Complex Objects Pose Estimation based on Image Moment Invariants

Omar Tahri and François Chaumette
IRISA/INRIA Rennes
Campus de Beaulieu, 35 042 Rennes-cedex, France
Firstname.Name@irisa.fr

***Abstract*— Moments are generic (and usually intuitive) descriptors that can be computed from several kinds of objects defined either from closed contours or from a set of points. In this paper, image moments are used in two new methods for the pose estimation of a planar object observed through full perspective model. The first method is based on an iterative optimization scheme formulated as virtual visual servoing, while the second is based on an exhaustive but efficient optimization scheme of the two most critical parameters. It allows to avoid local minima. We finally present some experimental results to validate the theoretical developments presented in this paper.**

***Index Terms*— Pose estimation, moment invariants.**

I. INTRODUCTION

Many approaches have been developed to estimate the pose between a known object and a camera using an image of this object. Pose estimation has many application in robotics, such as robot localization using a vision sensor, or position-based visual servoing. The geometric features considered for the estimation of the pose are often points [1], segment,s contours [2], or conics. Another important issue is the registration problem. Purely geometric or numerical and iterative [1] approaches may be considered. Linear approaches use a least-square method to estimate the pose. Full-scale non-linear optimization techniques consist of minimizing the error between the observation and the back-projection of the model [11]. Minimization is handled using numerical iterative algorithms such as Newton-Raphson or Levenberg-Marquartd. The main advantage of these approaches is their accuracy. The main drawback is that they may be subject to local minima and, worse, divergence. Therefore they usually require a good guess of the solution to ensure correct convergence. Furthermore, those methods generally have the basic requirement of feature matching.

In this paper, we are concerned with the determination of the pose of planar objects with complex shape using 2D moment invariants. A method to estimate the pose from a single 2D view has been proposed in [12]. It was based on 2D moments but only rotational motions were considered. In [13], translational motions are also considered. Unfortunately, this method assumes that the motion of planar objects is equivalent to an affine transformation in the image, which is not true in the general case of perspective transformation.

The method we propose is based on virtual visual servoing (VVS) using moment invariants as features and initialized by an exhaustive optimization scheme. In other words, we consider the problem of the pose computation as similar to the positioning of a virtual camera using features in the image [11]. This method is equivalent to non-linear methods that consists in minimizing a cost function using iterative algorithms. As mentioned above, it is well known that non-linear iterative methods give an accurate estimation of the pose. However, convergence towards local minima or divergence may appear if the difference between the initial value and the correct value of the pose is large. In fact, the domain of convergence depends on the features used to compute the pose. This domain is very large using the moment invariants proposed in [16] for visual servoing. The same features are used in this paper for pose computation. Furthermore, a second method based on an exhaustive but efficient optimization scheme of the two most critical parameters is proposed to get an accurate initialization to the iterative approach.

In the next section, we present the pose estimation problem formulation and some basic concepts on visual servoing and moment invariants. We also describe the method we propose, as well as experimental results. In Section III, the principle of the pose estimation using an exhaustive search is given. A generalization of the methods we propose to free model partial pose estimation is given in Section IV. Finally, the whole results obtained for continuous case are generalized to discrete objects in Section V.

II. POSE ESTIMATION BY VVS

A. Principle

1) Problem definition: The problem of pose estimation consists in determining the rigid transformation ${}^c\mathbf{M}_o$ between the object frame $\mathcal{F}_o$ and the camera frame $\mathcal{F}_c$ in unknown position using the corresponding object image. It is well known that the relation between an object point with coordinates $\mathbf{X}_c = (X_c, Y_c, Z_c, 1)$ in $\mathcal{F}_c$ and $\mathbf{X}_o = (X_o, Y_o, Z_o, 1)$ in $\mathcal{F}_o$ can be written:

$$\mathbf{X}_c = {}^c\mathbf{M}_o \, \mathbf{X}_o = \begin{pmatrix} {}^c\mathbf{R}_o & {}^c\mathbf{t}_o \\ 0_{1\times 3} & 1 \end{pmatrix} \mathbf{X}_o \qquad (1)$$

