République **A**lgérienne **D**émocratique et **P**opulaire
Ministère de L'**E**nseignement **S**upérieur et de la A **R**echerche **S**cientifique

Université **M**ouloud **M**ammeri De **T**izi-**O**uzou

Faculté De **G**énie **E**lectrique Et D'informatique
DEPARTEMENT D'AUTOMATIQUE

Mémoire de Fin d'Etude de MASTER ACADEMIQUE

Spécialité : **commande des systèmes.**

Présenté par
HAMMAD NOURIA
HOCINI OULHADJ

Mémoire dirigé par **PROFESSEUR DIAF**

Thème

ETUDE ET REALISATION D'UN SYSTÈME DE DETECTION ET SUIVI D'UN OBJET EN MOUVEMENT

Mémoire soutenu publiquement le 25 septembre 2017 devant le jury composé de :

Mme Sadia HAMMOUCHE
MCA, UMMTO, Présidente

M Moussa DIAF
Professeur, UMMTO, Rapporteur

M Takfarinas CHILLI
MAB, UMMTO, Exa

Avant-propos

Ce travail a été réalisé au Laboratoire Vision Artificielle et Automatique des Système (LVAAS)du département Automatique (LVAAS),FGEI,UMMTO.

Il fait suite au travail d'un mémoire de Master réalisé par monsieur IRATNI Salim soutenu et réalisé dans ce même laboratoire.
Nous avons apporté des modifications en dans ce qui concerne la commande des moteur de la camera et de la méthode de détection et de suivi d'objet en mouvement.
Nous tenons à remercier vivement monsieur DIAF moussa, professeur pour nous avoir proposé ce sujet et pour nous avoir guidé et dirigé durant ce travail.
Nos vifs remerciements aussi à monsieur YOUNSI Marzouk, YESLI Samir et IRATNI Salim, doctorants au LVAAS pour nous avoir aidés et encouragés durant ce projet.

Nous ne manquerons pas de remercier Madame HAMMOUCHE Sadia MCA au département Automatique, FGEI, UMMTO, pour nous avoir fait honneur en acceptant de présider le jury de ce mémoire.

Nos sincères remerciements vont aussi à Monsieur CHELLI Takfarinas, MAB au département Automatique, FGEI, UMMTO pour avoir accepté de faire partie du jury d'examen de notre mémoire.

Sommaire

Liste des figures

Introduction générale

De nos jours, les performances est L'augmentation constante de la puissance des ordinateurs la généralisation de l'utilisation des images, l'analyse du mouvement dans les vidéos s'est révélée être un outil indispensable pour des applications aussi divers que la vidéo surveillance, l'imagerie médicale, la robotique,…etc. En effet l'un des plus grands rêves de l'homme est de pouvoir faire exécuter par des acteurs autres que lui-même des tâches ou activités qu'il considère comme, dangereuses ou simplement ennuyeuses et non gratifiantes. Ce qu'il a commencé à se réaliser avec l'émergence de la robotique au cours des années 1960.

Depuis leur apparition, les robots industriels réalisés des tâches classiques qui peuvent toujours se ramener à des interactions précises avec leur environnement. De ce fait, une modification de ce dernier peut entraîner de graves conséquences sur le déroulement de la tâche à accomplir. C'est ce qui a poussé les chercheurs à aller vers des systèmes avec capteurs de vision, afin de lui permettre d'appréhender son environnement et de réagir à d'éventuels changements de ce dernier. Ceci a permis d'améliorer de manière significative leurs performances et d'aider à développer de nouvelles applications d'une part et l'émergence d'un nouveau domaine de recherche sous le nom d'asservissement visuel.

Un autre domaine aussi qui a fait un objet de recherche durant ces trente dernières années et qui reste toujours un domaine où de nouvelles approches sont à prospecter est le domaine de détection de mouvements, est de suivie d'objet en mouvement donc L'analyse du mouvement est un vaste sujet qui englobe un certain nombre de problématiques :

- La détection des objets en mouvement, c'est-à-dire la détection d'un ensemble de régions d'intérêt en mouvement dans la scène observée

- Le suivi de cible ou de régions ciblées, dont le but est de déterminer la position de chaque cible ou région ciblée dans l'image à chaque instant.

La problématique détection des objets en mouvement, est en général une première étape pour des outils automatiques de vision par ordinateur, et la vision artificielle. Ces outils peuvent avoir pour vocation, soit uniquement de détecter, soit de détecter et reconnaître, soit de détecter et

suivre des objets, par exemple, analyser le comportement ou la trajectoire de ces objets en mouvement.

L'autres problématiques est aussi importante et nécessite la mise en place de méthodes simples et robustes. Tous ces sujets font l'objet d'un grand nombre de travaux, mais il n'existe pas, à l'heure actuelle, d'algorithmes aboutis s'adaptant à n'importe quelle situation.

Le travail réalisé dans ce mémoire s'inscrit dans le domaine de détection et de suivie d'objet en mouvement et d'asservissement visuel. L'objectif consiste à utiliser la configuration caméra déportée et la technique Look and Move statique pour réaliser un asservissement visuel 2D d'un système articulé. En effet, le travail se limite à l'étude, sur le plan théorique et pratique, de la commande d'un système comportant une caméra, un système articulé qui pivote la caméra sur axes et un ordinateur connecté à une carte d'acquisition et de commande qui est une ARDUINO UNO. Le fonctionnement de cet ensemble consiste à détecter puis suivre en temps réel, un objet en mouvement dans le champ de vision de la caméra et guider cette dernière en temps réel de telle manière à ce que cet objet soit toujours suivi par l'organe visuel et de le placer au milieu du champ de vision.

Pour réaliser la commande en position de cette caméra, trois parties ont été effectuées dans, la première, nous avons effectué une liaison série entre le carte ARDUINO UNO et le logicielle matlab. Puis, en deuxième partie nous avons programmé grâce au logicielle matlab le programme qui s'en charge de la détection et de suivie c'est-à-dire le tracking en temps réel de n'importe quelle objet en mouvement puis en 3éme et dernier partie nous avons programmé grâce aux logicielle arduino un programme qui s'en charge de commander directement les servomoteur est de récupéré les donnée nécessaire c'est-à-dire le centre du rectangle de l'objet en mouvement ciblée pour donner a ce dernier la position nécessaire aux servomoteur pour qu'il puisse tourner avec exactitude.

Ainsi, le mémoire est organisé comme suit :

Le premier chapitre est dédié à la détection de mouvements et le suivie d'objet en mouvement et la méthode choisie pour ce tracking. Le deuxième chapitre présente des généralités sur l'asservissement visuel et les lois de commande liées à ce dernier. Le troisième chapitre est consacrée à expliquer les différentes étapes de la réalisation du système de suivie, en divisant ce dernier en trois étapes : partie matérielle, partie software et partie résultats expérimentaux.

Chapitre 1

Détection et suivi automatiques de mouvements

1.1. Introduction

La détection et le suivi automatiques d'objets en temps réel,prend un essor important dans de nombreux domaines d'applicationscomme la télésurveillance civile et militaire qu'elle soit terrestre, aérienne ou marine, la biologie ou encore la robotique mobile. Ce domaine a été très actif depuis les années 1970 et grâce à la puissance des ordinateurs, ce domaine ne cesse de croître depuis les années 1990. La problématique proposée dans ce chapitre consiste à détecter puis suivre en temps réel un objet de taille et de texture quelconque. Pour ce faire nous allons présenter les différentes méthodes de détection et de suivi utilisées dans ce type de problème. Un bref aperçu de l'algorithme utilisé sera présenté par la suite.

1.2. Séquence d'images

Une séquence d'images est une succession d'images bidimensionnelles qui montre l'évolution temporelle d'une scène. La cadence est de 25 images par seconde, ce qui correspond au seuil à partir duquel l'œil humaine perçoit la séquence comme un stimulus continu, grâce à la persistance rétinienne [1]. On appelle ces images «trame» ou «plan».

1.3. Détection de mouvement

La détection du mouvement est l'étape initiale pour chaque algorithme de suivi d'objets. Il s'agit de déterminer parmi les pixels de l'image courante lesquels appartiennent à l'arrière-plan de la scène et lesquels représentent des objets en mouvement. C'est une étape critique et difficile car elle doitêtre

robuste aux variations de la luminosité de la scène ainsi que la présence des ombres [2]. Pour ce faire plusieurs méthodes ont étés proposées. Ces méthodes ont été classées selon le schéma de la figure (1.1) [3] :

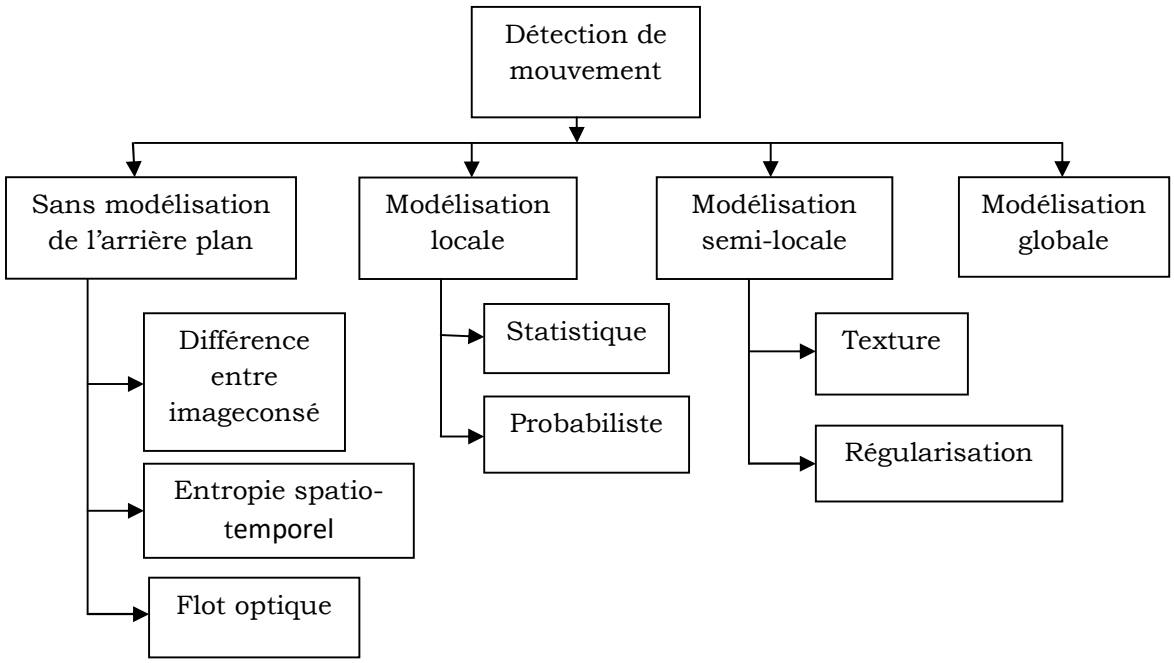


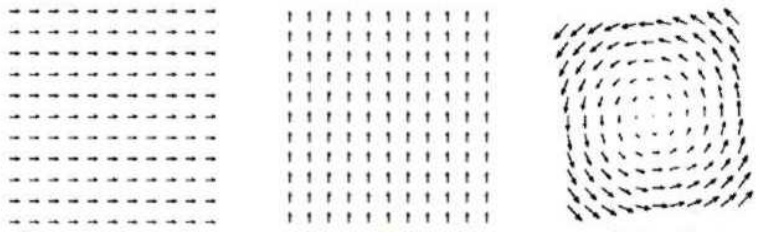
Fig.1.1 Méthodes de détection de mouvements [3]

1.3.1 Détection sans modélisation de l'arrière plan

Il s'agit de calculer une quantité mathématique en tout point de l'image fonction de l'intensité ou de la couleur de l'ensemble des pixels qui est censée refléter l'importance du mouvement visible dans la scène sans prendre en considération aucune information de l'arrière plan. Comme exemple on peut citer: la dérivée temporelle, l'entropie spatio-temporelle et le flot optique[3].

Le flot optique

Le flot optique est un champ de vecteur à deux dimensions représenté par la projection de mouvement réel sur le plan image.Cette méthode est très coûteuse en temps de calcul ce qui nécessite d'adopter des algorithmes plus rapides. Parmi les algorithmes de base du flot optique, on peut citer celui de Lucas et Kanade [4]qui est fondée sur l'hypothèse de conservation de l'intensité lumineuse de la scène (fig.1.2).



(a) translation x (b) translation y (c) rotation

Fig.1.2 Exemple de champ de mouvement

La dérivée temporelle

Le principe de la méthode de la dérivée temporelle consiste à mesurer le changement d'apparence des pixels entre deux trames consécutives dans un flux vidéo.La première utilisation de cette méthode dans l'analyse de séquences vidéo est généralement attribuée à Jain et Nagel [5]. L'estimation de la dérivée temporelle instantanée du signal au temps t est donnée par :

$\forall(x, y) \in E \forall t > 0 \quad It(x, y, t) \approx |I(x, y, t) - I(x, y, t - \Delta t)|$ (1.1)

Si $It(x, y, t) \neq 0$, alors le pixel a bougé sinon pas de mouvement.

Cette méthode se montre peu robuste face à des phénomènes tels que les mouvements lents ou saccadés, les arrêts brefs d'un objet en mouvement, ou encore la présence de trames redondantes dans certaines séquences vidéo, ce qui nécessite d'effectuer un lissage temporel de la séquence.

La soustraction de l'arrière plan

Il s'agit de détecter la région de mouvement en soustrayant pixel par pixel l'image courante à l'image de fond [2]. Cette dernière est crée de plusieurs manières comme :

- la moyenne des N premières trame du flux vidéo.
- L'image de fond qui a peu le changement ou qu'il n'y a pas de mouvement pendant une durée longe.
- la première image dans le flux vidéo.

Quelle que soit la façon utilisée pour créer l'image de fond, la méthode de soustraction de l'arrière plan reste très avantageuse car elle détecte l'objet complètement et elle est utilisées dans plusieurs applications(fig.1.3).

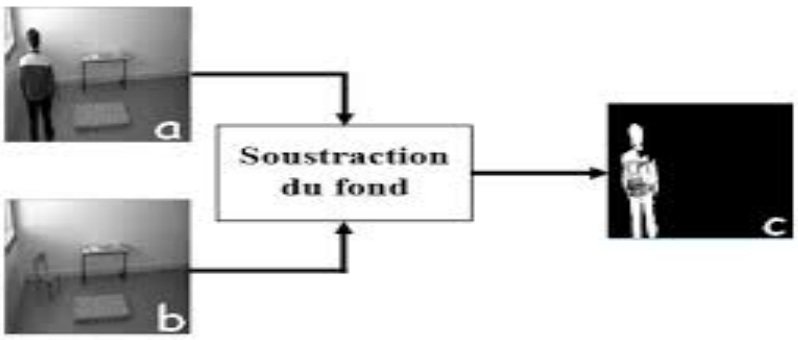


Fig.1.3 Soustraction de fond

1.3.2 Modélisation locale de l'arrière plan

C'est une méthode qui consiste à associer, à tout pixel de l'image, une valeur ou une fonction permettant de modéliser l'apparence de l'arrière-plan en ce pixel. Le modèle d'apparence de l'arrière-plan en un point ne dépend que des observateurs qui ont eu lieu en ce point [6]. Les autres pixels de l'image n'interviennent pas. La modélisation peut être statistique (ensemble de paramètres d'une loi, ou ensemble d'échantillons) ou probabiliste.

1.3.3 Modélisation semi-locale de l'arrière plan

Cette méthode est très semblable à celle de la catégorie précédente à la différence prés que la modélisation de l'arrière-plan en un point dépend des observations qui ont eu lieu dans un certain voisinage de ce point, ou dans la région de l'image à laquelle il appartient [3].

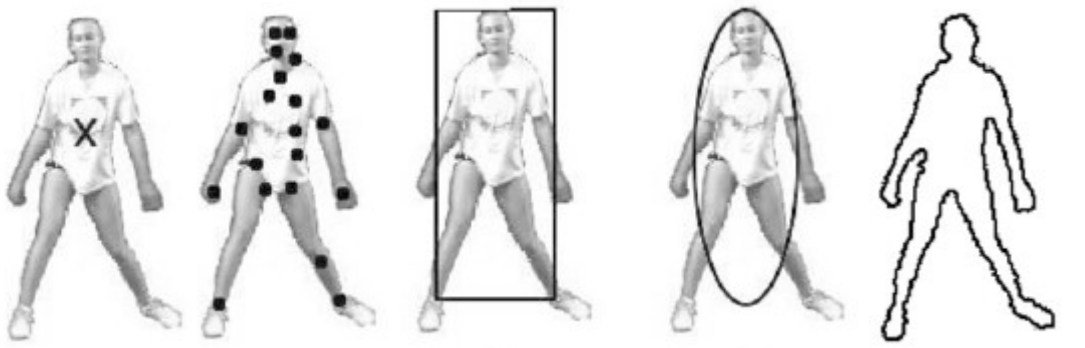
1.3.4 Modélisation globale de l'arrière plan

Ces méthodes utilisent à chaque instant l'ensemble des observations pour construire un modèle de l'ensemble de l'arrière-plan [3].

1.4 Suivi d'objet

Dans la littérature, de nombreuses méthodes ont étés proposées pour résoudre le problème de suivi qui a pour but d'estimer au fil du temps la localisation d'un ou plusieurs objets en mouvement dans une séquence d'image. C'est une étape très délicate surtout quand l'objet est non rigide et le fond n'est pas fixe[7].

Ces méthodes se distinguent par la représentation de la forme et de l'apparence de l'objet. Les représentations par forme sont classées en trois familles [8]: représentation par points, représentation par silhouettes et représentation par fenêtres englobantes (fig.1.4).



(a) (b) (c) (d) (e)

Fig.1.4 Représentation d'objet. (a) centroïde, (b) ensemble de points, (c)fenêtre englobante rectangulaire, (d) fenêtre englobante elliptique, (e)contour.

