

ASSERVISSEMENT VISUEL D'ENGINS AGRICOLES PAR COMMANDE REFERENCEE VISION.

(TECHNIQUE OF THE VISUAL SERVOING APPLIED ON AGRICULTURAL MOBILE ROBOTS.)

C. DEBAIN, M. DERRAS

Cemagref
Domaine des palaquins
03150 Montoldre, France
Tel. : 70 45 03 12 Fax. : 70 45 19 46

D. KHADRAOUI, P. MARTINET, P. BONTON.

Université Blaise Pascal de Clermont-Ferrand
LASMEA U.R.A. 1793 of the C.N.R.S.
Les Cézeaux, 24 Avenue des Landais
63177 Aubière cedex, France
Tel. : 73 40 72 43 Fax. : 73 40 72 62

Summary : The aim of this article is to present the techniques we have implemented to control our mobile machines along an interface (for example a white line or a limit between a mowed and unmowed vegetation zone). The detection of this limit is realized by an algorithm of segmentation in region using information of a CCD camera. This algorithm is based on the theory of Markov fields applied on texture parameters, deduced from co-occurrence matrices, and luminance parameters obtained from normalized local histograms. From the visual information we have elaborated a control law whose originality, resides in the fact that it uses data of the captor space. We have undertaken different tests on simulator and in real conditions allowing to show the robustness of our algorithms.

Résumé : L'objectif de cet article est de montrer les techniques que nous avons mises en œuvre afin d'asservir un engin mobile sur une interface quelconque (par exemple une ligne blanche ou la limite entre une zone fauchée et une zone non-fauchée d'un couvert végétal). La détection de cette interface est assurée par un algorithme de segmentation en région traitant des informations issues d'une caméra CCD. Cet algorithme est basé sur la théorie des champs de Markov appliquée à des paramètres de texture et de luminance déduits respectivement de matrices de cooccurrences et d'histogrammes locaux normalisés. A partir des informations visuelles extraites, nous avons élaborés une loi de commande permettant d'asservir notre machine. L'originalité de notre loi de commande réside dans le fait qu'elle utilise des données de l'espace du capteur, elle s'affranchit donc de toute reconstruction de la scène. Nous avons effectués différents essais sur simulateur et en conditions réelles permettant de montrer la robustesse de nos algorithmes.

A/ INTRODUCTION.

Dans le domaine de la robotique mobile, le guidage automatique est un problème à multiple facettes. Une des phases les plus importantes est la capture et le traitement

des informations relatant une partie de l'univers du robot. De ce traitement qui dans notre cas concerne une image, nous extrayons une primitive visuelle qui va servir de base à une loi de commande calculée dans l'espace du capteur.

B/ TRAITEMENT D'IMAGES.

I ACQUISITION ET TRAITEMENT DE L'IMAGE.

L'acquisition de l'image se fait au moyen d'une caméra CCD placée à l'avant et sur le côté gauche de la machine. Pour segmenter cette image nous avons choisi une analyse de texture, dont l'essentiel de l'information est contenue dans un voisinage de petite dimension : le texel, d'une taille de 16.16 pixels. Pour déterminer le choix des paramètres de l'image, nous nous sommes orientés vers deux méthodes. La première est basée sur des outils statistiques du premier ordre, et plus particulièrement l'histogramme des niveaux de gris dont deux paramètres ont été extraits. La deuxième concerne des outils statistiques du second ordre, pour laquelle Haralick[1] propose d'utiliser les matrices de cooccurrences dont nous pouvons déduire 14 paramètres. Suite à des essais sur différentes surfaces naturelles, nous en avons retenus deux en fonction de leur robustesse et de leur pouvoir discriminant. Il s'agit des facteurs d'homogénéité et d'entropie.

1.1 Texture et matrices de cooccurrences.

Parmi les nombreuses méthodes d'analyse de la texture d'une image, les matrices de cooccurrence occupent une place importante [1], [18]. Cette méthode part de l'hypothèse que toute l'information texturale est contenue dans une matrice qui retranscrit la distribution spatiale des variations de niveaux de gris entre pixels voisins. Ces matrices sont en fait le reflet des relations qui existent entre les pixels interconnectés d'une région. Ces régions forment à leur tour des primitives de niveaux de gris possédant des caractéristiques particulières qui sont accessibles par la définition de paramètres de texture.

R.M. Haralick définit 14 facteurs qui sont calculés à partir de la matrice de cooccurrence. Pour notre part nous avons mis en oeuvre 4 de ces paramètres dont nous avons mesuré l'efficacité dans le cadre de nos applications. Les 10 autres coefficients étaient inadaptés à notre modèle [15], [17]. Par ailleurs, il nous a paru intéressant de remplacer les coefficients de contraste et d'inertie (qui traduisent des effets de variation de niveaux de gris), par des critères issus d'histogrammes de régions pour réduire les temps de calculs. Par conséquent nous n'avons utilisé que deux de ces caractéristiques. Le coefficient d'homogénéité qui nous donne une idée de l'état de surface de la zone étudiée, et l'entropie qui reflète une notion d'ordre. Ces différentes valeurs sont calculées dans notre cas sur des zones élémentaires de l'image de taille 16.16 pixels :

$$\begin{aligned}
 \text{- Homogénéité} &= \sum_i \sum_j \left\{ \frac{1}{1+(i-j)^2} \cdot P(ij) \right\} \\
 \text{- Entropie} &= \sum_i \sum_j \{ P(ij) \cdot \log\{P(ij)\} \}
 \end{aligned}$$

I 2 Niveaux de gris et histogrammes de régions;

Pour mettre à profit les propriétés de réflectance des surfaces naturelles, mises en évidence lors d'une étape préliminaire à ce projet [12], nous avons défini deux paramètres issus d'histogrammes de régions élémentaires 16.16. Tout d'abord la valeur maximale d'histogramme (valeur du mode), qui traduit la nuance de gris prédominante dans le texel, puis le moment d'ordre 2 qui rend compte de l'amplitude de cette prédominance.

II SEGMENTATION DE L'IMAGE.

A partir de ces quatre paramètres nous avons développé une méthode de segmentation d'images en régions. Zucker [3] et Monga[4] ont défini des critères dans le but d'obtenir une segmentation en régions optimale. De nombreuses méthodes d'analyse de données [5], [6], [8] et [9] nécessitent, à divers niveaux un choix a priori incompatible avec nos applications (nombre de classes d'éléments, seuil de comparaison). Nous nous sommes donc orientés vers des méthodes de segmentation non supervisées. Une des solutions à notre problème est celle apportée par la modélisation par champ de Markov [10], [11]. Cette méthode présente l'intérêt d'être bien adaptée au traitement de séquences d'images, c'est à dire une prise en compte des résultats du traitement de l'image précédente, ce qui est essentiel lorsque l'on traite des séquences d'images avec, à terme, un objectif "temps réel".