Matrix ${}^c\mathbf{M}_o$ can be estimated by minimizing the error module:

$$e = ||\mathbf{s}({}^\mathbf{c}\mathbf{M}_\mathbf{o}) - \mathbf{s}^*|| \qquad (2)$$

where $\mathbf{s}^*$ is the value of a set of visual features computed in the image acquired in the camera unknown position and $\mathbf{s}({}^\mathbf{c}\mathbf{M}_\mathbf{o})$ is the value of the same set of features computed from the object model, the transformation

${}^c\mathbf{M}_o$, and the camera model. In this paper, the coordinates (x, y) of an image point are given by the full perspective camera model (i.e. $x = X_c/Z_c, y = Y_c/Z_c$). The minimization of the error e can be treated as the positioning of a virtual camera using a virtual visual servoing scheme (VVS) [11]. Indeed, VVS consists in moving a virtual camera from a known initial pose (referenced by frame $\mathcal{F}_i$ on Fig. 1) to the final unknown pose (referenced by frame $\mathcal{F}_c$ on Fig. 1) where e is minimized. In this paper, we will use moment invariants as visual features to, first, consider object of complex shape, and then obtain a large convergence domain.

2) Visual servoing: In few words, we recall that the time variation $\dot{\mathbf{s}}$ of the visual features $\mathbf{s}$ can be expressed linearly with respect to the relative camera-object kinematics screw $\mathbf{v}$ by $\dot{\mathbf{s}} = \mathbf{L_s}\mathbf{v}$ where $\mathbf{L_s}$ is the interaction matrix related to $\mathbf{s}$. The control scheme is usually designed to try to ensure an exponential decoupled decrease of the visual features to their desired value $\mathbf{s}^*$, from which we deduce if the object is motionless:

$$\mathbf{v_c} = -\lambda \widehat{\mathbf{L_s}}^{+}(\mathbf{s} - \mathbf{s}^*) \tag{3}$$

where $\widehat{\mathbf{L_s}}$ is a model or an approximation of $\mathbf{L_s}$, $\widehat{\mathbf{L_s}}^{+}$ the pseudo-inverse of $\widehat{\mathbf{L_s}}$, λ a positive gain tuning the time to convergence, and $\mathbf{v_c}$ the velocity of the virtual camera. In the following, we denote respectively $\boldsymbol{\upsilon}$ and $\boldsymbol{\omega}$ the translational and the rotational components of the kinematic screw, so that $\mathbf{v_c} = (\boldsymbol{\upsilon}, \boldsymbol{\omega}) = (\upsilon_x, \upsilon_y, \upsilon_z, \omega_x, \omega_y, \omega_z)$. As for classical visual servoing schemes, a first necessary condition of convergence is that the interaction matrix must be not singular. Hence, a good choice of the features must allow to obtain a large domain where the matrix $\mathbf{L_s}$ has full rank 6. A good way to ensure this condition is to design a decoupled control scheme, i.e. to try to associate each camera dof with only one visual feature. Such control would make easy the determination of the potential singularities of the considered task, as well as the choice of $\widehat{\mathbf{L_s}}$. Unfortunately, a perfect decoupling is probably impossible to reach. It is however possible to decouple the translational motion from the rotational one using moment invariants [16].

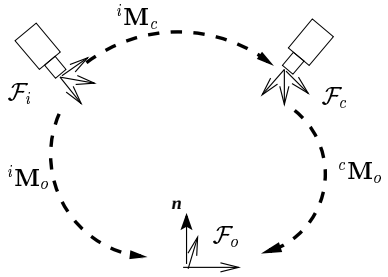

Fig. 1. Principle of VVS

In fact, the main difference between VS and VVS is that the current visual features are computed in VVS while they are extracted from acquired images in VS. The potential problems of virtual visual servoing are then almost the same as visual servoing ones (i.e. local minima, interaction matrix singularities). To avoid those problems, the same features proposed in [16] could be used. However, we will see that a better algorithm can be designed. In the next paragraph, we recall some moment definitions and the features proposed in [16].

