Détection des panneaux de limitation de vitesse

1.1. Motivations

La détection et la reconnaissance des panneaux signalétiques (*TSR - Traffic Sign Recognition*) a connu un grand engouement ces dernières années. Cela est dû à la grande quantité d'applications qu'un tel système est capable de fournir :

1. Maintenance des panneaux

De nos jours, un opérateur humain doit visionner des films afin de vérifier la présence et l'aspect des panneaux sur les routes. C'est une tâche pénible car les panneaux n'apparaissent que de temps en temps et parce que l'opérateur doit être extrêmement attentif.

2. Inventaire des panneaux

Répertorier les panneaux présents sur les routes permet entre autre de pouvoir réaliser des bases de données des informations utiles au conducteur (c'est notamment ce qu'utilisent les systèmes GPS embarqués).

3. Aide à la conduite

Les signes apparaissent toujours clairement dans l'environnement. Cependant, si ils sont distraits ou s'ils ont un manque de concentration, il arrive que les conducteurs ne voient pas un panneau. Or le dépassement de la vitesse limite autorisée est l'une des causes principales d'accident de la route. Il est alors important qu'un système puisse leur fournir l'information qu'ils viennent de rater et de les prévenir si ils roulent trop vite ou trop lentement.

4. Véhicules intelligents autonomes

Pour des véhicules complètement autonomes (sans conducteur humain), c'est un point primordial que de reconnaître les panneaux signalétiques qui fournissent un grand nombre d'informations utiles à la navigation.

1.2. Etat de l'art

Les recherches dans le domaine du TSR (*Traffic Sign Recognition*) ont commencé à partir des années 1980. La première approche naïve et directe est d'appliquer un filtre de corrélation croisée normalisée aux images brutes afin à la fois de détecter et de reconnaître les panneaux. Cependant cette méthode par brute force demande un temps de calcul beaucoup trop important pour être réellement envisageable (les panneaux pouvant avoir n'importe quelle taille et se trouver n'importe où dans l'image).

C'est ainsi que très vite, dans les années 1990 et encore de nos jours, des approches généralistes basées sur des méta-heuristiques (des algorithmes d'optimisation stochastique, hybridés avec une recherche locale) ont été utilisées afin de ne pas parcourir tout l'espace de recherche (i.e. la position et la taille des panneaux dans une image) et de trouver une solution optimale au problème en un temps raisonnable. Ont notamment été utilisées les approches par recuit simulé (Betke, et al., 1994), ainsi que par les algorithmes génétiques (Escalera, et al., 2003) (Aoyagi, et al., 1996). Cependant les méta-heuristiques ne garantissent pas de fournir la solution optimale au problème ni même une solution et un compromis doit être trouvé entre le temps d'exploration de l'espace de recherche et la qualité de la solution, comme le relève (Gavrila, 1999).

En revanche, il est possible d'effectuer le processus de corrélation de manière complètement optique, c'est-à-dire à la vitesse de la lumière en ne manipulant que l'onde lumineuse (Horache, 2001). Cet avantage a motivé beaucoup d'auteurs pour des applications telle que la reconnaissance de formes en temps réel et notamment la détection automatique de panneaux routiers (Guibert, 1995) (Petillot, 1996) (Guibert, et al., 1995). Malheureusement les corrélateurs optiques souffrent de sévères inconvénients tels que le coût et la fragilité des composants optiques, l'encombrement de ces architectures, la nécessité d'un alignement précis pour les traitements effectués et le manque de robustesse mécanique de l'optique pour des applications embarquées.

D'autres approches plus utopiques ont aussi été proposées pendant les années 90 comme dans (Pacheco, et al., 1994) où ils émettent l'idée d'ajouter des codes barre de couleur sous les panneaux afin d'aider à leur reconnaissance. De nos jours on pourrait vouloir rajouter des puces RFID aux panneaux, mais ces solutions demanderaient trop de temps et d'argent pour remplacer tous les panneaux.

Mise à part la méthode de corrélation croisée, toutes les recherches actuelles découpent le processus de TSR en trois parties comme le soulève (Gavrila, 1999):

1. Détection
2. Reconnaissance
3. Intégration temporelle

1.2.1. Détection

Les recherches dans la détection des panneaux signalétiques dans l'image, c'est-à-dire trouver des régions d'intérêts (ROI) où ils se situent, peuvent être classées en deux grandes catégories.

a) Segmentation par couleur

Comme relevé par (Bahlmann, et al., 2005), la segmentation par la couleur est la méthode la plus couramment utilisée dans l'étape initiale qu'est la détection des panneaux. Naïvement, de nombreux panneaux étant cerclés de rouge, de nombreux travaux (de la Escalera, et al., 1997) (Kim, et al., 1997) (Zadeh, et al., 1998) (Torresen, et al., 2004) effectuent, pour détecter ceux-ci, un seuillage sur la composante rouge du modèle RGB. Dans de rares cas, l'utilisation du modèle YUV est utilisé, comme dans (Miura, et al., 2000) où ils extraient les régions blanches de l'image (qui correspondent aux zones intérieures des panneaux). Cependant, ces modèles de couleurs sont sensibles aux changements de conditions lumineuses. C'est ainsi que (Miura, et al., 2000) utilise plusieurs seuils différents pour binariser ses images, il en résulte que pour chaque image de nombreuses autres doivent être traitées, ce qui alourdit considérablement le temps de calcul en rendant l'algorithme non utilisable pour le temps réel. La plupart des travaux (Escalera, et al., 2003) (Hsien, et al., 2003) (Piccioli, et al., 1996) (Arnoul, et al., 1996) (Hibi, 1996) (Fang, et al., 2003) (Fang, et al., 2003) se basent donc généralement sur le modèle HSV (ou HSI), ou même sur l'espace de couleur Luv (Kang, et al., 1994), qui sont invariants aux changements de conditions lumineuses. Cependant, comme le stipulent (Loy, et al., 2004) (Garcia, et al., 2006), l'image provenant de la caméra n'est pas complètement invariante aux changements chromatiques de la lumière qu'elle reçoit. En effet, la composante Hue change avec les ombres, les conditions climatiques et de plus comme l'explique (Thomson, et al., 2001) la couleur des panneaux s'estompe avec le temps. Ces méthodes de segmentation par couleur peuvent donc poser problème pour la robustesse d'un système TSR et sont de plus plus coûteuses (3 canaux à traiter).

b) Segmentation par contours de forme

Les panneaux de limitation et de fin de limitation de vitesses européens sont de forme circulaire. La méthode la plus connue et la plus robuste dans les techniques de reconnaissances de formes dans le domaine du traitement d'image afin de détecter des cercles est l'algorithme de Hough qui est utilisé dans (Garcia, et al., 2006) (Miura, et al., 2000). Cependant cette technique peut s'avérer fort lente. C'est pourquoi de nombreux travaux essayent de développer d'autres. A partir de la *distance transform* (DT) (Borgefors, 1986) qui calcule la distance pour chaque pixel de l'image au contour le plus proche, il est possible d'effectuer une corrélation « rapide » (Gavrila, 1999) pour identifier la position et la tailles de panneaux potentiels dans une image. Cette technique souffre quand même d'un temps de calcul largement trop important (dû au fait de calculer la DT des images et du processus de corrélation) pour être employé en temps réel. Une variation de l'algorithme de Hough a été proposée par (Barnes, et al., 2004) basée sur les travaux de (Loy, et al., 2003) qui propose d'avoir non pas un accumulateur comme dans l'algorithme classique, mais deux où le second accumule l'amplitude du gradient. Cette approche ne peut donc pas être plus rapide que l'algorithme classique. Une autre variation est proposée dans (Ajdari Rad, et al., 2003) où l'idée est de retrouver simplement quelques paires de vecteurs sur l'image de gradient, appariés selon le diamètre des cercles

recherchés. Malheureusement cette technique, qui est plus robuste au bruit que l'algorithme de Hough, implique d'avoir des images fortement contrastées, ce qui n'est pas le cas dans la réalité pour des images routières. Dans (de la Escalera, et al., 1997), ils détectent les coins dans les images (après seuillage sur la composante rouge, donc ils souffrent toujours du changement de conditions lumineuses) par un processus de convolution avec des masques spécifiques, afin de trouver les formes des panneaux souhaités. Cette méthode implique que tous les coins doivent être détectés et donc elle ne peut fonctionner dans le cas d'occlusion partielle des panneaux. De plus, comme le même masque est utilisé à la fois pour détecter les panneaux rectangulaires et circulaires, il est impossible de différentier un panneau carré d'un panneau rond, ce qui rend la méthode encore plus difficilement utilisable en pratique.

Il est à noter que certaines approches essayent de combiner les deux techniques (détection par couleur et par contours de forme) soit de façon parallèle (Fang, et al., 2003), soit successive. Cela revient à effectuer la seconde méthode uniquement sur les régions d'intérêts trouvées par la première (de la Escalera, et al., 1997) (Miura, et al., 2000) (Priese, et al., 1993). Ces techniques, en incluant l'approche par segmentation couleur, souffrent toujours essentiellement des changements de conditions lumineuses. Par exemple, (Priese, et al., 1993) utilise l'algorithme de croissance de régions pour trouver les cercles à partir d'une pré-segmentation des couleurs dans l'image.

Il est aussi possible d'utiliser les méthodes issues du domaine de l'apprentissage numérique (ou de l'intelligence artificielle) afin d'apprendre à un algorithme à trouver les régions d'une image contenant des panneaux. Ainsi, (Bahlmann, et al., 2005) utilise la récente technique du boosting dont l'idée est de combiner beaucoup de petits classificateurs faibles (c.à.d. qui arrivent à discriminer deux classes avec un taux légèrement supérieur à 50%) afin d'obtenir un classificateur fort (c.à.d. robuste). Dans cette technique, à l'issue de la phase d'apprentissage, les 4 meilleurs classificateurs induits travaillent sur le canal rouge et le 5ème sur l'image en niveau de gris. Cet algorithme souffre donc (à moindre mesure) des changements de variations lumineuses. L'algorithme AdaBoost est utilisé dans (Escalera, et al., 2004) mais cette fois-ci seulement sur des images en niveau de gris. Des réseaux de neurones, de type perceptron multicouche (PMC) sont aussi utilisés dans (Fang, et al., 2003) sur les gradients des images et sur les régions rouges (dans l'espace de couleur HSI), et de type cellular neural network dans (Adorni, et al., 1996). Cependant, apprendre à reconnaître des panneaux dans une image via des réseaux de neurones implique d'avoir un modèle géométrique sur les caractéristiques de ces panneaux (par exemple la taille des anneaux rouges), or celles-ci varient selon la distance des panneaux, ce qui rend ces techniques très difficile à utiliser sur ce type de problème. On peut aussi se demander si les modèles mathématiques induits par ces algorithmes d'apprentissage, lorsqu'ils sont utilisés sur les gradients des images, ne réinventent pas tout bonnement l'algorithme de Hough.