Toute la difficulté de notre tâche consiste à segmenter l'image (étiquetage de régions distinctes) à partir de la

seule connaissance de ses divers descripteurs [19], [20]. Traduit sous forme mathématique, le problème consiste à maximiser la probabilité a posteriori $P(X/Y)$, c'est à dire la probabilité de l'objet cherché conditionnellement aux mesures effectuées [21], avec :

- X résultat de la segmentation (champ des étiquettes),
- Y descripteur de l'image.

En appliquant le théorème de Bayes, cette probabilité a posteriori peut s'exprimer sous la forme :

$$P(X/Y) = \frac{P(X) \cdot P(Y/X)}{P(Y)} \quad (1)$$

avec :

- $P(X)$ probabilité a priori sur la connaissance du champ des étiquettes,
- $P(Y/X)$ probabilité conditionnelle de la mesure par rapport à l'une des segmentations possibles. Cette dernière décrit entièrement le processus de transposition "données-étiquettes",
- $P(Y)=1$ (les données sont connues).

Notre image se présente donc sous la forme d'un champ de données (4 paramètres par région élémentaire 16.16 pixels) que nous cherchons à modéliser par un champ des étiquettes qui, à terme, sera notre segmentation de l'image. En nous appuyant sur la théorie des champs de Markov nous limitons les effets de chaque élément sur l'ensemble (effet global), à une interaction locale entre sites voisins. Ces sites voisins sont reliés entre eux par des "cliques" d'ordre divers qui définissent la forme et la taille des systèmes de voisinages considérés.

Nous cherchons ensuite à exprimer sous une autre forme cette première équation de Bayes. Pour cela, le théorème de Hammersley-Clifford nous permet de faire intervenir les distributions de Gibbs qui s'expriment de la manière suivante :

$$P(\omega) = \left(\frac{1}{Z} \right) \cdot \exp\{-U(\omega)\} \quad \text{avec } \omega \in \Omega$$

avec :

- Z constante de normalisation,
- $U(\omega)$ fonction d'énergie. Cette distribution décrit, en quelque sorte, l'équilibre de notre système et de façon plus générale l'équilibre de systèmes physiques présentant des énergies U dépendantes des états ω ,
- ω état particulier du système,
- Ω ensemble des états possibles.

Par conséquent, chacun des termes de l'équation (1) peut s'exprimer sous la forme d'une distribution de Gibbs. Donc :

$$P(X/Y) = P(X) \cdot P(Y/X) \quad \text{devient :}$$

$$\left(\frac{1}{Z}\right) \cdot \exp\{-U(X/Y)\} = \left(\frac{1}{Z}\right) \cdot \exp\{-U(X)\} \cdot \left(\frac{1}{Z}\right) \cdot \exp\{-U(Y/X)\}$$

En utilisant les logarithmes népériens l'expression se réduit alors à :

$$U(X/Y) = U(X) + U(Y/X) \quad (2)$$

L'objectif est de réaliser le calcul des deux termes de la fonction pour minimiser l'énergie $U(X/Y)$ avec :

- $U(X)$ contraintes liées au champ des étiquettes (information a priori),
- $U(Y/X)$ fonction d'énergie liée aux interactions "données-étiquettes".

II 1 Développement de la segmentation.

Pour mettre en oeuvre la méthode de segmentation de l'image en utilisant la modélisation Markovienne nous nous sommes servis de l'algorithme déterministe I.C.M. (Iterated Conditional Mode) [22]. Pour chaque site, les différentes valeurs possibles de la fonction d'énergie sont calculées et seul l'état correspondant à l'énergie minimale est retenu.

Les deux termes de la fonction d'énergie a posteriori $U(X/Y)$ seront donnés dans la pratique par :

- l'énergie a priori,
- l'énergie de transposition.

II 2 Energie a priori.

$U(X)$ fonction dite de "pénalité" (ou a priori). Si nous considérons le voisinage d'un site étudié, nous pouvons alors définir une fonction qui permet de prendre en considération le nombre et le type des voisins de ce site :

$$f(e_s) = A \cdot \left(\sum_{r \in V_s} V_r(e_s) \right)$$

ou V_s est le voisinage du site étudié (x) et A un facteur de pondération.

Si nous prenons l'exemple suivant avec $A=1$:

e1	e1	e1
e2	.x.	e1
e2	e3	e3

alors nous pouvons écrire que :

$$f(e1)=4, f(e2)=2, f(e3)=2.$$

Cette fonction est alors utilisée pour définir un "estimateur de coût", qui permet de quantifier au mieux la pénalité liée au voisinage d'un site étudié. L'un des estimateurs les plus répandus est le M.A.P (Maximum A Posteriori) mais nous pouvons également citer le M.P.M (Maximum of the a Posteriori Marginal). Nous avons, par conséquent utilisé en premier le M.A.P, qui correspond à une loi linéaire décroissante, pour aboutir à des résultats de segmentation en régions très grossières. Il était donc nécessaire de déterminer une fonction de pénalité conservant l'allure décroissante du M.A.P. qui nous fournirait des résultats plus convenables. Nous avons donc utilisé une loi de type $(-1/x)$ tout en éliminant l'asymptote problématique par une fonction de pénalité du type :

$$U(X) = \exp(-f(e_s))$$

Le facteur de pondération A de $f(e_s)$ est fixé à 4.38 pour conserver le plus possible la courbure de la fonction de type $(-1/x)$.

II 3 Energie de transposition.

$U(Y/X)$ fonction de transposition représentée par la distance du CHI-2 entre vecteurs de données (décrit par 4 paramètres). Dans la pratique, les différents termes de la fonction du CHI-2 seront obtenus par :

- $K_i = \sum_{j=1}^n K_{ij}$ est une sommation des données pour chaque descripteur de l'image,
- $K_j = \sum_{i=1}^n K_{ij}$ est une sommation des données pour chaque région de l'image,
- $K = \sum_{j=1}^n \sum_{i=1}^p K_{ij}$ est une sommation de la totalité des données, $n=1024$ et $p=4$.

avec :

- n le nombre de régions (ici 1024, correspondant également au nombre de vecteurs de données),
- p le nombre de descripteurs (ici 4).

La distance entre deux vecteurs j et j' est alors donnée par :

$$d^2(j, j') = \sum_{i=1}^p \left(\frac{K_i}{K_j} \right) \cdot \left(\frac{K_{ij}}{K_j} - \frac{K_{ij'}}{K_{j'}} \right)$$

Dans notre cas, il sera question de définir, plus particulièrement, la distance entre chaque site et une référence générale dans l'image, qui se présente sous la forme de vecteurs de données de même dimension.

III APPLICATION : RECHERCHE DE LA LIMITE DE FAUCHE.