1.4.1 Représentation par points

On représente un objet par un point sauf si ce point est son centroïde[9], ou par un ensemble de points [10]. Cette représentation convient pour suivre des objets qui occupent de petites régions dans une image. Dans cette famille, les approches peuvent être déterministes ou probabilistes.

L'approche déterministe

Dans l'approche déterministe, le suivi s'effectue en minimisant une distance calculée sur certaines caractéristiques de l'objet. Ces caractéristiques sont l'apparence (Similarité de forme et/ou de contenu photométrique et/ou de mouvement). Les modèles d'objet basés sur l'apparence peuvent être des densités (histogrammes de couleur ou de contour), une carte de contours (contour ouvert ou fermé de l'objet) ou une combinaison de ces modèles [11].

L'approche probabiliste

Les méthodes probabilistes permettent de gérer les fluctuationscausées par des variations de mouvement et d'apparence des objets au fil du temps et les cibles candidates qui sont souvent corrompues par du bruit, en ajoutant une incertitude au modèle de l'objet et aux modèles des cibles candidates.

Ces méthodes ont un faible coût calculatoire, mais elles ont un inconvénient majeur parce qu'elles dépendent exclusivement de la qualité de la détection. En cas de fausse détection, le suivi est détérioré[12].

1.4.2 Représentation par silhouettes

Un contour définit les frontières d'un objet. La région à l'intérieur du contour représente sa silhouette.Les méthodes basées sur le contour et la

silhouette modélisent de manière efficace les objets à la formecomplexe en utilisant l'information encodée à l'intérieur de la région objet. Lebutest de déterminer la région occupée par l'objet au moyen d'un modèle généré en utilisant les images précédentes. Elles sont utilisées lorsque l'on souhaite extraire la silhouette de l'objet et que celle-ci se déforme au cours du temps**.**
Ce type de suivi peut se réaliser avec des méthodes utilisant des modèles d'étatou avec des méthodes minimisant des fonctions d'énergie sur le contour dont le terme d'attache aux données favorise le positionnement du contour estimé le long des zones à fort gradient dans l'image[12].

Les approches explicites

La plupart des méthodes de suivi par contour cherchent à représenter celui-ci par un ensemble de paramètres et à le suivre par une méthode de filtrage appropriée. Ces paramètres jouent le rôle du modèle d'état de l'algorithme de filtrage. Ces méthodes peuventêtre mises en œuvre avec un coût calculatoire assez faible, mais elles suivent plus difficilement les changements topologiques, la division ou la fusion des régions[12].

Les approches implicites

Contrairement aux méthodes précédentes, les méthodes fondées sur une représentation implicite sont robustes aux changements de topologie des objets grâce a la minimisation d'une fonctionnelle d'énergie qui permet de suivre un contour ou une région malgré des changements de topologie [13].Mais la minimisation de ces fonctions est généralement plus coûteuse et la convergence vers un minimum globale n'est pas nécessairement assuré.

1.4.3Modèle d'apparence sur une région

Il s'agit d'une région bornée telle qu'un rectangle ou une ellipse. Les techniques basées sur cette représentation sont fondées sur la conservation de l'apparence (couleur et/ou luminance) de l'objet pendant au moins deux instants consécutifs [14].
Dans un système de suivi fondé sur une représentation par boîte englobante, la boite peut être boîte rectangulaire avec les côtés parallèles aux bords de l'image. Dans ce cas, on peut définir l'état θ de l'objet comme étant le vecteur composé du centre de la boîte (x^c, y^c) et du vecteur $(x^{cs}, y^{cs}, x^{ci}, y^{ci})$ où (x^{cs}, y^{cs}) est le coin supérieur gauche de la boîte et (x^{ci}, y^{ci}) le coin inférieur droit [14]. L'objet à suivre dans une image de référence I_{ref} est représenté par une boîte$b^*_{n,t}$. A l'intérieur de cette boîte englobante est calculé un descripteur caractérisant l'apparence de l'objet.

1.5 Modèles d'apparence pour le suivi

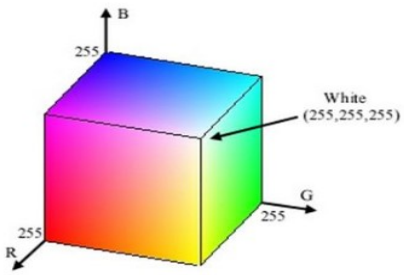
Dans le cas d'une approche par boîte englobante, la qualité de suivi est fortement conditionnée par le choix d'un modèle d'apparence adapté. Il ne s'agit pas de coder toute l'information véhiculée par l'objet mais de se concentrer sur l'information qui permet de définir un critère objectif efficace pour déterminer la position réelle de l'objet suivi dans l'image. Sans dissocier l'extraction des attributs de la structuration de l'information sous forme de descripteurs et de l'utilisation de cette représentation pour la localisation[12], une des clés de la représentation efficace est l'identification des caractéristiques primaires en accord avec le type et le but du suivi visé par le système telles que la couleur, la texture, la forme, lagéométrie et la topologie.

1.5.1 Caractéristiques de couleur

La couleur est l'un des attributs les plus utilisés en représentation d'objets. Elle joue un rôle fondamental dans la distinction des objets constituant une scène/image.

Les auteurs se concentrent autour de quelques grands thèmes tels que l'identification de l'espace couleur le plus discriminant [15], l'étude des problèmes d'invariance aux conditions d'illumination [16] ainsi que la combinaison avec des attributs complémentaires tels que la texture [17] ou l'information spatiale. Le premier point concerne le choix de l'espace colorimétrique. La littérature dense sur cet aspect montre qu'il n'a y a pas d'espace couleur idéal pour la représentation .Si l'espace RGB est utilisé par plusieurs auteurs, l'espace HSV séparant l'information relative à la teinte, la saturation et l'intensité est communément utilisé (fig1.5 et 1.6).

D'une manière générale, quel que soit l'espace couleur considéré, un pixel *p* d'une image est décrit par trois valeurs *C1(p), C2(p) et C3(p).* Ceci permet de représenter le pixel par un point dans l'espace et de percevoir l'image comme un nuage de points.



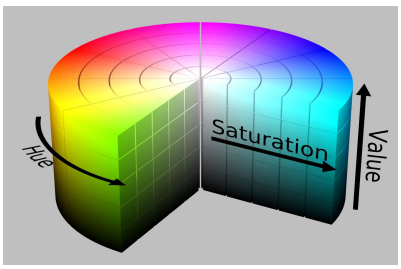


Fig1.5 cube de Maxwel**Fig.1.6** Représentation de l'espace HSV

1.5.2 Caractéristiques de texture

La texture est un facteur fondamental dans la perception de l'environnement et la reconnaissance de ses objets. Contrairement à la couleur, la texture reste difficile à définir de manière préciseetgénérique[18]. Cependant, l'information qu'elle véhicule concerne la manière dont les couleurs sont organisées sur la surface d'un objet en répétition d'éléments, cette explication est parfois considérée insuffisante étant donné qu'elle est indépendante du comportement de l'observateur humain[18]. Le problème de l'utilisation de la texture est le coût calculatoire qu'elle engendre. En effet, en général, les algorithmes proposés ne fonctionnent pas en temps réel. De nombreuses méthodes sont référencées dans la littérature pour la décomposition de l'image et le calcul de caractéristiques texturalesdont on peut citer la décomposition en ondelettes [19].

1.5.3 Caractéristiques de forme

Comme pour la texture, l'information de forme est complémentaire de celle de la couleur. Elle nécessite au préalable une segmentation dont les caractéristiques sont dépendantes del'application visée. Une fois un masque de segmentation obtenu sur la région d'intérêt, la forme de ce masque peut être caractérisée afin d'associer un descripteur à la région qui a donné lieu à ce masque[12]. Les descripteurs fondés sur l'intérieur du masque de segmentation caractérisent l'intégralité de la forme d'une région, par exemple à travers les moments invariants [20]. Ces attributs sont robustes aux transformations géométriques comme la translation, la rotation et le changement d'échelle. Les descripteurs fondés sur le contour du masque font référence aux descripteurs de Fourier [21].

1.6 Descripteurs statistiques de la couleur

Ce sont les modèles qui se fondent sur la représentation statistique de la couleur tel que l'histogramme couleur, l'histogramme pondéré, le spatiogramme et les histogrammes régionaux.
Dans cette partie nous nous intéressant aux deux premiers modèles.

1.6.1 Histogramme couleur

L'histogramme couleur est le descripteur de couleur le plus courant et le plus rencontré dans la littérature. Il permet de représenter statistiquement la distribution des couleurs des pixels, c'est-à-dire la proportion de pixels répartis sur un ensemble de classes de couleurs (fig.1.7).
Cet histogramme a été utilisé dans plusieurs applications telles que la reconnaissance d'objets. C'est un outil important dans les systèmes de suivi [22]. Le succès des approches par histogramme provient de leur faible complexité calculatoire associée à une bonne robustesse vis-à-vis du bruit et de leur invariance aux rotations et aux changements d'échelle.

Dans la plupart des applications, l'histogramme est calculé sur une région d'intérêt définie par une forme géométrique simple (rectangle, ellipse) englobant un objet.
Ainsi, soient$R$=(x_1, x_2,…, x_n)^T cette région et I (x_i), la couleur du pixel i et soit U un ensemble de m classes de l'espace de couleur utilisé (*RVB*, *HSV*, *LUV*, etc.). La $u^{ième}$classe est caractérisée par sa fonction indicatrice $\delta_u(c)$ qui vaut *1* pour une couleur c associée au bin u et 0 sinon [12].
Avec ces définitions, l'histogramme pour un bin u est défini par :

$$P_U = C\sum_{i=1}^{n} \delta_u(I(x_i)) \qquad (1.2)$$

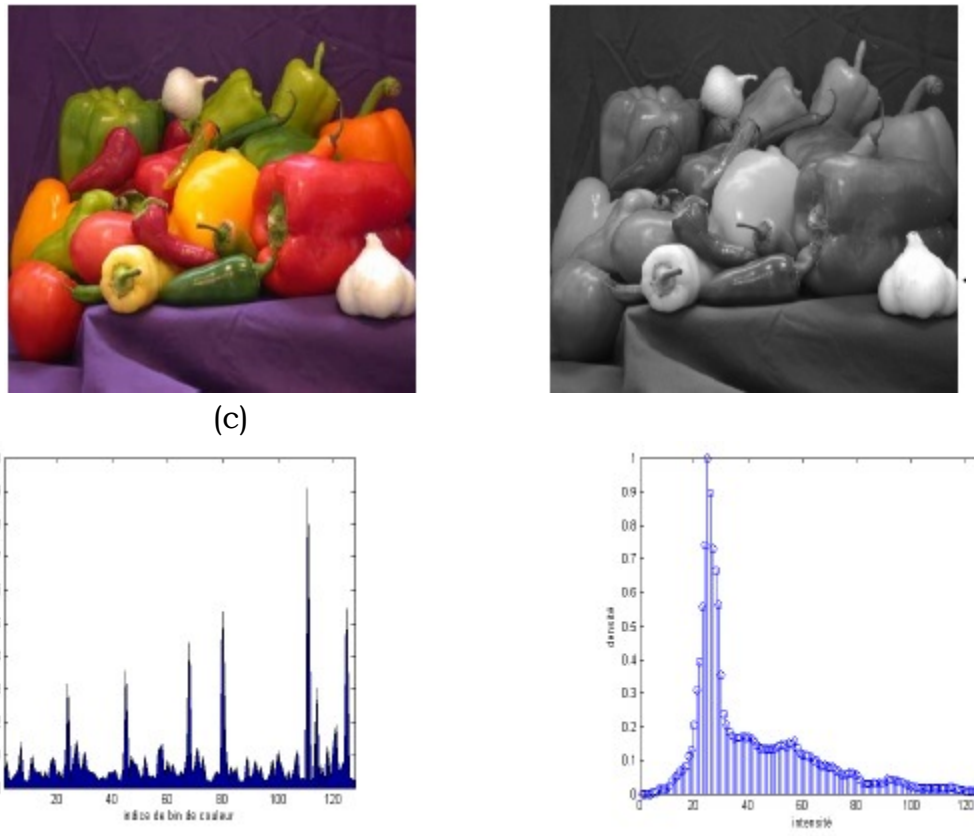
où C est un facteur de normalisation tel que :$\sum_{i=1}^{m} P_U = 1$

1.6.2 Histogramme pondéré

Lors de suivi l'histogramme couleur expliqué ci-dessus n'est pas vraiment valable et cela due à la région d'intérêt qui dans certains cas englobe l'objet d'intérêt et des éléments du fond. Pour remédier à ce problème il est possible de calculer un histogramme pondéré. Plus les pixels sont loin du centre de l'objet, plus le poids pris en compte dans l'histogramme final est faible.
L'histogramme pour un bin u est défini par :

$$q_u = C\sum_{i=1}^{n} \boldsymbol{K}(x_i)\delta_u(I(x_i)) \qquad (1.3)$$

oùKest un noyau spatial associant un poids à chaque position spatiale x_i: les noyaux typiques sont le noyau uniforme sur la région d'intérêt, le noyau d'Epanechnikov[23] ou le noyau gaussien.
Pour les pixels du centre, le poids vaut 1 et en s'éloignant du centre, le poids diminue jusqu'a arriver aux pixels des bordures où ce poids s'annule.



(a) (c)

(b) (d)

Fig.1.7 Histogramme : (a) : image RGB, (b) : histogramme RGB, (c) :image en niveaux de gris, (d) : histogramme en niveau de gris.

1.7 Suivi robuste par la méthode *camshift*

La méthode*camshift* (*Continuously Adaptive Mean Shift*, en anglais) est un algorithme desegmentation d'imagescouleur introduit parGary Bradskien 1998[24].C'est un algorithme qui a été développé pour le suivi de visage dans les interfaces perceptrices de l'utilisateur, sa partie principale représente une technique non-paramétrique.Dans la littérature, cette approche est appelée l'algorithme du meanshift[25].

1.7.1 Principe de la méthode de *MeanShift*

La méthode du *Mean-shift*est une approche non-paramétrique basée sur les caractéristiques comme la couleur, permettant detrouver les maxima d'une distribution de probabilité, où plus exactement le maxima le plus proche de sa position initiale[26].Les modes à trouver sont les maxima locaux de la distribution spatiale $f(x,y)=P(Peau/x, y)$ de la probabilité de teinte chair dans l'image.L'approche du mean-shift est une procédure itérative consistant à déplacer sur la distribution*f* une fenêtre **W**, de taille fixe, de sa position(x_0,y_0) à la position (x_c,y_c) moyenne des points de *f* inclus dans **W** pondérés par leur valeur de probabilité.

. Toutefois, il nécessite de connaître la taille de l'objet recherché, qui est définie à l'initialisation. Les étapes du meanshiftpeuvent se résumer ainsi (fig.1.8).

Après avoir le modèle et la taille (fenêtre **W**) de l'objet à suivre à partir de l'étape d'initialisation, elle vient l'étape de la recherche de la position de ce dernier dans les différentes trames qui suit :

1. Positionner la fenêtre **W** sur la trame qui suit à la même position qu'elle a occupée dans la trame précédente.

2. Déplacer W selon le vecteur de meanshift m(x) jusqu'à avoir m(x)=0.pour chaque itération est calculé :

   - Le moment d'ordre zéro

     $M_{00}=\sum_{x,y} I(x,y)$(1.5)

     $M_{10}=\sum_{x,y} xI(x,y)$(1.6)

     $M_{01}=\sum_{x,y} yI(x,y)$(1.7)

   - Le barycentre

     $x_c=\frac{M1}{M0}$ $y_c=\frac{M01}{M00}$(1.8)

   - Le vecteur du menshift

$m(x,y)=[x_c\_x, y_c\_y]$(1.9)

Fig.1.8Étapes de meanshift

Le nombre maximal d'itérations dans l'algorithme de meanshift est habituellement pris entre 10 et 20 itérations [24].L'ensemble de ces itérations est peut être représenté dans les figures suivantes :