3) Decoupled control using image moments: if we consider an object $\mathcal{O}$ in the image, its 2D moments m_{pq} of order $p + q$ are defined by:

$$m_{pq} = \iint_{\mathcal{O}} x^p y^q dx dy \tag{4}$$

Of course, moments cannot be computed in the degenerate case where the planar object reduces to a segment in the image. As all pose estimation methods of planar objects, the methods presented in this paper do not consider this degenerate case.

The centered moments μ_{pq} are computed with respect to the object centroid (x_g, y_g). They are defined by:

$$\mu_{pq} = \iint_{\mathcal{O}} (x - x_g)^p (y - y_g)^q dx dy \tag{5}$$

where $x_g = m_{10}/a$ and $y_g = m_{01}/a$, $a = m_{00}$ being the object area. In [16], the following visual features have been proposed to control the six degrees of freedom of a camera:

$$\mathbf{s} = (x_n, y_n, a_n, r_i, r_j, \alpha) \tag{6}$$

$$\text{where } \begin{cases} x_n = a_n x_g \ , \ y_n = a_n y_g \ , \ a_n = Z^* \sqrt{\frac{a^*}{a}} \\ \alpha = \frac{1}{2} \arctan(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}) \end{cases}$$

a^* is the desired value of the object area, Z^* the desired depth of the object, α the orientation of the object in the image, and r_i, r_j two invariants obtained by combining three kinds of moment invariants: invariant to translation, to 2D rotation and to scale. For instance, r_i, r_j can be chosen as:

$$r_1 = I_{n_1}/I_{n_3} \ , \ r_2 = I_{n_2}/I_{n_3} \tag{7}$$

with:

$$\begin{cases} I_{n_1} = (\mu_{50} + 2\mu_{32} + \mu_{14})^2 + (\mu_{05} + 2\mu_{23} + \mu_{41})^2 \\ I_{n_2} = (\mu_{50} - 2\mu_{32} - 3\mu_{14})^2 + (\mu_{05} - 2\mu_{23} - 3\mu_{41})^2 \\ I_{n_3} = (\mu_{50} - 10\mu_{32} + 5\mu_{14})^2 + (\mu_{05} - 10\mu_{23} + 5\mu_{41})^2 \end{cases}$$

Complete details on how r_i and r_j have been determined can be found in [15]. When the object is parallel to the image plane, the interaction matrix $\mathbf{L}_\mathbf{s}^{\parallel}$ related to the above six features has the following form [16]:

$$\mathbf{L}_\mathbf{s}^{\parallel} = \begin{bmatrix} -1 & 0 & 0 & x_{n_{wx}} & x_{n_{wy}} & y_n \\ 0 & -1 & 0 & y_{n_{wx}} & y_{n_{wy}} & -x_n \\ 0 & 0 & -1 & -3y_n/2 & 3x_n/2 & 0 \\ 0 & 0 & 0 & r_{i_{wx}} & r_{i_{wy}} & 0 \\ 0 & 0 & 0 & r_{j_{wx}} & r_{j_{wy}} & 0 \\ 0 & 0 & 0 & \alpha_{wx} & \alpha_{wy} & -1 \end{bmatrix}$$

As expected, we can notice the invariance of the last three selected features with respect to any 3D translational motion (when the image and the object planes are parallel), and the invariance of r_i and r_j with respect to ω_z. We can also note the very nice form of the interaction matrix for the three first features (note that upper left 3×3 block of $\mathbf{L}_\mathbf{s}^{\parallel}$ is equal to $-\mathbf{I}_3$). From this property, we can deduce

that when the rotational motion vanishes, the translational motion can be computed by:

$$\left\{ \begin{array}{rcccl} t_x & = & -\int_{x_n}^{x_n^*} dx_n & = & x_n - x_n^* \\ t_y & = & -\int_{y_n}^{y_n^*} dy_n & = & y_n - y_n^* \\ t_z & = & -\int_{a_n}^{a_n^*} da_n & = & a_n - a_n^* \end{array} \right. \quad (8)$$

