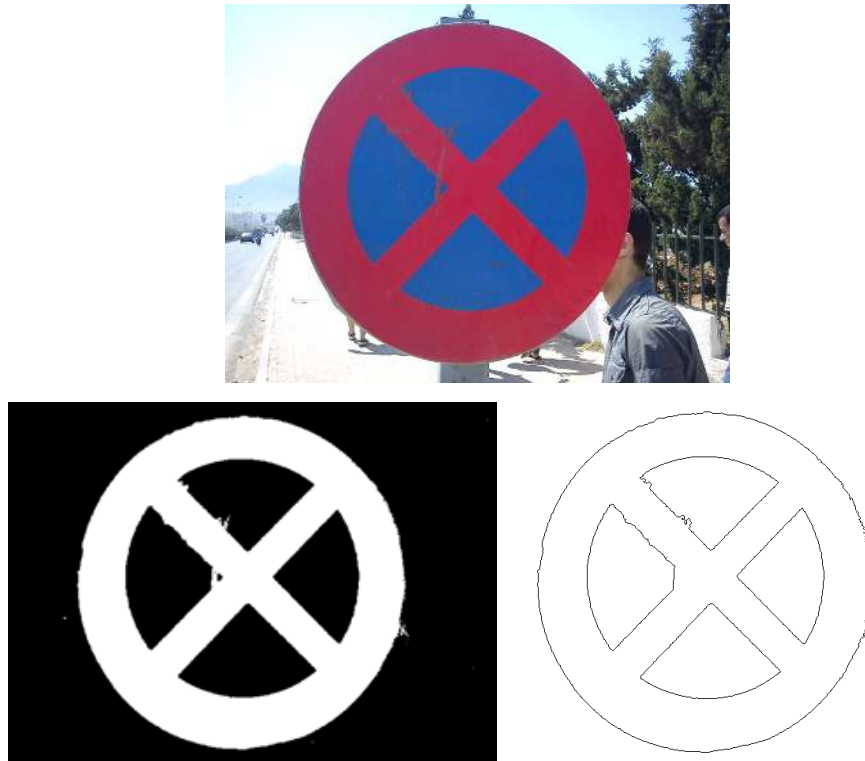


Figure 1.14. Image originale binarisé et son contour.

1.3.4- Opérations morphologiques

Les opérations morphologiques telles que la dilatation, l'érosion, la fermeture et l'ouverture sont utilisés pour l'amélioration des images binaires en éliminant les faux pixels considérés comme des bruits.

- **Dilatation**

La dilatation consiste à éliminer les points noirs isolés : on dilate les parties blanches ce qui élimine les points noirs. L'opération de dilatation est défini par [21]:

$$I \oplus S = \{x | [(S)_x \cap I] \neq \emptyset\} \quad (I-1)$$

Ou I est l'image qui sera dilatée, S est un ensemble de points de coordonnées connues. L'effet de base de l'opérateur de dilatation sur une image binaire est d'agrandir progressivement les limites des régions noires de l'image. Donc les régions blanches de l'image grandissent en dimensions.

- **Erosion :**

Il s'agit ici d'éliminer les points blancs isolés. La méthode est assez similaire à celle de la dilatation. L'opération d'érosion est définie par [21]:

$$I \ominus S = \{x | [(S)_x \subseteq I]\} \quad (1-2)$$

L'effet de base de l'opérateur de l'érosion est d'éroder les limites des régions de couleur blanche. Donc les régions d'images élémentaires de couleur blanche diminuent en dimensions, et les régions élémentaires de couleur noire deviennent plus grandes. La dilatation et l'érosion peuvent être combinées pour créer d'autres opérateurs morphologiques plus complexes pouvant résoudre une variété de problèmes [21].

- L'ouverture est une opération d'érosion suivie d'une dilatation, elle est définie par [20]:

$$B \circ I = (B \ominus I) \oplus I \quad (1-3)$$

- La fermeture est une opération de dilatation suivie d'une érosion, elle est donnée par [20]:

$$\Lambda \bullet I = (A \oplus I) \ominus I \quad (1-4) \quad [7]$$

- **Filtre dérivées premières.** Le filtre le plus simple consiste à calculer les différences entre pixels voisins sur les horizontales puis sur les verticales. Chaque extremum correspond à un point d'un contour.
- **Filtre de Sobel :** la technique précédente est améliorée en remplaçant le filtre rectangulaire par un filtre triangulaire.
- ✓ **Filtre dérivées secondes :** celles-ci se calculent simplement en différences finies et c'est maintenant un changement de signe qui correspond à un point d'un contour. On les utilise généralement à travers leur somme qui est le laplacien.

1.3.5- Filtrage

Le principe du filtrage est de modifier la valeur des pixels d'une image, généralement dans le but d'améliorer son aspect. En pratique, il s'agit de créer une nouvelle image en se servant des valeurs des pixels de l'image d'origine. On peut citer quelques types de filtrage

α -Filtres linéaires

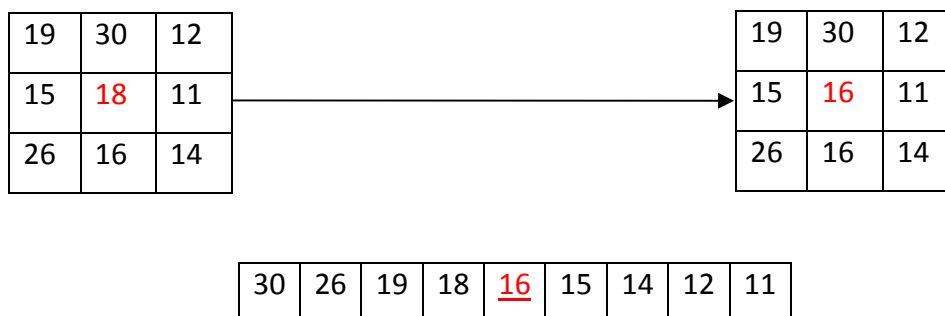
Un filtre linéaire transforme un ensemble de données d'entrée en un ensemble de données de sortie selon une opération mathématique appelée convolution. Lorsqu'il s'agit de données numérisées comme dans le cas du traitement d'image, la relation entre les valeurs des pixels de sortie et celle des pixels d'entrée est décrite par un tableau de nombres, généralement carré, appelé matrice de convolution. Le temps de calcul est souvent réduit lorsqu'on veut séparer un filtre en deux filtres dont la convolution mutuelle permet de le reconstituer. Cette remarque est utilisée en particulier pour créer un filtre à deux dimensions à partir de deux filtres à une seule dimension (vecteurs) dans le sens horizontal et le sens vertical.

b-Filtres non linéaires

Ils sont conçus pour régler les problèmes des filtres linéaires, sur tout pour ce qui concerne la mauvaise conservation des contours. Leur principe est le même que celui des filtres linéaires, il s'agit toujours de remplacer la valeur de chaque pixel par la valeur d'une fonction calculée dans son voisinage, la seule différence c'est que cette fonction n'est plus linéaire mais une fonction quelconque (elle peut inclure des opérateurs de comparaisons).

✓ *Filtre médian*

Sur un voisinage à huit, le nouveau niveau de gris du pixel centre est choisi comme étant la valeur médiane de tous les pixels de la fenêtre d'analyse centrée sur ce dernier. Son avantage est qu'il garde la netteté des éléments qui constituent l'image sans étaler les transitions.



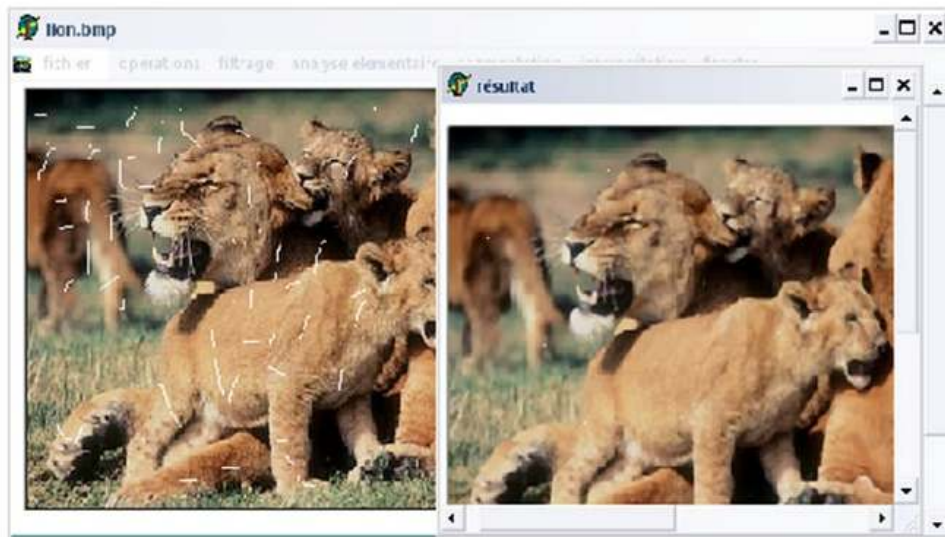


Figure I.15. Principe du filtre médian

✓ **Filtre maximum**

Même principe mais la valeur choisie est la valeur maximale.

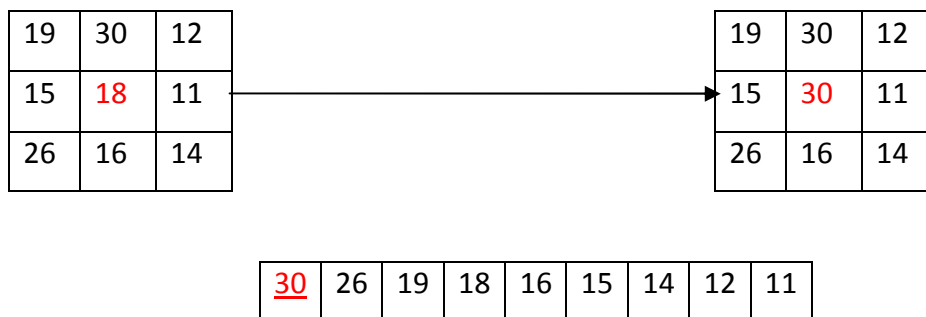


Figure I.16 : Principe du filtre maximum

✓ **Filtre minimum**

Même chose, sauf que la valeur choisie est la valeur minimale.

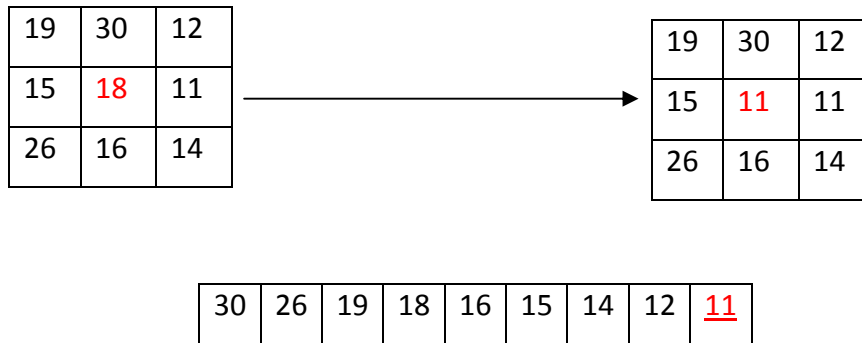


Figure 1.17: Principe du filtre minimum

c-Lissage

Ceux-ci sont des filtres passe-bas qui coupent plus ou moins les plus hautes fréquences. Ils sont utilisés pour atténuer les bruits d'origines les plus diverses qui polluent l'information, en particulier dans la détection de contours considérée ci-après.

Techniquement, il s'agit de traductions discrètes de filtres continus qui, comme ceux-ci, ne modifient pas le niveau global du signal. Les termes de la matrice de convolution sont donc généralement des entiers à diviser par leur somme.

I.4 Reconnaissance d'objets

La reconnaissance d'objets est une branche de la vision artificielle et un des piliers de la vision industrielle. Elle consiste à identifier des formes pré-décrites dans une image numérique, et par extension dans un flux vidéo numérique.

Il ne faut pas confondre reconnaissance d'objets (en anglais : « Object recognition » ou « Shape recognition ») et reconnaissance de formes (« pattern recognition » en anglais). La première s'attache à reconnaître des formes géométriques dans une image, alors que la seconde cherche à identifier des motifs dans des données statistiques. La confusion vient du fait qu'on utilise souvent la reconnaissance de formes comme technique appliquée à la reconnaissance d'objets.

Tout d'abord objet d'algorithmes dirigés par l'homme, jusqu'en 1995 (tentatives de reproduire par un algorithme un raisonnement humain d'identification, comme par exemple dans « un vélo possède deux roues, un cadre ... »), la reconnaissance d'objets

a fait l'objet de progrès importants par la suite au travers de la mise en œuvre de techniques d'apprentissage, comme par exemple les séparateurs à vaste marge. Ces techniques visent à faire exploiter des bases d'exemples positifs et négatifs (contre-exemples) par un algorithme de recherche de critères discriminants, c'est-à-dire de critères permettant de séparer au mieux les exemples des contre-exemples.

1.5 Conclusion

Après avoir présenté quelques notions sur la vision par ordinateur, ainsi que quelque technique de traitement d'image, on détaille au chapitre suivant Une des objectifs de la vision assisté par ordinateur qui est la reconnaissance des panneaux routier. On fait un état de l'art des dernières méthodes Utilisée.

Chapitre II Reconnaissance de panneaux de signalisation routière



II.1.Introduction

Parmi les outils les plus difficiles en vision par ordinateur et les plus utilisés pour ces véhicules, la reconnaissance automatique d'objets. Elle est en même temps une étape primordiale pour la mise en œuvre de plusieurs applications actuelles qui nécessitent une interprétation de haut niveau d'images. Par conséquent, il existe un intérêt croissant sur ce domaine de recherche dans les dernières années et une vaste littérature.