Nous avons appliqué cette théorie à la recherche d'interface fauchée/non fauchée. Différentes étapes sont présentes dans le processus de segmentation Markovien. L'étape n°1, correspondant à l'initialisation du processus, démarre sans aucune connaissance sur la forme ou le nombre de textures dans l'image. Par conséquent, nous formons l'hypothèse que l'image est constituée d'une seule région. Le champ des étiquettes est donc initialisé à 0 et un premier découpage est réalisé à partir de la comparaison de l'énergie locale de chaque site avec la valeur de l'énergie globale moyenne calculée sur la totalité de l'image. Pour cela, nous utilisons un vecteur prototype P_1 dont les composantes sont les moyennes de chaque descripteur pour la totalité de l'image :

$$P_1(H_1, E_1, M_1, Mo_1)$$

$$U_1(X/Y) = \left(\frac{1}{N}\right) \sum_{i=1}^{32} \sum_{j=1}^{32} \{U_{ijP_1}(Y/X) + U_{ij}(X)\}$$

avec :

- H_1, E_1, M_1, Mo_1 , respectivement les moyennes des valeurs d'homogénéité, d'entropie, des valeurs maximales d'histogrammes de région et des moments d'ordre 2 de ces mêmes histogrammes,
- P_1 prototype initial,
- $U_{ijP_1}(Y/X)$ distance du site ij au prototype P_1 ,
- $U_{ij}(X)$ fonction de pénalité du site ij ,
- N nombre de régions définies sur l'image,
- $U_1(X/Y)$ énergie globale moyenne de l'image.

A partir de P_1 et $U_1(X/Y)$:

- nous calculons la fonction d'énergie locale pour chaque site :

$$U_{ij}(X/Y) = U_{ijP_1}(Y/X) + U_{ij}(X)$$

- $U_{ij}(X/Y)$ est alors comparé à $U_1(X/Y)$ tel que si $U_{ij}(X/Y) > U_1(X/Y)$ l'étiquette du site ij est mise à 1, sinon elle reste inchangée à 0.

A l'issue de l'initialisation, nous aurons une classe dans le cas d'une image homogène, deux classes dans les autres cas. L'étape n°2 sert à mettre de "l'ordre" dans le champ des étiquettes. Les fonctions d'énergie relatives aux différentes étiquettes sont calculées et l'état de chaque site étudié est alors modifié en fonction de la région fournissant la valeur d'énergie la plus faible. Ce procédé est réitéré au cours de plusieurs balayages de l'image jusqu'à ce qu'il n'y ait plus de changement dans le champ des étiquettes.

Les étapes n°3 et 4 sont en fait des généralisations des deux premières étapes à l'ensemble des classes détectées.

L'objectif consiste à discerner la présence d'éventuelles nouvelles zones à l'intérieur de celles définies lors des étapes 1 et 2. Comme nous l'avons fait lors de l'étape 1, nous calculons les énergies globales moyennes de chaque zone en présence, ainsi que les prototypes qui leur sont liés. Nous possédons, par conséquent, P_k le prototype de la zone k et $U_k(X/Y)$ l'énergie globale moyenne de la zone k et procédons alors comme suit :

- Les énergies locales correspondant aux étiquettes présentes dans le voisinage du site étudié sont calculées. De toutes ces énergies seule la plus faible est retenue,
- Nous comparons alors $U_{ijk}(X/Y)$ avec $U_k(X/Y)$.

avec :

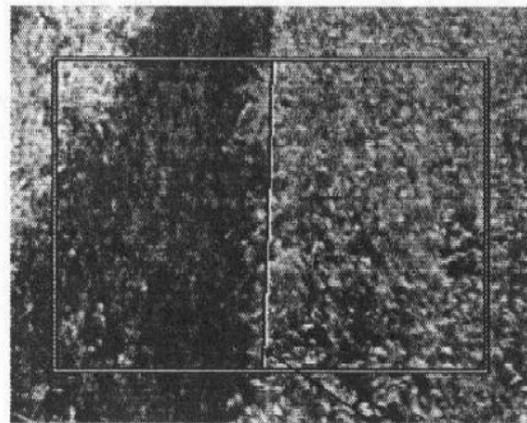
- $U_{ijk}(X/Y)$ l'énergie minimale locale du site ij qui correspond à la zone d'étiquette k ,
- $U_k(X/Y)$ l'énergie globale moyenne de la zone d'étiquette k .

Si $U_{ijk}(X/Y) > U_k(X/Y)$ alors l'étiquette du site ij prend une valeur nouvelle dans le champ des étiquettes, sinon son étiquette reste inchangée.

L'étape n°4 correspond à l'étape n°2 étendue à la totalité des classes détectées. Le processus s'arrête lorsque le nombre de classes détectées ne change plus.

En dynamique, le champ des étiquettes issu de la segmentation de l'image à l'instant T est utilisé comme information a priori pour traiter l'image suivante à l'instant $T+Dt$. Ceci a pour principal objectif de limiter les temps de calcul à une simple réactualisation des étiquettes.

La figure n°1 montre un exemple de résultat de détection d'une limite de fauche. On observe un positionnement correct de l'interface dont les coordonnées polaires (θ et ρ) sont envoyées au programme de commande.



Interface détectée

Figure n°1 : Résultat du traitement d'une image naturelle.

C/ SYNTHÈSE DE LA LOI DE COMMANDE.

A partir des informations visuelles (θ et ρ), extraites par l'algorithme de traitement d'images, nous avons cherché la loi de commande la mieux adaptée à notre application. Nos engins se déplaçant la plupart du temps sur des terrains bosselés, ces derniers induisent un bruit important dans la reconstruction de la scène effectuée à partir des informations 2D issues de la caméra.

Nous nous sommes donc intéressés à des lois de commande ne nécessitant pas cette reconstruction. L'une d'elle, appelée commande référencée vision (CRV), est appliquée avec succès au contrôle des robots manipulateurs [23]. Différentes réalisations ont montré que cette loi de commande était très robuste vis à vis du bruit inclus dans les mesures.

Cependant la CRV n'était pas applicable directement aux robots mobiles du fait de leur non-holonomie. L'objectif de cette deuxième partie sera donc de montrer comment nous avons pu résoudre le problème de la commande dans l'espace image des robots mobiles non holonomes. Pour cela, nous proposons, dans un premier temps, de présenter l'élément sur lequel est fondée la CRV c'est-à-dire le torseur d'interaction. Puis nous expliquerons comment nous avons pu relier les consignes calculées grâce au torseur d'interaction, aux effecteurs du robot, et ce, malgré sa non holonomie.

1 CALCUL DU TORSEUR D'INTERACTION [24], [25],[26]

Le torseur d'interaction (L_{of}^T) est une matrice permettant de relier les vitesses d'un point P dans la scène (données par le torseur cinématique T) à celles de son projeté dans l'image (données par u et v).