Enfin, certaines approches utilisent des connaissances a priori sur la géométrie de la route, par exemple en supposant que la route est à peu près droite, cela permet de supprimer une large partie de l'image pour l'analyse et la détection des panneaux afin d'accélérer le processus (Hsu, et al., 2001). Cela dit cette supposition se révèle fausse dans les courbes, ou lors de passage de ralentisseurs de type dos d'âne. Une approche plus sophistiquée est de détecter la route et/ou le ciel afin d'éliminer de larges régions. Par exemple (Piccioli, et al., 1996) suggère que les grandes régions

uniformes dans l'image représentent le ciel et la route, et donc de ne rechercher les panneaux qu'entre ces deux zones. Malheureusement, cette approche ne peut fonctionner en environnement urbain où des immeubles sont présents et dans les environnements où les routes sont bordées d'arbres. De plus, n'effectuer la détection que sur les bords de route ne permet pas de détecter les panneaux temporaires (lors de travaux par exemple) posés à même le sol. Une autre idée émise par (Fang, et al., 2003) est de ne rechercher les nouveaux panneaux que près du point de fuite dans l'image. Ceci est vrai, les nouveaux objets apparaissant près de ce point, cela dit, ils seront tellement petits dans l'image qu'ils seront tout bonnement indétectables.

1.2.2. Reconnaissance

Cette étape consiste à reconnaître, dans les zones d'intérêts (ROI) de l'image trouvées précédemment, quels sont effectivement les panneaux si panneau il y a, ou à supprimer les faux positifs (des ROI où aucun panneau n'est en fait présent).

Une grande majorité des travaux (Piccioli, et al., 1996) (Barnes, et al., 2004) (Guibert, 1995) (Seitz, et al., 1991) (Miura, et al., 2000) (Escalera, et al., 2004) se contentent d'effectuer le processus de corrélation croisée, l'algorithme le plus simple à mettre en œuvre, qui sera ici plus rapide car les zones à couvrir sont bien moindres que la taille de l'image originale. Cependant, la taille variable des panneaux à rechercher est toujours un problème, ce qui requiert d'avoir plusieurs masques à différentes échelles. Cela dit, si la première étape retourne des ROI plus ou moins bien cadrés sur les panneaux, la taille du masque à appliquer s'induit naturellement (Guibert, 1995).

D'après (Liu, et al., 2008) qui ont testé différents algorithmes d'apprentissage bien connus du domaine (kNN, MLP, RBF, PC, SVM) pour la reconnaissances de caractères, il apparaît que, sur une base de chiffres manuscrits (donc un peu plus complexe à reconnaître que des chiffres typographiés), tous les algorithmes donnent des résultats satisfaisants (plus de 98.5% de taux de bonne reconnaissance), mis à part le processus de corrélation croisée (ce qui est confirmé par (Simard, et al., 2003)). Ainsi comme le souligne (Piccioli, et al., 1996), la phase de classification est beaucoup moins critique que celle de la détection et le choix de l'algorithme d'apprentissage à utiliser revient à choisir celui avec lequel on est le plus à l'aise à manipuler. Toutefois comme le font remarquer (Liu, et al., 2008) (Simard, et al., 2003), les Convolutional Neural Networks (CNN) se placent un cran au dessus dans la qualité de classification. En effet, la grande force de cette technique est qu'ils apprennent la topologie spatiale des caractères et ne voient donc plus chaque exemple comme juste un ensemble de pixels juxtaposés.

Dans le domaine de la reconnaissance de panneaux, les réseaux de neurones, dans leurs différents types de topologies, sont les plus utilisés parmi les algorithmes de plus « haut niveau ». Une carte de Kohonen est utilisée dans (Luo, et al., 1992), tandis que dans (Adorni, et al., 1996) un réseau de neurone cellulaire est utilisé, de type ART1 dans (Escalera, et al., 2003) et ART2 dans (Fang, et al., 2003), mais la topologie la plus couramment utilisée reste le perceptron multi couche (PMC, ou multi layer perceptron - MLP) (Aoyagi, et al., 1996) (de la Escalera, et al., 1997) (Kang, et al., 1994) (Garcia, et al., 2006) (Torresen, et al., 2004). L'entrée du PMC est une imagette normalisée (sa taille est fixe et son histogramme de niveaux de gris est normalisé), par exemple dans (Aoyagi, et al., 1996) les

images en entrée du réseau de neurone font 18x18 pixels de cotés. Une autre topologie largement utilisée sont les RBF, radial base neural network (Gavrila, 1999) (Janssen, et al., 1993), voire des classificateurs à base de noyaux laplaciens (Paclik, et al., 2000). Curieusement les SVM, bien qu'ayant eu un grand engouement scientifique ces dernières années faces aux réseaux de neurones, ne sont que très peu utilisés ; (Liu, et al., 2008) explique cela par le fait que cette technique est d'une grande complexité pour l'apprentissage.

L'un des problèmes majeurs de ces types d'apprentissage se situe dans le fait qu'ils nécessitent d'avoir une base de données d'exemples (i.e. la base d'apprentissage) assez conséquente afin d'obtenir de bons résultats. (Liu, et al., 2008) stipule qu'utiliser des classificateurs statistiques, tels que les mixtures modèle (ex : mixture de Gaussiennes comme dans (Bahlmann, et al., 2005)), permet une meilleur généralisation lors de l'utilisation de petites bases d'apprentissage, mais les modèles neuronaux les dépassent largement sur de plus vastes.

(Soetedjo, et al., 2005) note cette même nécessité de disposer d'une base de données assez conséquente. Ils développent donc une technique basée sur la corrélation d'histogramme, ce qui leur permet de n'avoir besoin que de quelques exemples dans leur base d'apprentissage. Comme deux images différentes peuvent avoir deux histogrammes identiques (cas bien connus en traitement d'image), ils partitionnent leurs images en sous parties circulaires concentriques et n'effectuent le processus de reconnaissance que sur les histogrammes de chacune de ces sous parties. L'inconvénient majeur de leur méthode résulte dans le fait que l'histogramme doit être calculé sur une image en niveau de gris invariant à la lumière et que cela n'existe pour ainsi dire pas.

La grande majorité des cas, voir tous mise à part (Torresen, et al., 2004) se contentent d'effectuer l'apprentissage et la reconnaissance directement sur les ROI détectées lors de la première étape, c'est-à-dire sur tout un panneau (plus ou moins bien cadré). Cela impose plusieurs problèmes, par exemple si la ROI est mal centrée, il faut que l'algorithme d'apprentissage soit invariant aux translations, à la taille,… De plus cela impose une plus grande quantité d'informations en surplus à traiter par l'algorithme d'apprentissage ce qui influe forcément sur la complexité et les résultats de cet algorithme. Dans le même genre d'idées, bien que les panneaux soient normalement placés perpendiculairement à la route et droits (i.e. les chiffres sont alignés sur une droite parallèle au sol), certains d'entres eux peuvent être mal installés ou avoir subi des dommages (accidents mineurs) et donc apparaître avec une rotation.

Afin de gérer les déformations, rotations,… il existe trois grandes techniques comme le note (Cecotti, et al., 2004). Il est possible de rajouter des images dans la base d'apprentissage en transformant les exemples déjà présents en leur rajoutant par exemple du bruit, en les déformants, en changeant leur luminance,… comme le font (de la Escalera, et al., 1997) (Simard, et al., 2003). Il est aussi possible de redresser les images comme dans (Escalera, et al., 2004) où ils détectent des ellipses et non des cercles dans l'image, ce qui leur permet de remettre à plat un panneau qui pourrait apparaître déformé dans l'image. Enfin, la dernière technique est d'effectuer la reconnaissance sur la transformée de Fourier des ROI, transformée qui est invariante au bruit, à la rotation, à la translation et à l'échelle des panneaux détectés, comme le fait (Kang, et al., 1994). A noter que d'après (Cecotti, et al., 2004), la transformée de Fourrier-Melin est plus performante, pour cette application, que la transformée simple de Fourrier. (Cecotti, et al., 2004) conclut sur ces trois techniques que la

première et la dernière (ajouter des exemples ou utiliser la transformée de Fourrier) donnent des bons résultats. Cependant, ajouter des exemples donne de meilleurs résultats, car la reconnaissance s'effectue sur toute l'imagette et donc toute l'information est conservée. A noter que si l'on ne se soucie que de la rotation, il est possible de passer en coordonnée polaire où la rotation n'est plus qu'une simple translation (Cecotti, et al., 2004). Une dernière technique pour gérer les rotations est d'effectuer la reconnaissance en utilisant la distance de Hausdorff (qui est invariante à la rotation) comme le fait (Wu, et al., 1999), cependant un Perceptron Multicouches – PMC (réseau de neurones) est quand même utilisé en aval pour les cas non reconnus, ce qui traduit une certaine faiblesse de cette technique.

Enfin, afin d'améliorer les performances de ces algorithmes, comme le stipule (Soetedjo, et al., 2005), il est possible de combiner différents algorithmes d'apprentissage / reconnaissance (technique du boosting), ce qui permet de mieux identifier et rejeter des exemples ambigus, et / ou aussi de générer un plus grand nombre d'exemples comme déjà stipulé dans le paragraphe précédent.