✓ Les panneaux de signalisation routière

Les panneaux routiers sont aussi vieux que les routes. Les premiers signes furent ceux donnant les directions. L'Empire romain a érigé des bornes, en colonnes de pierres, le long des routes pour indiquer les distances jusqu'à Rome. Au Moyen Âge, les signaux directionnels devinrent courants, indiquant la direction des villes aux carrefours. Inscrites sur des colonnes ou des bornes, les indications sont progressivement, à partir du XVIII^e et au XIX^e siècle, placées sur des plaques (en fonte, à partir de 1835), dites plaques de cocher, fixées sur les murs, les croix ou obélisques des carrefours, ou sur des poteaux métalliques.

La signalisation devint beaucoup plus importante avec le développement de l'automobile. Les principes de base de la plupart des signalisations furent définis le 11 octobre 1908 lors du premier congrès international de la route à Rome. [8]

II.2.Reconnaissance de panneaux de signalisation routière

II.2.1.Introduction

La conduite assisté par ordinateur a pour objectif d'aider le conducteur a prendre les bonnes décisions dans des circonstances difficiles (mauvaise vision, fatigue...) pour mieux contrôler son véhicule et augmenter sa sécurité et la sécurité des autres conducteurs, piétons, etc. prenant part a la circulation sur les routes.

II.2.2. Problématique

Parmi l'ensemble de difficulté, en cite les point suivant :

- Différence La luminance.
- Faux signaux (ressemblance de couleur exemple : feu de stop d'une voiture avec une plaques rouge).
- L'état et positionnement des plaques en Algérie
- L'environnement (caméra OUTDOOR)

II.2.3.Historique

Les premières publications sur la détection et la reconnaissance des panneaux routiers datent des années quatre vingt, de notre précédent siècle. Des états de l'art complets et détaillés ont été réalisés par Lalonde et Li [9] du CRIM (centre de recherche en informatique de Montréal), suivi par Stéphane Estable du LASMEA (Laboratoire des Sciences et Matériaux pour l'Electronique et d'Automatique à Aubière), de Pavel Paclik [19] du CTU (Czech Technical University in Prague) et de Rozenn Dahyot [10] du LRS (Laboratoire Régional des Ponts et Chaussées de Strasbourg) reprenant l'idée de Lalonde et Li sur une description des travaux réalisés en France, Allemagne, Italie, Canada, États-Unis et Japon.

De nombreux groupes de recherches concentrent leurs travaux sur la détection et reconnaissance des panneaux routiers, nous citons quelques projets en cours comme le projet ARGO de l'Université de Parma [11], le projet VITA II du centre de recherche de Daimler Benz AG à Ulm ainsi que le projet Européen PROMETHEUS (PROgram for an European Traffic with Highest Efficiency and Unprecedented Safety).

De nombreux centres de recherche, comme CMM, INRIA, IRISA, LIFIA, UBP Clermont-Ferrand, USTL Lille, UTC Compiègne, y consacrent leurs travaux à l'élaboration du véhicule PROLAB2 en partenariat avec les constructeurs automobiles Renault, PSA (Peugeot-Citroën), Matra, Jaguar, Fiat et Saab afin de contribuer à l'étude et la recherche de systèmes de sécurité active par un système temps réel embarqué.

L'EUCAR [12] (European Council for Automotive R&D) regroupe également des projets dédiés à la sécurité active : CHMAC (coopération homme machine pour l'aide à la conduite) de l'UTC de Compiègne, ARCOS (action de recherche pour une conduite sécurisée) de Peugeot-Citroën, PAROTO [13] (projet anticollision radar et optronique pour l'automobile) collectifs Lasma, Inrets, Team et Sagem, ADASE (advanced driver assistance systems in Europe) de DaimlerChrysler, ROADSENSE (road awareness for driving via a strategy that evaluates numerous systems) de Renault et PSA, RADARNET [14] (Multifonctional Automotive Radar Network) du collectif, DaimlerChrysler, Volvo, Jaguar, Fiat et Siemens et Infineon. Tous ces laboratoires de recherche ont élaboré de nombreux algorithmes pour la détection et la reconnaissance des panneaux routiers basés sur une classification statistique ou neuronale.

Très souvent cité, le système le plus avancé est le TSR développé par le Laboratoire d'Identification d'Image de l'Université Koblenz Landau. La détection par segmentation des couleurs a été réalisée à l'aide d'une architecture parallèle en temps réel (PowerPC et Transputers) pour analyser trois images par seconde. Daimler-Chrysler a développé le système RSR où la reconnaissance des panneaux est réalisée par un réseau de neurones.

Le laboratoire de recherche de l'Université de Gênes en Italie a élaboré une méthode robuste pour la détection et l'identification des panneaux routiers par un processus de corrélation statistique à l'aide d'un DSP, le TMS320C40. Le groupe de recherche de la Faculté des Sciences de Transport, CTU Prague développe depuis 1995 le système RS2

à base d'architecture parallèle muni du DSP TMS320C80 pour y implémenter un classificateur statistique.

Le principal obstacle décrit par la majorité des chercheurs est la segmentation des images couleurs. De nombreux travaux ont été effectués pour essayer de créer des régions dans une image où chaque région est définie par un ensemble de critères de connexité, d'homogénéité ou appartenant à une même classe. La segmentation la plus souvent citée est la segmentation récursive par l'étude des différents histogrammes associés à chaque région.

II.3. Techniques de reconnaissance de panneaux de signalisation routière

Parmi les objectifs de la vision assistée par ordinateur la reconnaissance de forme, en détaille dans ce qui suit quelques techniques de reconnaissance.

II.3.1 Descripteur de contour

✓ Transformée de Fourier

L'idée d'exprimer une fonction de forme compliquée comme une combinaison linéaire de fonction élémentaire de forme simple est très utile.

Par exemple, dans un intervalle $[u_1, u_2]$, une fonction $\theta(u)$ peut être représentée par :

$$\theta(u) = \sum \alpha_i \varphi_i(u) \dots \dots \dots (2.1)$$

Où les fonctions $\varphi_i(u)$ forment un ensemble de fonctions élémentaires de forme simple. Si ces fonctions sont orthogonales, alors les coefficients α_i sont indépendants les uns des autres on parle dans ce cas d'un développement en série de fonctions orthogonales.

Parmi les développements en séries orthogonales, la série de Fourier est sans doute celle dont on fait le plus grand usage en traitement des signaux.

On peut généraliser la notion de série de Fourier pour représenter une fonction dans un intervalle infini. Cette généralisation aboutit à la transformation intégrale de Fourier

$$x(f) = \int_{-\infty}^{+\infty} x(t) \exp(-j2\pi ft) dt \dots \dots \dots (2.2)$$

La transformée inverse est donnée par :

$$x(t) = \int_{-\infty}^{+\infty} x(f) \exp(j2\pi ft) df \dots \dots \dots (2.3)$$

La transformée de Fourier des signaux numériques est définie par :

$$x(f) = \sum_{k=-\infty}^{+\infty} x(k) \exp(-j2\pi fk) \dots \dots \dots (2.4)$$

Soit $f(x)$ une fonction continue de la variable réelle x . La transformée de Fourier notée $F\{f(x)\}$ est définie par l'équation :

$$F\{f(x)\} = F(u) = \int_{-\infty}^{+\infty} f(x) \exp(-j2\pi ux) \dots \dots \dots (2.5)$$

De même, à partir de $F(u)$ on peut revenir à $f(x)$

$$F^{-1}\{f(x)\} = f(x) = \int_{-\infty}^{+\infty} F(u) \exp(j2\pi ux) \dots \dots \dots (2.6)$$

Généralement la transformée de Fourier d'une fonction réelle est complexe et s'écrit :

$$F(u) = R(u) + jI(u) \dots \dots \dots (2.7)$$

Où $R(u)$ et $I(u)$ sont respectivement les parties réelles et imaginaires de $F(u)$

$$F(u) = |F(u)| \exp(j\phi(u)) \dots \dots \dots (2.8)$$

Ou

$$|F(u)| = [R^2 + I^2]^{1/2} \dots \dots \dots (2.9)$$

Et

$$\phi(u) = \text{arctg}[I(u)/R(u)] \dots \dots \dots (2.10)$$

La variable apparaissant dans la transformée de Fourier est souvent dite variable fréquentielle (ou fréquence). Ceci provient de l'utilisation de la formule d'Euler des nombres complexes $\exp [2\pi ux]$ peut être exprimé sous la forme :

$$\exp[-j2\pi ux] = \cos[j2\pi ux] - j \sin j2\pi ux \dots \dots \dots (2.11)$$

Si nous interprétons l'intégrale de l'équation (2.11) comme une sommation limitée de termes discrets, il est évident que $F(u)$ est composé d'une somme infinie de termes en sinus et cosinus et chaque valeur de u détermine la fréquence correspondant à une paire sinus-cosinus.[15]

✓ **Transformée de Fourier discrète**

Supposons qu'une fonction continue est discrétisée en séquences

$$\{f(x_0), f(x_0 + \Delta x), f(x_0 + 2\Delta x), \dots, f(x_0 + (N - 1)\Delta x)\} \dots \dots \dots (2.12)$$

En prenant **N** unités Δx .

On peut prendre **x** comme variable en écrivant :

$$f(x) = f(x_0 + x\Delta x) \dots \dots \dots (2.13)$$

Ou **x** prend des valeurs discrètes 0, 1, 2, ..., N-1

Ceci nous donne la relation de la DFT suivante :

$$F(u) = 1/N \sum_{x=0}^{N-1} f(x) \exp[-j2\pi ux/N] \quad u = 0, 1, 2, \dots, N - 1 \dots \dots (2.14)$$

Et

$$f(x) = 1/N \sum_{u=0}^{N-1} F(u) \exp[j2\pi ux/N] \quad x = 0, 1, 2, \dots, N - 1 \dots \dots (2.15)$$

Les valeurs $u = 0, 1, 2, \dots, N - 1$ pour la transformée discrète correspondent à : $0, \Delta u, 2\Delta u, (N - 1)\Delta u$, pour la transformée continue.

En d'autres termes **F(u)** représente **F(uΔu)**.

Pour une fonction à deux variables la DFT est donnée par :

$$u = 0, 1, 2, \dots, M - 1 \quad , \quad v = 0, 1, 2, \dots, N - 1$$

Et

$$F(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} f(u, v) \exp[-j2\pi(ux/M + vy/N)] \dots \dots \dots (2.16)$$

Dans le cas d'une matrice carrée $M = N$

$$F(u, v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \exp[-j2\pi(ux/M + vy/N)] \dots \dots \dots (2.17)$$

$$u = v = 0, 1, 2, \dots, N - 1$$

✓ **Descripteur de Fourier**

L'une des techniques les plus promettant de description de forme est basée sur les descripteurs de Fourier.

Supposons que nous avons **N** points de la frontière d'une région. Nous pouvons considérer la région comme placée dans le plan complexe, avec l'axe des ordonnées comme axe imaginaire et l'axe des abscisses comme l'axe réel. Alors les coordonnées **x**, **y** de chaque point du contour à analyser peuvent être représentées comme des nombres complexes ($x+jy$).

Le contour peut être écrit sous la forme de séquence de nombres complexes.

$$Z_i = x_i + jy_i \quad i = 0,1,2, \dots, N - 1$$

La séquence est périodique pour chaque complet du contour. Les descripteurs de Fourier sont définis comme suit :

$$A(u) = 1/N \sum_{i=0}^{N-1} Z_i \exp[2\pi u i/N] \quad u = 0, 1, 2, \dots, N - 1 \dots \dots \dots (2.18)$$

La transformation étant réversible

$$Z_i = \sum_{u=0}^{N-1} A(u) \exp[j2\pi u i/N] \quad i = 0, 1, 2, \dots, N - 1 \dots \dots \dots (2.19)$$

Avant d'utiliser les descripteurs de Fourier il faut éliminer leur dépendance de la position, la dimension, l'orientation et le point de départ, comme suit :

- 1 - Le changement de la position du contour altère $A(0)$ seulement.
- 2 - En multipliant la dimension du contour par une constante, les descripteurs de Fourier sont multipliés par une constante également.
- 3 - La rotation du contour dans le domaine spatial requiert la multiplication de chaque coordonnées par $\exp(j\theta)$ où θ est l'angle de rotation. La linéarité de la transformée de Fourier fait que la multiplication des coefficients du domaine fréquentiel produit le même effet.
- 4 - Le changement du point de départ du contour dans le domaine spatial (temporel) correspond à multiplier le $k^{\text{ième}}$ coefficient du domaine fréquentiel par $\exp(jkT)$ où **T** est

une fraction de période par laquelle le point de départ a été décalé (T varie de 0 à 2π , le point de départ traverse le contour une fois).[15]

On normalise les DF en prenant $A(0)$ égal à 0 (pour rendre les DF indépendantes de la position) et on divise chaque coefficient par l'amplitude de la première DF $A(1)$ (pour normaliser leur taille).