1.1/ Calcul du torseur d'interaction pour un point

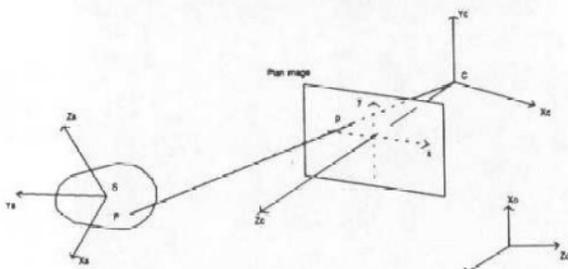


Figure n°2 : Projection du point P dans les différents référentiels.

Soit une scène S ayant comme référentiel R_s et une caméra C ayant pour référentiel R_c , le tout étant placé dans un référentiel galiléen R_0 . Soit P un point de la scène. Nous allons nous intéresser à l'expression de la vitesse de P dans les différents référentiels (Figure n°2). R_c est le référentiel lié à la caméra, R_s étant liée à la scène. Les différents éléments décrits ci-dessus sont en mouvement les uns par rapport aux autres. Exprimons la vitesse de P dans les différents référentiels :

on a :

$$\begin{aligned} \vec{V}_{R_c}(P) &= \vec{V}_{R_c}(C) + \vec{V}_{R_c}(P) + \vec{\Omega}_{R_c/R_s} \wedge C\vec{P} \\ &\text{de même :} \\ \vec{V}_{R_0}(C) &= \vec{V}_{R_0}(S) + \vec{V}_{R_c}(C) + \vec{\Omega}_{R_c/R_0} \wedge S\vec{C} \end{aligned}$$

Prenons comme hypothèse l'immobilité de la scène et de ces objets, on a :

$$\vec{V}_{R_c}(P) = \vec{0} ; \vec{\Omega}_{R_c/R_0} = \vec{0} ; \vec{V}_{R_0}(S) = \vec{0}$$

$$\Leftrightarrow \vec{V}_{R_c}(P) = -\vec{V}_{R_c}(C) - \vec{\Omega}_{R_c/R_c} \wedge C\vec{P} \text{ et } \vec{V}_{R_0}(C) = \vec{V}_{R_c}(C)$$

on obtient donc :

$$\begin{aligned} \vec{V}_{R_c}(P) &= -\vec{V}_{R_0}(C) - \vec{\Omega}_{R_c/R_0} \wedge C\vec{P} \\ &\text{avec :} \\ \vec{\Omega}_{R_c/R_c} &= \vec{\Omega}_{R_c/R_0} - \vec{\Omega}_{R_c/R_0} = \vec{\Omega}_{R_c/R_0} \end{aligned} \quad (3)$$

Les données que nous voulons exploiter doivent être exprimées dans le plan image qui est perpendiculaire à l'axe Z_c et situé à une distance f du centre optique. Ecrivons les équations traduisant cette projection de l'univers 3D dans le plan image :

$$\begin{cases} x = F \frac{X_{R_c}}{Z_{R_c}} \\ y = F \frac{Y_{R_c}}{Z_{R_c}} \end{cases} \quad (4)$$

Différencions ces expressions afin d'avoir la vitesse du point dans le plan image :

En prenant $F = 1$:

$$\begin{cases} \dot{x} = u = \frac{\dot{X}_{R_c}}{Z_{R_c}} - \frac{X_{R_c}}{Z_{R_c}^2} \dot{Z}_{R_c} & (5) \\ \dot{y} = v = \frac{\dot{Y}_{R_c}}{Z_{R_c}} - \frac{Y_{R_c}}{Z_{R_c}^2} \dot{Z}_{R_c} & (6) \end{cases}$$

avec :

$$P_{R_c} \begin{pmatrix} X_{R_c} \\ Y_{R_c} \\ Z_{R_c} \end{pmatrix} \text{ et } \bar{V}_{R_c}(P) \begin{pmatrix} \dot{X}_{R_c} \\ \dot{Y}_{R_c} \\ \dot{Z}_{R_c} \end{pmatrix}$$

Explicitons (3) :

$$\begin{pmatrix} \dot{X}_{R_c} \\ \dot{Y}_{R_c} \\ \dot{Z}_{R_c} \end{pmatrix} = \begin{pmatrix} -V_{R_0}(C)_x - Z_{R_c} \cdot \Omega_{R_c/R_0 y} + Y_{R_c} \cdot \Omega_{R_c/R_0 z} \\ -V_{R_0}(C)_y - X_{R_c} \cdot \Omega_{R_c/R_0 y} + Z_{R_c} \cdot \Omega_{R_c/R_0 x} \\ -V_{R_0}(C)_z - Y_{R_c} \cdot \Omega_{R_c/R_0 x} + X_{R_c} \cdot \Omega_{R_c/R_0 y} \end{pmatrix} \quad (7)$$

avec :

$$V_{R_0}(C) \begin{pmatrix} V_{R_0}(C)_x \\ V_{R_0}(C)_y \\ V_{R_0}(C)_z \end{pmatrix} \text{ et } \Omega_{R_c/R_0} \begin{pmatrix} \Omega_{R_c/R_0 x} \\ \Omega_{R_c/R_0 y} \\ \Omega_{R_c/R_0 z} \end{pmatrix}$$

Remplaçons (7) dans (5) :

$$\begin{aligned} u = \dot{x} = & -\frac{V_{R_0}(C)_x}{Z_{R_c}} - \Omega_{R_c/R_0 y} + \frac{Y_{R_c}}{Z_{R_c}} \Omega_{R_c/R_0 z} \\ & + \frac{X_{R_c}}{Z_{R_c}^2} V_{R_0}(C)_z + \frac{X_{R_c} Y_{R_c}}{Z_{R_c}^2} \Omega_{R_c/R_0 x} \\ & - \frac{X_{R_c}^2}{Z_{R_c}^2} \Omega_{R_c/R_0 y} \end{aligned}$$

D'après (4) on a avec $f = 1$:

$$\begin{cases} X_{R_c} = x \cdot Z_{R_c} \\ Y_{R_c} = y \cdot Z_{R_c} \end{cases}$$

$$\begin{aligned} u = & -\frac{V_{R_0}(C)_x}{Z_{R_c}} - \Omega_{R_c/R_0 y} + y \cdot \Omega_{R_c/R_0 z} + \frac{x}{Z_{R_c}} V_{R_0}(C)_z \\ & + x \cdot y \cdot \Omega_{R_c/R_0 x} - x^2 \cdot \Omega_{R_c/R_0 y} \end{aligned}$$

$$\begin{aligned} u = & -\frac{1}{Z_{R_c}} V_{R_0}(C)_x + \frac{x}{Z_{R_c}} V_{R_0}(C)_z + x \cdot y \cdot \Omega_{R_c/R_0 x} \\ & - (1+x^2) \cdot \Omega_{R_c/R_0 y} + y \cdot \Omega_{R_c/R_0 z} \end{aligned} \quad (8)$$