Un dernier problème qui survient lors de la reconnaissance est qu'il est presque impossible de différentier les panneaux lorsqu'ils sont loin, c'est-à-dire lorsqu'ils sont extrêmement petits dans l'image. Afin de surmonter ce problème, (Miura, et al., 2000) utilisent deux caméras, une conventionnelle ayant un grand angle avec laquelle ils effectuent la recherche des ROI (la première étape), puis ils utilisent une caméra téléobjectif afin d'obtenir une image en grande résolution de ces ROI lointaines. Ce qui amène un coût plus important pour équiper leur voiture, une plus grande fragilité de leur système (la caméra à téléobjectif étant motorisée afin de pouvoir observer n'importe quelle zone) et surtout la nécessité d'avoir un processus de suivi temporel robuste (la caméra motorisée se déplaçant lentement comparé à la vitesse de la voiture).

Tous les travaux actuels effectuent l'apprentissage sur l'ensemble du panneau et n'essayent pas de segmenter l'information pertinente (i.e. les chiffres pour un panneau de limitation de vitesse) à l'intérieur de ce panneau.

1.2.3. Segmentation de caractères

Bien qu'un seul travail (Torresen, et al., 2004) en détection et reconnaissance des panneaux de limitation de vitesse n'ait envisagé cette approche, segmenter et reconnaître chacun des caractères à l'intérieur des panneaux peut être une solution intéressante. C'est pourquoi sont présentées ici les techniques classiques de segmentation de caractère.

Comme stipulé ci-dessus et confirmé par (Casey, et al., 1996), la segmentation de caractères est la partie la plus importante dans un système d'OCR (optical character recognition), et est nécessaire afin d'améliorer le taux de reconnaissance des algorithmes. Cette partie est souvent cachée dans les tests par l'utilisation de base de données de caractères bien segmentés (comme dans (Liu, et al., 2008)).

Il existe trois grands types de méthode de segmentation de caractères (Casey, et al., 1996) (Casey, et al., 1995):

1. classique où la segmentation se fait par rapport à l'histogramme, ce qui peut poser de nombreux problèmes (par exemple quand les lettres s'entremêlent).
2. par reconnaissances successives de lettres via des fenêtres de taille variable que l'on fait glisser le long du texte, ce qui n'est pas envisageable pour un processus temps réel.
3. des méthodes holistiques, où la segmentation se fait mot par mot, ce qui implique, pour la reconnaissance, d'avoir un dictionnaire fini et assez restreint de mots.

Afin d'améliorer la méthode classique, il est possible d'effectuer une sur-segmentation et d'effectuer ensuite un processus de reconnaissance en regroupant les segments (ce qui revient à peu près à la seconde méthode). Il est aussi possible d'utiliser les méthodes de suivi de contour, par exemple celui de la goutte d'eau qui tombe (drop-fall) vers le haut ou vers le bas, voire combiner les deux en analysant la convexité du point de contact (Khan, 1998). Une autre idée intéressante de (Khan, 1998) pour segmenter un nombre est de compter le nombre de minima locaux afin d'en déduire le nombre de chiffres qui le composent. D'une manière générale, l'extraction de composantes connexes donne de meilleurs résultats que la segmentation sur l'histogramme (Casey, et al., 1996).

Enfin, si on connaît a priori la taille et/ou la position des caractères à segmenter, il est possible de forcer la segmentation (Casey, et al., 1996), comme c'est le cas par exemple pour la reconnaissance des plaques minéralogiques (Zhang, et al., 2003) et non celui des panneaux de limitations de vitesse (la taille variant selon les villes, les pays, selon qu'il s'agisse d'un panneau urbain ou autoroutier,…).

Cela dit, il apparaît qu'on ne puisse pas vraiment comparer les différentes méthodes de segmentation, chacune étant élaborée pour un système, une tâche précise (Casey, et al., 1996). En revanche, il faut toujours tenir compte du contexte.

C'est ainsi que (Torresen, et al., 2004) segmente les caractères en extrayant, après avoir converti les ROI sur 3 couleurs (rouge, blanc et noir), les composantes connexes de couleurs noires avant d'en effectuer la reconnaissance via un PMC. L'inconvénient de cette approche, comme déjà stipulé, est que lorsque les chiffres s'entremêlent, il n'est alors plus possible de les reconnaître.

1.2.4. Intégration temporelle

Comme le font si bien remarquer (Piccioli, et al., 1996) et (Fang, et al., 2003) la détection des panneaux signalétiques en ne considérant qu'une seule image à la fois (donc sans intégration temporelle) soulève trois problèmes :

1. Cela ne permet de prédire la position et la taille des panneaux pour réduire l'espace de recherche et le temps d'exécution.
2. Il est difficile de correctement détecter un panneau quand il y a une occlusion temporaire.
3. La véracité de la détection est dure à vérifier sur une image seulement.

C'est ainsi que quelques approches emploient un processus de fusion temporelle basée sur la détection image par image afin d'obtenir une meilleure détection globale. Ces approches peuvent être divisées en deux catégories.

1. Il est tout d'abord possible de ne se soucier que de la succession des détections à chaque image. Ainsi plus un panneau sera détecté de façon successive, meilleur sera son indice de

confiance dans sa reconnaissance globale. C'est ce que fait (Bahlmann, et al., 2005) par exemple.

2. La deuxième approche consiste, à partir des détections antérieures, à prédire la position des panneaux dans les futures images (tracking temporel). Ceci peut avoir lieu en supposant que le mouvement du véhicule est localement droit et ainsi prédire la position dans une image à partir de 2 prédictions précédentes (l'idée est de tracer la droite entre les 2 anciennes prédictions, la suivante sera sur cette droite à une distance régie par la vitesse du véhicule) comme le fait (Miura, et al., 2000). Il est possible d'améliorer ce processus en y incorporant un filtre de Kalman (Piccioli, et al., 1996) (Zadeh, et al., 1998) (Garcia, et al., 2006) (Fang, et al., 2003).

   Bien évidemment, ceux qui emploient cette seconde technique utilisent aussi tous la première.

Une autre technique pour gérer les cas d'occlusion partielle consiste à rajouter dans la base d'apprentissage des nouveaux exemples soumis à diverses translations et tronqués (Yang, et al., 2003). Il est aussi possible de repérer les cas d'occlusions partielles en trouvant les zones non rouges, blanches et noires dans l'image du panneau et d'utiliser cette information pour modifier dynamiquement les règles de l'algorithme de reconnaissance (Soetedjo, et al., 2005).

Enfin, certains élaborent des règles de plus haut niveau afin d'améliorer la robustesse de leur système de reconnaissance de panneaux de limitations de vitesse. Ainsi (Torresen, et al., 2004) établit que par exemple, après avoir reconnu une limitation à 80, qu'il n'est pas possible d'avoir directement une limitation à 10, mais il est obligatoire de passer par la vitesse limite de 60 (en Norvège tout au moins).

Pour résumer, un système embarqué de détection et de reconnaissance des panneaux de limitation de vitesse est basé sur 3 grands modules comme le rappelle la Figure 3.

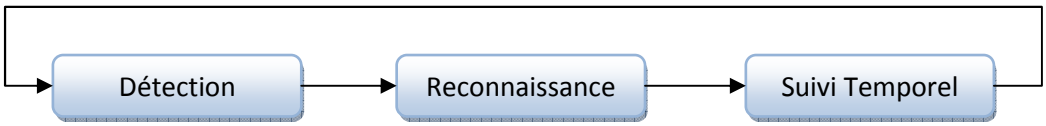


Figure 3 - Les 3 étapes de la détection des panneaux de limitation de vitesse

1.3. Détection et reconnaissance des panneaux

1.3.1. Conception générale

Le système développé doit pouvoir fonctionner sous n'importe quelle condition climatique (ensoleillé, par brouillard,…), lumineuse (de jour, de nuit, …) et quel que soit l'état du panneau (neuf avec des couleurs criantes et ancien avec des couleurs estompées). Il est donc de ce fait exclu

d'envisager d'utiliser un processus de segmentation par couleur en première étape (détection des zones d'intérêt), car, bien que rapide, cette méthode ne pourrait pas fonctionner dans tous les cas. Mais le système doit aussi pouvoir être exécuté en temps réel car embarqué au sein de véhicule pour aider les conducteurs (et donc avoir une réactivité accrue). L'enjeu principal de cette première étape de segmentation réside donc dans le développement d'un module basé sur la détection de contours qui soit robuste et rapide.

Pour l'étape de reconnaissance des zones d'intérêt, la question réside dans le fait de savoir sur quelle partie de l'image la reconnaissance se fera. Soit sur toute la zone d'intérêt, représentant donc tout le panneau, ou, pour le cas spécifique des panneaux de limitation de vitesse, essayer de n'effectuer la reconnaissance que sur chacun des chiffres à l'intérieur du panneau. Les deux solutions ont chacune des avantages et des inconvénients :

- La reconnaissance de toute la zone d'intérêt peut être sujette à une détection peu précise (c.à.d. une zone d'intérêt mal calée sur le panneau) qui pourrait fausser les résultats de l'algorithme de reconnaissance, mais elle n'entraine pas de traitement particulier en plus qui pourrait alourdir l'exécution du système (et donc son temps de traitement).
- La reconnaissance des chiffres implique de pouvoir extraire chaque caractère à l'intérieur du panneau détecté ce qui peut induire plus d'erreur (de segmentation) et un traitement plus lourd pour une exécution embarquée devant fonctionner en temps réel. En revanche, l'algorithme de classification n'aurait plus qu'à reconnaître qu'une dizaine de classes (chiffres de 0 à 9) au lieu d'un plus grand nombre de panneau de limitation de vitesse (de 10km/h à 130 km/h en France). De plus, la segmentation de caractère peut donner des zones plus précises à reconnaître. Il en résulte que le résultat de la classification basée sur cette méthode doit être meilleur.

De plus, en pratique, il n'existe pas de norme stricte nationale et encore moins européenne sur l'aspect physique des panneaux de limitation de vitesse. Ils sont certes tous cerclés de rouge avec le nombre en noir au milieu, mais ni la taille du panneau, ni la typographie des chiffres ne sont imposées. Il en résulte une grande variabilité des panneaux au sein d'un même pays (voire d'une même ville) et entre pays différents (Figure 4).