✓ **Algorithme utilisant les descripteurs de Fourier**

Soit $(x(m), y(m))$ les coordonnées d'un pixel appartenant au contour ; avec $0 \leq m \leq L-1$

En prenant la transformée de Fourier discrète (DFT) de la donnée $(x(m), y(m))$ $0 \leq m \leq L-1$, on obtient les coefficients de Fourier pour $0 \leq k \leq L-1$

$$A(k) = 1/L \sum_{m=0}^{L-1} Z(m) \exp(-j2\pi km/L) \dots \dots \dots (2.19)$$

$$\exp(j\theta_0) = \cos \theta_0 + j \sin \theta_0 \dots \dots \dots (a)$$

$$Z(m) = x(m) + jy(m) \dots \dots \dots (b)$$

D'après l'équation (2.18) et (2.19) :

$$A(k) = 1/L \sum_{m=0}^{L-1} [x(m) + jy(m)] [\cos(2\pi km/L) - j \sin(2\pi km/L)] \dots (2.20)$$

$$a(k) = \text{Re}(A(k)) = 1/L \sum_{m=0}^{L-1} [x(m) * \cos(2\pi km/L) + y(m) * \sin(2\pi km/L)] \dots (2.21)$$

$$b(k) = \text{Im}(A(k)) = 1/L \sum_{m=0}^{L-1} [-x(m) * \sin(2\pi km/L) + y(m) * \cos(2\pi km/L)] \dots (2.22)$$

On distingue les composants DC ; $a(0)$ et $b(0)$ puisqu'il donne l'information seulement sur la position du centre de l'image, On laisse

$$|A(k)| = \sqrt{|a(k)|^2 + |b(k)|^2} \dots \dots \dots (2.23)$$

$$S(k) = A(k)/A(1) \quad k = 1, \dots, L-1 \dots \dots \dots (2.24)$$

Où $|a(k)|, |b(k)|$ de note la valeur absolue des nombres complexes $a(k)$ et $b(k)$. Alors, il devient facile de voir que $A(k)$ est invariant pour la rotation, la translation et plus loin $S(k)$ est invariant pour le changement d'échelle.

Les descripteurs invariants $s(k)$ sont appelés descripteurs de Fourier. Noter que pour $L_1, L_2 < e_1 < e_2 < (L-1)/2$ $S(1), \dots, S(e_1)$ sont les basses fréquences de l'image alors que $S(e_2), \dots, S[(L-1)/2]$ sont les hautes fréquences, où $[(L-1)/2]$ représente la partie entière de $(L-1)/2$. La partie de basses fréquences de $S(k)$ est déterminée par la forme d'un objet, alors que les hautes fréquences donnent les détails. Puisque les hautes fréquences sont dégradables par le bruit et ne contribuent pas beaucoup à la reconnaissance de l'objet composé aux basses fréquences, souvent on les ignore. Cette compression significative de la donnée est l'une des raisons de l'utilisation des descripteurs de Fourier.

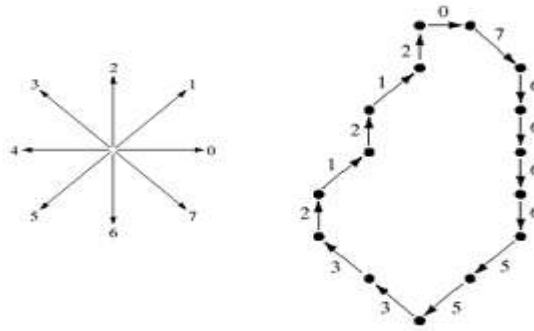
Dans une étude d'une image qui contient plus de 500 points, seulement les 32 premiers composants ($S[1], \dots, S[32]$) ont été utilisés comme vecteur de forme sur plus de 500 coefficients.

II.3.2 Codage Freeman

✓ Définition

Le codage de Freeman sont générées par localisation du pixel de contour, également il fixe le premier pixel de départ, ensuite il déplace au long de la frontière de contour soit dans le sens horaire ou antihoraire, cherche le pixel suivant du contour et assigne le nouveau pixel par un code en fonction de son emplacement par rapport au pixel précédent. La procédure cherche le pixel suivant est se termine lorsque le pixel de départ est rencontré.

Les codes peuvent être 4-directionnel ou 8-directionnel selon 4-connectivité ou 8-connectivité d'un pixel à son pixel voisin de contour. Une chaîne 8-directionnelle d'une image codé est illustrée par la (Figure 2.7). [16]



Chain code: 076666553321212

Figure 2.1 : chaîne code de 8-connectivité

On constate que la chaîne code pour différents chiffres donne différentes longueurs de code et chaque longueur de chaîne code dépend de la taille de la référence de l'objet.

Dans certain cas la taille de la référence de l'objet est très élevée, et pour résoudre ce problème il faut normaliser la chaîne code comme suit :

Supposer que cette chaîne code est générée pour un contour traversant en sens anti horaire (sens horaire).

V1= [1 2 2 2 1 2 1 2 1 1 2 0 1 0 1 0 0 0 0 0 7 7 0 0 0 1 0 0 0 0 0 0 0 0 0 7 0 0 7 0
 7 6 6 6 7 6 6 7 6 6 6 6 6 7 6 6 6 6 5 6 6 6 5 6 6 5 5 6 5 4 4 5 4 5 4 4 4 5 4 4 4 4 4 3 4 4
 4 3 4 4 3 4 3 4 4 3 2 3 3 2 3 2 3 2 3 2 2 2 2 2 2 2 2 2]

Calculer la fréquence des codes 0, 1, 2 7, du vecteur V1 pour obtenir le vecteur fréquence V2 comme ci-dessous.

$$V2 = [23 \ 8 \ 19 \ 10 \ 20 \ 9 \ 20 \ 8]$$

La fréquence normalisée représentée par le vecteur V3, calculée par la formule suivante :

$$V3 = \frac{V2}{|V1|} \quad |V1| = \sum V2 \quad \dots\dots\dots (2.25)$$

Pour l'exemple considéré ci-dessus, on a :

$$V3 = [0.1966 \ 0.0684 \ 0.1624 \ 0.0855 \ 0.0015 \ 0.0769 \ 0.1709 \ 0.0684]$$

Finalement, la concaténation de la vecteur V2 et V3 on obtient un vecteur caractéristique de taille 16 comme ci-dessous.

$$V4 = [23 \ 8 \ 19 \ 10 \ 20 \ 9 \ 20 \ 8 \ 0.1966 \ 0.0684 \ 0.1624 \ 0.0855 \ 0.0015 \ 0.0769 \ 0.1709 \ 0.0684]$$

II.4. Classification

La majorité des techniques de reconnaissance des signaux routiers utilise un modèle statistique entraîné avec des vecteurs de caractéristiques calculés sur de nombreux exemples d'images d'objets. Le modèle détermine ainsi la fonction de décision qui permet de séparer au mieux les exemples positifs (représentant un objet) des exemples négatifs (le reste).

Les modèles discriminants cherchent à déterminer la meilleure frontière de décision permettant de trouver l'hyperplan optimal séparant les vecteurs positifs des vecteurs négatifs dans l'espace de représentation des vecteurs. Les machines à vecteur de support font partie des modèles les plus employés et les plus efficaces dans ce domaine. Ils ont été utilisés pour la détection de personnes avec un noyau linéaire, des noyaux gaussiens ou polynomiaux [38]. Les performances en détection dépendent des caractéristiques et des

classifieurs faibles utilisés. Les réseaux de neurones peuvent également être employés, par exemple un perceptron multicouche modifié afin que les neurones cachés ne reçoivent leurs entrées que d'une région spatiale locale [17].

II.4.1 Méthodes de classification

On distingue principalement deux types de classification :

a-Méthode statistique

L'extraction des statistiques produit des valeurs numériques qui sont confrontées aux modèles statistiques caractérisant chaque classe. [18] voir **Figure 2.2**.

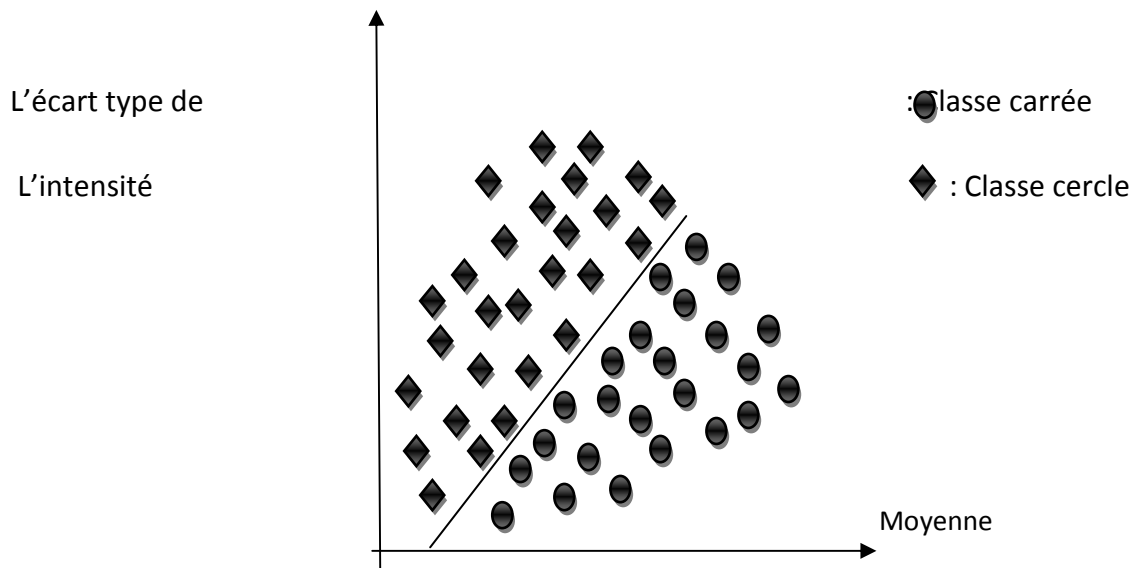


Figure 2.2 : extraction de deux classes

b- Méthode syntaxique (structurelle)

L'extraction des primitives produit des valeurs symboliques des relations qui font l'objet d'une analyse structurelle ou syntaxique.

II.4.2 Mesure de distance entre deux points

Comment mesurer la distance entre deux points $d(x_1, x_2)$:

➤ **Distance Euclidienne**

$$d^2(x_1, x_2) = \sum_i (x_{1i} - x_{2i})^2 \dots\dots\dots (2.19)$$

➤ **Distance de Manhattan**

$$d(x_1, x_2) = \sum_i |x_{1i} - x_{2i}| \dots\dots\dots (2.20)$$

➤ **Distance de Sebstyen**

$$d^2(x_1, x_2) = (x_1 - x_2) \cdot \omega \cdot (x_1 - x_2) \dots\dots\dots (2.21)$$

ω : Matrice diagonal de Pondération

➤ **Distance de Mahalanobis**

$$d^2(x_1, x_2) = (x_1 - x_2) C^{-1} (x_1 - x_2) \dots\dots\dots (2.22)$$

C : Matrice de covariance

II.4.3 Mesure de distance entre deux classes

C'est la différence entre deux vecteurs.

Comment mesurer la distance entre deux classes D (C_1, C_2)

- **Distance de plus proche voisin (Nearest Neighbors : NN)**

$$\min(d(i, j), i \in C_1, j \in C_2) \dots \dots \dots (2.23)$$

- **Distance maximum**

$$\text{Max}(d(i, j), i \in C_1, j \in C_2) \dots \dots \dots (2.24)$$

- **Distance moyenne**

$$\frac{\sum d(i, j)}{n_1 n_2} \dots \dots \dots (2.25)$$

- **Distance des centres de gravité**

$d(u_1, u_2)$ u_1 : Vecteur moyen de la classe 1.

- **Distance de Ward**

$$\sqrt{\frac{n_1 n_2}{n_1 + n_2}} d(u_1, u_2) \dots \dots \dots (2.26)$$

II.5. Evaluation et performances

Afin de connaître précisément les performances des différents algorithmes, des bases d'images et de vidéo contenant des plaques routières ont été constituées et la position de chaque plaque annotée manuellement. Ceci permet de juger les performances de l'algorithme de reconnaissance, par rapport à une « vérité terrain » estimée par un humain.

Certaines de ces bases ont été rendues publiques, ce qui permet la comparaison entre plusieurs algorithmes.

La mesure des performances peut se faire par image ou par fenêtre. La mesure par image consiste à mesurer la performance finale dans l'image. La mesure par fenêtre est spécifique aux algorithmes fonctionnant par fenêtre glissante : on mesure les performances dans chaque fenêtre testée, qui est soit positive ou négative.

La reconnaissance des signaux routiers, est une tâche assez difficile, les performances des différents algorithmes restent assez modestes.

Les performances dépendent toutefois beaucoup de la base de test et de la base d'apprentissage. Les algorithmes ont des difficultés à gérer les occultations, les faibles résolutions, les objets de ratio largeur/hauteur non standards. Les méthodes obtenant les meilleurs résultats sont celles qui utilisent plusieurs types de caractéristiques, avec des temps de traitement trop élevé

II.6.Conclusion

Après cette modeste étude sur les différents techniques de reconnaissance d'objets; On a choisi la méthode de KPPV pour l'implémenter et reconnaître les signaux routier

On a pris en considération pour choisir cette méthode plusieurs facteurs importants :

- Possibilité d'implémentation sous c/c++ builer.
- Simple à utiliser.

On détaillera cette méthode ainsi que son implémentation sous C/CC+ builder

Chapitre III : Implémentation, test et résultats

III.1 Introduction

L'algorithme est conçu pour reconnaître des signaux rouges triangulaires, rouges circulaires et bleus circulaires. Faisant ce choix des signaux routiers à reconnaître, on couvre une grande majorité des signaux routiers usuels en Algérie et aussi les signaux les plus importantes qui annoncent un danger sur la route, une interdiction (limitation de la vitesse) ou qui imposent au conducteur un certain comportement (signaux bleues).



Figure3.1. Echantillons de panneaux de signalisation routière.

III.2. Mise en situation

III.2.1 Définition de l'objectif

L'objectif de l'application est de permettre à un véhicule de reconnaître les signaux routiers dans un champ de vision (caméra embarquée), sachant qu'il évolue dans un milieu urbain, avec fonds d'images très variés et en présence de plusieurs objets fixes et mobiles : voiture , piétons, vélo, arbresetc.