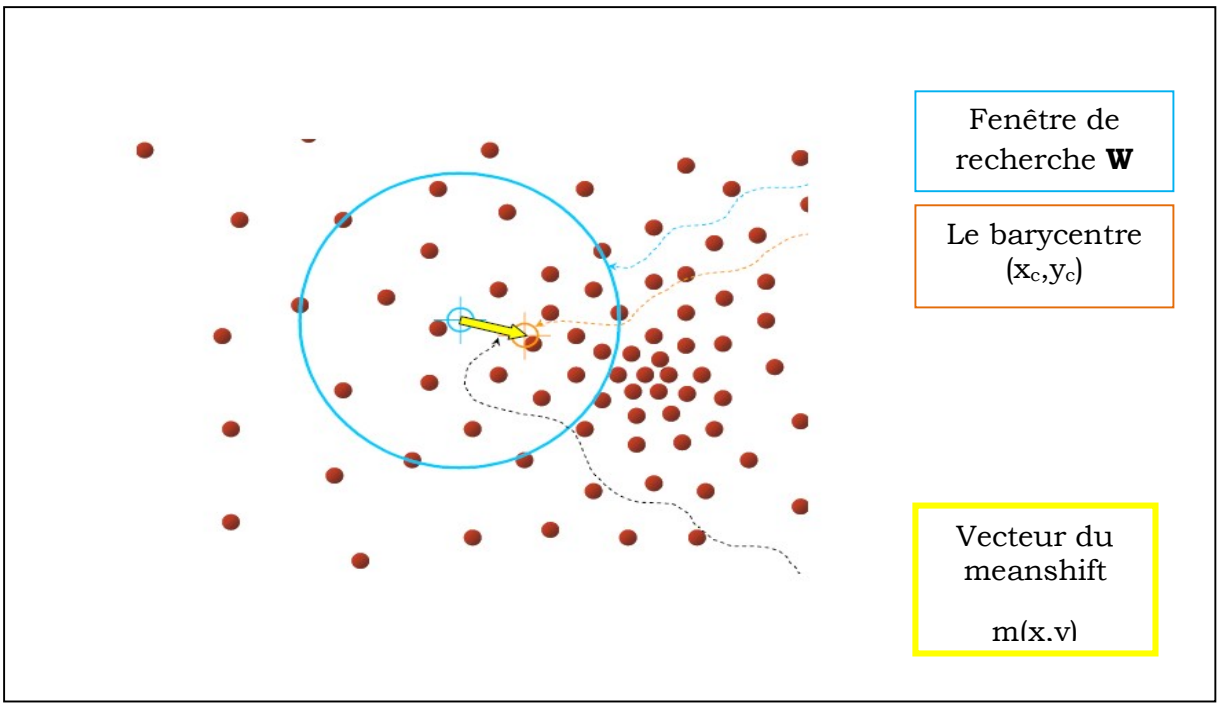


Fig.1.9première itération du meanshift

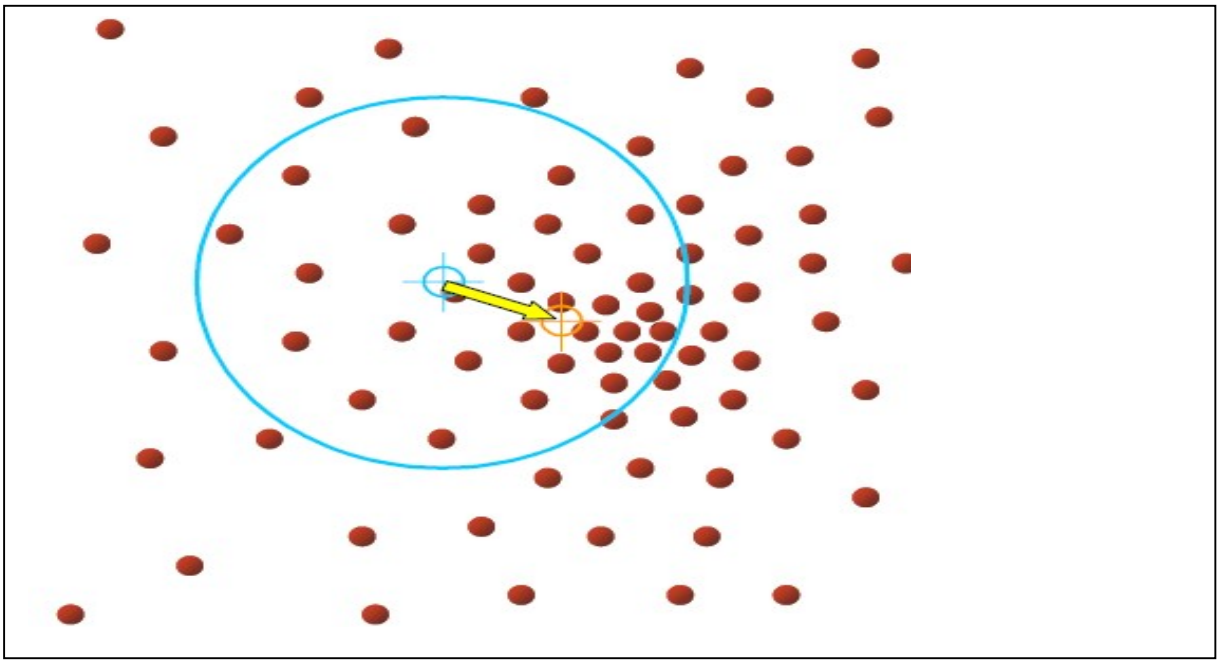


Fig.1.10 Deuxième itération du meanshift

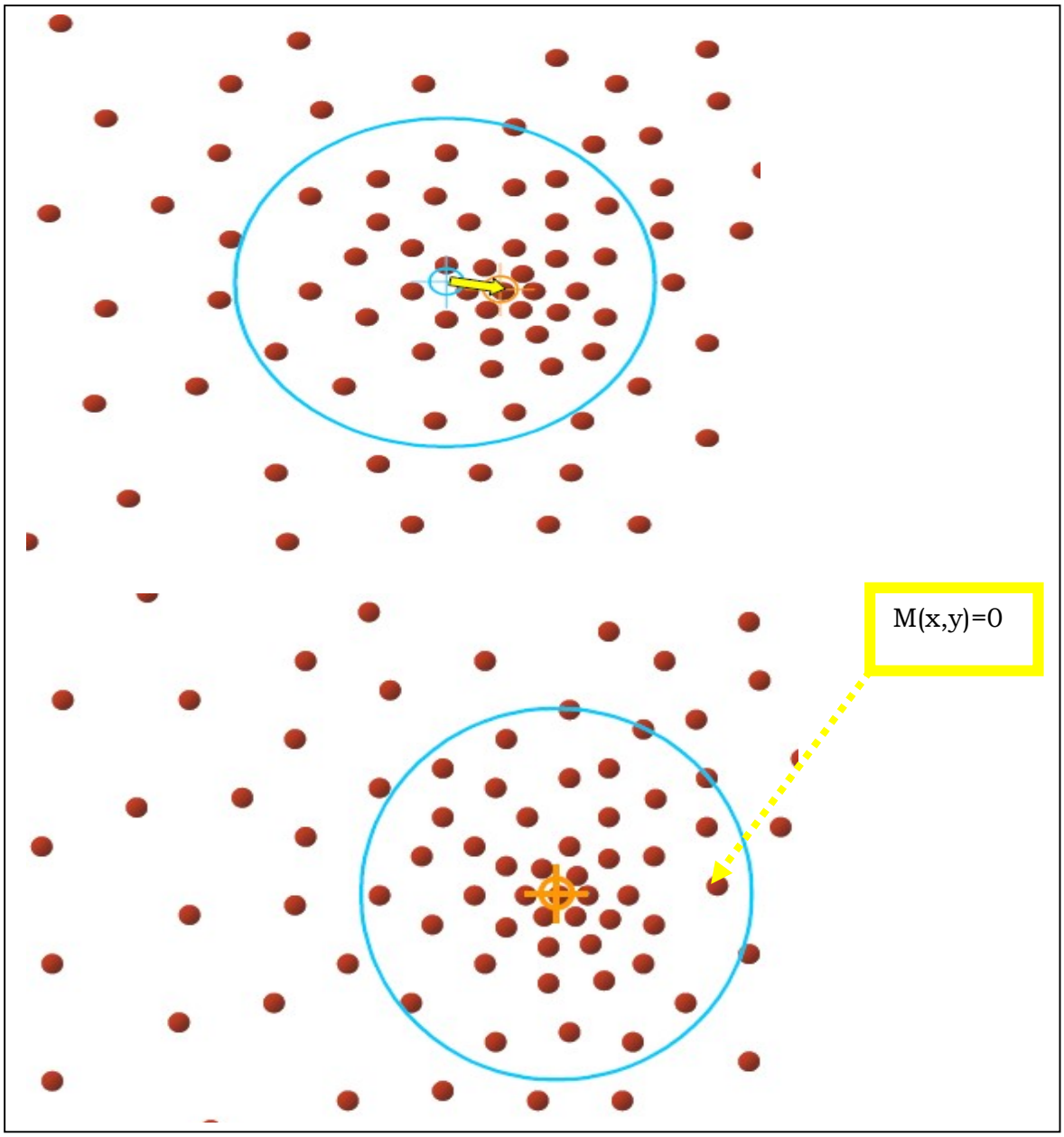


Fig.1.11 Dernière itération du meanshift (convergence)

Cet algorithme seul n'est pas efficace comme algorithme de suivi car la dimension de la fenêtreest fixe.et ne change pas de taille en fonction de la position de l'objet (loin ou proche de la camera)[27]: c'est-à-dire unefenêtre de petite dimension peut être perdu pour une grande apparition de l'objet dans la scène, et une fenêtre de grande dimension peut inclure d'autre objet ou autre personne. Pour remédier à ce problème nous avons opté à l'algorithme de camshift.

1.7.2 Algorithme de *Camshift*

Bradsky encapsule l'algorithme de meanshift dans celui de camshift en faisant varier la taille de la fenêtre englobante selon la taille de la cible dans les différentes trames successives [28]. Cette méthode peut être utilisée pour la segmentation d'un objet dans une image fixe comme elle peut être utilisée pour le suivi d'un objet dans une séquence d'images. Dans notre approche on s'intéresse à la deuxième utilisation c.-à-d. : le suivi d'objet en temps réel. Lesdifférentes étapes du camshift sont ainsi les
Suivantes [29] :

Pour chaque trame de la séquence d'image et après que la position de la cible (x_c,y_c) a été déterminée par la méthode du meanshift c.-à-d. :la dernière itération de meanshift (convergence),la dimension de la fenêtre de recherche sera adaptée par la méthode du camshift pour la prochaine trame de la séquence. Cette adaptation est faite avec le moment d'ordre zéro tel que :

La largeur est :W=2.$\sqrt{\frac{M00}{256}}$ et la hauteur est : h=1,2.W

La position de la cible, sa position, sa hauteur et aussi son orientation peuvent être aussi dérivées à partir des moments statiques :

La largeur : $W=2.\left(\frac{(a+c)-\sqrt{b^2+(a-c)^2}}{2}\right)^{\frac{1}{2}}$ (1.10)

La hauteur : $h=2.\left(\frac{(a+c)+\sqrt{b^2+(a-c)^2}}{2}\right)^{\frac{1}{2}}$ (1.11)

Tel que les coefficients a,b et c sont calculés en utilisant les moments d'ordre 0,1 et 2.

$$\begin{cases} a = \frac{M20}{M00} - x_c^2 \\ b = 2.(\frac{M1}{M00} - x_c.y_c) \\ c = \frac{M0}{M00} - y_c^2 \end{cases} \quad (1.12)$$

Pour ce qui concerne l'orientation de la cible Bradsky a utilisé les moments centraux de premier et deuxième ordre :

$\varphi = \frac{1}{2}\tan^{-1}\left(\frac{2\mu_{11}}{\mu_{20}-\mu_{02}}\right)$(1.13)

tel que le moment central d'ordre i pour x et j pour y est définie par :

$\mu_{i,j} = \sum_{x,y}(x - x_c)(y-y_c).I(x,y)$(1.14)

Fig.1.12 Étapes de camshift

1.8. Conclusion

Dans ce chapitre, nous avons présenté les méthodes de détection de mouvement ainsi que les différentes représentations de la forme et de l'apparence d'objet utilisés dans les algorithmes de suivi en temps réel. Ainsi, la méthode de *Camshift* a été présentée de façon un plus détaillée.
Ces représentations nous ont permis de constater que le choix de l'algorithme de suivi dépend de plusieurs paramètres tels que la taille, la forme et la nature de l'objet à suivre ainsi que des conditions environnementales. Aussi, peut-on constater que tout algorithme de suivi nécessite, en initialisation, une bonne méthode de détection pour ne pas avoir des confusions entre l'objet à suivre et le fond de la scène. En respectant tout ces paramètres, le résultat du suivi sera robuste.

Chapitre 2

Généralité sur l'asservissement visuel

1.1. Introduction

L'asservissement visuel est le résultat de la combinaison de différentes spécialités telles le traitement d'image, la cinématique, la dynamique, l'automatique et l'informatique. C'est un asservissement dont la boucle de réaction est constituée d'un système de vision artificielle dont le rôle est de ramener un ensemble de mesures pour permettre les mouvements appropriés ou suivre une trajectoire selon la loi de commande élaborée. En ce qui concerne le système de vision artificielle, l'image est acquise avec ses propres caractéristiques de forme, de couleur et de texture. Un traitement adéquat de l'image permet, par exemple, de déterminer la position de l'objet dans un environnement par rapport à cette caméra.
Dans ce chapitre nous allons présenter un bref historique de l'asservissement visuel ainsi le principe de fonctionnement et la loi de commande élaborée pour ce type d'asservissement.

2.2 Historiques de l'asservissement visuel

Le terme d'asservissement visuel (*visual servoing* en anglais), fut introduit pour la première fois par Hill et Park en 1979 [30] pour distinguer leur approche d'expérimentations antérieures dites «blocks world» dans les quelles les systèmes alternaient les étapes «prises d'images »puis «bouge».
Les premier travaux apportés sur l'asservissement visuel remontent a 1980.les progrès dans l'art de commander un robot par la vision ont été laborieux, cependant, depuis quelques année, le domaine a connu un regain d'intérêt, comme le prouve l'abondant littérature qui y est consacré.

En 1986, Dzialo et Schalkoff discutent de l'effet de la perspective sur la commande «pan-tilt» d'une caméra montée sur une tête robotique pour les applications de poursuites de cibles [31]. En 1988 Kabuka, présente un système qui utilise les projections d'objet dans une image binaire [32]. Il applique une transformation de Fourier sur les axes verticaux et horizontaux pour centrer une cible dans le plan caméra et calculer l'orientation, La caméra est asservie sur une plate-forme à deux axes par ordinateur et met 30 s, pour être centrée sur la cible Son approche est décrite de manière générale, mais seules des expérimentations dans le plan ont été menées. En 1990, Venkatesan et Archibald ont proposé un système pour asservir un robot à cinq degrés de liberté à partir d'informations d'un système à deux lasers montée sur une tête robotique [33].

2.3 Principe de l'asservissement visuel

Dans l'asservissement visuel, l'erreur est calculée entre les valeurs courante et désirée et les informations visuelles. Le but est de déterminer la loi de commande qui va permettre d'amener l'image d'une position initiale vers la position cible (fig.2.1)

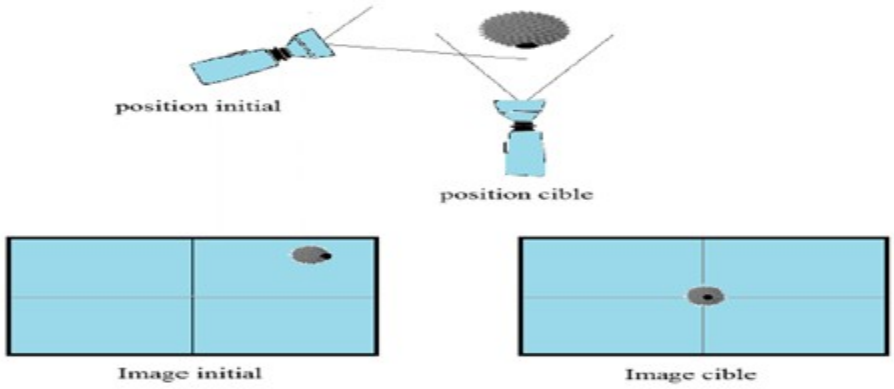


Fig.2.1 Illustration du principe de l'asservissement visuel

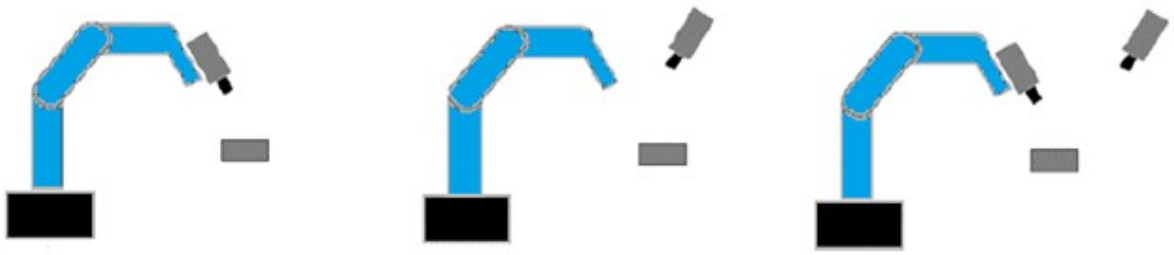
2.4 Classification des asservissements visuels

Les asservissements visuels peuvent être vus en fonction de la position de(s) caméra(s), du type de commande et du type de mesure ou encor en fonction de la rapidité

2.4.1 En fonction de la position de(s) caméra(s),

Les capteurs visuels plus essentiellement les caméras peuvent être classées selon trois types de configurations rebot-caméra (fig.2.2).En effet, la caméra peut être montée sur un support fixe ou mobile et regardant à la fois

le robot et l'objet d'intérêt ou embarquée, c'est à dire montée sur l'organe terminal du robot et regardant l'objet d'intérêt, ou encore mixte an utilisant une caméra portée sur l'organe terminal et une autre portée par un autre robot. Le choix de la configuration dépend essentiellement de la tâche à exécuter. Par exemple, pour un suivi de cible, il est judicieux de placer la caméra sur le robot de manière à ce que l'image vue soit sensiblement toujours la même quels que soient les mouvements du robot et de la cible. Il peut être intéressant de déporter la caméra pour observer à la fois l'outil et l'objet d'intérêt. Néanmoins, cette solution, si la caméra est fixe, implique que les objets de la scène restent dans l'image, ce qui signifie que le robot doit évoluer dans une zone de travail relativement restreinte. Cette configuration se prête particulièrement bien à la robotique chirurgicale car la zone de travail y est souvent très petite.
Notons que la plupart des applications utilisent la configuration où la caméra est embarquée, Pour facilitée la conception de la loi de commande.



(a) (b) (c)

Fig.2.2 Configuration de la caméra : (a) caméra embarquée,(b) caméra déportée,(c) deux caméra une embarquée et l'autre déportée(mixte)

2.4.2 En fonction du type de commande

Les asservissements visuels sont sous deux types : asservissement visuel direct et asservissement visuel indirect.

Asservissement visuel direct

Les axes du robot sont asservis en vitesse ou en couple par les variateurs. Les commandes du correcteur de l'asservissement visuel sont directement envoyées aux variateurs. L'asservissement de position du robot est réalisé grâce au retour de l'information visuelle. Ce type d'architecture est utilisé pour les asservissements visuels rapides (fig2.3).