B. Method

The principle of our method is different from the method described above. Indeed, to maintain the good decoupling properties obtained when the object is parallel to the image plane, we consider that the unknown camera pose corresponding to the current image is the initial one, while the desired camera pose is known and such that its image plane is parallel to the object (for example the camera position referenced by frame $\mathcal{F}_p$ on Fig. 2 and defined by the transformation ${}^p\mathbf{R}_o = \mathbf{I}_3$, ${}^p\mathbf{t}_o = (0, 0, 1)$). The image of the object for this position can be obtained using the object model (if available) and the camera model. Otherwise, it is possible to determine this image for a parallel position without any model knowledge from two different images as we propose in Section IV. In this case, an approximation of the depth between the object and the camera can be used. A wrong value of the depth (Z^*) will change only the estimated translational motion vector module. According to Fig 2, we have ${}^c\mathbf{M}_o = {}^c\mathbf{M}_p {}^p\mathbf{M}_o$. Thus we have only to estimate ${}^c\mathbf{M}_p$ to determine the pose ${}^c\mathbf{M}_o$.

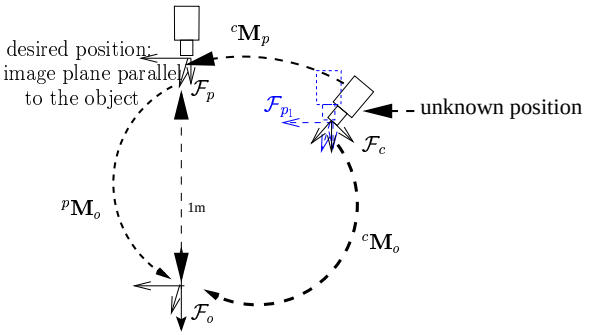

Fig. 2. Rotation estimation principle

The determination of the visual features after a translational motion requires the knowledge of the object plane parameters. Since the camera initial position is unknown, those parameters are also unknown. For this reason, the method we propose is divided in two steps: firstly, we determine the rotation between $\mathcal{F}_c$ and $\mathcal{F}_p$ using (r_i, r_j, α) in a VVS loop. We will see in the following that the computation of (r_i, r_j, α) after a rotational motion does not require the knowledge of any 3D information. Once the rotation is determined, the translation is directly obtained using (8).

Let $\mathcal{F}_{p_1}$ be the frame that has the same orientation than the frame $\mathcal{F}_p$ and the same origin than $\mathcal{F}_c$ (see Fig. 2). The visual features proposed to control the rotational motions (i.e. invariants r_i, r_j and α) are invariant to 3D translational motions when the object and the image plane are parallel. Since only a translational motion exists between $\mathcal{F}_p$ and $\mathcal{F}_{p_1}$, those invariants have then the same value for those two camera positions.

The rotational motion estimation is realized as follow:

- computation of the features $\mathbf{s}^* = (r_i^*, r_j^*, \alpha^*)$ for the position referenced by $\mathcal{F}_{p_1}$ (computed from the image corresponding to pose $\mathcal{F}_p$).
- computation of the features $\mathbf{s} = (r_i, r_j, \alpha)$ for the pose referenced by $\mathcal{F}_c$.
- initialization of the rotation: the rotation matrix is initialized by the value that corresponds to a rotation around the optical axis of the camera by the angle $\alpha^* - \alpha$: ${}^{c_1}\mathbf{R}_c = \mathbf{R}_{z_{(\alpha^* - \alpha)}}$.
- determination of rotational motion by VVS. It is obtained using iteratively the following steps:

1) determination of the rotational velocity:

$$\boldsymbol{\omega} = -\lambda \widehat{\mathbf{L}_\mathbf{s}}^{-1} (\mathbf{s} - \mathbf{s}^*) \text{ with } \widehat{\mathbf{L}_\mathbf{s}} = (\frac{\mathbf{L}_{\boldsymbol{\omega}}(s) + \mathbf{L}_{\boldsymbol{\omega}}(s^*)}{2})$$

$$\text{and } \mathbf{L}_{\boldsymbol{\omega}} = \begin{bmatrix} r_{i_{wx}} & r_{i_{wy}} & 0 \\ r_{j_{wx}} & r_{j_{wy}} & 0 \\ \alpha_{wx} & \alpha_{wy} & -1 \end{bmatrix}$$

This choice of $\widehat{\mathbf{L}_\mathbf{s}}$ ensures a fast convergence of the algorithm [16], [9].