Figure 4 - Illustration de la variabilité de l'aspect des panneaux de limitation de vitesse : En gris des panneaux allemands (à gauche de chaque paire) comparés à leur équivalent français - En couleur à droite 2 variantes d'un même panneau français.

Le système développé, qui vise à être le plus robuste possible, est donc naturellement basé sur la segmentation de caractères afin de palier aux problèmes de variabilité des panneaux et de réduire le nombre de classes à reconnaitre. Curieusement, dans la littérature, seul les travaux de (Torresen, et

al., 2004) utilisent une technique similaire, tous les autres travaux effectuent une reconnaissance globale. L'utilisation de cette technique pourrait nous aider à adapter facilement nos travaux pour la reconnaissance des panneaux de limitation de vitesse américains qui « souffrent » eux d'une grande variabilité de leur contenu et de leur taille (Figure 5).

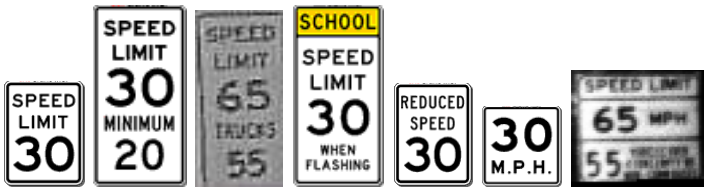


Figure 5 - Illustration de quelques uns des pannaux de limitation de vitesse américains qui prouvent la complexité d'utiliser un processus de reconnaissance globale des panneaux dans ce pays.

Le processus de segmentation qui fonctionne le mieux, est d'après la littérature l'extraction des composantes connexes, qui sera donc utilisée dans le système. Le choix de l'algorithme de reconnaissance d'image / de classification (SVM, RN,…) est beaucoup moins critique, ces algorithmes donnant de nos jours des résultats plus que satisfaisants. De plus, par le choix de cette méthode, il n'est pas exclu de pouvoir utiliser des algorithmes d'OCR (reconnaissance de caractère) issus par exemple de la reconnaissance des documents scannés, ayant prouvé leur bon fonctionnement depuis des années.

Afin de se rendre compte de la complexité de la tâche, une première version du système est développée en utilisant les algorithmes bien connus issus du traitement d'image. Son fonctionnement, utilisant une caméra monochromatique embarquée dans le véhicule, est schématisé sur la Figure 6.

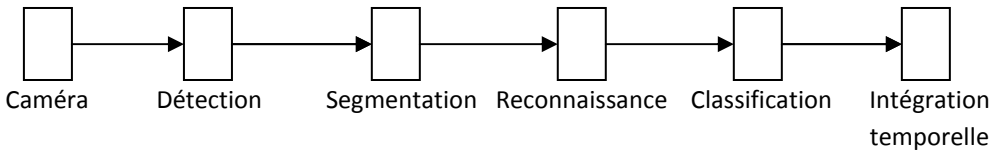


Figure 6 - Fonctionnement général du système

1. Un flux d'image en niveau de gris est acquis via une caméra embarquée (*Figure 7*).

2. De ce flux d'images sont extraits les panneaux potentiels via l'algorithme de Hough (i.e. sont détectés tous les cercles dans chacune des images) pour les panneaux circulaires de type européen (*Figure 8*).

3. Chaque imagette issue d'un cercle subit une normalisation d'histogramme suivie d'un seuillage binaire local adaptatif (*Figure 9*).

4. Une segmentation des caractères, via une extraction de composantes connexes (blobs) est effectuée sur chacune de ces imagettes en noir et blanc (*Figure 10*).

5. Les composantes respectant certains critères géométriques (ex : ratio hauteur/largeur) sont analysées via un réseau de neurones (de type perceptron multicouche) afin d'en reconnaître le chiffre (*Figure 11*).

   La topologie du réseau de neurone a été trouvée de façon empirique (Tableau 1) : il apparaît qu'avec 20 neurones sur la couche cachée, l'apprentissage est le meilleur (i.e. aussi bon qu'avec plus de neurones, et meilleur qu'avec moins). Pour des raisons historiques, la taille de l'entrée est de 64 neurones pour reconnaître des images de 8x8 pixels (une première version de la base de chiffre existait déjà). Le réseau de neurones a été entrainé sur cette base de données agrémentée de chiffres extraits de nos enregistrements (qui constituent donc de vrais exemples et non des exemples artificiels) dont 2772 images de chiffres (provenant de panneaux français, allemands et italiens) et 2790 exemples négatifs (des non-chiffres). L'outil utilisé (*Levis*) pour l'entrainement et le choix de la topologie du réseau de neurone a été développé spécifiquement et est décrit au chapitre 5.

6. Le flux de chiffres reconnus est classifié (i.e. assemblé et validé 1 + 1 + 0 → 110 ou invalidé 1 + 9 + 0 → 190, limitation qui n'existe pas) puis intégré temporellement afin d'extraire un niveau de confiance géométrique et temporel de la détection pour valider ou invalider la détection d'un panneau (*Figure 12*).

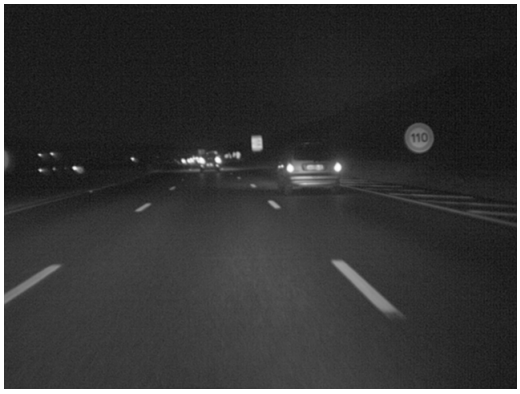


Figure 7 - Image originale

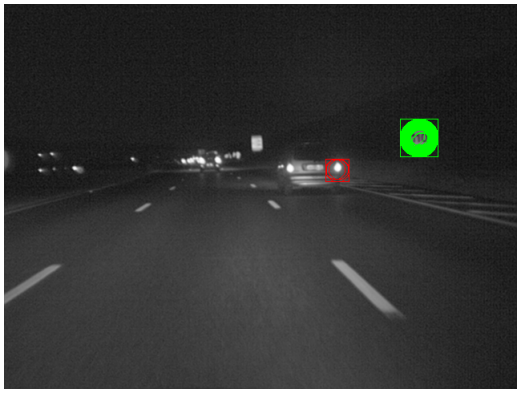


Figure 8 - Détection de cercles

Figure 9 - Image binaire

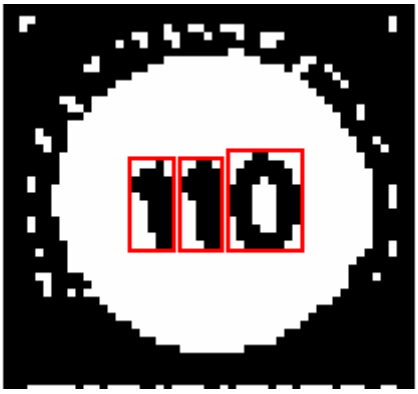
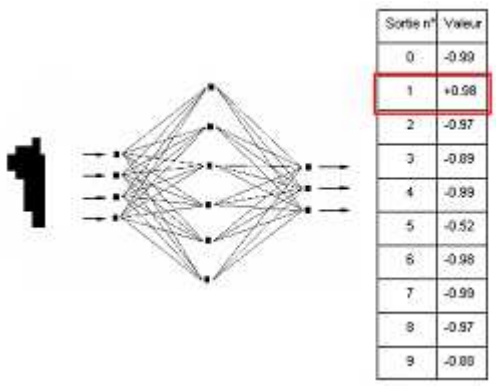


Figure 10 - Extraction des blobs



Sortie n°	Valeur
0	-0.99
1	+0.98
2	-0.97
3	-0.89
4	-0.89
5	-0.52
6	-0.98
7	-0.99
8	-0.97
9	-0.88

Figure 11 - Reconnaissance des blobs

Figure 12 - Résultat

Taille de la couche cachée	Taux de bonne classification de la base d'apprentissage	Taux de bonne classification de la base de test
5	74%	72%
10	95%	93%
15	96%	95%
20	97%	96%
25	97%	96%
30	97%	96%
35	97%	96%

Tableau 1 - Taux de bonne classification du réseau de neurone sur la base de chiffre

Quelques commentaires sur cette première version s'imposent. Le système fonctionne globalement bien et est très prometteur, mais il n'est, pour l'instant, pas assez efficace pour être utilisé tel quel dans un système d'aide à la conduite automobile. En effet il souffre de certains problèmes. Ainsi, comme listé dans l'état de l'art, la segmentation de caractère par extraction de composante connexe est robuste, même quand les panneaux sont inclinés et non droits (Figure 13), mais inefficace quand les chiffres s'entremêlent (ce qui arrive quand le panneau à reconnaître est loin du véhicule). La détection des cercles est, elle aussi, efficace mais l'algorithme de Hough est inexploitable en temps réel et un compromis a dû être trouvé pour le faire fonctionner à une fréquence raisonnable. C'est ainsi que cet algorithme n'est appliqué que sur des images en demi-résolution, ce qui réduit forcément la distance à partir de laquelle un panneau peut commencer à être détecté. Par conséquence, dans les cas où le panneau est toujours trop petit dans l'image, ce qui arrive quand il est physiquement trop petit et / ou quand il est toujours loin du véhicule (par exemple sur une route à 3 voies, quand le panneau est à l'opposé du véhicule), le système ne peut le détecter.

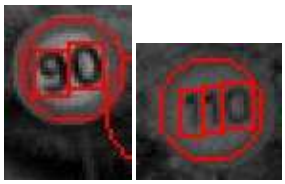


Figure 13 - Illustration de la robustesse de la segmentation par extraction de composantes connexe

Il apparaît donc que les choix initiaux soient en corrélation avec l'état de l'art et que le travail mis en place ne soit pas tombé dans les pièges mis en exergue par la facilité, cependant le système souffre de problèmes inhérents aux choix qui ont été faits. Les améliorations apportées afin de rendre le système efficace et exploitable en temps réel sont décrites dans les sections suivantes.