S'ajoute à ca, la différence de la luminance entre les endroits due a l'ombre par exemple, le changement climatique, l'heure de reconnaissance.

Ce travail présente une première étape pour la mise en place d'un module de reconnaissance.

III.2.2 Matériels utilisés

Le matériel utilisé pour les expérimentations de cette application est composé d'un ordinateur portable et d'une caméra (web Cam).



Figure3.2. Matériels utilisés.

III.3 Environnement de logiciels

Pour l'implémentation et le développement de l'application on a utilisé le langage suivant :

III.3.1 c/c++ builder 6

C++Builder est un environnement de programmation visuel orienté objet pour le développement rapide d'applications (RAD). En utilisant C++Builder, nous pouvons créer des applications Windows 32 bits très efficaces, avec un minimum de codage manuel. C++Builder fournit tous les outils qui sont nécessaires pour développer, tester, déboguer et déployer des applications, incluant une importante bibliothèque de composants réutilisables, un ensemble d'outils de conception, des modèles d'applications et de fiches, ainsi que des experts de programmation. Ces outils simplifient le prototypage et réduisent la durée du développement.

Pourquoi le Builder C++6 ?

- Développé des programmes Windows C++ avec plus de facilité et de rapidité qu'auparavant.
- Crée des applications console Win 32 ou des programmes Win 32 GUI (Interface Utilisateur Graphique). (Voir **figure 3.3**)
- Analyse, exploration et visualisation des données.

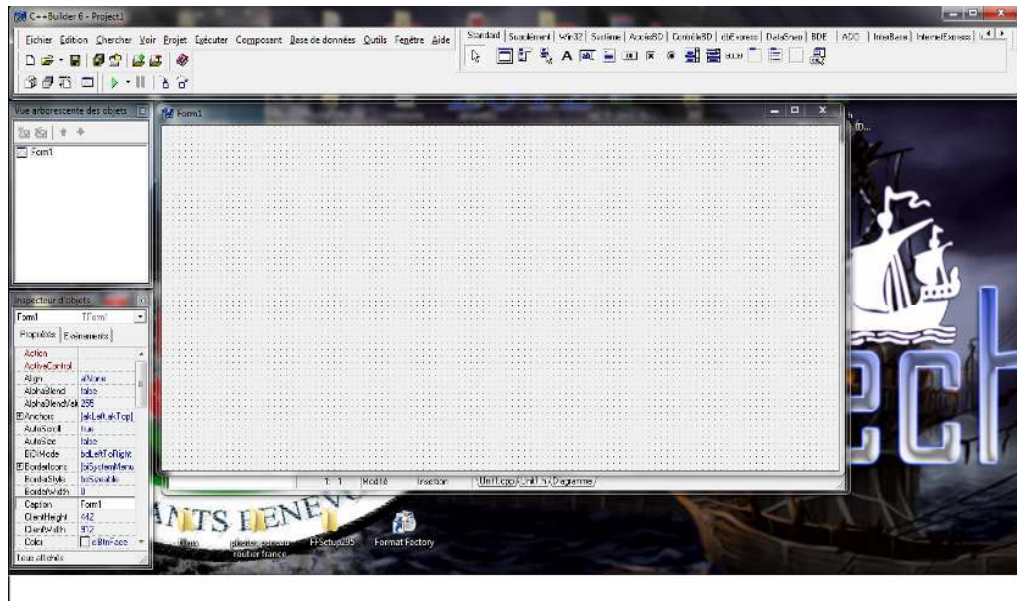


Figure3.3. C/C++ Builder 6.

- **Module**

Un module permet d'intégrer les fonctions de traitement sous la forme de services dans une structure standard, et d'accéder à ces services ainsi qu'aux données produites par le biais d'interfaces standards. Ces fonctions sont contrôlées au moyen de requêtes, c'est dans cette partie du module que nos algorithmes sont introduits. Lorsqu'un service, requis au moyen d'une requête, se termine, une réponse est retournée au client à la quelle est associé un bilan d'exécution qui caractérise la façon dont s'est déroulé le service.

On distingue deux types de requête : les requêtes d'exécution démarrent effectivement un service, alors que les requêtes de contrôle contrôlent leur exécution. Couche fonctionnelle IHM et Couche de contrôle Robot logique (modification de paramètres, interruption). Ces données peuvent être transmises à la fin de l'exécution par la réplique, ou durant l'exécution par des posters. Un poster est une structure C qui est mise à jour par une activité et qui est accessible en lecture par n'importe quel élément du système (un autre module, un opérateur,...). Toutes les données transmises par et vers le module (via les requêtes/répliques et les posters) sont mémorisées dans une base de données interne dénommée Structure de Données Interne Fonctionnelle ou SDIF.

III.3.2 La librairie OpenCV :

Un des objectifs buts d'OpenCV est d'aider les utilisateurs à construire rapidement des applications sophistiquées de vision à l'aide de simples opérations de vision par ordinateur.

OpenCV (Open Source Computer Vision) est une bibliothèque proposant un ensemble de plus de 2500 algorithmes de vision par ordinateur, elle a été écrite en C et C++, puis des interfaces ont été développées pour Python, Ruby, Matlab et autre langages. Elle est distribuée sous une licence BSD (libre) pour les plates-formes

Windows, GNU/Linux, Android et MacOS. Cette bibliothèque est orientée vers des applications en temps réel [19].

Initialement écrite en C il y a 10 ans par des chercheurs de la société Intel,

OpenCV est aujourd'hui développée, maintenue, documentée et utilisée par une communauté de plus de 40 000 membres actifs. C'est la bibliothèque de référence pour la vision par ordinateur, aussi bien dans le monde de la recherche que celui de l'industrie.

III.4 Reconnaissance des panneaux de signalisation routiers

Le travail que nous avons développé dans le cadre de ce projet de fin d'étude est basé essentiellement sur différentes étapes représentées par la *figure.3.4*.

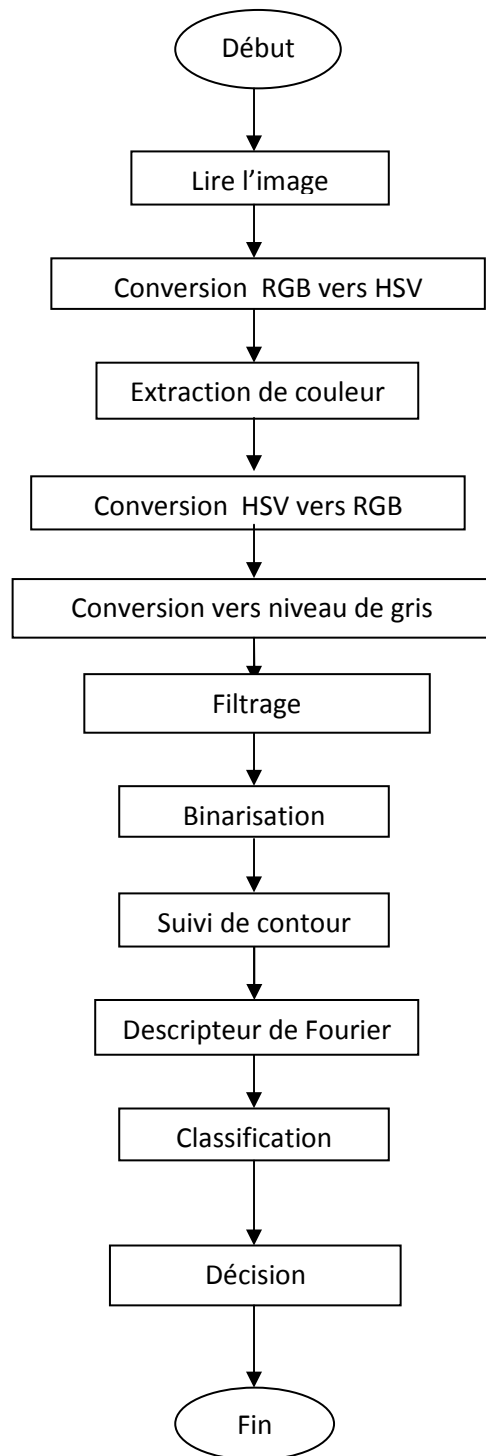


Figure3.4. Organigramme du projet.

III.4.1 Acquisition d'image

Une seule image de taille VGA (640x480) est captée du flux vidéo qui suivra le traitement illustré par la figure en dessous (**figure3.5**).

```
img1=cvLoadImage(OpenDialog->FileName.c_str(),1);
```

III.4.2 Conversion RGB/HSV

Il existe plusieurs manières de spécifier les couleurs (on dit qu'il existe plusieurs *espaces de couleur*).

On utilise principalement RGB (rouge/vert/bleu), mais il est nettement moins intuitif que d'autres tels que HSL (Hue/Saturation/Lightness, Teinte/Saturation/Lumière). Les espaces de couleur comme HSL sont également utile pour extraire une plage de couleur ou une valeur qui correspond une couleur facilement

III.4.3 Extraction de couleurs

Dans l'image on a besoin que de la couleur du panneau (rouge ou bleu) alors on laisse que la plage de la couleur rouge ou bien celle du bleu par une simple condition comme illustre la (**figure.3.5**)



Figure.3.5.Image original d'un panneau danger et sont extraction de couleur (rouge).



Figure.3.6 Image original d'un panneau d'interdiction et son extraction de couleur (rouge).



Figure.3.7 Image original d'un panneau obligation et son extraction de couleur(bleu).

III.4.4 Conversion HSV/RGB

Une fois le panneau est détecté on doit le mettre en niveau de gris. Ce qui veut dire qu'il doit être converti tout d'abord en RGB.

III.4.5 Niveau de gris :

Cette étape nous permet de passer d'une image de 3 plans à une image d'un seul plan.

$$G = \alpha R + \beta G + \gamma B.$$

Parmi ces avantages :

- Minimiser la taille de l'image.
- Réduire le temps d'exécution.
- facilite les manipulations.

III.4.6 Binarisation

On applique un seuil à l'image de niveau de gris et les pixels qui ont une valeur supérieure auront la valeur max égale à 255 et ceux qui ont une valeur inférieure auront la valeur min égale à 0 ou le contraire. Le résultat de l'application du module de la binarisation est représenté par la figure 3.8.

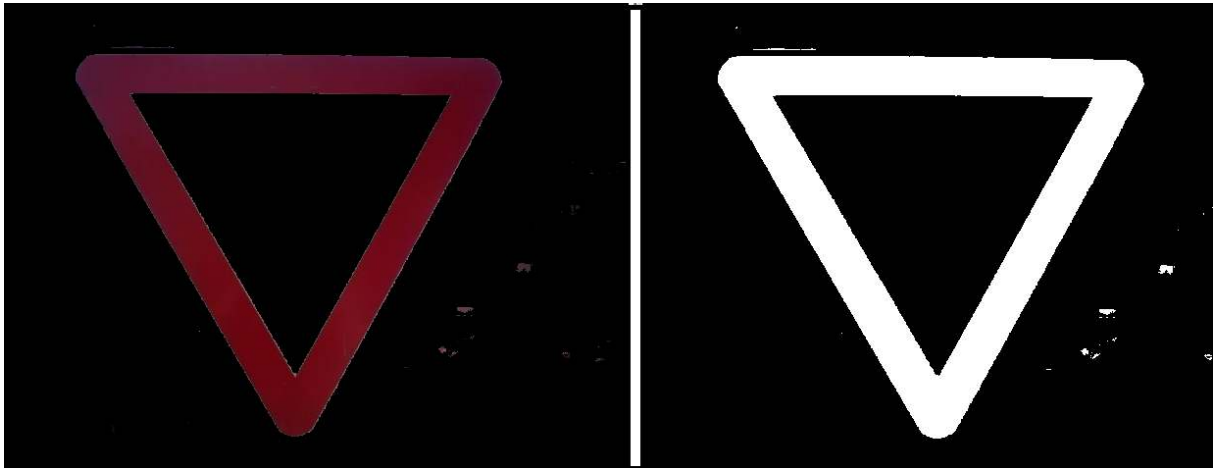


Figure.3.8 Image d'un panneau extrait et sa binarisation

➤ Méthode d'Otsu

Dans la méthode d'Otsu, le seuil qui minimise la variance inter-classe est recherché à partir de tous les seuillages possibles :

$$\sigma_{\omega}^2(t) = \omega_1(t)\sigma_1^2(t) + \omega_2(t)\sigma_2^2(t) \dots \dots \dots (3.1)$$

Les poids ω_i représentent la probabilité d'être dans la n^{ieme} classe, chacune étant séparée par un seuil t . Finalement, les σ_i^2 sont les variances de ces classes.

Otsu montre que minimiser la variance inter-classe revient à maximiser la variance intra-classe:

$$\sigma_b^2(t) = \sigma^2 - \sigma_{\omega}^2(t) = \omega_1(t)\omega_2(t)[\mu_1(t) - \mu_2(t)]^2 \dots \dots \dots (3.2)$$

Qui est exprimée en termes des probabilités de classe ω_i et des moyennes de classes μ_i qui à leur tour peuvent être mises à jour itérativement. Cette idée conduit à un algorithme efficace.

➤ Les étapes de la méthode

Cette méthode de binarisation nécessite au préalable le calcul de l'histogramme. Puis la séparation en deux classes est effectuée.

a Calcul de l'histogramme

Le calcul de l'histogramme est très simple.

On initialise un tableau hist avec des 0 Généralement, ce tableau est constitué de 256 cases correspondant aux 255 niveaux de gris d'une image.