Remplaçons (7) dans (6) :

$$\begin{aligned} v = & -\frac{1}{Z_{R_c}} V_{R_0}(C)_y + \frac{X_{R_c}}{Z_{R_c}} \Omega_{R_c/R_0 z} + \Omega_{R_c/R_0 x} \\ & - \frac{Y_{R_c}}{Z_{R_c}^2} (-V_{R_0}(C)_z - Y_{R_c} \Omega_{R_c/R_0 x} + X_{R_c} \Omega_{R_c/R_0 y}) \\ v = & -\frac{1}{Z_{R_c}} V_{R_0}(C)_y + \frac{y}{Z_{R_c}} V_{R_0}(C)_z + \Omega_{R_c/R_0 x} \cdot (1+y^2) \\ & - x \cdot y \cdot \Omega_{R_c/R_0 y} - x \Omega_{R_c/R_0 z} \end{aligned} \quad (9)$$

On cherche :

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} u \\ v \end{pmatrix} = L_{of}^T \cdot T \text{ avec :}$$

$$T(V_{R_0}(C)_x; V_{R_0}(C)_y; V_{R_0}(C)_z; \Omega_{R_c/R_0 x}; \Omega_{R_c/R_0 y}; \Omega_{R_c/R_0 z})^T$$

donc :

$$L_{of}^T = \begin{pmatrix} -\frac{1}{Z_{R_c}} & 0 & \frac{x}{Z_{R_c}} & xy & -(1+x^2) & y \\ 0 & -\frac{1}{Z_{R_c}} & \frac{y}{Z_{R_c}} & (1+y^2) & -xy & -x \end{pmatrix} \quad (10)$$

1.2/ Torseur d'interaction pour le cas d'une droite.

Dans un univers 3D une droite est représentée par l'intersection de 2 plans non parallèles et nous est donnée dans le référentiel lié à la caméra sous la forme :

$$h(X, Y, Z, Q) = \begin{cases} a_1 \cdot X + b_1 \cdot Y + c_1 \cdot Z + d_1 = 0 \\ a_2 \cdot X + b_2 \cdot Y + c_2 \cdot Z + d_2 = 0 \end{cases}$$

Projetons ces deux équations dans le plan image grâce aux formules de projections :

$$\begin{cases} x = X/Z \\ y = Y/Z \end{cases}$$

$$\begin{cases} 1/Z = -\frac{(a_1 \cdot X + b_1 \cdot Y + c_1)}{d_1} \\ 1/Z = -\frac{(a_2 \cdot X + b_2 \cdot Y + c_2)}{d_2} \end{cases} \quad (11)$$

On peut donc par égalité écrire :

$$\frac{(a_2 \cdot X + b_2 \cdot Y + c_2)}{d_2} + \frac{(a_1 \cdot X + b_1 \cdot Y + c_1)}{d_1} = 0$$

$$\Leftrightarrow (a_1 \cdot d_2 - a_2 \cdot d_1) \cdot x + (b_1 \cdot d_2 - b_2 \cdot d_1) \cdot y + (c_1 \cdot d_2 - c_2 \cdot d_1) = 0$$

L'équation cherchée est de la forme : $Ax + By + C = 0$

$$\begin{cases} A = a_1 \cdot d_2 - a_2 \cdot d_1 \\ B = b_1 \cdot d_2 - b_2 \cdot d_1 \\ C = c_1 \cdot d_2 - c_2 \cdot d_1 \end{cases}$$

Cette représentation n'est pas minimale ; c'est pourquoi on va choisir une autre représentation n'utilisant que 4 paramètres.

Représentation polaire

$$g(\underline{X}, \underline{q}) = x \cos \theta + y \sin \theta - \rho = 0 \quad (12) \quad \text{avec } \underline{q} = (\rho, \theta) \text{ et } \underline{X} = (x, y).$$

Si nous considérons la représentation de la droite :
 $Ax + By + C = 0$
 on obtient :

$$\left. \begin{matrix} A = \cos \theta \\ B = \sin \theta \end{matrix} \right\} \theta = \arctan\left(\frac{B}{A}\right) \text{ et } \rho = -\frac{C}{\sqrt{A^2 + B^2}}$$

en dérivant $g(\underline{X}, \underline{q}) = 0$ on obtient :

$$\frac{\partial g}{\partial q}(\underline{X}, \underline{q}) \dot{q} = -\frac{\partial g}{\partial X}(\underline{X}, \underline{q}) \dot{X}$$

$$\dot{\rho} + \dot{\theta}(x \cdot \sin \theta - y \cdot \cos \theta) = \dot{x} \cdot \cos \theta + \dot{y} \cdot \sin \theta \quad (13)$$

en utilisant (12) remplaçons x en fonction de y :

$$x = \frac{\rho - y \sin \theta}{\cos \theta}$$

(13) devient :

$$\dot{\rho} + \left[\left(\frac{\rho - y \sin \theta}{\cos \theta} \right) \cdot \sin \theta - y \cdot \cos \theta \right] \cdot \dot{\theta} = \dot{x} \cdot \cos \theta + \dot{y} \cdot \sin \theta$$

$$\dot{\rho} + [(\rho - y \sin \theta) \cdot \tan \theta - y \cdot \cos \theta] \cdot \dot{\theta} = \dot{x} \cdot \cos \theta + \dot{y} \cdot \sin \theta$$

$$\begin{aligned} \dot{\rho} + \rho \cdot \tan \theta \cdot \dot{\theta} - [y \sin \theta \cdot \tan \theta + y \cdot \cos \theta] \cdot \dot{\theta} \\ = \dot{x} \cdot \cos \theta + \dot{y} \cdot \sin \theta \end{aligned}$$

$$\begin{aligned} \dot{\rho} + \rho \cdot \tan \theta \cdot \dot{\theta} - \left[y \frac{\sin^2 \theta}{\cos \theta} + y \frac{\cos^2 \theta}{\cos \theta} \right] \cdot \dot{\theta} \\ = \dot{x} \cdot \cos \theta + \dot{y} \cdot \sin \theta \end{aligned}$$

$$\left[-\frac{\dot{\theta}}{\cos \theta} \cdot y + \dot{\rho} + \rho \cdot \tan \theta \cdot \dot{\theta} \right] = \dot{x} \cdot \cos \theta + \dot{y} \cdot \sin \theta \quad (14)$$