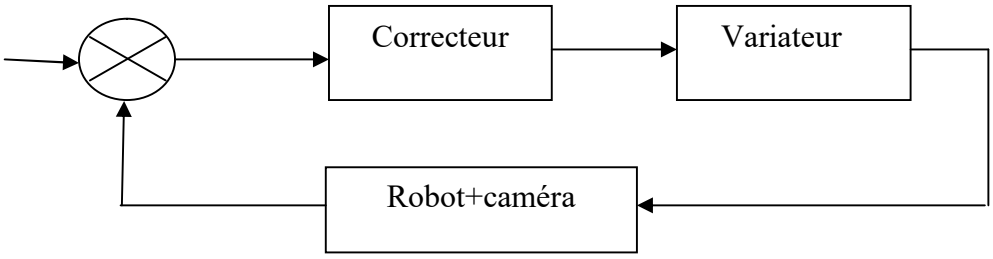


Fig.2.3 Schéma d'un asservissement visuel direct

Asservissement visuel indirect

Les positions articulaires du robot sont asservies par le contrôleur fourni par le fabriquant. Le correcteur de la boucle de vision fourni des consignes de vitesse qui sont transformées en consignes de positions articulaires par un générateur de trajectoire. Ce type d'architecture est utilisé lorsque la période d'échantillonnage de l'asservissement visuel est grande (fig.2.4).

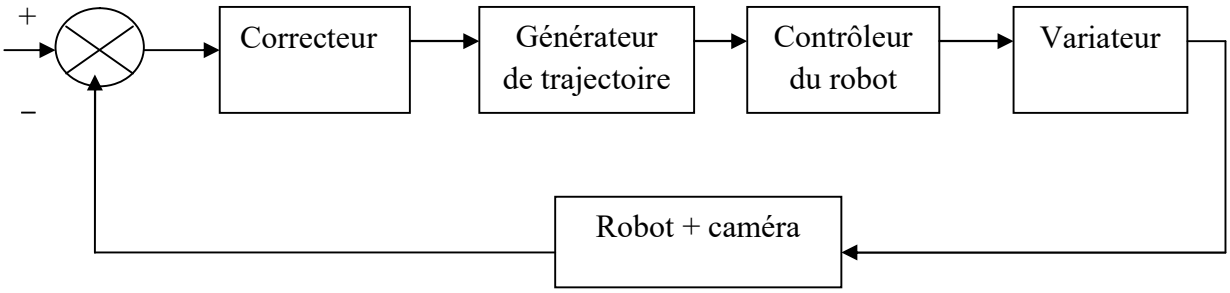


Fig.2.4 Schéma d'un asservissement visuel indirect

2.4.3 En fonction du type de mesure

L'asservissement visuel peut basée (espace 3D, image (2D), hybride ($2D_{1/2}$), dynamique (d2D/dt)), lent et visuel rapide.

Asservissement visuel basé espace (3D)

L'approche 3D est historiquement, la première. Elle suppose que le modèle de la scène est connu par le système de vision. La figure (2.5) donne le schéma-bloc d'un asservissement visuel 3D. La consigne p* est comparée à la mesure estimée à partir de l'image.

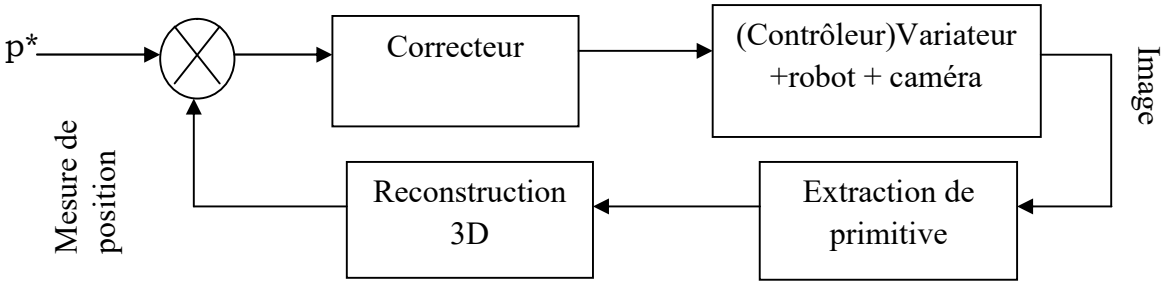


Fig.2.5 Schéma d'un asservissement visuel 3D

L'avantage de cet asservissement est la maîtrise des trajectoires du robot dans l'espace opérationnel. Son principal inconvénient est la nécessité d'avoir à recourir à un modèle de la scène.

Asservissement visuel basé image (2D)

En 2D, la mesure et la consigne de l'asservissement visuel sont directement exprimées en termes de coordonnées de primitives.

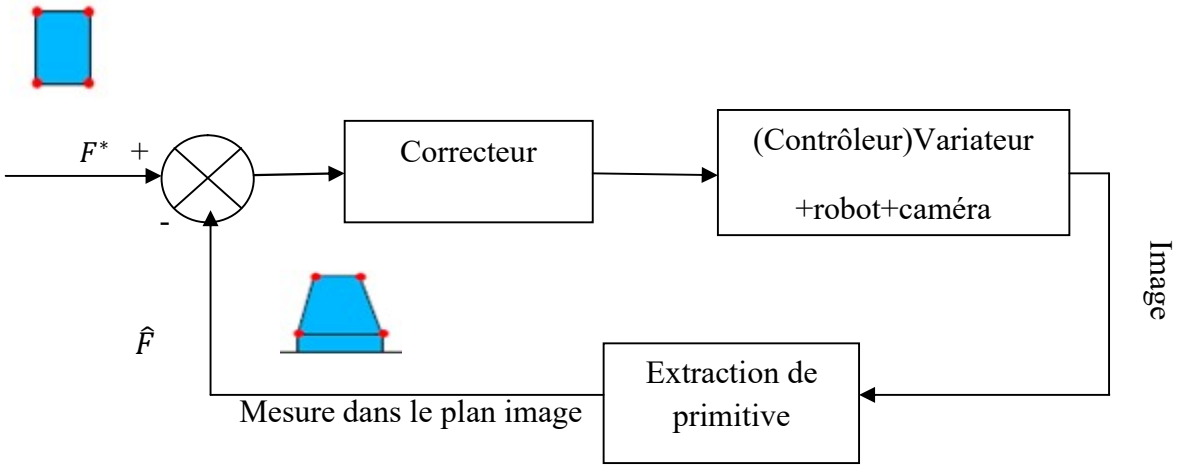


Fig.2.6 Schéma d'un asservissement visuel 2D

La figure (2.6) montre le schéma-bloc d'un asservissement visuel 2D. Si on compare les figures ci-dessus, on s'aperçoit que le bloc «reconstruction 3D» n'est pas présent dans l'approche 2D. En effet, ici, la consigne F* et la mesure sont directement exprimées dans l'image. Le but d'un asservissement 2D est de faire converger l'image courante vers une image de référence. Cette commande permet donc de s'affranchir de modèle de la scène est donc adaptée aux cas des environnements peu connus (en général l'environnement industriel ne rentre pas dans cette catégorie). C'est là son principal avantage. Son inconvénient majeur vient des trajectoires non maîtrisées dans l'espace opérationnel.
Le robot est fondamentalement un dispositif de déplacement opérationnel. Afin de le commander à partir de données purement visuelles, il est nécessaire d'avoir recours à une conversion de l'espace 2D du capteur vers l'espace opérationnel du robot.

Asservissement visuel hybride (2D1/2)

Dans la nouvelle approche d'asservissement caractérisé de $2D_{1/2}$ développée par E. Malis dans le cadre de sa thèse [34], on utilise, à la fois, des informations directement exprimées dans l'image 2D pour l'asservissement de la position de l'organe terminal et des informations

exprimées dans le repère de la caméra 3D pour asservir l'orientation de l'organe terminal. Les deux boucles d'asservissement sont découplées.est représenté dans la figure (2.7)

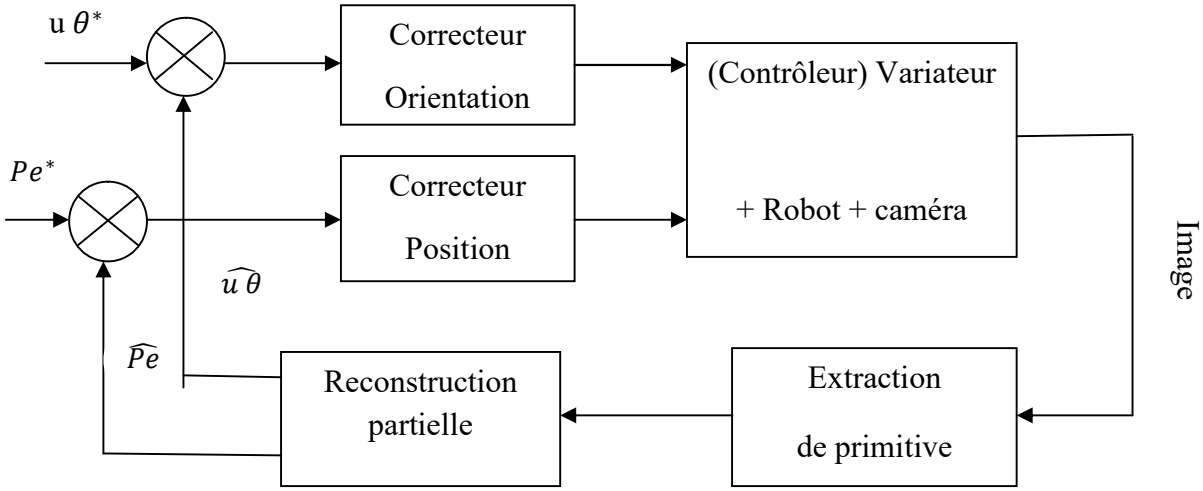


Fig.2.7 Schéma d'un asservissement visuel hybride

Asservissement visuel dynamique (d2D/ dt)

La grandeur asservie est un champ de vitesse dans le plan image. Le principe de la commande consiste à contrôler les mouvements de la caméra de telle sorte que le mouvement 2D mesuré atteigne un champ de vitesse désiré, d'où l'appellation d'asservissement visuel d2D/dt. Comme pour l'asservissement visuel 2D, il s'agit de contraindre les mouvements de la caméra de manière à ce que les informations visuelles acquises par celle-ci atteignent des valeurs souhaitées. sa différence réside dans le fait que dans le cas 2D, il est nécessaire de passer par une étape d'extraction des primitives, alors que dans le cas d2D/dt, on souhaite seulement atteindre un champ de vitesse désiré.

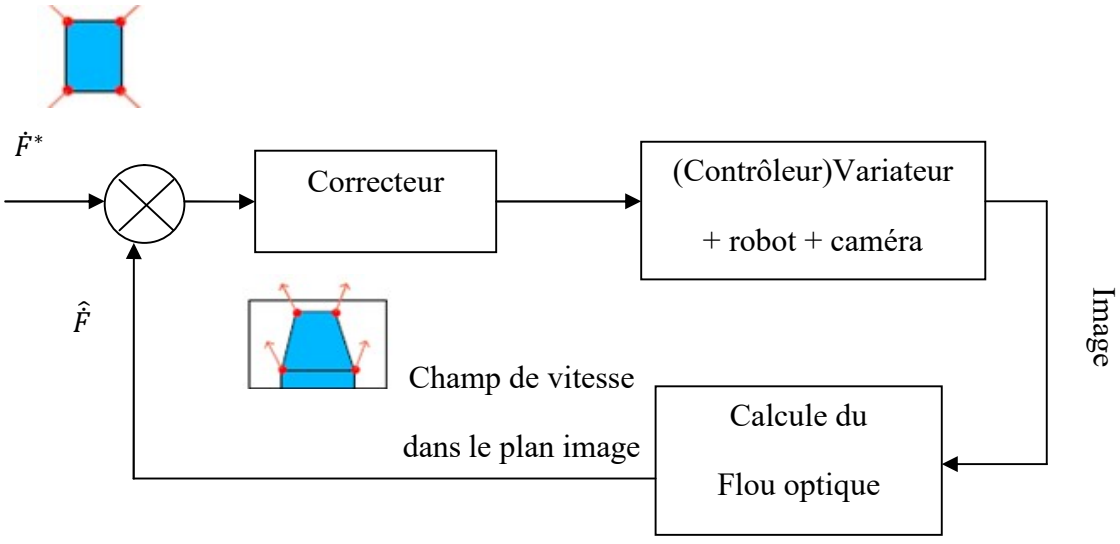


Fig.2.8 Schéma d'un asservissement visuel dynamique

2.4.4 En fonction de la rapidité

L'asservissement visuel Asservissement visuel lent et Asservissement visuel rapide

Asservissement visuel lent

Cet asservissement est aussi appelé asservissement visuel cinématique. On approxime la fonction de transfert du robot entre consigne de vitesse et la mesure de vitesse par un simple gain. Cette approximation est valable pour des déplacements lents du robot et lorsque la commande est du type indirecte (asservissements imbriqués de position articulaire). Le correcteur est un simple gain ce qui permet de démontrer une convergence exponentielle de la mesure vers la consigne

Asservissement visuel rapide

Cet asservissement est encore appelé asservissement visuel dynamique. Le modèle dynamique du manipulateur est pris en compte dans la synthèse du correcteur. Étant donné le comportement dynamique moyen de la plupart des manipulateurs, on considère que 50 Hz fixe la limite à partir de laquelle il est nécessaire de tenir compte du modèle dynamique pour améliorer les performances. La commande est du type direct. Il peut s'agir d'une vitesse ou d'un couple.

2.5 Les fondamentaux de l'asservissement visuel

Pour un robot qui suit une cible, la faculté de perception visuelle permet à un robot de s'informer sur l'état de son environnement et de s'adapter aux possibles variations de ce dernier. Le mouvement d'un robot équipé d'une caméra peut être commandé à partir de sa perception visuelle en utilisant la technique d'asservissement visuel. L'asservissement visuel consiste à utiliser la perception pour l'action dans une boucle fermée, comme le montre la figure (2.9).

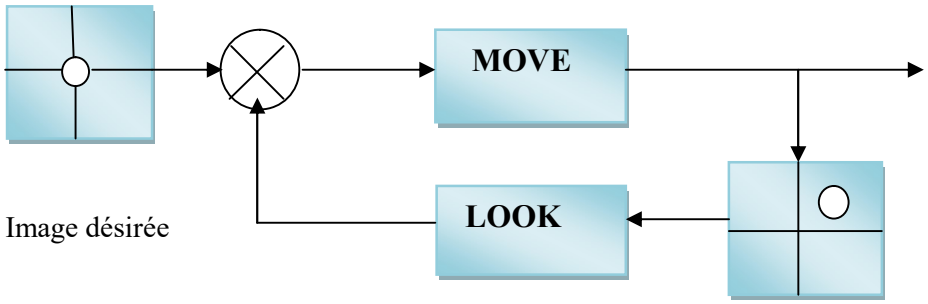


Fig.2.9 Perception pour agir

La perception visuelle d'un objet peut être définie comme l'abstraction d'un ensemble de mesures visuelles sur l'objet cible. Par exemple, si nous voulons essayer de déterminer les coordonnées d'un point de l'image, la perception se traduit par la position de l'objet par rapport au robot. L'action, quant à elle, peut être vue comme un mouvement de robot pour effectuer une tâche. La perception pertinente de l'environnement du robot est un facteur déterministe. Atteindre cette perception pertinente, revient à une modélisation d'informations visuelles adéquates, c'est-à-dire, permettant au robot de s'approcher d'un comportement idéal pendant une tâche soit effectuer. La question qui se pose est comment exploiter judicieusement la grande quantité d'informations qu'ils captent afin d'améliorer le comportement d'un robot commandé par asservissement visuel.

2.6 Schéma fonctionnel de l'asservissement visuel

La boucle d'asservissement visuel peut être représentée par le schéma bloc de la figure suivante :

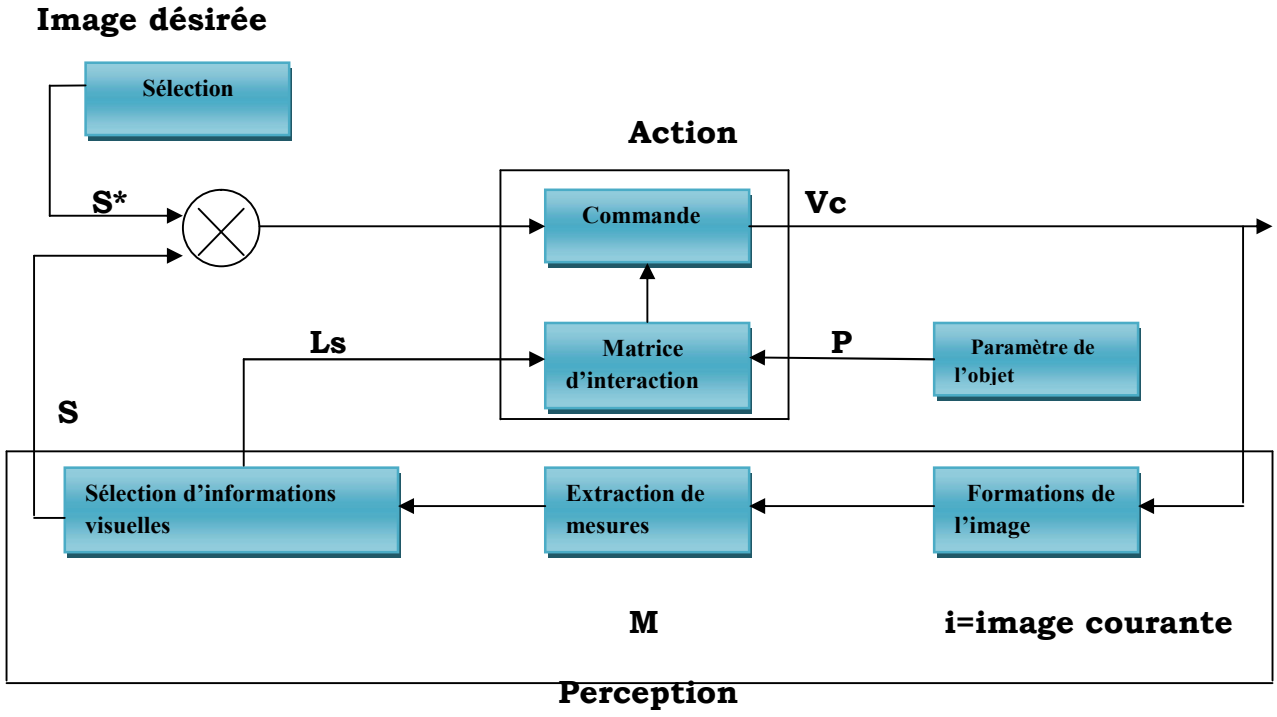


Fig.2.10 Schéma bloc de la boucle de l'asservissement visuel

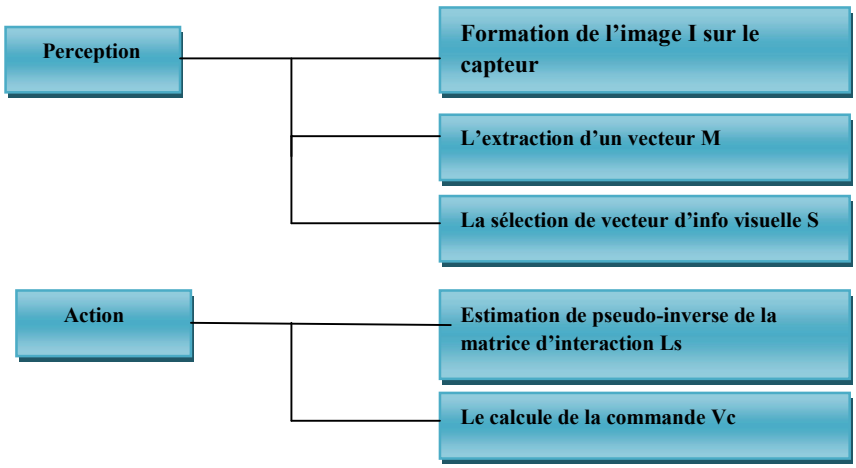


Fig.2.11 Diagramme explicatif de la partie Perception et Action

La matrice «*ls*» est fondamentale en asservissement visuel car elle représente la relation vision-commande. Dans la suite, nous décrivons comment la matrice «*ls*» intervient dans le calcul de la commande «*vc*», ce qui nous permet de définir les propriétés souhaitables de la matrice «*ls*» et par conséquent de définir des critères pour choisir le vecteur « *s* » d'informations visuelles idéal.