En suite, si $p(i, j)$ fréquence représente la valeur du pixel au point (i, j) , on balaye toute l'image et on compte le nombre de fois ou un niveau de gris apparait.

b Séparation en deux classes

La séparation se fait à partir des moments des deux premiers ordres : la moyenne et l'écart type. Pour que le procédé soit indépendant du nombre de points dans l'image N , on normaliser l'histogramme : $p_i = n_i/N$ ou n_i représente le nombre de pixels de niveau i .

On peut calculer alors les deux moments utilisés :

$$\mu(k) = \sum_{i=1}^{i=k} i * P_i \quad \text{et} \quad \omega(k) = \sum_{i=1}^{i=k} P_i \dots \dots \dots (3.3)$$

On note $\mu_T = \mu(256-1)$, où 256 est le nombre totale de niveaux de gris.

Si on appelle ω_0 la probabilité de la classe C_0 et ω_1 la probabilité de la classe C_1 alors:

$$\omega_0 = \omega(k^*) \quad \text{où } k^* \text{ représente le niveau de seuil.}$$

Et

$$\omega_1 = 1 - \omega(k^*)$$

Si on note de même μ_1 et μ_0 avec :

$$\mu_0 = \mu(k^*) / \omega(k^*) \text{ et } \mu_1 = (\mu_T - \mu(k^*)) / (1 - \omega(k^*)) \dots \dots \dots (3.4)$$

Or l'image totale conserve certaines propriétés, d'où on peut tirer les relations :

$$\omega_0 * \mu_0 + \omega_1 * \mu_1 = \mu_T \text{ et } \omega_0 + \omega_1 = 1$$

En introduisant un paramètre pour évaluer la qualité du niveau de seuillage, on obtient :

$$S^2 = \omega_0 * \omega_1 (\mu_1 - \mu_0)^2 \dots \dots \dots (3.5)$$

La valeur précédente est fonction de k .

On calcule donc cette valeur pour les 256 niveaux de gris de l'image. En fait on peut déjà enlever les valeurs 0 et 255 qui correspondent à affecter tous les pixels à la même classe.

A partir de $\omega(k)$ et $\mu(k)$ on calcule donc :

$$S^2(k) = (\mu_T * \omega(k) - \mu(k))^2 / \omega(k) * (1 - \omega(k)) \dots \dots \dots (3.6)$$

La valeur du seuil k^* est obtenue pour le maximum de S^2 .

Il ne reste plus qu'à comparer la valeur de tous les pixels de l'image au seuil ainsi trouvé.

III.4.7 Suivi de contour

Algorithme du suivi de contour

Cette méthode est très rapide car elle ne nécessite que peu de calculs à chaque étape. Elle présente l'avantage de détecter les pixels frontières suivant toutes les directions. Tous les pixels seront traités une seule fois [20].

Le programme de suivi ainsi réalisé, permet de détecter les contours d'objets d'images binaires pouvant être représentées de la façon suivante :

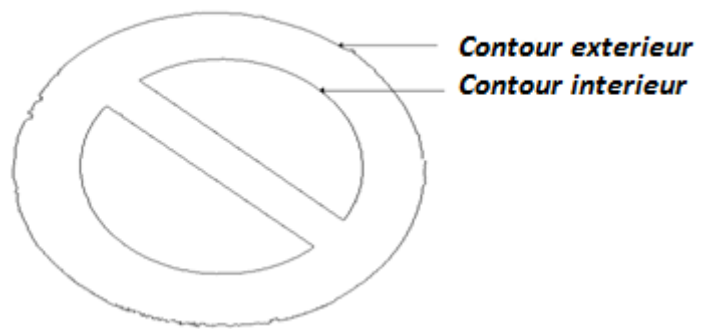
- Les objets sont constitués de pixels « noirs », la valeur « 1 » leur sera attribuée, tandis que l'extérieur (fond) est formé de pixels « blancs » de valeur « 0 ».

Ces contours sont classés selon deux catégories :

- Ceux qui englobent un objet. Ils sont dits extérieurs, c'est la silhouette de l'objet.
- Ceux qui sont noyés dans un objet, ce sont les contours intérieurs.

Après la détection du premier pixel contour, la nature de la transition détermine son type :

- Si cette transition est de nature fond forme (0 – 1), c'est un contour de type externe.
- Sinon (1 – 0), le contour est de type interne voir



- **Figure 3.9** Contour d'un panneau.

Cette méthode se résume à balayer l'image jusqu'à trouver une transition. C'est le premier pixel contour.

Etiqueter les huit voisinages de ce pixel. Tester si ces pixels appartiennent au contour. a chaque fois qu'un pixel objet est rencontré, on lui étiquette ses huit voisinages et continuer jusqu'à la rencontre du premier pixel trouvé.

- **Implémentation de la méthode**

Les principales étapes de l'algorithme sont les suivantes :

1^{er} étape :

Balayage de l'image jusqu'à la rencontre d'un pixel objet

Premier pixel contour à (i_1, j_1) Le pixel qui le précède est noté par (i_D, j_D) .

2^{eme} étape :

A partir du pixel (i_D, j_D) en tournant autour du pixel (i_1, j_1) de façon à balayer ses voisins dans le sens horaire, on numérote les sept autres voisins du pixel (i_1, j_1) comme 2, ..., 8.

3^{eme} étape :

Evaluer les coordonnées $Lx(k)$, $Ly(k)$ du k^{ieme} voisin de (i_1, j_1) en utilisant le (*Tableau 3.1*)

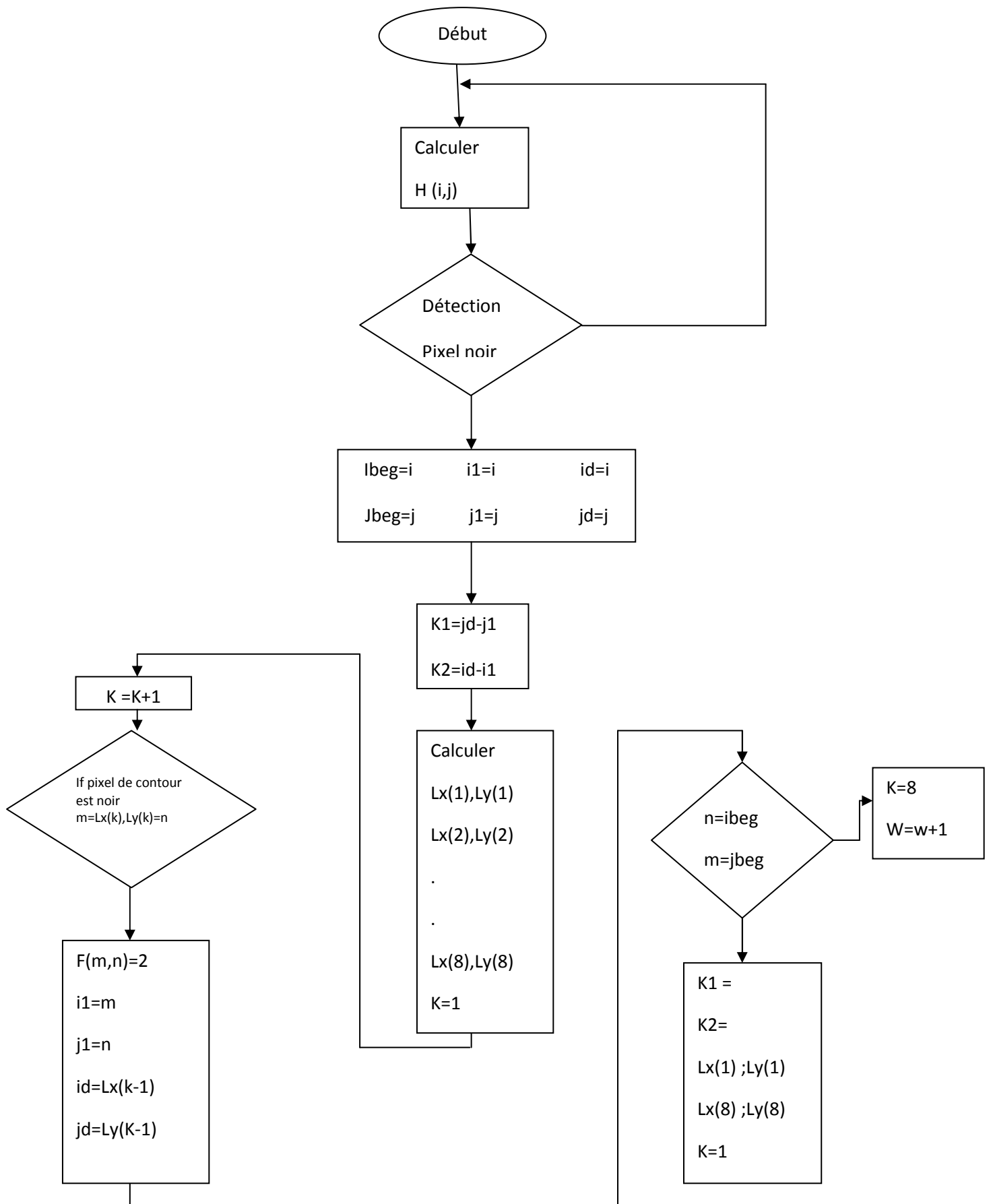


figure 3.10. Organigramme de suivi de contour.

1	$Lx(1)=i$	$Ly(1)=j-1$
2	$Lx(2)=i-1$	$Ly(2)=j-1$
3	$Lx(3)=i-1$	$Ly(3)=j$
4	$Lx(4)=i-1$	$Ly(4)=j+1$
5	$Lx(5)=i$	$Ly(5)=j+1$
6	$Lx(6)=i+1$	$Ly(6)=j+1$
7	$Lx(7)=i+1$	$Ly(7)=j$
8	$Lx(8)=i+1$	$Ly(8)=j-1$

Tableau 3.1 Coordonnées des 8 voisins

Coordonnées du pixel contour (i_1, j_1)

Coordonnées du 1^{er} pixel voisin (i_d, j_d)

$$K1 = j_d - j_1$$

$$K2 = i_d - i_1$$

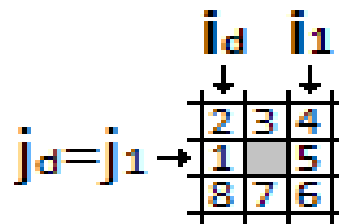


figure 3.11. Pixel de contour

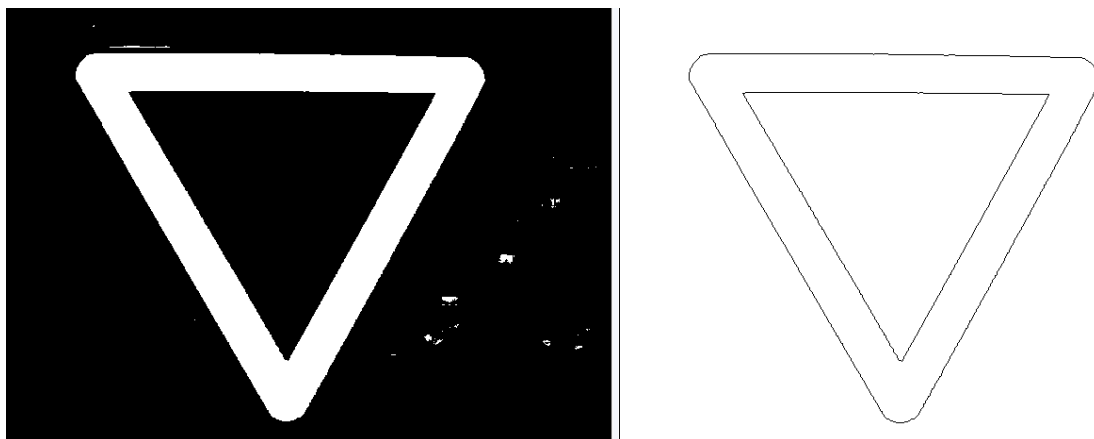


Figure 3.12. Image d'un panneau binarisé et son suivi de contour.

III.4.8 Descripteur de fourrier

Ce dernier est basé sur les coordonnées X, Y des pixels de suivi de contour de chaque forme et après une série des formules (voir chapitre2) on aura un vecteur a la fin correspondant a la forme utilisée. Dans ce projet on a travaillé sur 16 premières valeurs, vu que c'est suffisant d'après des tests, ce qui est comme avantage de cette technique pour réduire les calculs et gagner un peu de temps d'exécution.

III.4.9 Classification

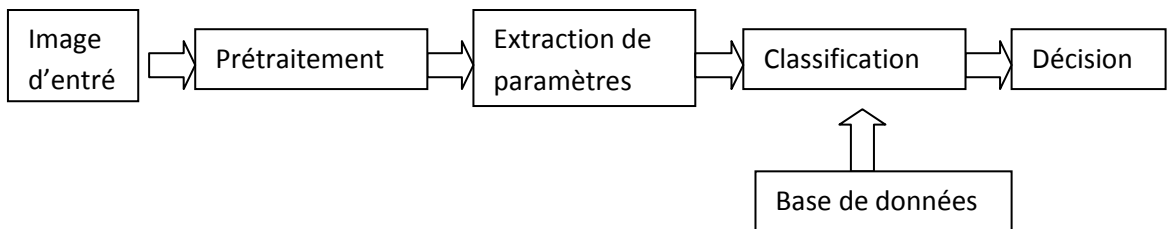


Figure 3.13.. Schéma synoptique de méthode de traitement

- **Méthodes de classification utilisées**

- **Règle du plus proche voisin**

On note par $t_k^i = [t_{k1}^i, t_{k2}^i, \dots, t_{km}^i]$ le k^{ieme} vecteur caractéristique dans la classe i pour $1 \leq m \leq 16$ on note aussi par le \bar{t}_m^i et σ_m^i la valeur de la moyenne et l'écart type de la m ième composante du vecteur caractéristique dans la classe i respectivement

En plus on aura : $\bar{t}_m^i = [:\bar{t}_1^i : \bar{t}_2^i \dots \dots : \bar{t}_{16}^i]$

le vecteur caractéristique moyen de la classe (Feature vector)

La distance entre une forme de test

$X=[x_1, x_2, \dots, x_{16}]^T$ et t_k^i est donnée par :

$$d(X, t_k^i) = \sum_{m=0}^{16} \frac{|x^{(m)} - \bar{t}_m^i|}{\sigma_m^i} \dots \dots \dots (3.1)$$

K : nombre de motif par classe

σ_{tm}^i : L'écart type de m^{ieme} composant de la classe i

Avec la règle du plus proche voisin, la classe i est assigné ou affecté a X si le vecteur caractéristique dans la classe i a le minimum de distance a partir de X parmi toutes les formes d'apprentissage (training).