2) computation of the rotation matrix: ${}^{c_{k+1}}\mathbf{R}_c = \Delta\mathbf{R}^{\mathbf{c_k}}\mathbf{R_c}$ where $\Delta\mathbf{R}$ is computed using the Rodrigues formula:

$$\Delta\mathbf{R} = \mathbf{I_3} + \sin\theta \, [\mathbf{u}_\theta]_\times + (\mathbf{1} - \cos\theta) \, [\mathbf{u}_\theta]_\times{}^\mathbf{2}$$

with $\theta = \sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2}$, $\mathbf{u}_\theta = \frac{\boldsymbol{\omega}}{\theta}$ and $[\mathbf{u}_\theta]_\times$ is the antisymmetric matrix defined from $\mathbf{u}_\theta$.

3) Computation of the new value of m_{ij} from their initial value and ${}^{c_{k+1}}\mathbf{R}_c$. Let us denote $\mathbf{X}_t$ and $\mathbf{X}$ the coordinates of a 3D point after and before the virtual rotational motion. We have of course:

$$\mathbf{X}_t = {}^{c_{k+1}}\mathbf{R}_c \mathbf{X} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \mathbf{X} \quad (9)$$

The moments after a rotational motion can thus be obtained by [16]:

$$m_{t_{pq}} = \iint_{\mathcal{O}} \frac{(r_{11}x + r_{12}y + r_{13})^p (r_{21}x + r_{22}y + r_{23})^q}{(r_{31}x + r_{32}y + r_{33})^\gamma} dxdy \quad (10)$$

where $\gamma = p + q + 3$, $x = X/Z$ and $y = Y/Z$. We can note that transformation (10) can be computed directly from the rotation and the image moments. An estimation of the coordinates of the 3D point is thus useless. The pose estimation by VVS requires several iterations to converge to the desired solution. The direct use of (10) to compute the moments after a rotational motion is time consuming. We thus propose to use a Taylor series expansion of the term $(r_{31}x + r_{32}y + r_{33})^{-\gamma}$. Indeed, if $r_{31}x + r_{32}y \ll r_{33}$ (which is the case in practice), the moments after a rotational motion can be obtained by [16]:

$$\begin{aligned} m_{t_{pq}} \approx \frac{r_{13}^p r_{23}^q}{r_{33}^\gamma} \sum_{k_1=0}^{p} \sum_{l_1=0}^{k_1} \sum_{k_2=0}^{q} \sum_{l_2=0}^{k_2} \binom{k_1}{p} \binom{l_1}{k_1} \binom{k_2}{q} \binom{l_2}{k_2} \\ \left(\frac{r_{11}}{r_{13}}\right)^{l_1} \left(\frac{r_{12}}{r_{13}}\right)^{k_1 - l_1} \left(\frac{r_{21}}{r_{23}}\right)^{l_2} \left(\frac{r_{22}}{r_{23}}\right)^{k_2 - l_2} \\ (m_{l,k-l} - \frac{(r_{31} m_{l+1,k-l} + r_{32} m_{l,k-l+1})}{r_{33}} + \ldots) \end{aligned} \quad (11)$$

with $k = k_1 + k_2$ and $l = l_1 + l_2$.

4) Computation of the new visual features $\mathbf{s} = (r_i,\, r_j,\, \alpha)$ from the moments computed at step 3.