2.7 Conclusion

Dans ce chapitre, les notions de base sur l'asservissement visuel ont été proposées et nous a permis de dresser un état historique du domaine de en précisant les diverses relations mathématiques sous-jacentes aux asservissements en position et dans l'image. Il été également introduit, par souci d'exhaustivité, que cette étude est récente et inspire nombreux chercheurs. Les applications relatives à cet axe sont nombreuses et utilisent parfois des concepts très poussés en matière de vision 3D, de navigation, d'évitement d'obstacles, poursuite et bien d'autres encore. Etant donné l'aspect théorique et réalisation pratique de notre travail, nous nous sommes fixés le but de développer un système qui gère le fonctionnement d'une poursuite d'un objet en mouvement pour que ce dernier reste dans le champ de vision de la caméra. Cependant, toutes ces méthodes se basent sur des caméras perspectives. Or ces caméra ont un champ de vision restreint, ce qui limite fortement la quantité d'information visuelle. C'est en partant de ce constat que des travaux ont été effectués sur des caméras Omnidirectionnelles.

Chapitre 3

Etude et réalisation d'un système de suivid'un objet en mouvement

3.1 Introduction

Dans ce chapitre nous allons présenter le projet qui a été réalisé au dans le cadre de notre mémoire de fin d'études qui est destiné au suivi d'un objet en temps réel. Le système est composé d'un ensemble mécanique articulé qui garantit l'orientation d'une caméra placée dans l'espace 3D en utilisant l'asservissement visuel qui a pour but de positionner la camera de façon à avoir l'objet détecté au centre du champ de vision. C'est le principe de l'asservissement visuel qui a été expliqué dans le deuxième chapitre. Ce système est piloté par un microordinateur via une carte d'acquisition et de restitution de signaux de type ARDUINO.

Dans la première section de ce chapitre, nous allons présenter la partie matériel du projet,et dans la deuxièmesection, c'est la partie software qui sera présentée. Des tests et résultats de ce système ont été effectués.

3.2 Partie matériel du projet

Les éléments qui composent cette partie sont deux servomoteurs de type MG996R. Leur rôle principal est de faire déplacer la caméra dans l'espace 3D en suivant l'objet en mouvement.Le modèle MG996R est un servonumérique. C'est une version améliorée du servomoteur MG995. Ce sont des servos standard de haut moment de rotation peuvent tourner approximativement à 180 dégrée (90° dans chaque direction).Ils s'adaptent avec tout type de code de servomoteur en utilisant le logiciel Arduino.

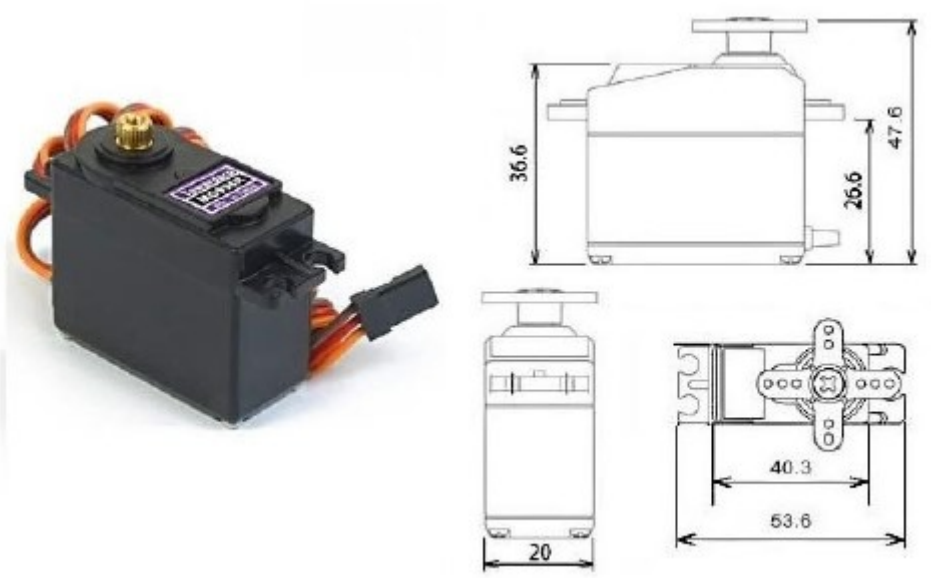


Fig.3.1 Servomoteur MG996R

3.2.1 Fonctionnement des servomoteurs

Un servomoteur (souvent abrégé en « servo », provenant du latin *servus*qui signifie « esclave ») est un moteur capable de maintenir une opposition à un effort statique et dont la position est vérifiée en continu et corrigée en fonction de la mesure [36]. C'est donc un système asservi.Ils sont très fréquemment employés dans les applications de modélisme pour piloter le safran d'un bateau, le gouvernail d'un avion ou bien même les roues d'une voiture téléguidée etc.

3.2.3 Composition d'un servomoteur

Le servomoteur est un ensemble de mécanique et d'électronique [35]composé de plusieurs éléments visibles et d'autres invisibles.

Les éléments visibles:

- Les fils, qui sont au nombre de trois.
- L'axe de rotation sur lequel est monté un accessoire en plastique ou en métal.
- Le boitier qui le protège.

Les éléments invisibles

- Un moteur à courant continu (CC) souvent de petite taille.
- une carte électronique d'asservissement (pour le contrôle de la position de l'axe et le pilotage du moteur à courant continu.
- un réducteur de vitesse qui fait deux choses. D'une part, il réduit la vitesse de rotation en sortie de l'axe du servomoteur, d'autre part il permet d'augmenter le couple en sortie du servomoteur pour les applications de positionnement mécanique.
- un potentiomètre sous forme d'un capteur de la position de l'axe du moteur car sa résistance est proportionnelle à la position angulaire de l'axe de sortie.
- un axe dépassant, hors du boîtier avec différents bras ou roues de fixation.

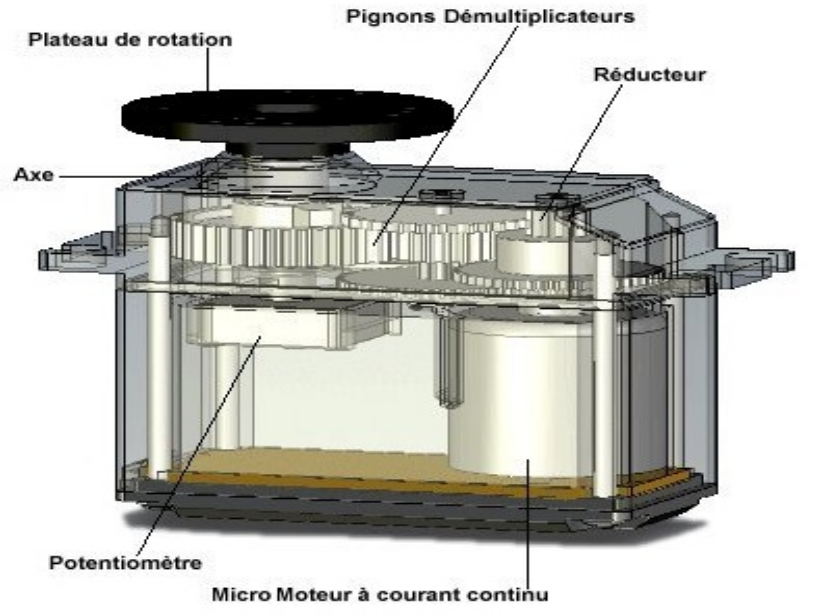


Fig.3.2 les éléments qui composent un servomoteur

3.2.4 L'électronique d'asservissement du servomoteur

Les servomoteurs ont l'avantage d'être asservis en position angulaire autrement l'axe de sortie du servomoteur respecte la consigne d'orientation envoyé sur son entrée en utilisant une électronique de commande. On peut la nommer *électronique d'asservissement*, car c'est elle qui va gérer la position du bras du servomoteur [35]. Cette électronique est constituée d'une zone de comparaison qui compare la position du bras du servomoteur au signal de commande. Le deuxième élément qui constitue cette électronique, est le capteur de position du bras qui est le potentiomètre couplé à l'axe du moteur qui mesure la tension au point milieu et permet d'obtenir une tension image de l'angle d'orientation du bras. Cette position est ensuite comparée à la consigne (le signal de commande) qui est ensuite transmise au servomoteur[36]. Après une rapide comparaison entre la consigne et la valeur réelle de position du bras, l'électronique de commande du servomoteur va appliquer une correction si le bras n'est pas orienté à l'angle imposé par la consigne pour annuler l'erreur.

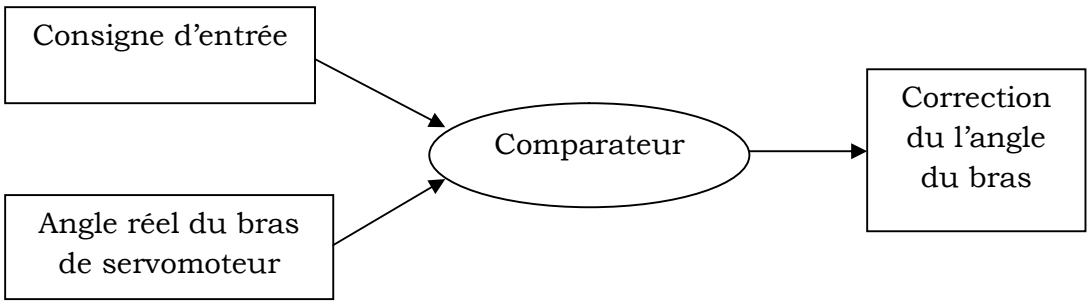


Fig.3.3 Synoptique de fonctionnement de l'asservissement du servomoteur

3.2.5 Carte arduino UNO

C'est un circuit imprimé comportant tous les composants électroniques nécessaires pour fairefonctionner un microcontrôleur (Atmega 328) associé à une interface USB lui permettant decommuniquer avec un ordinateur [36]. Cettecarte possède :

- 14 entrées/sorties numériques dont 6 peuvent être utilisées comme étant des sorties PWM (Pulse Width Modulation,
- 6 entrées analogiques avec un convertisseur Analogique/Numérique de 10 bits de résolution, un résonateur céramique (quartz) de 16 MHz,
- un connecteur ICSP (In Circuit Serial Programmation) qui permet d'injecter le bootloader à l'intérieur du microcontrôleur, un connecteur

jack pour une alimentation extérieur, un bouton de reset pour mettre le processus à zéro.

L'intérêt principal des cartes ARDUINO est leur facilité de mise en œuvre. ARDUINO fournit un environnement de développement s'appuyant sur des outils open-source. Le chargement du programme dans la mémoire du microcontrôleur se fait de façon très simple par port USB. En outre, des bibliothèques de fonctions sont également fournies pour l'exploitation d'entrées-sorties, gestion des convertisseurs ADC, génération de signaux PWM, exploitation de bus I2C. La carte ARDUINO peut être alimentée soit viala connexion USB qui fournit 5V 500mA ou à l'aide d'une alimentation externe. Dans notre application, la carte ARDUINO est alimentée via le port USB [7].

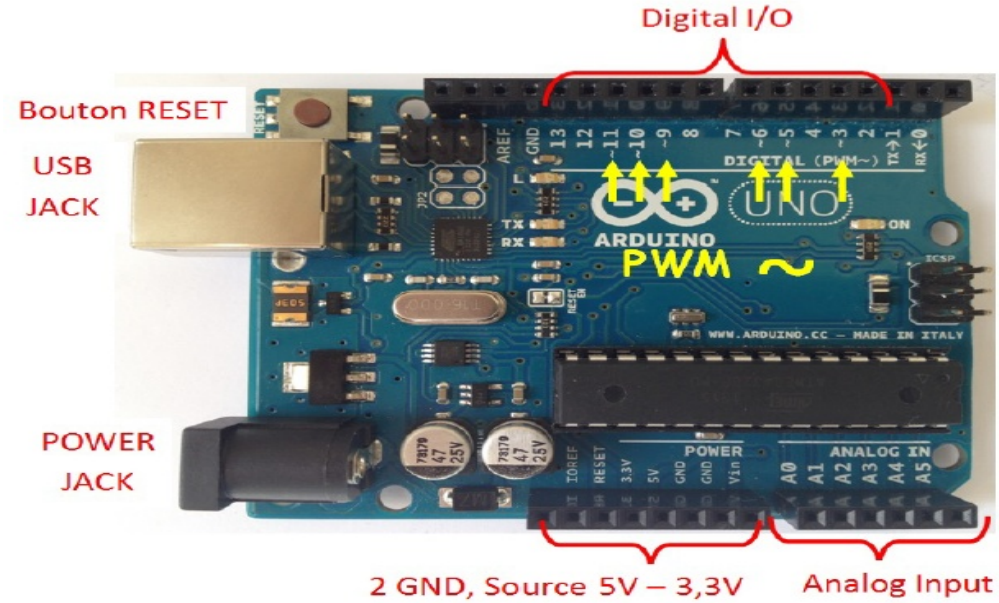


Fig.3.4 Description de la carte arduino UNO

3.2.6 Branchement des servomoteurs a arduino

Le branchement des servomoteurs à la carte Arduino ne va pas être question de pont en H ou d'autre circuit comme les autres moteurs. C'est un branchement qui est très simple du fait que le servomoteur a besoin de trois fils de connexion pour fonctionner. Deux fils servent à son alimentation et le troisième étant celui qui reçoit le signal de commande.

- *rouge* : pour l'alimentation positive (4.5V à 6V en général).
- *noir ou marron* : pour la masse (0V).
- *orange, jaune, blanc, …* : entrée du signal de commande.

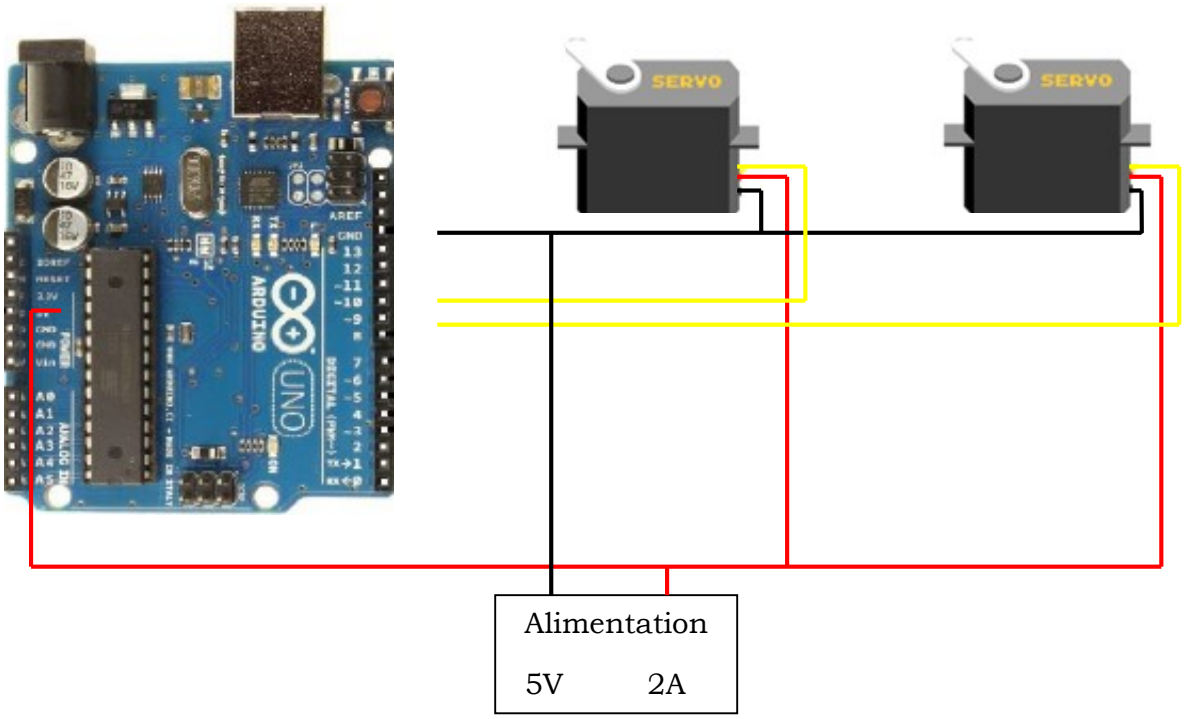


Fig.3.5 Branchement des servomoteurs à la carte Arduino

Pour une utilisation manuelle du système, nous avons utilisé un joystick. Le branchement de ce dernier est montré à la figure suivante :

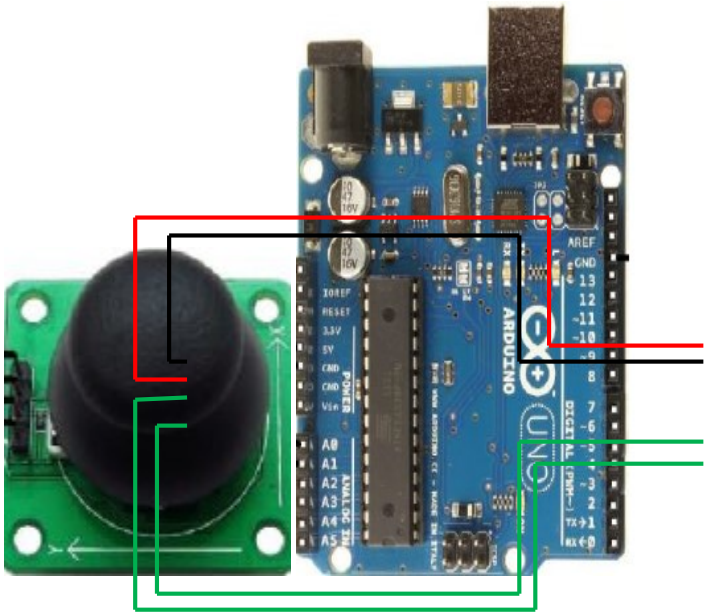


Fig.3.6 Branchement Arduino avec le joystick

3.3 Partie software du projet

Dans cette section, nous allons présenter des aperçus des différents algorithmes utilisés dans notre projet, en commençant par la partie image (détection et suivi d'un objet en mouvement en temps réel et en terminant par la partie commande (positionnement de la camera).