(le nombre de classe X est le nombre de motif) c a d

X classe i si $d(X, t_k^i) \leq d(X, t_p^j)$

$1 \leq j \leq M$ et $1 \leq p \leq K$

\uparrow \uparrow
 Nbr de classe Nbr de motif par classe

Mais avec la règle de distance moyenne minimale, la classe i est affecté à X, si le vecteur caractéristique moyen de la classe i a la distance minimale a partir de X parmi tous les vecteurs moyens :

C a d:

X classe i, si $d(X, \bar{t}_k^i) \leq d(x, \bar{t}_p^j)$ tel que $0 \leq j \leq M$

$$d(X, \bar{t}_k^i) = \sum_{m=0}^{16} \frac{|x^{(m)} - \bar{t}_m^i|}{\sigma_m^i} \dots \dots \dots (3.2)$$

➤ **Distance Euclidienne**

$$d^2(x_1, x_2) = \sum_i (x_1 - x_2)^2 \dots \dots \dots \text{(voir page33)}$$

III.4.9.1 Création de la base d'apprentissage

L'apprentissage consiste à présenter au classifieur plusieurs exemples de panneaux de signalisation et de lui indiquer à chaque fois de quel type de panneau il s'agit.

Tous ces exemples sont placés dans une base appelée ensemble d'apprentissage.

Un exemple est formé d'un vecteur d'entrée dont les composantes sont les valeurs des paramètres caractéristiques et d'un vecteur de sortie qui code la classe de l'exemple.

III.5 Implémentation

III.5.1. Description de l'interface

Nous allons découvrir l'interface de notre application (*figure 3.14*) dans la partie supérieure nous avons les différents boutons et afficheurs qui nous permettent d'entamer les différentes procédures de l'analyse de notre image.

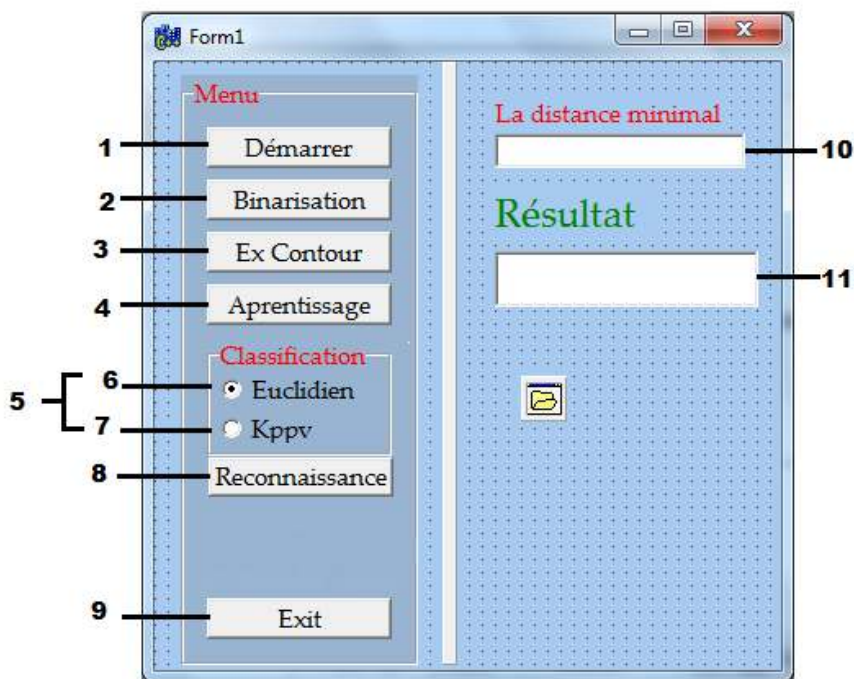


Figure 3.14 : Interface principale.

- 1- Bouton pour l'acquisition d'image.
- 2- Bouton pour binariser l'image.
- 3- Bouton pour extraire le contour.
- 4- Bouton pour l'apprentissage.
- 5- Espace pour la classification.
- 6- Bouton radio pour choisir la méthode Euclidienne de la classification.
- 7- Bouton radio pour choisir la méthode Kppv de la classification.
- 8- Bouton pour la reconnaissance.
- 9- bouton pour quitter l'application.
- 10- afficheur pour la distance minimale.
- 11- afficheur pour la décision.

a-L'acquisition d'image

Pour commencer on doit sélectionner l'image a traiter par le bouton démarrer.

(voir *figure3.15*)

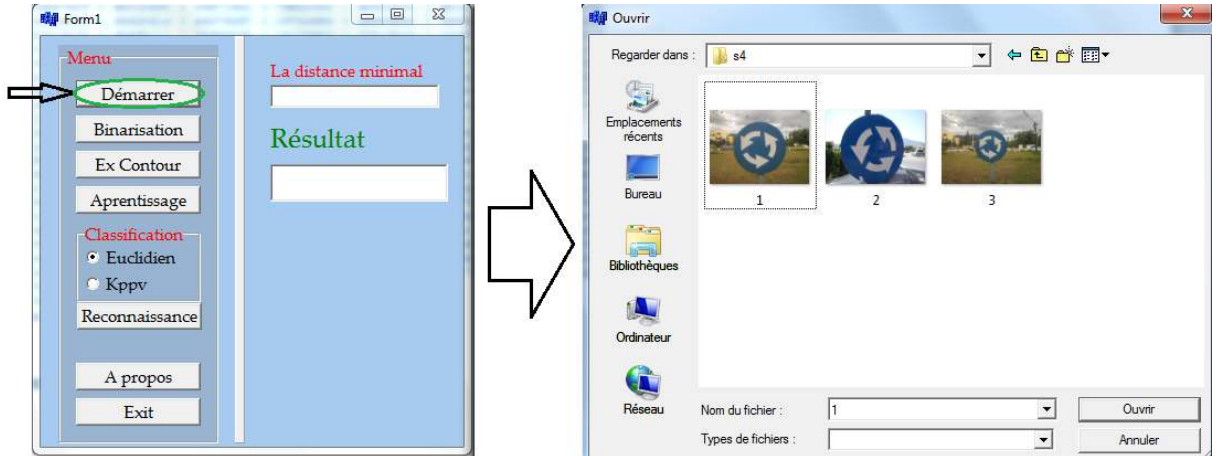


Figure 3.15 acquisition d'image.

b-Binarisation

On utilise dans notre application la binarisation par seuillage (voir page 44). Le résultat de binarisation est représenté dans la figure suivante.

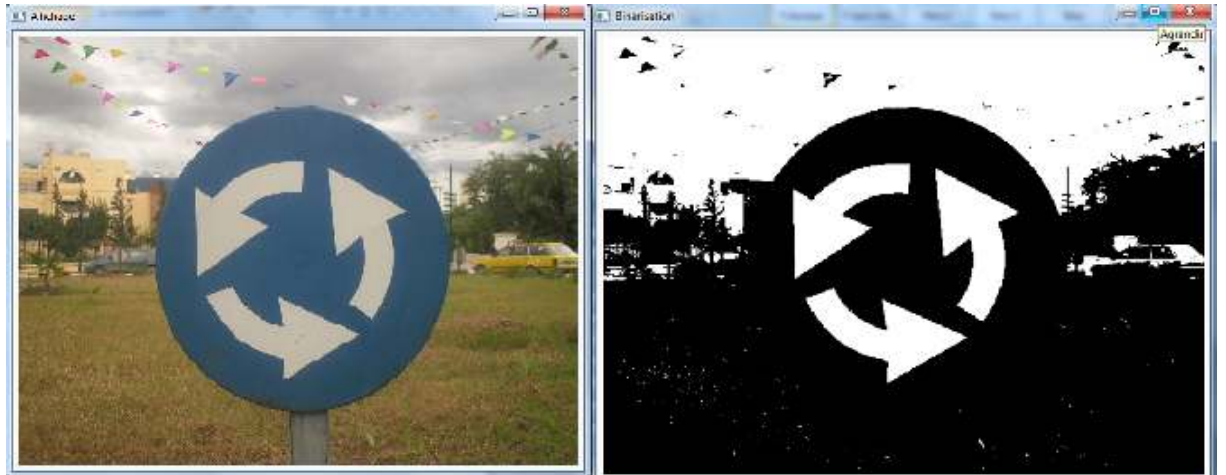


Figure 3.16. Binarisation..

c-L'extraction de contour

Dans cette étape on applique la conversion RGB/HSV, l'extraction de couleur, conversion HSV/RGB, conversion en niveau de gris, binarisation, puis extraction de contour (voir pages 43,44).

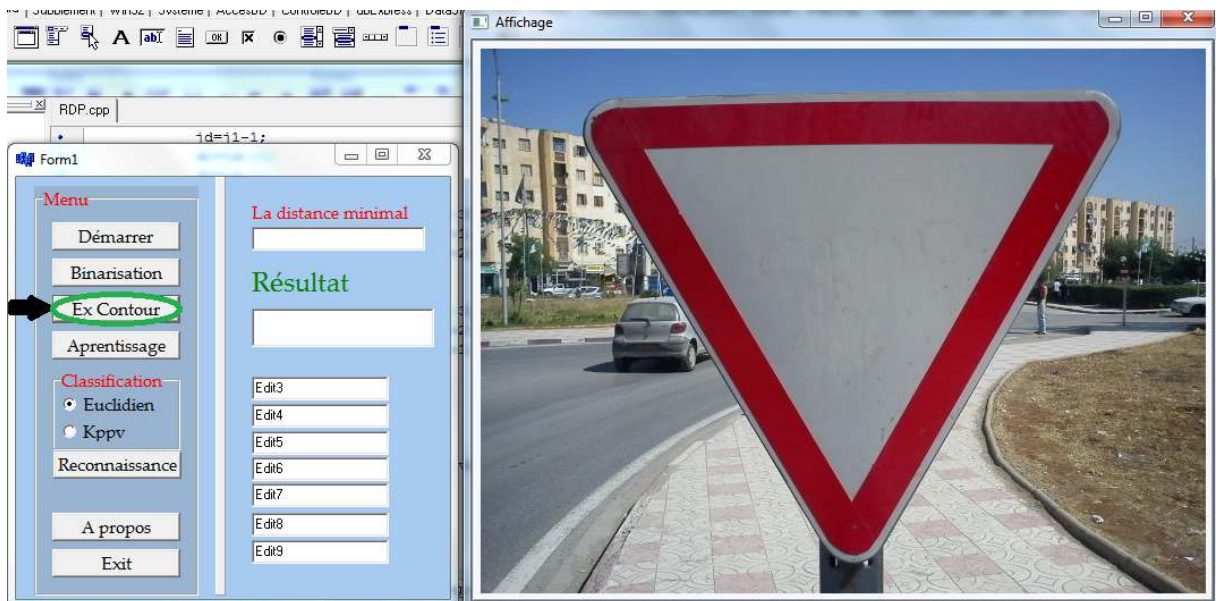


Figure 3.17. Extraction de contour.

d-L'apprentissage

On fait l'apprentissage une fois le programme est démarré.