A partir de l'équation (12), on écrit X en fonction de Y si $\cos \theta \neq 0$ (ou Y en fonction de X sinon) et l'équation (14) peut alors s'écrire, en utilisant (10) et (11) :

$$\begin{aligned} -\left(\frac{\dot{\theta}}{\cos \theta} \right) \cdot Y + (\dot{\rho} + \rho \cdot \tan \theta \cdot \dot{\theta}) \\ = Y \cdot K_1 \cdot T + K_2 \cdot T \end{aligned}$$

on en déduit aussitôt :

$$\begin{cases} \dot{\theta} = -K_1 \cdot \cos \theta \cdot T \\ \dot{\rho} = (K_2 + K_1 \cdot \sin \theta) \cdot T \end{cases}$$

d'où :

$$\begin{pmatrix} \dot{\theta} \\ \dot{\rho} \end{pmatrix} = \begin{pmatrix} \lambda_\theta \cdot \cos \theta & \lambda_\theta \cdot \sin \theta & -\lambda_\theta \cdot \rho & -\rho \cdot \cos \theta & -\rho \cdot \sin \theta & -1 \\ \lambda_\rho \cdot \cos \theta & \lambda_\rho \cdot \sin \theta & -\lambda_\rho \cdot \rho & (1+\rho^2) \cdot \sin \theta & -(1+\rho^2) \cdot \cos \theta & 0 \end{pmatrix} \cdot T$$

$$\text{avec : } \lambda_\theta = \frac{1}{d_i} \cdot (a_i \cdot \sin \theta - b_i \cdot \cos \theta) \text{ et}$$

$$\lambda_\rho = \frac{1}{d_i} (a_i \cdot \rho \cdot \cos \theta + b_i \cdot \rho \cdot \sin \theta + c_i)$$

II/ RECHERCHE THEORIQUE DE LA LOI DE COMMANDE.

II.1/ Calcul du torseur d'interaction pour notre application :

L'objectif de notre travail est d'asservir nos engins sur une ligne symbolisant l'interface fauchée/non fauchée d'un espace naturel. Contrairement aux différents travaux qui ont été réalisés sur le suivi de lignes blanches [24], nous n'avons qu'une seule droite de référence. Calculons le torseur d'interaction pour cette droite :

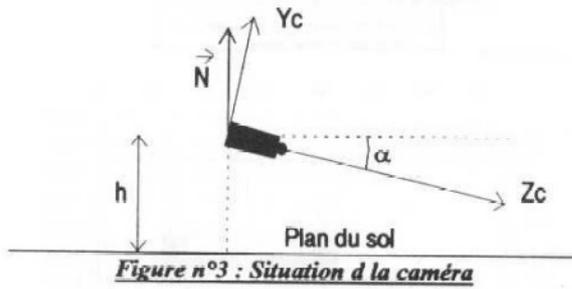


Figure n°3 : Situation d la caméra

Equation du plan du sol dans le repère caméra :

Soit un vecteur normal au plan du sol : $\vec{N} \begin{pmatrix} 0 \\ \cos \alpha \\ -\sin \alpha \end{pmatrix}$

L'équation du plan est de la forme :

$$(\cos \alpha) \cdot y - (\sin \alpha) \cdot z + cte = 0$$

Soit M, un point du plan : $M \begin{pmatrix} 0 \\ -h/\cos \alpha \\ 0 \end{pmatrix}$

$$\Rightarrow (\cos \alpha) \cdot \left(-\frac{h}{\cos \alpha}\right) + 0 + cte = 0 \Rightarrow cte = h$$

L'équation du plan du sol est :

$$\underline{(\cos \alpha) \cdot y - (\sin \alpha) \cdot z + h = 0}$$

Equation de la droite de référence en polaire

L'équation de la droite de référence dans le repère caméra est :

$$\begin{cases} (\cos \alpha) \cdot y - (\sin \alpha) \cdot z + h = 0 \\ x = 0 \end{cases}$$

Ce qui donne en coordonnées polaires dans le repère écran :

$$\begin{cases} \rho^* = 0 \\ \theta^* = 0 \end{cases}$$

D'après ces équations on trouve :

$$\begin{aligned} \Rightarrow \lambda_{\theta^*} &= \frac{1}{h} (-\cos \alpha) \\ \lambda_{\rho^*} &= \frac{1}{h} (-\sin \alpha) \end{aligned}$$

Le torseur d'interaction est donc égal à :

$$\Rightarrow L^T = \begin{pmatrix} -\frac{\cos \alpha}{h} & 0 & -1 \\ \frac{\sin \alpha}{h} & -1 & 0 \end{pmatrix} \cdot T_c \begin{pmatrix} V_x \\ \Omega_y \\ \Omega_z \end{pmatrix}$$

A partir de ce torseur nous déterminons la loi de commande pour une seule tâche :

$$\begin{aligned} T_c &= -\lambda \cdot L^T \cdot (S - S^*) & S^* &= \begin{pmatrix} \theta^* \\ \rho^* \end{pmatrix} \\ S &= \begin{pmatrix} \theta \\ \rho \end{pmatrix} & L^T &= L^T \cdot (L^T \cdot L^T)^{-1} \end{aligned}$$

S représente les coordonnées polaires de la ligne détectée et S* les coordonnées polaires de la droite de référence dans le repère écran.

En développant les calculs et en gardant le terme en Ω_y , nous obtenons :

$$\Omega_y = \lambda \cdot (\tan \alpha \cdot (\theta - \theta^*) + (\rho - \rho^*)) \quad (15)$$

Toute la difficulté est de relier cette rotation (Ω_y) aux effecteurs du robots.

II.2/ Equation cinématique des véhicules :

Les plateformes expérimentales que nous possédons sont de deux sortes. La première est du type chenillard (figure n°4), et la seconde est un engin équipée de deux roues directrices à l'arrière.

Le chenillard est contrôlé par la différence de vitesse entre ses roues droites et gauches ($\Delta V = V_2 - V_1$). Pour le deuxième engin nous contrôlons l'angle des roues arrières (δ) (figure n°6).

Ces deux véhicules, lorsqu'ils se déplacent à vitesse constante, décrivent des trajectoires en forme de cercle. Nous pouvons donc les modéliser par un seul essieu à deux roues qui décrirait ce même cercle.

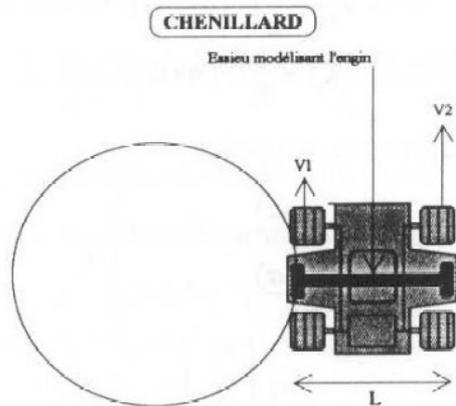


Figure n°4 : Trajectoire du chenillard.