3.3.1 Suivi d'objet en mouvement

Dans chaque système robotique destiné au suivi d'objets mouvement en temps réel, le choix de l'algorithme de suivi dépond de plusieurs paramètresenvironnementaux et des contraintes fondamentales. Dans notre projet, l'algorithme choisiest celui de Camshift qui a été exposé et expliqué dans le premier chapitre.L'algorithme original du Camshift utilise l'histogramme unidimensionel comme modèle de l'objet à suivre, c'est l'histogramme du canal (H) de l'espace couleur (HSV), mais dans notre cas nous avons utilisé l'histogramme en niveau de gris.
Tous les algorithmes de suivi nécessitent une bonne détection à l'initialisation car ils sont très sensibles à cette étape.C'est pourquoi nous avons choisi deux méthodes d'initialisation pour notre algorithme de Camshift. Lapremière méthode est une méthode automatique et la deuxième, une méthode manuelle.

Initialisation automatique

Pour ce type de d'initialisation, nous avons choisi la méthode de soustraction d'arrière plan où il s'agit de soustrairepixel par pixel l'image courante à l'image de fond qui est la moyenne des N premières trame du flux vidéo, dans notre cas N=50.

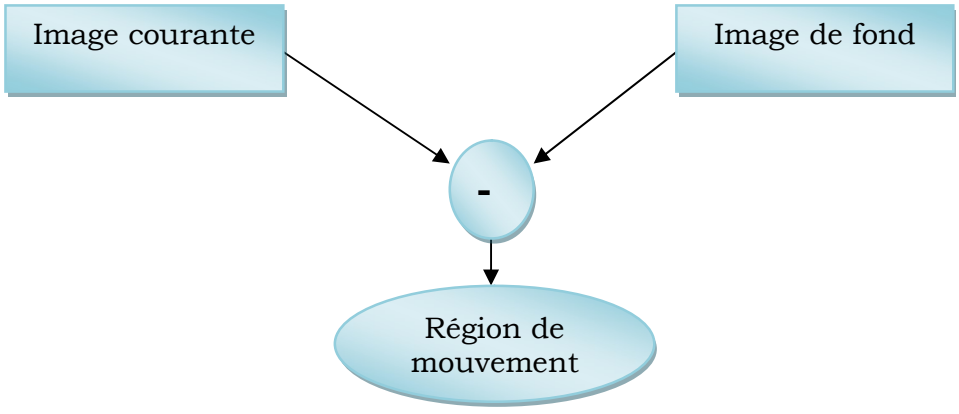


Fig.3.7 Schéma de la soustraction d'arrière plan

Initialisation manuelle

Il s'agit de sélectionner l'objet à suivre manuellement dans la première trame du flux vidéo en utilisant la souris. Cette méthode nécessite d'avoir l'objet à suivre dans le champ de vision dés la première trame.

Ces deux méthodes sont appliquées à l'initialisation du programme pour obtenir le modèle de l'objet à suivre. Une fois le modèleobtenu,on passe à l'étape de suivi qui est la recherche de la position de ce modèle dans les différentes trames qui suivent où on positionne la boite englobante au centre de la région la plus probable et cela jusqu'à la prochaine initialisation ou la fin de la séquence d'images. L'ensemble de ces opérations est résumé dans l'algorithme suivant :

Début
Déclaration des variables
Configuration de la caméra ;
Ouverture du port série de la carte arduino ;
Lire la séquence d'image ;
// $(I_{m,t})$ Image courante ;
// $(A_r P_t)$ Image de fond ;
// (AVP) La région du mouvement ;
Premier choix d'initialisation : Initialisation automatique
Quantifications des trames successives ;
Lire les 50 premières trames ;
$(A_r P_t)$ ← Calculer la moyenne de ces trames ;
$(I_{m,t})$ ← Lire la trame suivante ;
(AVP) ← Soustraire $(A_r P_t)$ de $(I_{m,t})$ pour trouver la région de mouvement ;
Convertir l'image en niveau de gris
Appliquer des opérations morphologiques ;
Calculer l'histogramme couleur de (AVP) ;
Appliquer un masque de pondération sur l'histogramme précédent ;
Deuxième choix d'initialisation : Initialisation manuelle
Sélectionner l'objet a suivre en utilisant la sourie dans la première trame ;
Fin d'initialisation
Pour tout trame
Calculer pour chaque pixel la probabilité d'appartenir à l'objet à suivre ;
Positionné la boite englobante au centre de la région la plus probable ;
Mettre à jour la taille et l'angle de la boite englobante ;
Calculer le centre de la boite englobante ;
Envoyer les coordonnées du centre via la liaison série avec arduino ;
Passer à la trame suivante ;
Fin pour tout

Fin

Fig.3.8 Algorithme du CAMSHIFT

3.3.2 commande des servomoteurs

Dans la section (3.2.4), nous avons expliqué l'électronique d'asservissement des servomoteurs utilisés dans notre application. C'est un asservissement simple a programmer, ou la commande de chaque servodépond de la position d'objet dans l'image par rapport au centre de champ de vision (image).On distingue quatre positionsdifférentes, ces position sont schématisée dans la figure suivante :

(fv

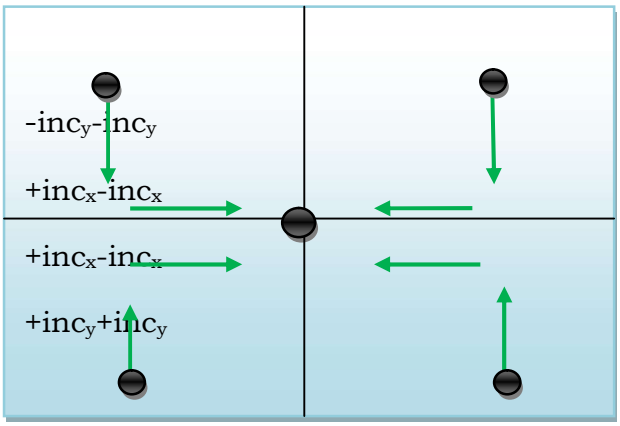


Fig.3.9 Positions possibles de l'objet dans l'image

Donc la commande utilisée est une boucle qui fait incrémenter ou décrémenter la position du servo jusqu'a avoir les coordonnées du centre de l'objet les même avec celles du centre de l'image. Cette boucle est implémentée dans un programme arduino qui récupère successivement les différentes positions de l'objet dans la scène a partir de la liaison serie.les étape de ce programme sont résumé dans l'algorithme ci-dessous :

Début
Partie déclaration des variables
<servo.h> // librairie pour servomoteur
Servox ; //servomoteur de l'axe x
Servoy ; //servomoteur de l'axe y
px ; //position du servox
py ; //position du servoy
cx ; cy ; //coordonné d'objet dans l'axe x et l'axe y

Début d'initialisation
px=90 ; //initialiser les deux servomoteurs aux positions milieux
py=90 ;
Connexion du port série ;
Attendre les donnés du port série ;
Lecture des coordonnés d'objet à suivre vx et vy;
Fin d'initialisation

Début d'exécution

Pour servox	*Pour servoy*
Si (vx< cx) faire	Si (vy<cy) faire
px = px + incx ;	py = py + incy ;
Sinon si (vx> cx) faire	Sinon si (vy>cy) faire
px = px - incx ;	py = py - incy ;
Sinon si (vx = cx) faire	Sinon si (vy = cy) faire
px = px ;py= py ;	

Fin d'exécution
FIN

Fig.3.10 Commande des servomoteurs

3.3.3 Interface graphique du projet

Grâce à l'outil GUIDE (Graphical User Interface Design Environment) de Matlab qui permet de créer facilement des interfaces graphiques. Nous avons put réaliser une interface pour notre système dans le but de faciliter l'utilisation. Cette interface est présentée dans la figure qui suit :

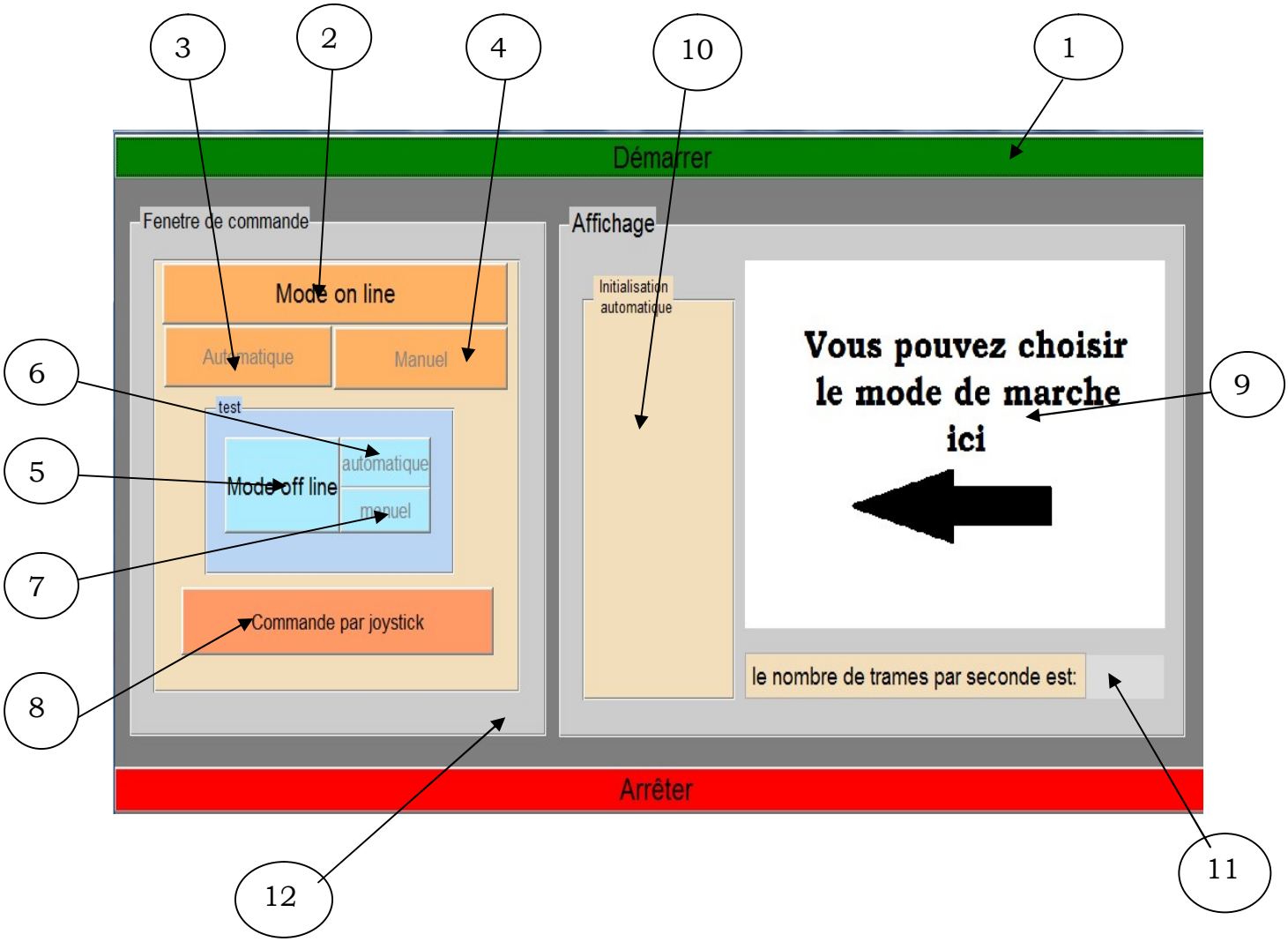


Fig.3.11 Interface graphique du projet

❖ ***Description des fonctions***

1.Le bouton « Démarrer » il permet de mettre en marche les autres boutons de commande.

2. Bouton de mode en line, le choix de marche en temps réel et il donne accès à choisir le mode d'initialisation.

3. Le bouton « automatique » il permet de choisir le mode d'initialisation automatique si on est en temps réel.

4. Le bouton « manuel » il permet de choisir le mode d'initialisation manuel si on est en temps réel.

5.Le bouton de mode off line il permet de passer au suivi sur une séquence d'image existante.

6. Le bouton « automatique » il permet de choisir le mode d'initialisation automatique si on est en off line.

7. Le bouton « manuel » il permet de choisir le mode d'initialisation manuel si on est en offline.

8.Le bouton « commande par joystick »pour une utilisation manuelle du système.

9. Une fenêtre pour afficher la séquence en temps réel ou en off line traiter par l'algorithme de suivi.

10.Une fenêtre pour afficher le rectangle d'initialisation si on est en mode automatique (soustraction d'arrière plan).

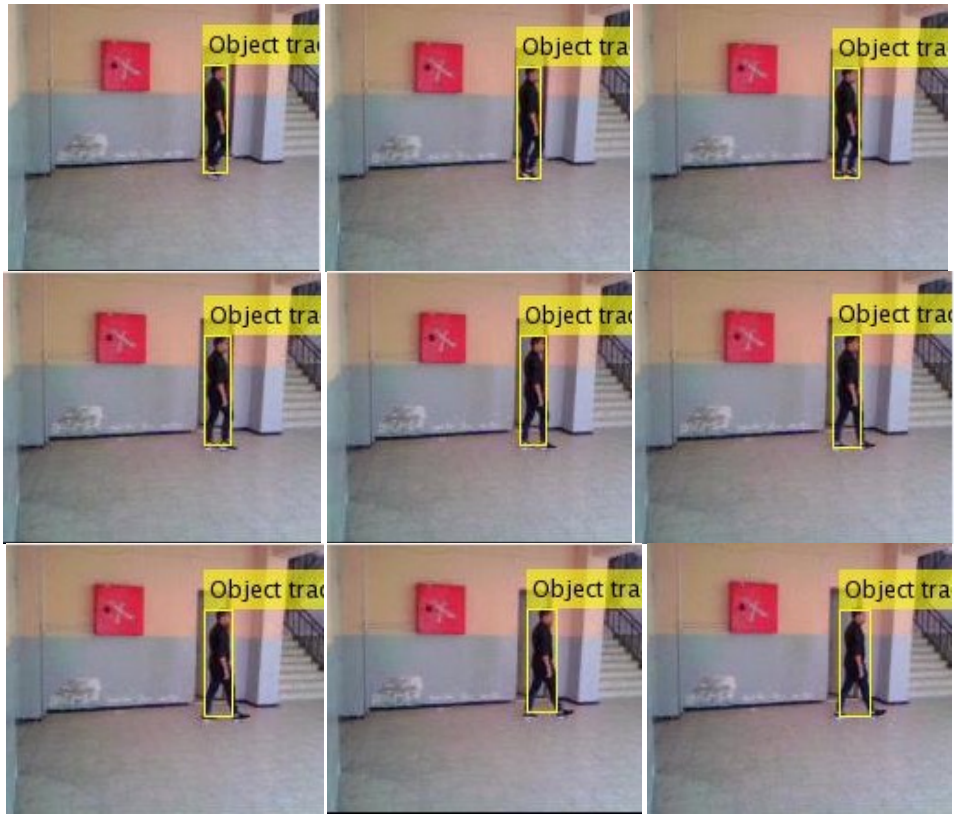
11. L'affichage de la vitesse de traitement en Tps (Traitement par seconde).

12.Le bouton « Arrêter » permet d'arrêter toutes les fonctions de l'application.

3.4 Tests et résultats

Afin d'évaluer notre approche, nous avons utilisé plusieurs tests en utilisant des différentesséquences d'images. Cette application a été réalisée sous Matlabqui offre un environnement interactif pour le calcul numérique, la visualisation et la programmation dans différents domaines d'application et particulièrement le traitement d'images.
Dans cette partie nous allons voir les tests et les résultatstrouvés par une séquence d'image qui a été prise au niveau de notre département en utilisant une camera dynamique (arrière plan n'est pas fixe) de résolution 320x240.
Comme première étape, nous avons affiché quelques trames originales de la séquence utilisée etles mêmes trames après l'application de l'algorithme de suivi.