1.3.2. Segmentation globale des chiffres

Dans la majorité des cas, la segmentation des caractères via extraction des composantes connexes fonctionne correctement. Cependant, pour plusieurs raisons, il arrive que les chiffres composant la limitation de vitesse sur l'image vue par la caméra s'entremêlent et ne forme donc plus qu'une seule grosse et même composante connexe non reconnaissable par le système.

Ces raisons sont :

1. Quand les panneaux sont détectés d'assez loin, la discrétisation effectuée par le processus d'acquisition numérique fait que la séparation des chiffres n'est plus visible.

2. Quand les panneaux sont naturellement petits, pour les mêmes raisons les chiffres s'entremêlent.

3. De nuit, la caméra est éblouie par la forte réflexion des phares sur les panneaux signalétiques ce qui a pour conséquence de rendre les images plus ou moins floues.

Une solution triviale serait d'utiliser un autre réseau de neurones pour apprendre à classifier ce genre de formes. Cependant, afin d'avoir un algorithme de reconnaissance robuste, il faut disposer de nombreux exemples. Or les formes des composantes connexes trouvées sont trop variées pour pouvoir créer une base d'exemples importante. Par exemple, *Figure 14*, à gauche ne sera extrait que le '100', tandis qu'à droite, les chiffres touchent aussi le bord du panneau, ainsi la composante connexe est formée du '100' et du bord du panneau. Parfois aussi seuls deux parmi trois chiffres sont « collés », ce qui augmente le nombre de cas de figure et rendrait très complexe une approche de ce type (Figure 15).

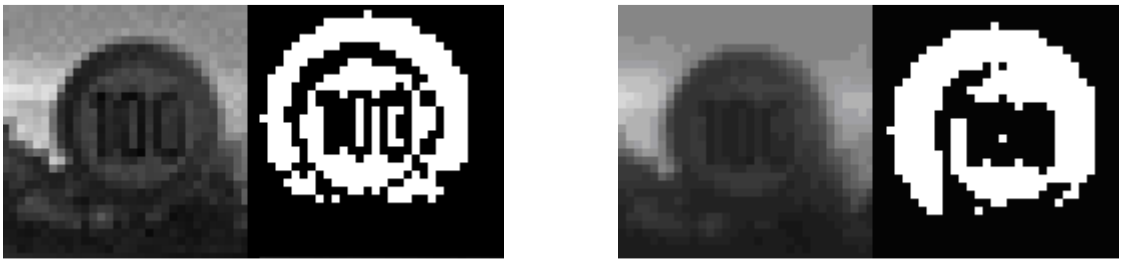


Figure 14 - Exemple d'images binaire où la segmentation ne fonctionne pas

Figure 15 - Exemple de cas où la segmentation n'est pas correcte

Il est donc nécessaire de mettre en place un processus de segmentation spécifique à notre problème.

Typiquement, dans le domaine du traitement d'image, la première idée serait d'effectuer une segmentation selon l'histogramme de nos images. Chaque chiffre, représentant un pic dans l'histogramme, il paraîtrait théoriquement facile de segmenter les chiffres qui s'entremêleraient. Cependant, avant la reconnaissance, il nous est impossible de savoir si notre image représente un panneau à trois chiffres (trois pics dans l'histogramme ?), ou à deux chiffres (donc deux pics dans l'histogramme ?). De plus, l'image binaire, dépend d'un nombre important de facteurs, comme par exemple la position du cercle détecté, l'environnement sur lequel le panneau se sur-imprime dans l'image… qui modifient quels pixels seront noirs ou blancs dans l'image binaire et donc qui modifient l'histogramme (et le nombre de pics présents dans celui-ci) comme le montrent les Figure 16, Figure 17 et Figure 18.

Il existe, comme nous l'avons vu plus haut, de nombreuses méthodes dans le domaine de la segmentation de caractères pour nous aider. Cependant, il faut proscrire les méthodes de type sur-segmentation, qui pourraient faire apparaître de fausses détections (par exemple segmenter les barres des '0' des panneaux allemands pourrait faire apparaître des '1') et de type segmentation par reconnaissance, nous avons déjà vu que cela revenait à peu près à la méthode de sur segmentation (via les fenêtres glissantes de taille variable). La dernière technique est d'effectuer une segmentation globale (i.e. extraire tout le nombre du panneau sans sa bordure), ce qui répondrait au cas de la *Figure 14*.

Le problème revient donc, pour une segmentation de type global, à trouver les limites horizontales (haute et basse) et verticales (gauche et droite) du nombre dans l'image représentant la limitation de vitesse dans le panneau.

Nous avons déjà vu qu'il est impossible de se baser sur l'histogramme pour trouver les limites verticales. La même question se pose pour les limites horizontales. Il apparaît que là aussi, comme le montre la Figure 18, il est impossible de se baser sur la technique du comptage de pics dans l'histogramme.

Figure 16 - Impossibilité de compter le nombre de pics verticaux dans l'histogramme pour segmenter les chiffres (panneau allemand)

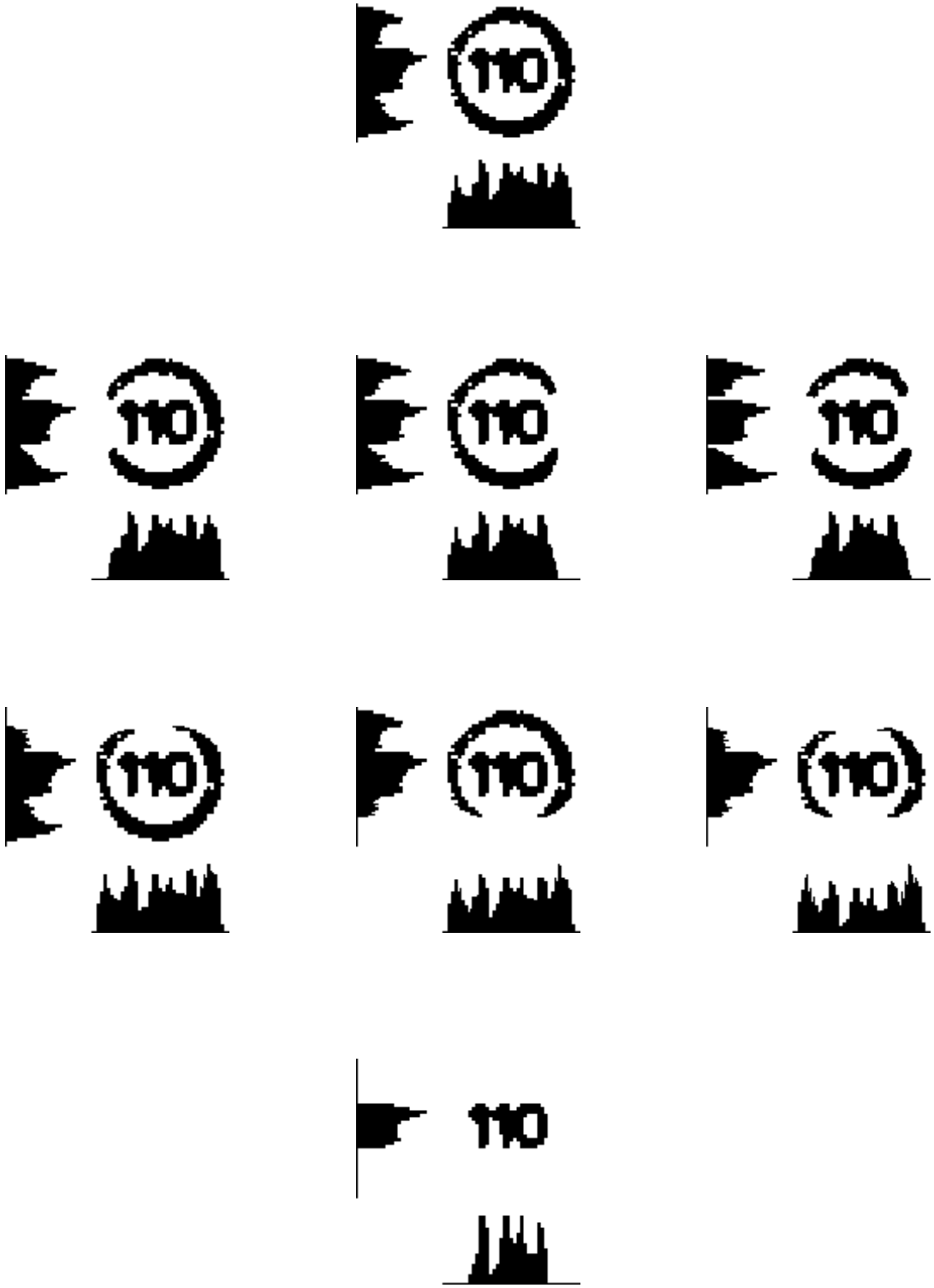


Figure 17 - Impossibilité de compter le nombre de pics verticaux dans l'histogramme pour segmenter les chiffres (panneau français)

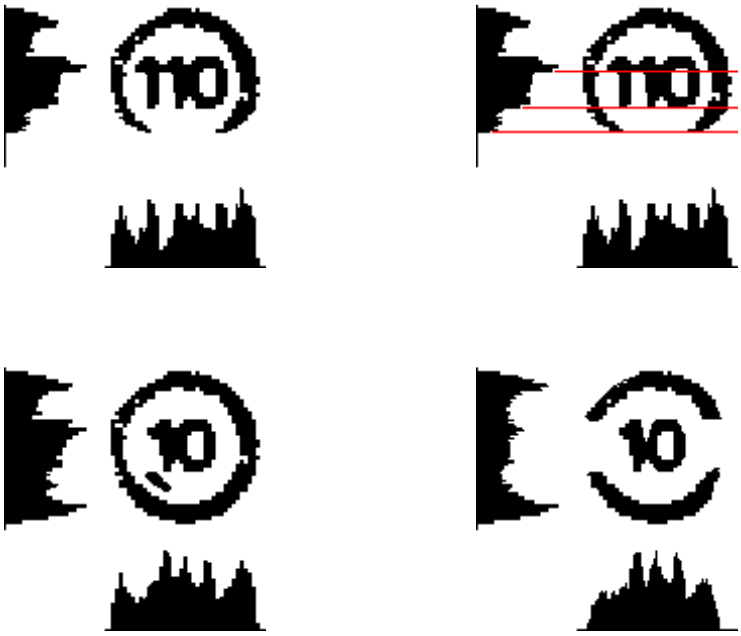


Figure 18 - Impossibilité de compter le nombre de pics horizontaux dans l'histogramme pour segmenter le nombre

La solution proposée pour trouver ces limites est l'algorithme *Global Segmentation for Speed-Limit Sign – GS-SLS* exposé ci-après. A partir de l'image binaire (Figure 19), et à partir d'une petite portion horizontale au milieu de cette image (Figure 20), l'algorithme consiste à parcourir l'image de telle sorte que l'on privilégie d'aller d'abord vers le centre de l'image (donc à droite quand on est à gauche et vice versa) puis vers le bas si on cherche la limite inférieure (ou vers le haut pour la limite supérieure) avec l'impossibilité de revenir sur ses pas et d'aller vers le haut quand on cherche la limite basse et vice versa (Figure 21). L'image binaire est donc divisée en quatre zones sur chacune desquelles on effectue le parcours décrit précédemment (Figure 22). Les limites horizontales sont les maxima trouvés par ces différents parcours (Figure 23).