L'orientation de cet essieu est liée à la vitesse de ces roues par la relation :

$$\Omega_y = \frac{V_2 - V_1}{L}$$

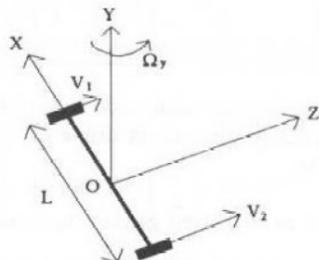


Figure n°5 : Cinématique de l'essieu modélisant le robot.

Pour le chenillard la variable $(V_2 - V_1)$ est directement contrôlable :

$$V_2 - V_1 = \Omega_y \cdot L$$

Pour le deuxième engin l'essieu qui le modélise est celui qui n'a pas de roues directrices. Cet essieu décrit lui aussi un cercle répondant à la même équation :

$$\Omega_y = \frac{V_2 - V_1}{L}$$

Engin à deux roues directrices

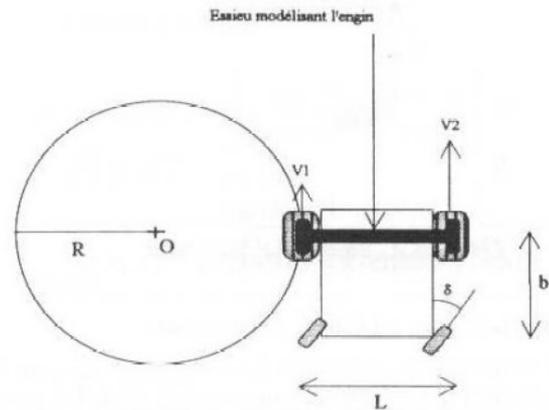


Figure n°6 : Trajectoire du véhicule à deux roues directrices.

Cependant sur cet engin la variable contrôlée est (δ) .

(δ) est lié à R par la relation :

$$\tan \delta = \frac{b}{R} \Rightarrow \delta = \frac{b}{R} \text{ car } \delta \text{ est petit.}$$

de plus avec V vitesse moyenne de l'engin :

$$R = \frac{L \cdot V}{V_2 - V_1} = \frac{L \cdot V}{\Delta V}$$

donc on obtient :

$$\delta = \frac{b \cdot \Omega_y}{V}$$

Donc d'après les équations qui précèdent, nous pouvons commander nos engins à partir de la donnée Ω_y .

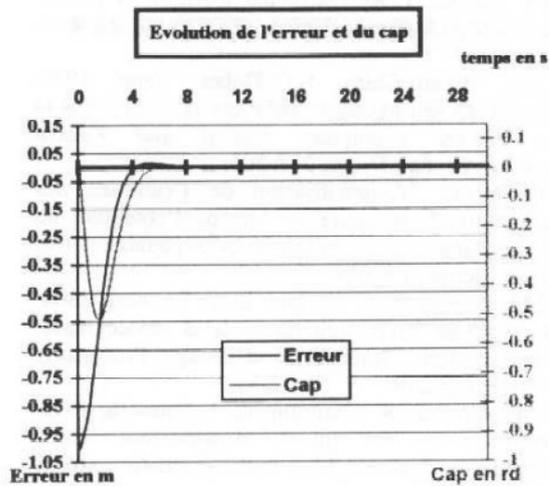
III/ ESSAIS EN SIMULATION.

Dans un premier temps nous avons simulé le comportement de nos machines en leur appliquant un échelon d'un mètre d'amplitude.

Ces simulations nous fait apparaître une différence très importante entre les termes dépendant de (ρ) et ceux dépendant de (θ) (θ étant exprimé en radian et ρ en pixel). Nous avons donc décidé d'introduire des gains dans l'équation n°15. Celle ci devient alors :

$$\Omega_y = K_1 \cdot \tan \alpha \cdot (\theta - \theta^*) + K_2 \cdot (\rho - \rho^*)$$

Les résultats donnés sur le graphe n°1 (pour le chenillard) montrent un asservissement correct de la machine et nous ont permis de régler nos gains ($K_1 = 15$ et $K_2 = 0.005$).



Graphe n°1 : Réponse à un échelon en simulation.

De la même manière pour le deuxième engin nous avons déterminé à l'aide du simulateur les gains les mieux adaptés à l'application à savoir ($K_1 = 8$ et $K_2 = 0.004$).

IV/ ESSAIS EN CONDITIONS REELLES.

Après avoir simulé avec succès le comportement de nos engins nous avons testé nos lois de commande en condition réelles. Pour cela, nous avons tracé sur le sol une ligne blanche avec un échelon de un mètre dans le but de comparer nos résultats avec ceux du simulateur figure n°7. La ligne blanche a été retenue car elle est facile à détecter, ce qui nous garantissait une excellente qualité de traitement d'images.

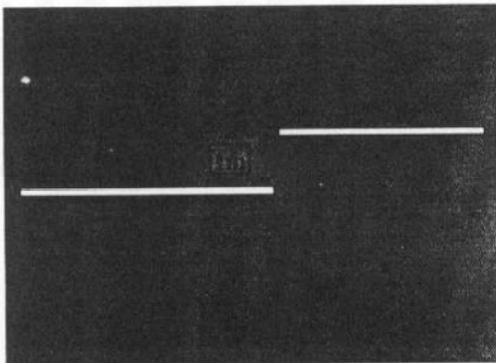


Figure n°7 : Vue du terrain d'essais.

Pour analyser la qualité de notre loi, de commande nous avons placé sur le côté de l'engin, une caméra de contrôle perpendiculaire au sol (figure n°8).

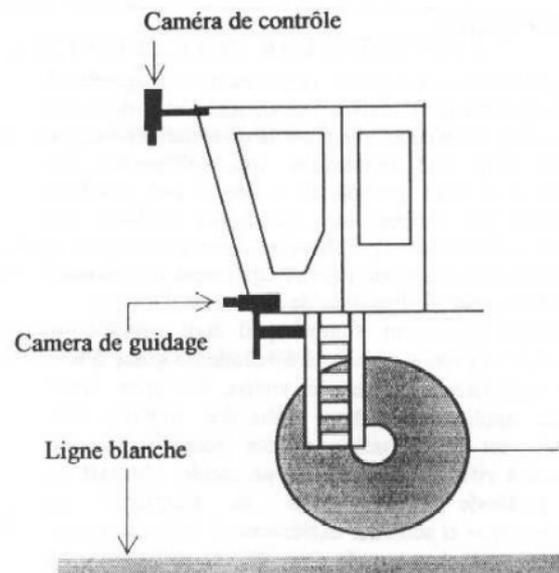
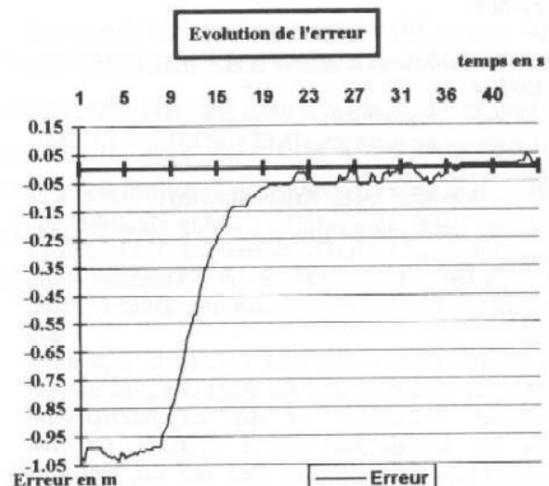


Figure n°8 : Vue des 2 caméras permettant le guidage et le contrôle de la qualité de l'asservissement.