(a)



(b)

Fig.3.12 Trames (80 à 88) de la séquence utilisée
(a):trames originales, (b):trames de Camshift

Dans la deuxième étape de ce test, nous avons calculé le vecteur d'erreur sur toutes les trames de notre séquence d'images afin de pouvoir évaluer l'algorithme utilisé. Pour ce faire nous avons créé un vecteur référence V_{ref} de dimension N_{trame}x2manuellement en sélectionnant l'objet par la souris dans les différents trames de la séquence d'images, .Ensuite, nous avons appliqué l'algorithme du Camshift sur la même séquence d'images et, au cours de l'exécution, nous avons enregistré le vecteur V_{cam} de même dimension que le précédent qui contient les coordonnés du rectangle de suivi. A la fin, nous avons pu soustraire élément par élément V_{ref}de V_{cam}et le résultat de la soustraction est un vecteur de même dimension : la première colonne est l'erreur selon l'axe x, et la deuxième colonneest l'erreur selon l'axe y.

$$e=|Vcam - Vref| \qquad (3.1)$$

$$\text{telque} : \begin{cases} e = \begin{bmatrix} e_{x1} & e_{y1} \\ \vdots & \vdots \\ e_{x_{Ntrame}} & e_{y_{Ntrame}} \end{bmatrix} \\ Vcam = \begin{bmatrix} x_{cam} & y_{cam1} \\ \vdots & \vdots \\ x_{cam_{Ntrame}} & y_{cam_{Ntrame}} \end{bmatrix} \\ Vref = \begin{bmatrix} x_{ref1} & y_{ref1} \\ \vdots & \vdots \\ x_{ref_{Ntrame}} & y_{ref_{Ntrame}} \end{bmatrix} \end{cases} \qquad (3.2)$$

La figure suivante est la représentation graphique du vecteur d'erreur selon x et y séparément.

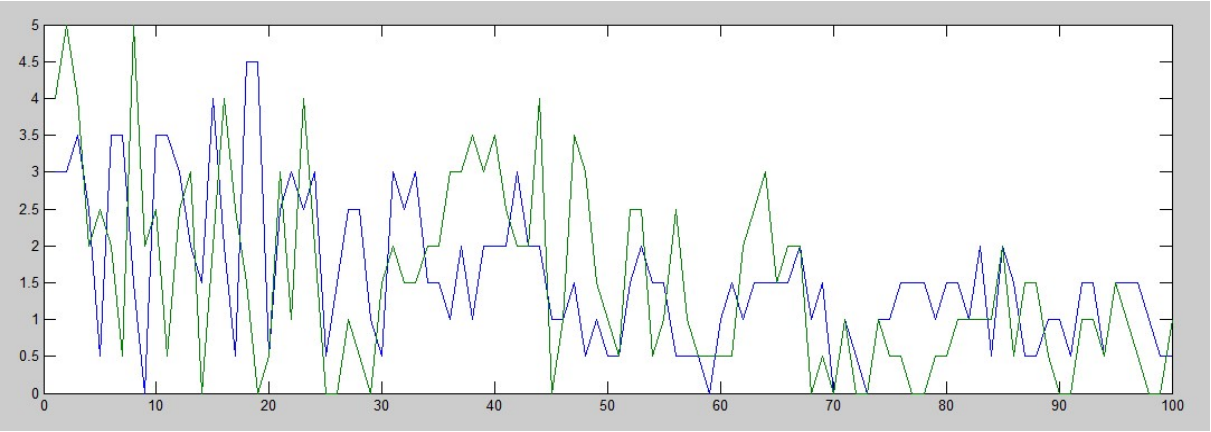


Fig.3.13représentation graphique du vecteur d'érreur ducamshift

Bleu(erreur selon l'axe x),Vert(erreur selon l'axe y)

Interprétation

Dans la figure (3.12), nous remarquons que le vecteur d'erreur varie dans un intervalle de 0 à 5 pixels.tel que, dans les premières trames, l'erreur est un peu élevée et elle n'a pas arrêté de diminuer jusqu'a sa convergence vers la valeur 0.

3.9 Conclusion

Après une description matérielle de notre projet, nous avons passé à la partie software qui est composée de deux parties : la partie image oùnous avons détaillé l'algorithme de suivi utilisé avec ses deux différentes méthodes d'initialisation, en se basant sur l'adaptation de l'approche de suivi par boite englobant avec l'algorithme de camshift. Ladeuxième partie qui est destinéeà l'algorithme de commande des servomoteurs et à la liaison série entre l'environnement Matlab et le logiciel ARDUINO.A la fin nous avons terminé par quelques tests et résultats du système de suivi. Une autre partie a été consacrée à la création de l'interface et la présentation du projet.

Conclusion générale

Dans ce mémoire nous avons traité la détection et le suivi automatique d'objets en mouvement sur le plan théorique et pratique en utilisant une caméra fixée sur support mais mobile dans l'espace 3 dimensions polaires. Le fonctionnement de cette ensemble est piloté par une carte de développement ARDUINO qui consiste à commander le système articulée monté sur deux servomoteurs de telle manière à ce que l'objet soit toujours dans le champ de vision de la camera.

Après avoir divisé notre travail en trois étapes, la première est consacrée à la liaison du logiciel ARDUINO avec Matlab via une communication série pour échanger les informations nécessaires, c'est-à-dire envoyer le centre de l'objet à suivre. Nous avons entamé la deuxième partie en programmant grâce aux logiciel Matalb, le programme de détections et de suivi d'un objet (CAMSHIFT) qui se base sur un algorithme de segmentation d'images couleur. En dernière partie, nous avons programmé grâce au logiciel ARDUINO, l'asservissement en positions des servomoteurs sur le mouvement de rotation du système articulé pour qu'il puisse attendre leur positions avec exactitude.

Pour valider notre approche de suivi (CAMSHIFT), nous avons effectué des tests en utilisant l'environnement Matlab sur une séquence d'images prise au niveau de notre département. En calculant le vecteur d'erreur sur les trames de notre séquence d'images avant et après l'application de l'algorithme, nous avons discuté les résultats obtenus. Après avoir calculé le vecteur d'erreur qui tend vers la valeur zéro, ce qui signifie une bonne détection et suivi de cible mais tout en respectent un certain nombre de conditions et de critère telle que :

- L'objet ciblé doit avoir au moins une couleur.
- L'objet ciblé ne change pas sa couleur au cours du temps.
- L'objet ciblé doit être de couleur différent de l'arrière plan.
- Ne pas avoir d'autre objet de même couleur que l'objet ciblée dans le champ de vision de la camera.
- Ne pas avoir un changement brusque de la luminosité de la scène.

L'inconvénient de cette méthode c'est de respecter ces critères et conditions. Toutefois, à l'heure actuelle il n'existe pas encore de méthode de détection et de suivi qui s'adapte à tout configuration rebot-camera.

Bibliographie

[1] Medjahed fatiha, Université des Sciences et de la Technologie d'Oran U. S. T.O.'Détection et Suivi d'Objets en Mouvement Dans Une Séquence d'Images', mémoire de Magister 2011/2012

[2] Brougui boumaraf.' Faculté des Nouvelles Technologies de l'Information et de la Communication de ouargla Mémoire de Master Académique 2015.

[3]Verbeke, N. (2007). Suivi d'objet en mouvement dans une séquence vidéo. *Thèse de doctorat Université de paris DESCARTES*, France.

[4] Thomas PENNE "Développement d'un système de tracking vidéo sur caméra robotisée",UNIVERSITÉ BLAISE PASCAL - CLERMONT-FERRAND II

[5] Jain, R. et Nagel, H. (1979). On the analysis of accumulative difference pictures from image sequences of real world scenes. Pattern Analysis and Machine Intelligence, pages 206 214.

[6] Salim Iratni.' Asservissement visuel : étude et réalisation d'un dispositif de suivi d'un objet en mouvement'.Université Mouloud Mammeri de Tizi-Ouzou.

[7]Medjahed fatiha, Université des Sciences et de la Technologie d'Oran U. S. T.O.'Détection et Suivi d'Objets en Mouvement Dans Une Séquence d'Images', mémoire de Magister 2011/2012

[8] Alper, Y.; Javed, O. & Shah, M. (2006), 'Object Tracking: A Survey', ACM Journal of Computing Surveys 38 (4).

[9] Veenman, C.; Reinders, M. & Backer, E. (2001), 'Resolving motion correspondence for densely moving points', IEEE Pattern Analysis and Machine Intelligence 23 (1), 54–72.

[10] Serby, D.; Koller, M.S. & Gool, L.V. (2004), 'Probabilistic object tracking using multiple features', 'In IEEE International Conference of Pattern Recognition (ICPR)', 184–187.

[11] Koller, D.; Danilidis, K. & Nagel, H. (1993), 'Model-based object tracking in monocular image sequences of road traffic scenes',International Journal of Computer Vision, 257-281.

[12] Mikram Mounia.' Suivi d'objets dans une séquence d'images par modèle d'apparence : conception et évaluation'.université de BORDEAUX I.

[13] Paragios, N. & Deriche, R. (1999), 'Geodesic active regions for motion estimation and tracking', 'IEEE International Conference on Computer Vision (ICCV)'.

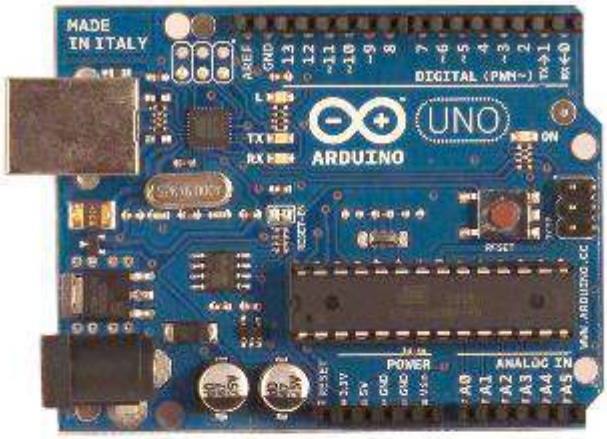
[14] Comaniciu, D.; Ramesh, V. & Meer, P. (2003), 'Kernel-based object tracking', IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (5), 564–577.
[15] Allen, J.; Xu, R. & Jin, J. (2004),'Object Tracking Using CamShift Algorithm and Multiple Quantized Feature Spaces', 'In Proceedings Pan-Sydney Area Workshop on Visual Information Processing (VIP)', Sydney, Australia.
[16] Lee, Y.; You, B. & Lee, S. (2001), 'A real time color based object tracking robust to irregular illumination variations', 'IEEE International Conference on Robotics and Automation', 1659–1664.
[17] Takala, V. & Pietikдinen, M. (2007), 'Multi-object tracking using color, texture and motion', 'Proc. Seventh IEEE International Workshop on Visual Surveillance (VS)', Minneapolis, USA.
[18] Dhouha ATTIA.′ Segmentation d'images par combinaison adaptative couleur/texture et classification de pixels ′.Thèse de doctorat. Université de Technologie de Belfort Montbéliard.
[19] Chang, P. & Krumm, J. (1999),'Object Recognition with Color Cooccurrence Histograms', 'IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', 498-504.
[20] Hu, M. (1962), 'Visual pattern recognition by moment invariants', IRE Transactions on Information Theory 8, 179-187.
[21] Persoon, E. & Fu, K. (1977), 'Shape discrimination using fourier descriptors', IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS 7(3), 629-639.
[22] Bradski, G. (1998), 'Computer vision face tracking for use in a perceptual user interface', Intel Technology Journal 2(2).
[23] Comaniciu, D.; Ramesh, V. & Meer, P. (2003), 'Kernel-based object tracking', IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (5), 564–577.
[24] John G. Allen, Richard Y. D. Xu, Jesse S. Jin ",Object Tracking Using CamShift Algorithm and Multiple Quantized Feature Spaces," Madsen, Building F09, University of Sydney, NSW 2006.
[25] G. Bradski "Computer Vision Face Tracking For Use in a Perceptual User Interface," in *Intel Technology Journal*, pp. 1–15, Q2. 1998.
[26] SAKET JOSHI, SHOUNAK GUJARATHI, ABHISHEK MIRGE"MOVING OBJECT TRACKING METHOD USING IMPROVED CAMSHIFT WITH SURF ALGORITHM"
[27] Gary R. Bradski, Microcomputer Research Lab, Santa Clara, CA, Intel Corporation"Computer Vision Face Tracking For Use in a Perceptual User Interface".

[28] Nicole M. Artner "A Comparison of Mean Shift Tracking Methods" Digital Media, Upper Austria University of Applied Sciences, Hagenberg, Austria

[29] G. R. Bradski. Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal*, 2, April–June 1998.
[30] = J.Hill and W.T.park,"Real time contorl of a robot with a mobile camera",presented at proceedings of the 9th ISIR,Washington D.C,USA1979.
[31] K. A. Dzialo and R. J. Schalkoff, "Control implications in tracking moving objects using timevarying
perspective-projective imagery," *IEEE Transactions on Industrial Electronics*, vol. IE-33, pp. 247-253, 1986.
[32] M. Kabuka, J. Desoto, and J. Miranda, "Robot vision tracking system," *IEEE Transactions on Industrial Electronics*, vol. 35, pp. 40-51, 1988.
[33] IE-33, pp. 247-253, 1986. S. Venkatesan and C. Archibald, "Realtime tracking in five degrees of freedom using two
wrist-mounted laser range finders," presented at International conference on robotics andAutomation, 1990
[34] Malis, E. (1998) Contributions à la modélisation et à la commande en asservissement visuel. *Thèse de doctorat, Université de Rennes1*, France.
[35] Robot commandé par une CMU CAM 4, *Projet de Physique P6 STPI/P6/2014 – 5*
[36] CHELLY Nizar.Formation arduino simulink /android.' Asservissement de vitesse d'un moteur à courant continu à l'aide de la carte Arduino UNO'

Annexes

Arduino UNO



Product Overview

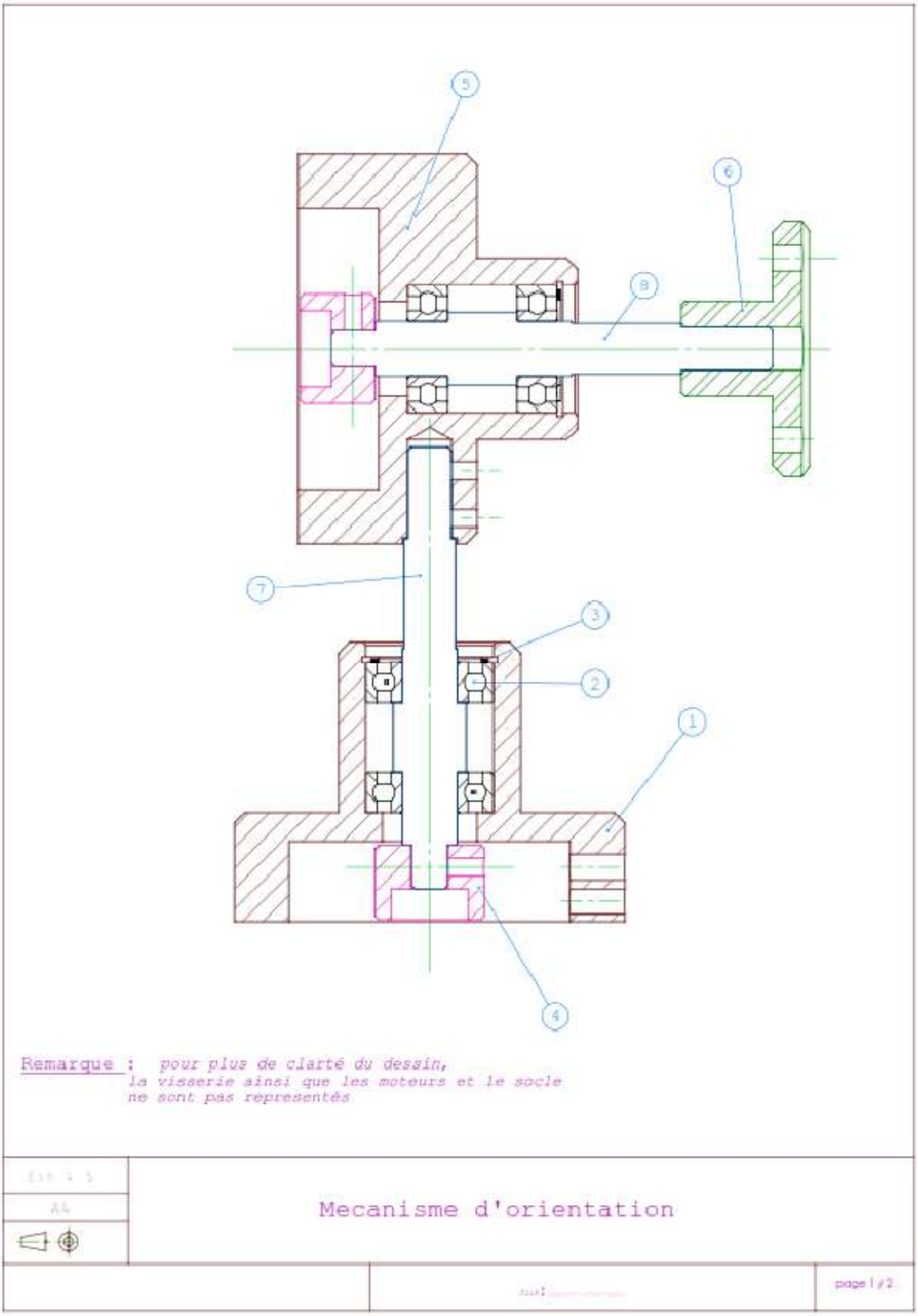
The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduno, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards.