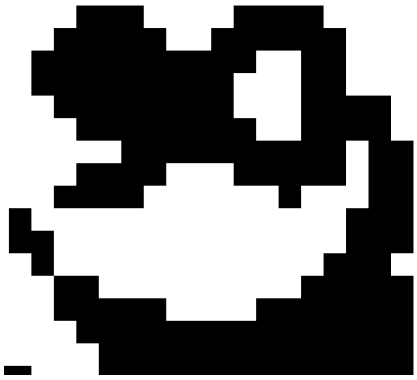


Figure 19 - Image binaire

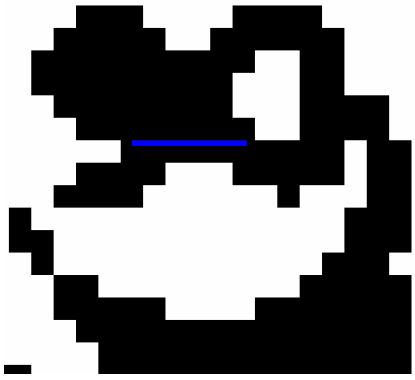


Figure 20 - Ligne de départ

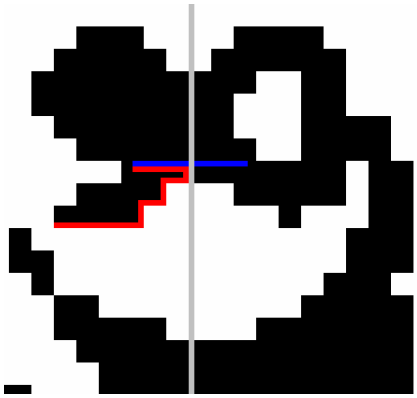


Figure 21 - Parcours gauche inférieur

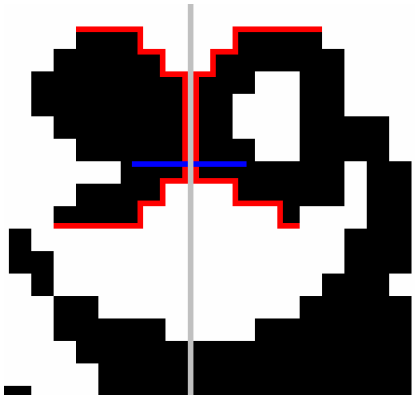


Figure 22 - Les 4 parcours

Ce parcours, pixels par pixels via un voisinage 3-connexes (4 voisins moins 1) est robuste au bruit. En effet, si l'un des points de départ est en fait un pixel de bruit (un pixel ou petit groupe de pixels noirs au milieu du fond blanc du panneau), la propagation à partir de cet endroit n'irait pas bien loin. De plus, le fait d'utiliser un voisinage 3-connexe, empêche le parcours de s'étendre dans le bruit vers le haut (ou vers le bas) à l'extérieur du nombre. En revanche, pour éviter au parcours de s'étendre le long de la bordure du panneau, une règle mise en place en place consiste à empêcher la propagation vers l'extérieur si on a dépassé une certaine limite (gauche ou droite), typiquement un pourcentage de la taille du panneau.

Il ne reste plus qu'à trouver les limites verticales du nombre dans le panneau. Pour cela, à partir du milieu de l'image, l'algorithme consiste à parcourir l'image colonne par colonne vers la droite ou vers la gauche (Figure 24, Figure 25, Figure 26 et Figure 27). Ce parcours s'arrête lorsque que la colonne de pixels noirs dépasse l'une des limites horizontales (Figure 28 et Figure 29).

Cet algorithme a été testé sur de nombreux exemples (vidéos) sous diverses conditions (de jour, de nuit, …) et semble fonctionner correctement (Figure 30).