Grâce à un algorithme automatique de dépouillement et à la connaissance de la position du centre de l'image par rapport à la machine nous pouvons mesurer à chaque instant la position de la machine par rapport à la ligne blanche. Les résultats donnés sur le graphe n°2 montrent un bon comportement de notre loi de commande. Le bruit présent sur la courbe est essentiellement dû aux vibrations de la caméra de contrôle et à la précision d'environ 2 cm de l'algorithme de dépouillement.



Graphe n°2 : Réponse à un échelon en conditions réelles.

D/ CONCLUSION

Dans cet article nous avons voulu montrer le potentiel important offert par l'analyse Markovienne et surtout son adaptabilité, notre problème étant la diversité des images acquises dans des conditions très différentes. La méthode que nous proposons a donné des résultats satisfaisant par rapport aux techniques utilisant des seuils de comparaison. L'utilisation d'une information a priori dans cette méthode permet également de résoudre le problème posé par l'analyse de séquences d'images.

Nous avons également montré qu'il était possible de commander un robot mobile non-holonyme grâce à une loi de commande référencée vision. De plus notre méthode appliquée à deux véhicules mobiles très différents est généralisable à tout robot mobile se déplaçant à vitesse constante sur un cercle. L'intérêt de cette méthode réside dans sa simplicité de programmation et son coût extrêmement faible en temps de calcul puisqu'un simple PC AT est suffisant pour assurer toute la partie commande. Cependant nos gains K_1 et K_2 qui sont déterminés arbitrairement, ne sont pas optimum et ne peuvent s'adapter aux variations de vitesse moyenne de l'engin. Ils résultent donc d'un compromis qui a ses limites. C'est pourquoi nous nous orientons vers des méthodes permettant de calculer ces gains en fonction du modèle cinématique du véhicule et de sa vitesse moyenne [27].

E/ BIBLIOGRAPHIE.

- [1] R.M. Haralick, may 1979. Statistical and structural approaches to texture. Proceeding of the IEEE, vol. 67, n°5, 786-804.
- [3] S.W. Zucker, 1976. Region growing : childhood and adolescence. Computer graphics and Image processing, n°5, 382-399.
- [4] O. Monga "Segmentation d'images : ou en sommes nous ?", Rapport de recherche INRIA n°1216, avril 1990, 63 pages.
- [5] S.C. Ahalt, A.K. Krishnamurthy, P. Chen, D.E. Melton, 1990. Competitive learning algorithm for vector quantization. Neural Networks, vol. 3, 277-290.
- [6] L. Lebart, A. Morineau, N. Tabard, 1977. Techniques de la description statistique. Dunod, Paris, 351 pages.
- [8] M. Derras, M. Berducat and P. Bonton, september 1991. Vision guided mower for the upkeep of natural environment. Poceedings of the 1st International seminar of on-machine vision systems for the agricultural and bio-industries (3rd and 6th technical Sections of the CIGR), Montpellier, 10 pages.
- [9] M. Popovic, F. Chantemargue, R. Canals, P. Bonton, september 1991. Several approaches to implement the merging step of the split and merge region segmentation. EUROGRAPHICS, Vienne.
- [10] J.R. Sullins, 1990. Distributed learning of texture classification", Computer Vision, ECCV 90, Antibes, 349-358.
- [11] Chen, Chaur Chin, R.C. Dubes, june 1989. Experiments in fitting discrete Markov random fields to texture. CVPR (Computer Vision and Pattern Recognition) 89, San-Diego, 248-303.
- [12] M. Derras, "Automatisation de l'entretien des espaces naturels", mémoire en vue de l'obtention du D.E.A. en "Electronique et systèmes", septembre 1990, Université Clermont II, 84p.
- [15] M.M. Trivedi, C.A. Harlow, R.W. Connors, 1984. Object detection based on gray level cooccurrence. Computer Vision, Graphics, and Image Processing, vol. 28, n°2, 199-219.
- [17] R.M. Haralick, K. Shanmugam, I. Dinstein, 1973. Textural features for image classification. IEEE Transactions on Systems, Man and Cybernetics, vol. 3, n°6, 610-621.
- [18] P.C. Chen, T. Pavlidis, 1979. Segmentation by texture using a cooccurrence matrix and a split and merge algorithm. Computer Graphics and Image Processing, vol. 10, 172-182.
- [19] H. Derin, H. Elliot, january 1987. Modeling and segmentation of noisy and textured image using Gibbs random fields. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-9, n°1, 39-55.
- [20] D. Genam, S. Genam, C. Graffigne, Ping Dong, july 1990. Boundary detection by constrained optimization. IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 12, n°7, 609-628.
- [21] R. Canals, P. Bonton, avril 1991. Segmentation spatio-temporelle : régularisation par champs markoviens. Rapport interne au Laboratoire d'Electronique de l'Université Blaise-Pascal, Clermont II, 31 pages.
- [22] M. Bernard, M. Sigelle, mai 1990. Champs de Gibbs et Champs markoviens. Télécom Paris, Département Images, Groupe Image.
- [23] Motyl G., Martinet P., Gallice J. (1993) : Visual Servoing with Respect to a Target Sphere Using a Camera/Laser-Stripe Sensor, in proceedings'93 International Conference on Advanced Robotics, Tokyo, Japan, Nov.1-2.
- [24] F. Chaumette : La commande référencée vision : une approche aux problèmes d'asservissement visuels en robotique. Thèse soutenue le 19 juillet 1990. Université de RENNES.
- [25] C. Samson, M. le Borgne, B. Epiau : Robot Control, The Task Function Approach. Oxford science publications, 1991, 364 p, ISBN 0-19-8538057.
- [26] R. Pissard-Gibollet, P. Rives : Asservissement Visuel appliqué à un Robot Mobile : état de l'art et modélisation cinématique. Rapport de recherche n° 1577, INRIA, Décembre 1991.
- [27] D. Khadraoui, P. Martinet, J. Gallice, "Linear Control of High-Speed Vehicle in Image Space", to appear at : International Conference on Industrial Automation, Nancy 7-9 june 1995