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

Annexe 1 Carte arduino UNO

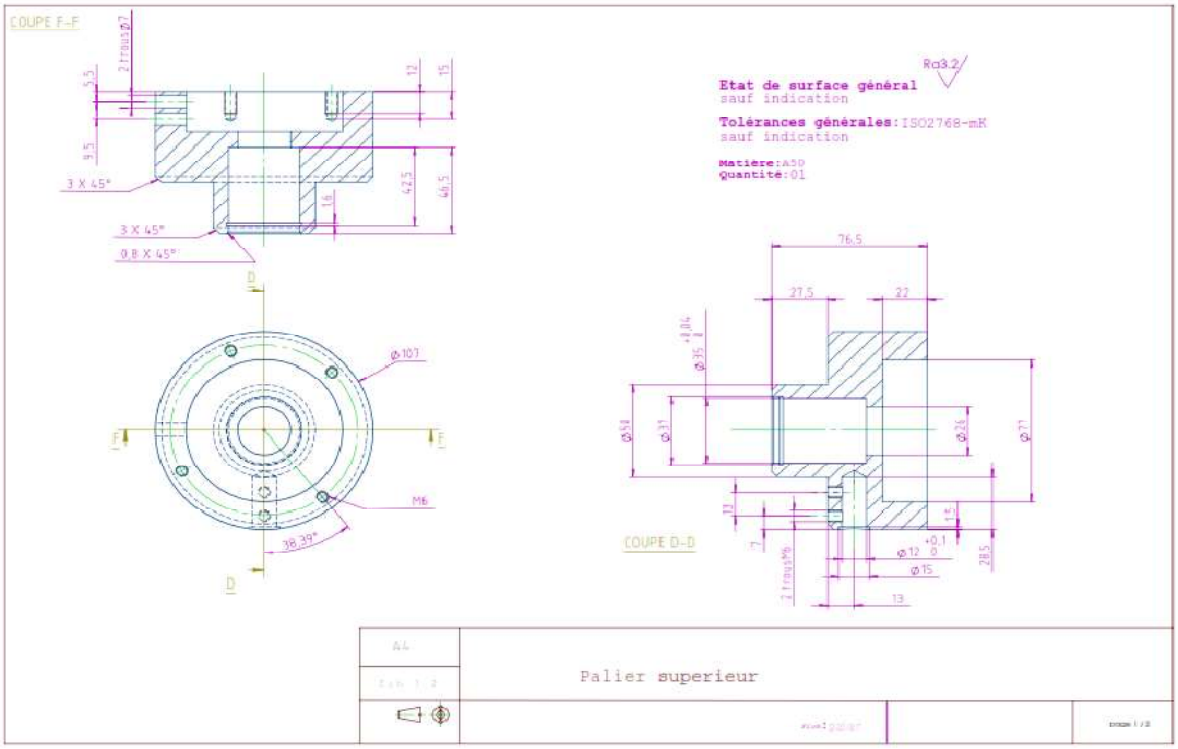


Annexe 2 Mécanisme d'orientation

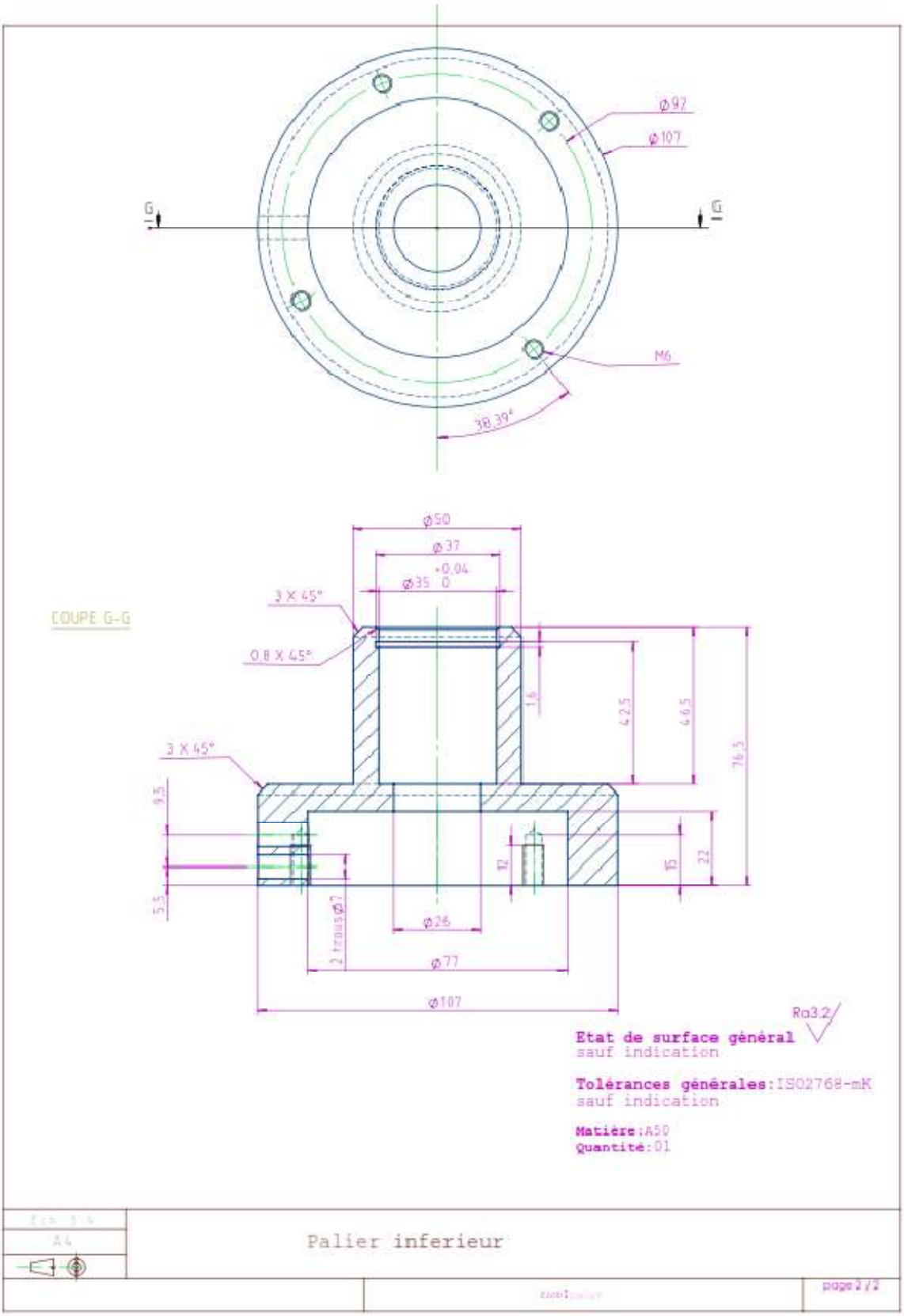
Rep	Nbr	designation	matiere	observation
8	1	Arbre horizontal	acier S235	
7	1	arbre vertical	acier S235	
6	1	atache	Aluminium EN AB-21000 [Al Cu 4 Mg Ti]	
5	1	palier superieur	Aluminium EN AB-21000 [Al Cu 4 Mg Ti]	
4	2	douille	acier S235	
3	2	circlips	-	
2	4	roulements SKF - 6202 - 10,DE,NC,10_68	-	
1	1	palier inferieur	Aluminium EN AB-21000 [Al Cu 4 Mg Ti]	

A4 — Nomenclature

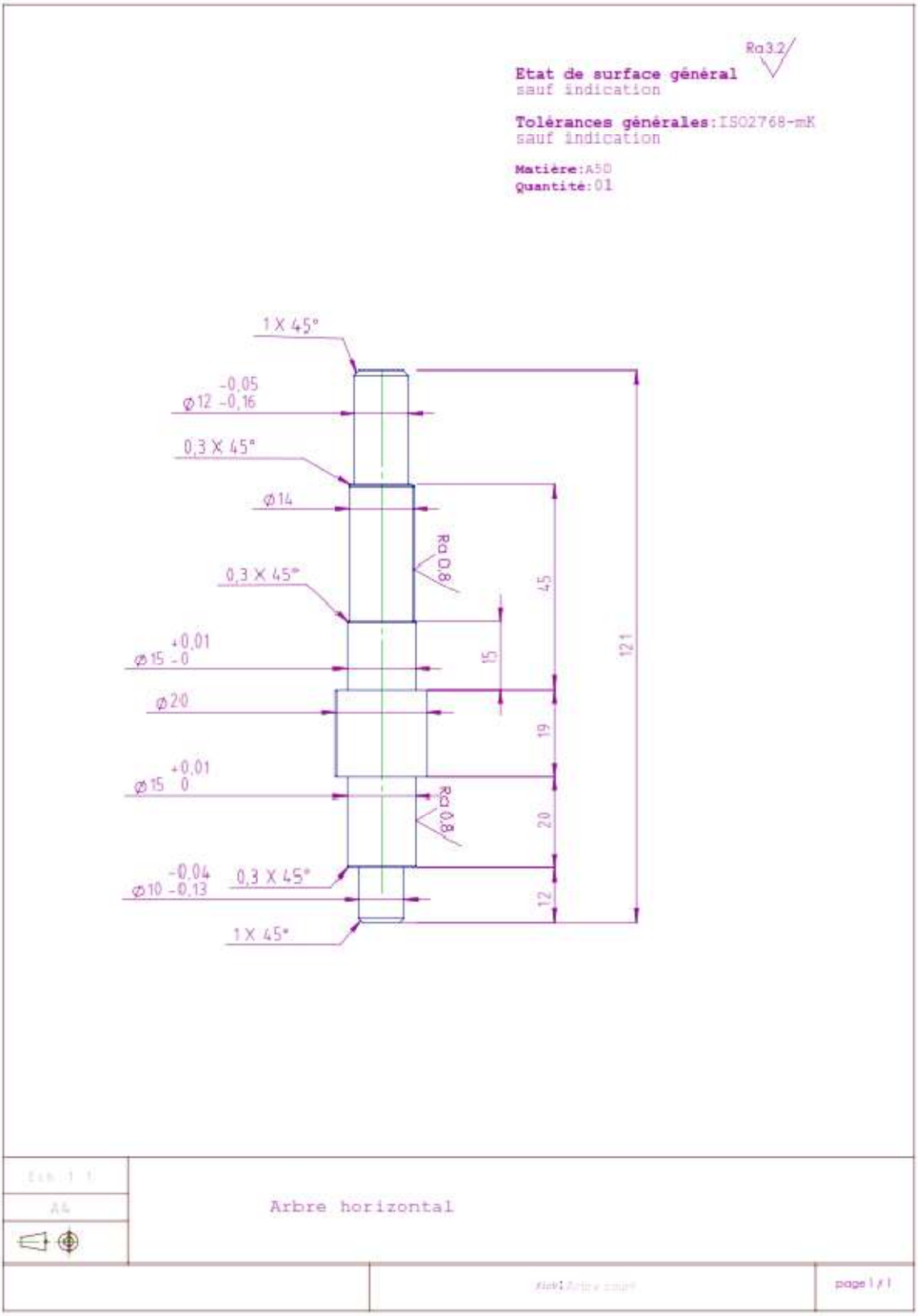
Annexe 3 Nomenclature de la conception mécanique



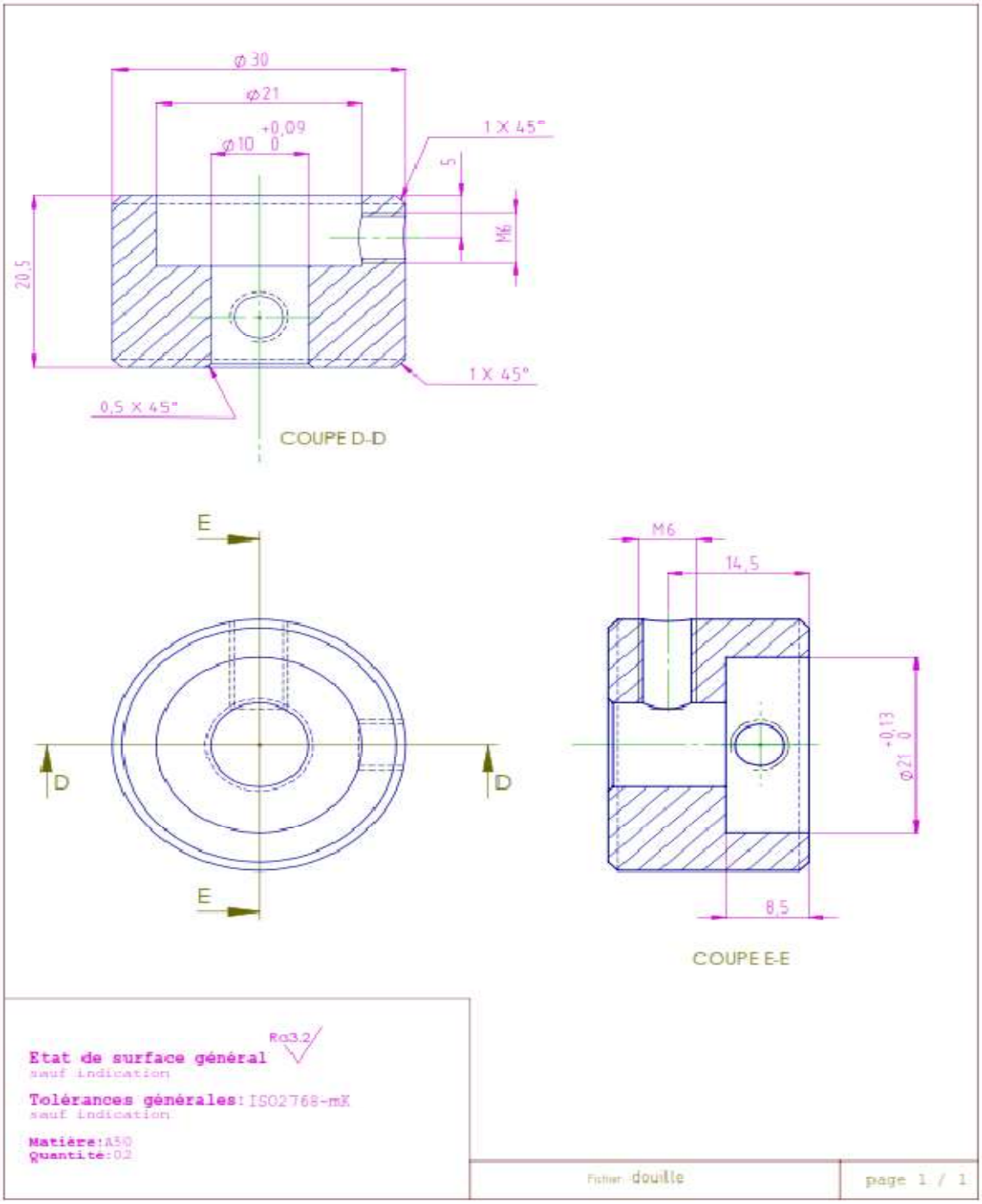
Annexe 4 Palier supérieur



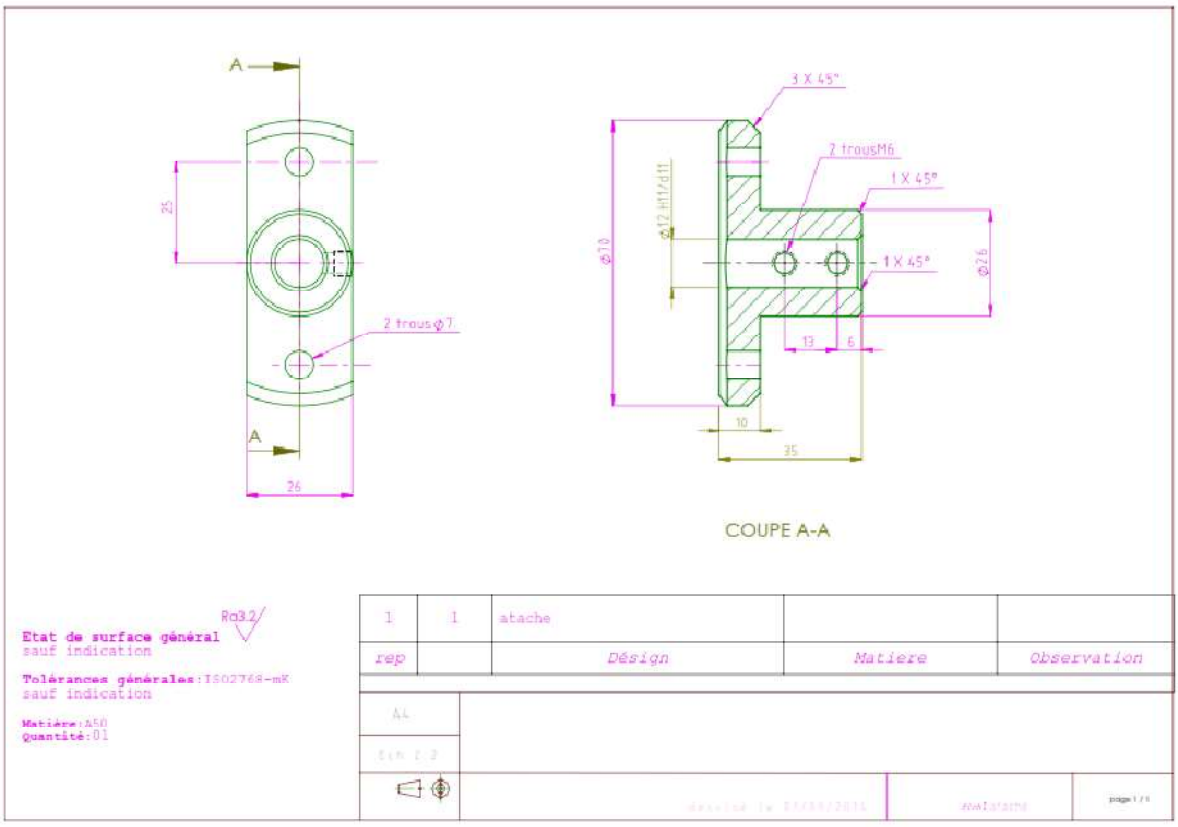
Annexe 5 Palier inférieur



Annexe 6 Arbre horizontal



Annexe 7 Douille



Annexe 8 Attache