Figure 23 - Limites horizontales

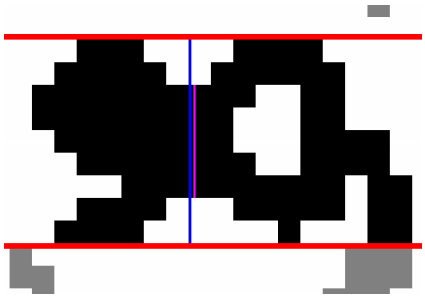


Figure 24 - Parcours par colonne

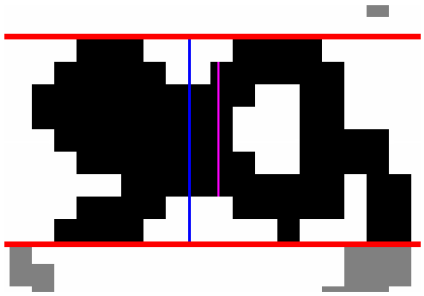


Figure 25 - Parcours par colonne

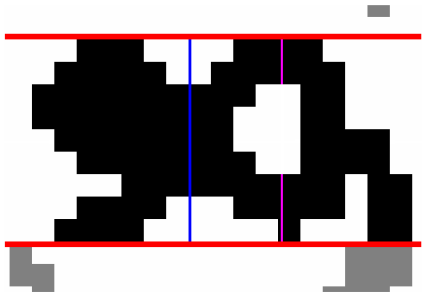


Figure 26 - Parcours par colonne

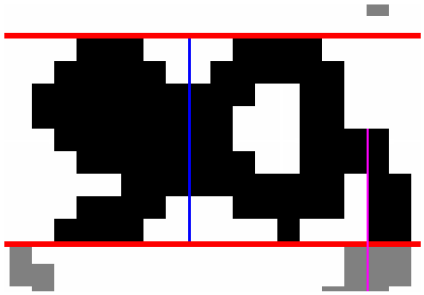


Figure 27 - Parcours par colonne

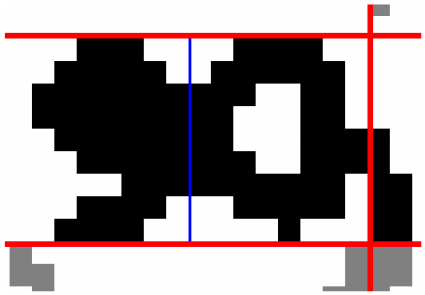


Figure 28 - Limite droite trouvée

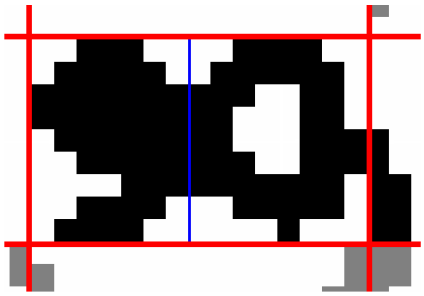


Figure 29 – Résultat

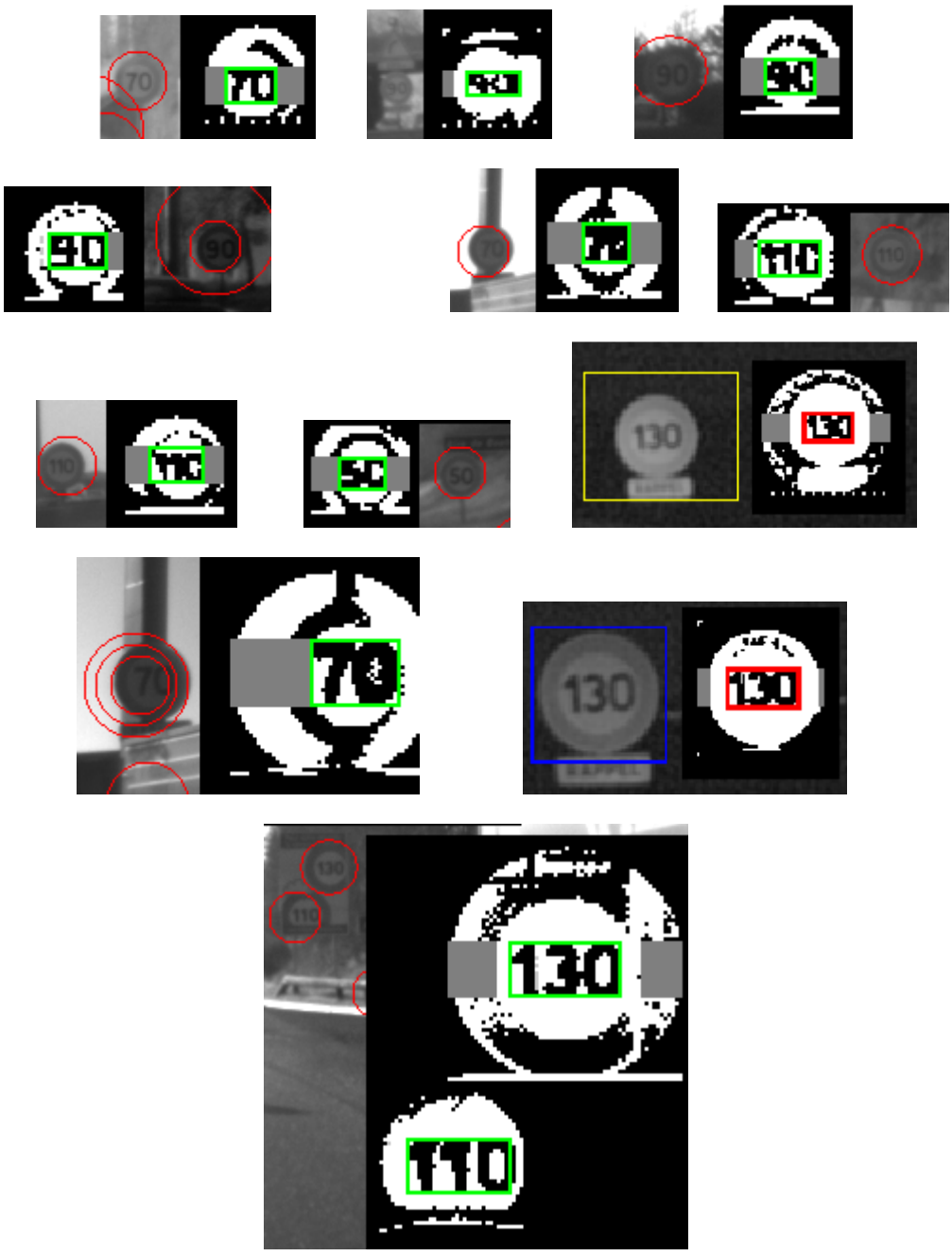


Figure 30 - Résultats de l'algorithme GS-SLS de segmentation sur divers cas

La segmentation des caractères du panneau par l'extraction des composantes connexes échouait car plusieurs chiffres se retrouvaient dans la même composante connexe. Cet effet indésirable est dans la majorité des cas dû au fait que le seuil calculé dans l'algorithme de seuillage adaptatif (pour rendre l'image binaire) est faussé car trop d'information est présente quand on considère le panneau en entier (i.e. la détection de cercle effectuée) : il y a dans cette zone des pixels représentant le cercle rouge, des pixels représentant l'extérieur du panneau (surtout si le cercle détecté n'est pas bien centré sur le panneau),… Il est maintenant possible d'effectuer le seuillage uniquement sur la zone trouvée par le nouvel algorithme de segmentation globale du nombre. Il en résulte que chaque chiffre est correctement représenté par une composante connexe (Figure 31). Néanmoins, un algorithme de segmentation basé sur l'histogramme a quand même été mis en place sur ces nouveaux résultats. En effet, il arrive que, sur cette nouvelle image binaire, les chiffres se retrouvent quand même liés par un ou deux pixels entre eux, ce qui est facilement segmentable.

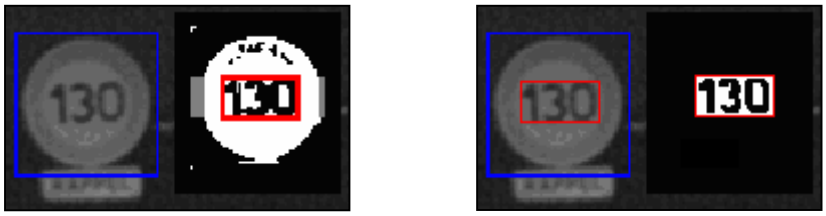


Figure 31 - Exemple d'amélioration apporté par la segmentation globale : Ancien résultat (à gauche) – Nouveau résultat (à droite)

Ce nouvel algorithme de segmentation, spécifique à l'extraction des nombres dans les panneaux de limitation de vitesse, impose quelques restrictions pour son utilisation. Dû au fait que la zone de départ soit une ligne horizontale centrée sur le panneau, il est primordial que cette zone contienne des portions des chiffres et aussi et surtout qu'elle ne chevauche pas la bordure du panneau (sinon la propagation se fera sur cette bordure et à l'extérieur des chiffres) comme le montre la Figure 32. Il faut donc que la pré-détection des cercles soit assez centrée sur le panneau et pas trop grande. Ce qui peut sembler assez contraignant face aux contraintes, plus laxistes de l'extraction des blobs.

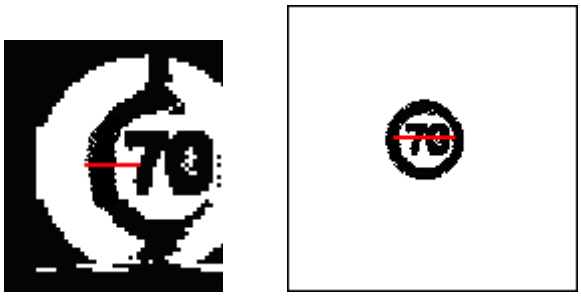


Figure 32 - Deux cas où la segmentation holistique ne fonctionne pas

1.3.3. Détection de cercles

Dans la première version de l'algorithme, afin de rendre l'algorithme de Hough pour la détection des cercles plus rapide et utilisable en temps réel, il n'est pas appliqué sur l'image dans sa taille originale, mais sur une version réduite (en demi-résolution) du flux vidéo. De plus, seuls les cercles dont les rayons qui sont compris dans un certain intervalle (évalué empiriquement de façon à ce que les cercles détectés restent assez grands / reconnaissables) sont recherchés.

Cependant, réduire la taille de l'image engendre évidemment une perte d'information rendant l'algorithme peut exploitable dans certains cas et notamment dans les cas à faible contraste et quand les panneaux à détecter sont loin / petits) comme on peut le voir Figure 33.

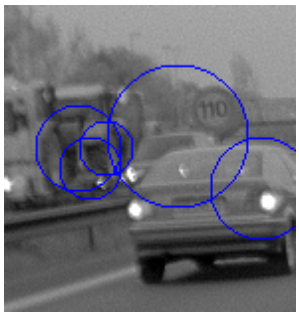


Figure 33 - Détection de cercle sur une image réduite

Même si cela ne semble pas poser beaucoup de problème dû au fait de l'extraction des composantes connexes à l'intérieur de chaque détection, il est nécessaire d'avoir une détection plus robuste (i.e. mieux cadré sur le panneau), afin d'améliorer la reconnaissance des panneaux, par exemple par l'utilisation de l'algorithme de segmentation décrit précédemment ou pour effectuer une reconnaissance globale du panneau (cf. les fins de limitation de vitesse comme on le verra plus tard).

SUITE DE LA PAGE TEMPORAIREMENT MASQUEE

PAGE TEMPORAIREMENT MASQUEE

PAGE TEMPORAIREMENT MASQUEE

PAGE TEMPORAIREMENT MASQUEE

PAGE TEMPORAIREMENT MASQUEE

PAGE TEMPORAIREMENT MASQUEE

PAGE TEMPORAIREMENT MASQUEE

PAGE TEMPORAIREMENT MASQUEE

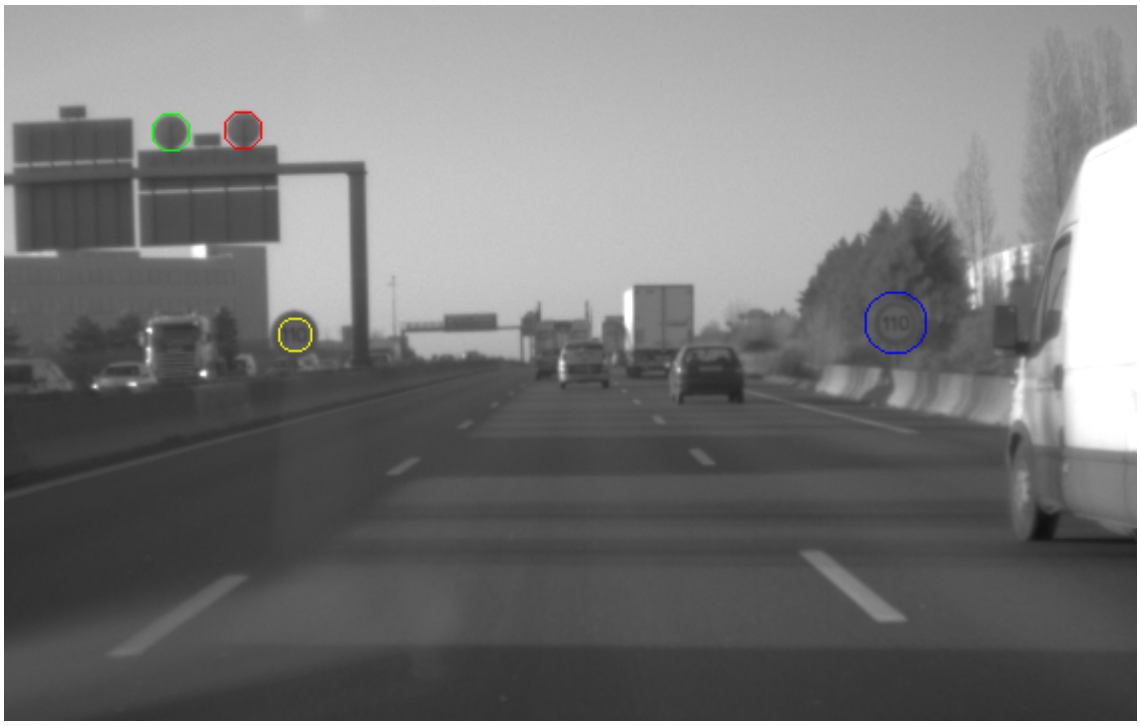


Figure 48 - Exemple de résultat de la nouvelle approche de l'algorithme de Hough

1.3.3. Intégration temporelle

La phase d'intégration temporelle initialement développée dans le système ne sert qu'à modifier l'indice de confiance globale que l'on a dans la détection et dans la reconnaissance d'un panneau. Un panneau réel sera en effet détecté dans plusieurs images consécutives, il suffit donc d'augmenter l'indice de confiance du panneau à chaque détection (et à l'inverse l'abaisser à chaque non détection), le panneau est validé quand son indice de confiance dépasse un seuil prédéfini. L'appariement d'un même panneau sur 2 images consécutives, dans le système initial, se fait en associant les panneaux les plus proches et de même limitation de vitesse.