Il s'agit de charger la base de données. Voici la base de données.

```
0.36735770 0.35090402 0.65955865 0.33522207 0.19262537 0.36899063
0.09628516 0.05622746 0.10820689 0.31340113 0.09422020 0.08558868
0.10291230 0.26420239 0.04239075 0.09190387 0.06418587 0.14284049
0.03110776 0.12526236 0.11076354 0.09337203 0.01930589 0.06292191
0.11260056 0.03331050 0.03499473 0.03026845 0.07225816 0.03547331
0.03678235 0.05336379 0.38782254 0.44278723 0.19040127 1.09197116
0.02254791 0.16156697 0.12569129 0.33662692 0.94567358 0.19708824
0.03540818 0.10061759 0.20135278 0.09275118 0.10866731 0.05705545
8.45498778e-003 0.28923079 0.18565285 0.04465470 0.04378912
0.01556433 0.14561196 0.03909945 0.02695004 0.01910442 0.01920208
0.21472456 0.02462390 0.01861865 0.02590348 0.02777055 1.66299796
0.68028879 3.60515738 0.09950746 0.96395677 0.28186041 1.10637498
0.09823452 0.24686471 0.13603179 0.39486930 0.26843005 0.05163567
0.08165496 0.52590919 0.10246214 0.04993474 0.23844291 0.39655435
0.07320047 0.10079713 0.07772660 0.39970943 0.16688207 0.12291998
0.09266680 0.34126151 0.07975759 0.14106648 0.11212763 0.25371307
0.02867429 0.44010580 0.39671901 0.73922002 1.55301201 1.46151042
1.50356555 0.33620614 0.43279397 0.23753893 0.26689360 0.09855632
0.81513125 0.35890111 0.05455525 0.06159629 0.07819009 0.10287503
0.21774547 0.25381771 0.05968061 0.02086548 0.20271119 0.06391136
0.03363743 0.34680274 0.01995461 0.21023253 0.02060873 0.02117046
0.03816358 0.12072840 0.18615447 0.14208241 0.55345160 0.65755987
0.08507510 0.28070578 2.23532963 0.42818680 0.07986943 0.12240670
0.15183915 0.01923669 0.14801086 0.66798490 0.30633816 0.06382930
0.10976875 0.10505477 0.04628330 0.08774286 0.31606808 0.19917749
0.04375842 0.08736837 0.07322235 0.05880214 0.10781583 0.20942986
0.15235275 0.05877537 0.06905753 0.01753423 0.02040248 0.66660386
0.58142984 1.54417455 0.09533124 0.23294243 0.30120161 0.37558362
0.08150084 0.11399012 0.22151345 0.05375706 0.03936956 0.06521598
0.27436054 0.05188315 0.07358150 0.08556268 0.10057066 0.02935219
0.06956974 0.13590831 0.03747072 0.01216539 0.02750969 0.14778957
0.01738927 0.04513032 0.04851131 0.10320126 0.02699312 0.04258562
0.08196422 0.44499192 0.37253335 0.44727373 0.44950411 0.27167997
0.08247995 0.11249343 0.10227821 0.02948860 0.07385282 0.11155074
0.06630563 0.03389977 0.02464210 0.04043144 0.04648380 0.06982911
0.04377510 0.02121115 0.02571961 0.02323982 0.02378541 0.03877660
0.03205257 0.02244232 0.01791191 0.02135306 9.86198429e-003
0.02597021 0.03243751 0.01061567 0.01489131</data></mat>
</opencv_storage>
```

e-La reconnaissance

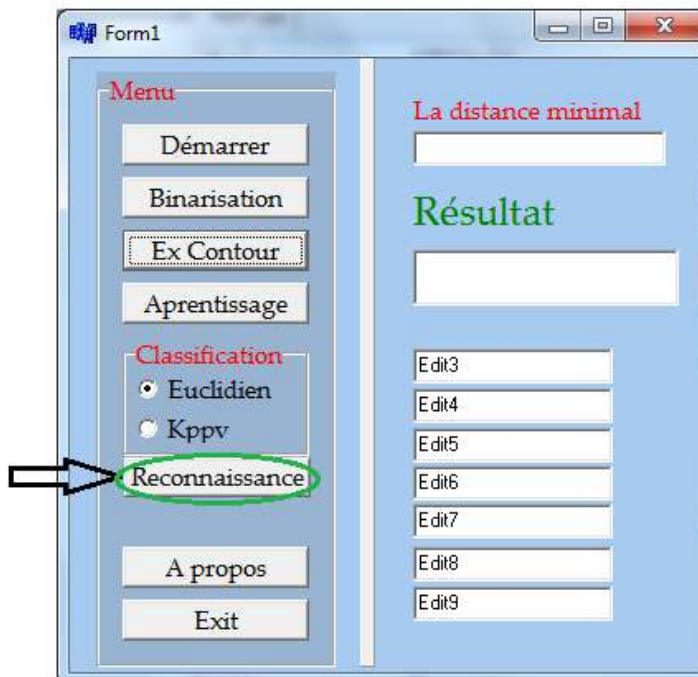


Figure 3.18. Reconnaissance.

III.5.2.Résultats

a-résultat d'acquisition d'image

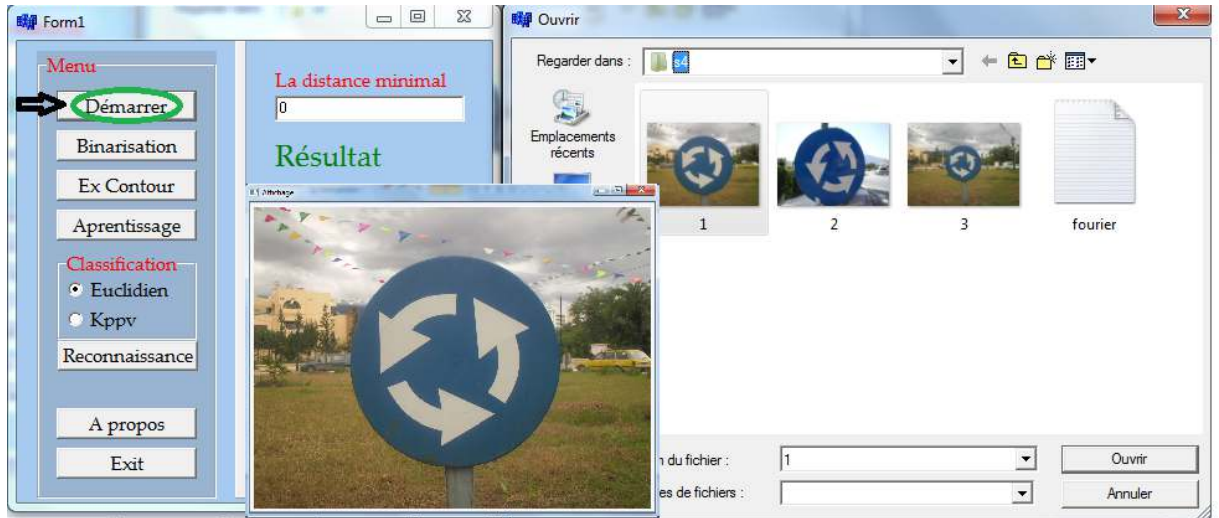


Figure 3.19 Résultat de l'acquisition

b-résultat d'extraction de contour

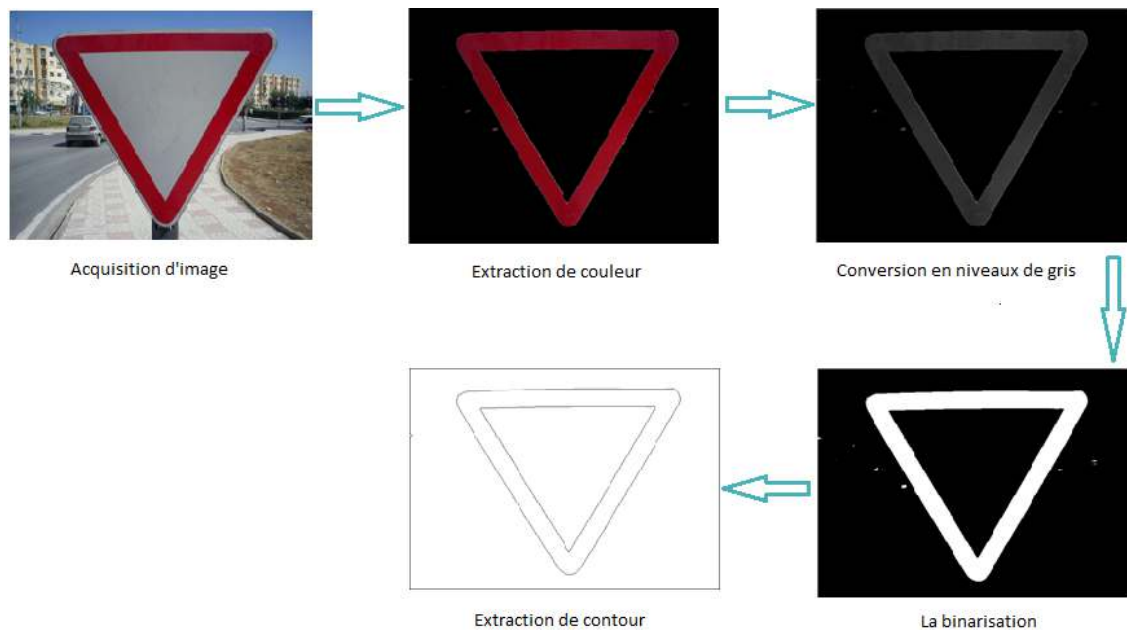


Figure 3.20 Extraction de contour.

c-Décision

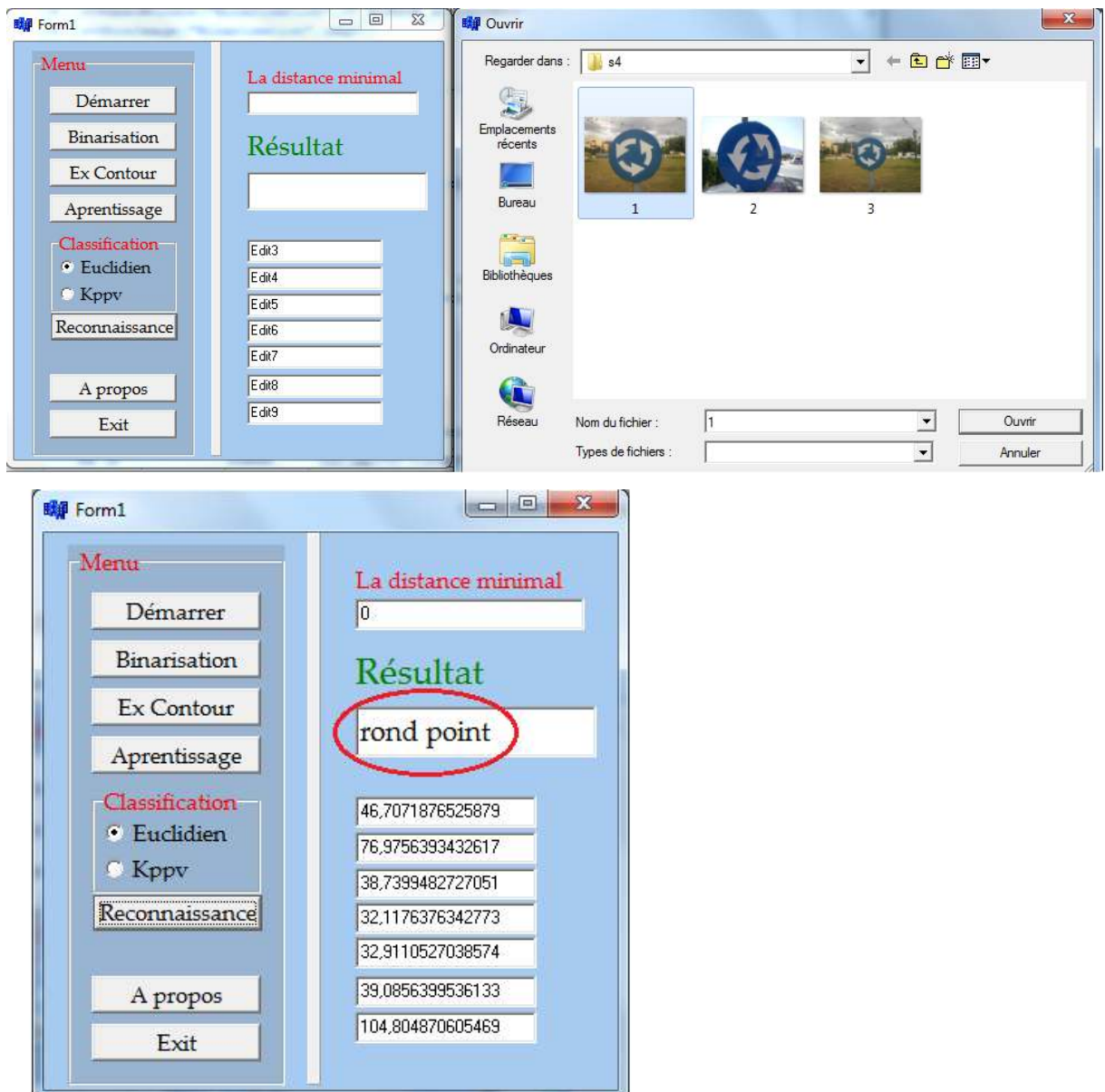


Figure 3.21 Décision finale

III.5.3 Tests

On a dévisé les panneaux de signalisation en 3 catégories : Interdiction, obligation, danger.

- **Les graphs de reconnaissance**

Après le test des panneaux de signalisation de la base de données on est arrivé à une reconnaissance de 100% pour chaque catégorie

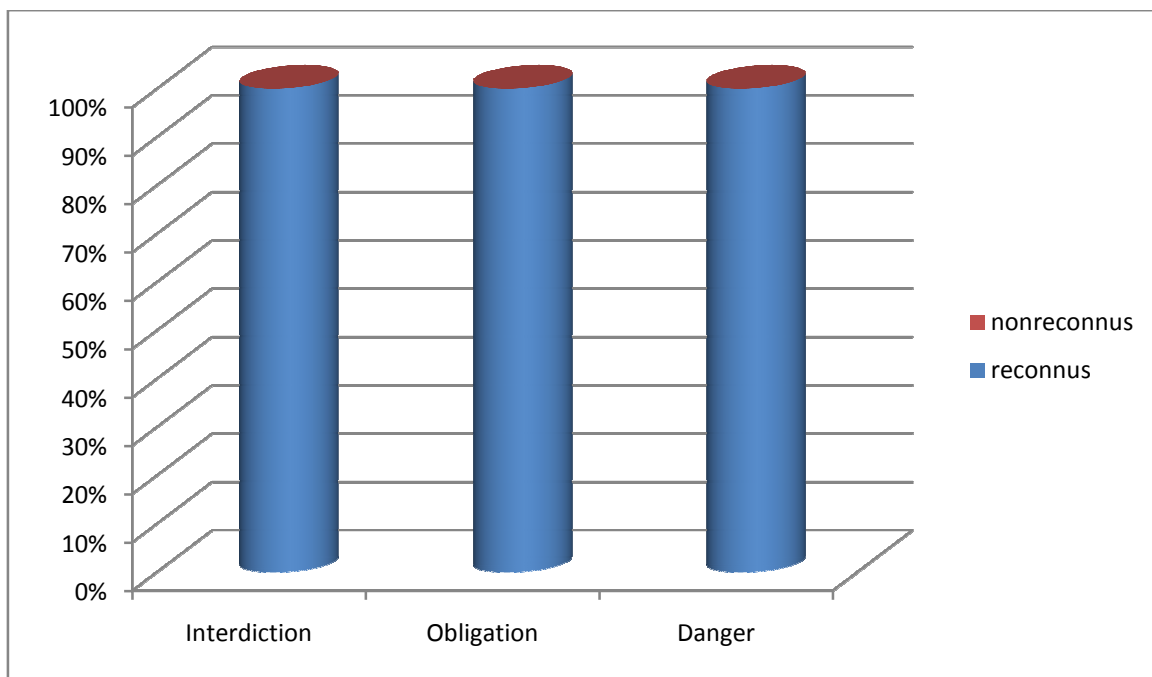


Figure 3.22 Base d'apprentissage

Après le test d'autres panneaux de signalisation (hors base de données) on a eu les résultats suivants :

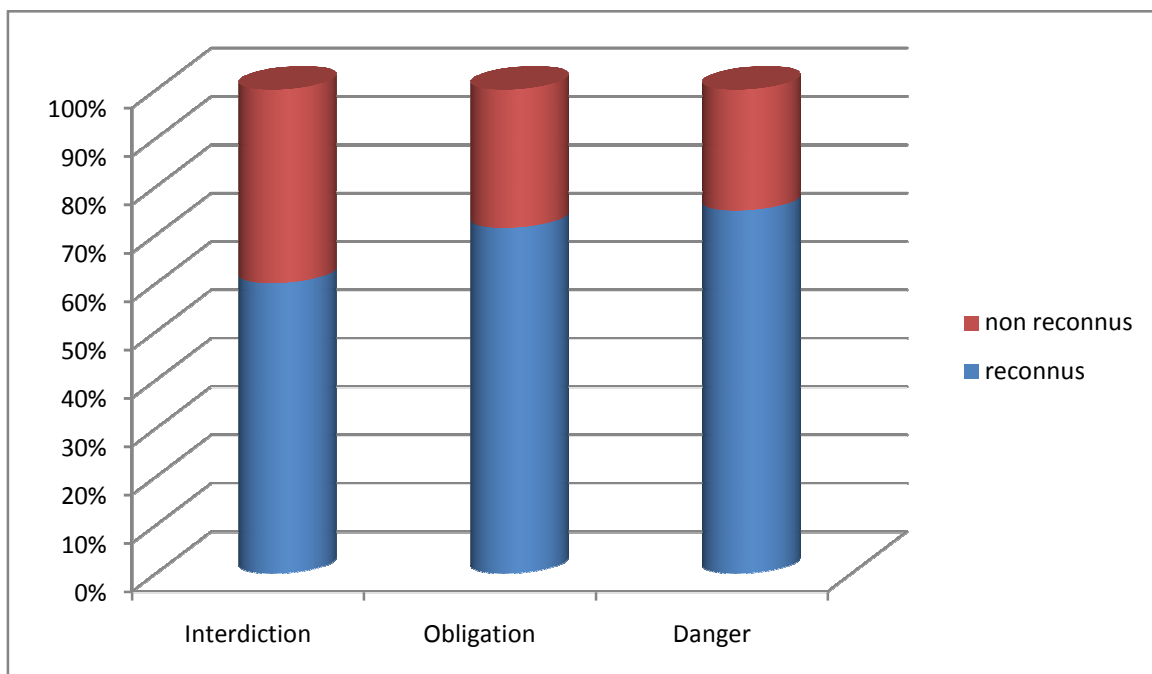


Figure 3.23 Ensemble test

III.6 Conclusion

Le passage de l'algorithme théorique à l'application pratique était la phase la plus importante et essentielle, la technique que nous avons utilisée pour la reconnaissance des panneaux routiers nous a donné de bons résultats malgré les contraintes auxquelles nous sommes confrontés (fausses alarmes, la luminance, les objets.....)

Après de longues séries de tests nous avons réussi à trouver la combinaison de paramètres qui nous convient.

Conclusion générale

Le travail présenté dans ce mémoire a été effectué au sein du département d'électronique université de Blida.

Pour réaliser un tel travail nous avons été confrontés à des problèmes liés à l'aspect expérimental. L'environnement étant très complexe, les premiers problèmes concernaient les occultations, le changement de luminosité, la grande variation de texture et surtout la grande différence entre les panneaux routiers (couleur, position par rapport à la caméra, etc.).

Nous avons présenté en premier lieu la vision assistée par ordinateur, sa définition, domaine d'utilisation puis on a détaillé quelques outils de traitement d'image qu'on a utilisé dans nos programmes.

En deuxième lieu, nous avons réalisé une étude sur les méthodes de reconnaissance d'objets, on se focalisant surtout sur celles qui reconnaissent les panneaux routiers. Vu la difficulté de cette tâche, peu de techniques récentes ont pu donner d'assez bons résultats. Après cela, nous les avons évalué et choisis quelques unes pour l'implémentation.

En troisième étape, on a conçu un algorithme et l'implémenté, avec la conception des bases de données qui était une tâche assez importante, ensuite, nous avons tenté d'améliorer notre application en testant ses performances pour différentes valeurs de paramètres

Une étape importante est l'évaluation des performances, on a testé cette application sur des bases de tests d'autres conçus sur des images réelles depuis notre caméra. Les résultats étaient assez satisfaisant, le temps de reconnaissance aussi.

Au cours de ce projet et à travers le travail fait, nous avons pu acquérir beaucoup de connaissances. Premièrement, nous avons utilisé les outils appris au cours de notre formation, et enrichi nos connaissances dans le domaine du traitement d'image en apprenant de nouveaux outils (OpenCV). On a aussi appris à utiliser le langage de programmation C/C++.

On est satisfait de notre contribution pour le projet, surtout qu'on a un bon taux de reconnaissance. Nous estimons que notre module pourra être exploité sur les voitures roulantes ou encore l'observation et la gestion du trafic.

Bibliographie

- [1] M. Badeche, *“Suivi de modèle et Incrustation d’Objets Virtuels pour la Réalité Augmentée ”*, Thèse de magistère, Université de Batna, 2006.
- [2] <http://www.map.toulouse.archi.fr>, *“Support de cours”*, Ecole nationale supérieure d’architecture de Toulouse, 2002.
- [3] D. Loulha& M. Boutheldja, *“Extraction d’informations et modélisation d’une carte de contours par la transformée de hough”*, PFE 2003
- [4] Daniel Pourcelot, *“Histogramme”*, Club Besançon Déclic image, octobre 2010.
- [5] Jeff Schewe, *“traitement d’images”*, comment ça marche.net, octobre 2008.
- [6] B. GUY, *“Analyse d’image”*, Axe Génie des Procédés, centre SPIN, Ecole des Mines de Saint-Etienne, 2010.
- [7] Vincent BARRA, Cours de *“traitement d’images”*.
- [8] http://fr.wikipedia.org/wiki/Panneau_de_signalisation_routi%C3%A8re
- [9] Lalonde, M., Li, Y., 1995. Road Sign Recognition - Survey of the State of the Art for Sub-Project 2.4,
CRIM Centre de recherche informatique de Montréal
<http://www.iro.umontreal.ca/~benallam/crim/roadsign.p>
- [10] Rozenn Dahyot - Analyse d'images séquentielles de scènes routières par modèles d'apparence pour la gestion du réseau routier, Rozenn Dahyot, PhDThesis University of Strasbourg France,
November 2001
- [11] Projet ARGO Université de Parma -
<http://millemgia.ce.unipr.it/ARGO/english/index.html>
274
Mohamed Bénallal 29 novembre 2002
- [12] TSR - Realtime Traffic Sign Recognition – University Koblenz Landau -
http://www.unikoblenz.de/~lb/lb_research/research.tsr.html
- [13] The Road Sign Recognition System - RS2 -
<http://euler.fd.cvut.cz/research/rs2/index.html>
- [14] Projet EUCAR European Council for Automotive R&D
<http://europa.eu.int/comm/research/growth/gcc/projects/eucar.html>

[15]support de cours " traitement de signal numérique"

[16] Freeman, H. (1977). *Computer processing of line drawing images*. *ACM Computer Survey*, 6 :57–97.

[17] Markus Enzweiler, Dariu Gavrilă, "Monocular Pedestrian Detection:Survey and Experiments", TPAMI, Vol 31, n° 12, p. 2179-2195, Décembre 2009.

[18]Support de cours Vision industriel Master2 système de vision et robotique.

[19] Claire MAHEO, "Méthodes de suivi d'un objet en mouvement sur une vidéo", Université Espagne du 15/06/09.

[20] A. Chottera and M. Shridhar, "Feature extraction of manufactured parts in the presence of spurious surface reflections,"*Canadian Electron. J.*, vol. 7, no. 4, pp. 29-33, 1982.

Annexe

La signalisation routière est aussi vieille que les routes. Les premiers signes furent ceux donnant les directions. L'empire romain a érigé des bornes, en colonnes de pierres, le long des routes pour indiquer les distances jusqu'à Rome. Au moyen âge, les signaux directionnels devinrent courants, indiquant la direction des villes aux carrefours. Inscrites sur des colonnes ou des bornes, les indications sont progressivement, à partir du XVIII^e et au XIX^e siècle, placées sur des plaques (en fonte, à partir de 1835), dites plaques de cocher, fixées sur les murs, les croix ou obélisques des carrefours, ou sur des poteaux métalliques.

La signalisation devint beaucoup plus importante avec le développement de l'automobile. Les principes de base de la plupart des signalisations furent définis le 11 octobre 1908 lors du premier congrès international de la route à Rome.

La représentation symbolique sur les panneaux de circulation (par des pictogrammes, des images de carrefours, etc.) en remplacement de textes, apparaît dès le début du siècle : 1902 en France, 1903 en Italie, 1907 en Allemagne. La convention internationale de Genève de 1909 standardise au niveau international quatre types de signaux : virage, croisement, cassis et passage à niveau¹.

Depuis cette époque, des changements considérables sont intervenus et le nombre de panneaux différents a considérablement augmenté. En France, le nombre est ainsi passé de 4 en 1909 à 384 en 2007.

Pour plus de détail sur l'évolution des signaux, voir

Article détaillé : Signalisation [routière / Histoire](#).

L'évolution de la nature des panneaux, considérés comme dispositifs sur lesquels sont implantés des signaux routiers, a quant à elle suivi l'évolution des techniques et des technologies.






Dimensions :

Les dimensions des panneaux de signalisation doivent être telles que le signal soit facilement visible de loin et facilement compréhensible quand on s'en approche. Elles tiennent compte de la vitesse usuelle des véhicules³.

Chaque pays définit sa propre réglementation concernant les dimensions de panneaux. En règle générale, il y a quatre catégories de dimensions pour chaque type de signal : petites, normales, grandes et très grandes dimensions

Les signaux de petites dimensions sont employés lorsque les conditions ne permettent pas l'emploi de signaux de dimensions normales ou lorsque la circulation ne peut se faire qu'à allure lente. Ils peuvent aussi être employés pour répéter un signal antérieur. Les signaux de grandes dimensions seront employés sur les routes de grande largeur à circulation rapide. Les signaux de très grandes dimensions seront employés sur les routes à circulation très rapide, notamment sur les autoroutes⁴.

Dimensions des panneaux de signalisation routière

Gamme et type	 A	 AB4	 B	 AB6	 C et CE
Exceptionnelle					1 500 mm
Supérieure					1 200 mm
Très grande	1 500 mm	1 200 mm	1 250 mm	1 050 mm	1 050 mm
Grande	1 250 mm	1 000 mm	1 050 mm	900 mm	900 mm
Normale	1 000 mm	800 mm	850 mm	700 mm	700 mm
Petite	700 mm	600 mm	650 mm	500 mm	500 mm
Miniature	500 mm	400 mm	450 mm	350 mm	350 mm

Position

Les panneaux de signalisation sont placés de manière à pouvoir être reconnus aisément et à temps par les conducteurs auxquels ils s'adressent. Habituellement, ils sont placés du côté de la route correspondant au sens de la circulation. Toutefois, ils peuvent être placés ou être répétés au-dessus de la chaussée⁵.

Tout signal placé du côté de la route correspondant au sens de la circulation devra être répété au-dessus ou de l'autre côté de la chaussée lorsque les conditions locales sont telles qu'il risquerait de ne pas être aperçu à temps par les conducteurs auxquels il s'adresse.

Les différents types de panneaux de signalisation

Interdiction



B12 Accès interdits véhicules dont la hauteur, chargement compris, excède le nombre indiqué.



B13 Accès interdits véhicules dont le poids total autorisé excède le nombre indiqué.



B14 Limitation de vitesse. Ce panneau a toute l'importance de dépasser la vitesse indiquée.



B14 Limitation de vitesse.



B16 Signal sonore interdit.

Obligation



B21-1 Obligation de tourner à droite au feu rouge.



B21c Direction obligatoire à la prochaine intersection : à droite.



B21c2 Direction obligatoire à la prochaine intersection : à gauche.



B21d1 Direction obligatoire à la prochaine intersection : à droite et à gauche.



B21d2 Direction obligatoire à la prochaine intersection : à droite et à gauche.



B21e Direction obligatoire à la prochaine intersection : à droite et à gauche.



B22a Parc ou bande obligatoire pour les cycles, trottinette ou remorque.



B22b Chemin obligatoire pour piétons.

Danger



A17 : Annonce de feux tricolores



A15c : Passage de cavaliers



A16 : Descente montagneuse



A19 : Risque de chute de



A20 : Descente sur un quai sur une berge



A18 : Circulation dans les deux sens