Cependant, afin de pouvoir apparier correctement deux détections du même panneau mais à deux instants différents, donc à deux places différentes dans l'image (dû au mouvement du véhicule

embarquant le système) et afin de changer correctement l'indice de confiance et aussi de ne pas mélanger les détections de plusieurs panneaux différents, il est nécessaire de prédire la position future des panneaux.

Une première méthode mise en place consiste, en supposant que la trajectoire des panneaux est linéaire, à utiliser la méthode des moindres carrés, appliquée aux centres des panneaux sur plusieurs image, pour réaliser l'interpolation, i.e. trouver les coefficients a et b de la droite y = ax + b estimant la trajectoire d'un panneau (Équation 1).

$$y = ax + b \text{ avec } \begin{cases} a = \dfrac{n\sum_i x_i y_i - \sum_i x_i \sum_i y_i}{n\sum_i x_i^2 - \left(\sum_i x_i\right)^2} \\ b = \dfrac{\sum_i x_i \sum_i y_i - \sum_i x_i \sum_i x_i y_i}{n\sum_i x_i^2 - \left(\sum_i x_i\right)^2} \end{cases}$$

Équation 1 - Interpolation par moindres carrés

Cette méthode, comme le montre la Figure 49 fonctionne correctement dans le processus d'appariement des panneaux.

Figure 49 – Résultat du suivi temporel par moindres carrés d'un panneau

Il est possible ici d'améliorer le système en gardant la même logique. En effet, l'intégration temporelle n'est réalisée que pour modifier l'indice de confiance que l'on a dans la détection d'un panneau.

Cependant il arrive qu'un panneau ne soit plus détecté pendant une ou plusieurs images (ce qui ne pose pas de problème au processus actuel qui met à jour les coordonnées des panneaux en mémoire

via la méthode des moindres carrés (Équation 1). Il apparaît qu'un panneau peut ne plus être détecté pour seulement deux raisons :

1. soit la détection du contour (cercle) du panneau dans l'image a échoué, et donc notre système, à l'état actuel, n'a pas pu essayer de segmenter les chiffres puis reconnaître la limitation de vitesse.

2. soit la segmentation ou la reconnaissance a échoué

Après étude, beaucoup de cas sont dus à la première proposition. L'idée est donc ici de forcer le processus de recherche / segmentation de caractères puis de leur reconnaissance à des endroits prédits selon la position des anciens panneaux détectés (Figure 50).

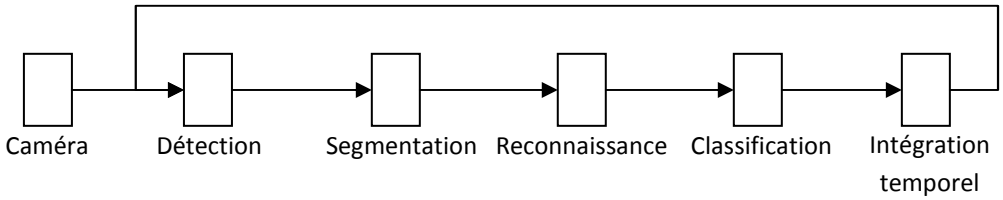


Figure 50 - Fonctionnement général du système avec suivi spatio-temporel

L'interpolation linéaire par la méthode des moindres carrés est certes très efficace, mais elle ne prend pas en compte le modèle de projection de l'image du monde réel dans la caméra.

En simplifiant le modèle physique de la caméra par le modèle sténopé (appelé aussi modèle *pin-hole* dans la littérature anglo-saxonne) il est possible de prédire la position et la taille d'un panneau dans l'image (Équations E7 et E8).

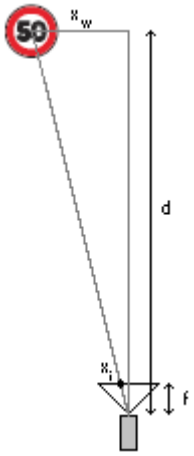


Figure 51 - Modèle sténopé

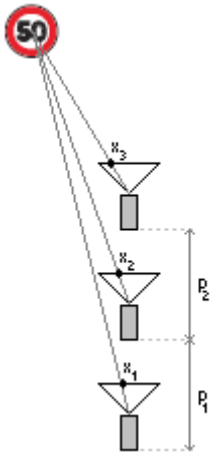


Figure 52 - Suivi spatio-temporel

Le modèle sténopé est régi par les équations de Thales, ainsi, Figure 51 $\frac{f}{x_i} = \frac{d}{x_w}$ où f est la focale de la caméra, d la distance du panneau à la caméra, x_i la position du centre du panneau dans l'image et x_w la position du panneau par rapport à la caméra.

Il est alors possible, d'après la Figure 52, d'écrire les équations suivantes, en supposant une trajectoire rectiligne de la voiture (i.e. qu'elle ne tourne pas) :

$\frac{f}{x_1} = \frac{d}{x_w}$	$\frac{f}{x_2} = \frac{d - p_1}{x_w}$	$\frac{f}{x_3} = \frac{d - (p_1 + p_2)}{x_w}$	E1
$fx_w = dx_1$	$fx_w = (d - p_1)x_2$	$x_3 = \frac{fx_w}{d - (p_1 + p_2)}$	E2
$x_3 = \frac{dx_1}{d - (p_1 + p_2)}$	$x_3 = \frac{(d - p_1)x_2}{d - (p_1 + p_2)}$		E3
$dx_1 = (d - p_1)x_2$ $dx_1 = dx_2 - p_1x_2$ $dx_2 - dx_1 = p_1x_2$ $d(x_2 - x_1) = p_1x_2$			E4
$d = \frac{p_1x_2}{(x_2 - x_1)}$		$x_3 = \frac{dx_1}{d - (p_1 + p_2)}$	E5
$x_3 = \frac{\frac{p_1x_1x_2}{(x_2 - x_1)}}{\frac{p_1x_2}{(x_2 - x_1)} - (p_1 + p_2)}$ $x_3 = \frac{p_1x_1x_2}{\left[\frac{p_1x_2}{(x_2 - x_1)} - (p_1 + p_2)\right][x_2 - x_1]}$ $x_3 = \frac{p_1x_1x_2}{p_1x_2 - (p_1 + p_2)(x_2 - x_1)}$ $x_3 = \frac{p_1x_1x_2}{p_1x_1 - p_2x_2 + p_2x_1}$ $x_3 = \frac{p_1x_1x_2}{(p_1 + p_2)x_1 - p_2x_2}$ $x_3 = \frac{(p_1x_1x_2)\frac{1}{p_1}}{[(p_1 + p_2)x_1 - p_2x_2]\frac{1}{p_1}}$			E6

$$x_3 = \frac{x_1 x_2}{(1+\frac{p_2}{p_1})x_1 - \frac{p_2}{p_1}x_2}$$

$$x_3 = \frac{x_1 x_2}{(1+\alpha)x_1 - \alpha x_2} \text{ avec } \alpha = \frac{p_2}{p_1} \quad \text{E7}$$

En supposant que la trajectoire du véhicule est localement droite entre 3 images successives (le processus d'acquisition étant à 10 images secondes, on suppose que la voiture sur un laps de temps de 30ms a suivi une trajectoire rectiligne, ce qui reste raisonnable), on peut écrire :

$$\alpha = \frac{p_2}{p_1} = \frac{v\delta t_2}{v\delta t_1} = \frac{\delta t_2}{\delta t_1}$$

Il est donc possible de prédire la position du panneau dans l'image, connaissant deux positions antérieures, sans connaître la focale f de la caméra ni même la vitesse du véhicule, ce qui corrobore les résultats de (Miura, et al., 2000).

De la même façon qu'ont été établies les équations pour suivre spatio-temporellement le centre du panneau, il est possible d'établir celles pour le rayon du cercle. Ainsi d'après la Figure 53:

$$\left.\begin{array}{l} \frac{x_1}{X_1} = \frac{f}{d} \rightarrow x_1 d = fX_1 \\ \frac{x_2}{X_2} = \frac{f}{d} \rightarrow x_2 d = fX_2 \end{array}\right\} \begin{array}{c} x_2 d - x_1 d = fX_2 - fX_1 \\ \rightarrow (x_2 - x_1)d = (X_2 - X_1)f \\ \frac{x_2 - x_1}{X_2 - X_1} = \frac{f}{d} \end{array}$$

Par suite, comme $x_2 - x_1 = 2r$ (le diamètre du cercle dans l'image) et $X_2 - X_1 = 2R$ (le diamètre réel du panneau), alors $\frac{r}{R} = \frac{f}{d} \leftrightarrow \frac{f}{r} = \frac{d}{R}$. On retrouve donc les équations de (E1). Ce qui nous permet de déduire que :

$$r_3 = \frac{r_1 r_2}{(1+\alpha)r_1 - \alpha r_2} \quad \text{E8}$$

Il est donc possible de prédire la taille du panneau dans l'image sans connaître sa taille réelle, ni, comme précédemment, la focale de la caméra ni la vitesse du véhicule.

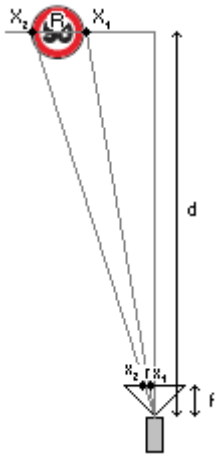


Figure 53 - Modèle sténopé

Les résultats de cette nouvelle approche (Figure 54 et Figure 55) montrent que la prédiction spatio-temporelle (à partir des deux précédentes détections) de la taille et de la position d'un panneau est correcte. Dans certains cas, cette prédiction est meilleure (mieux centrée, bonne taille) que la détection de cercle au même instant (Figure 55).

Figure 54 - Résultat de la prédiction spatio-temporelle

Figure 55 - Détection de cercle en haut - Résultat de la prédiction spatio-temporelle (en bas)