5) Go to step 1 while $e = (r_i^* - r_i)^2 + (r_j^* - r_j)^2 + (\alpha^* - \alpha)^2 > \epsilon$

Finally, we recall that, once the rotation is determined, the translation is directly obtained from (8), where $(x_n^*,\, y_n^*,\, a_n^*)$ are computed for the position referenced by $\mathcal{F}_p$ and $(x_n,\, y_n,\, a_n)$ are computed using (11) from the image acquired for the unknown position and the rotation that has been obtained.

C. Experimental results

The images used in this experiment are given on Figure 3. The real value and the estimated value of the pose between frames $\mathcal{F}_c$ and $\mathcal{F}_p$ are given respectively by P_1 to P_4, and $\widehat{P}_1$ to $\widehat{P}_4$ on Table I ($\mathbf{u}\theta$ is the angle of rotation by the normalized rotation vector). For the three first results, the estimated pose corresponds to its real value, which proves that the convergence domain of our method is large. However, in the critical case where the pose to estimate corresponds to the image given on Figure 3.e (which corresponds to pose P_4 that is a rotational motion equal to 75^o around the x axis), we can see on Table I that the method converges to a local minimum. In fact, the displacement between the initial pose and the pose to determine is too large. The problem of local minimum is generally an intrinsic problem of iterative approaches. To avoid this problem, we propose in the next section a new efficient pose estimation method based on an exhaustive optimization algorithm. This method can be useful to initialize the method using VVS.

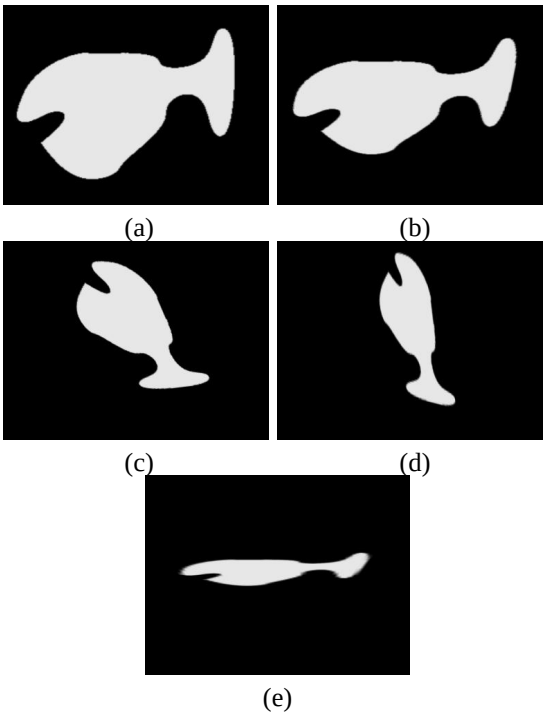

Fig. 3. Images used for pose estimation: (a) image obtained when camera and object planes are parallel, (b) image obtained after transformation P_1, (c) after transformation P_2, (d) after transformation P_3, (e) and after transformation P_4.

	$\mathbf{u}\theta$ (dg)	$\mathbf{t}$ (m)
P_1	$(-40\ 0\ 0)$	$(0\ -0.642\ 0.233)$
$\widehat{P}_1$	$(-40.42\ -0.30\ 0.33)$	$(0.007\ -0.647\ 0.238)$
$\widehat{P}_1$'	$(-40\ 0.\ 0)$	$(0.000\ -0.643\ 0.231)$
P_2	$(-40\ 0\ -69)$	$(-0.500\ -0.689\ -0.111)$
$\widehat{P}_2$	$(-39.90\ 0.25\ -69.29)$	$(-0.504\ -0.688\ -0.117)$
$\widehat{P}_2$'	$(-39.68\ 0.27\ -69.05)$	$(-0.491\ -0.694\ -0.120)$
P_3	$(-50\ -35.36\ -79.06)$	$(-0.141\ -1.149\ 0.229)$
$\widehat{P}_3$	$(-50.29\ -35.49\ -78.76)$	$(-0.160\ -1.163\ 0.216)$
$\widehat{P}_3$'	$(-49.84\ -35.76\ -78.59)$	$(-0.148\ -1.165\ 0.211)$
P_4	$(-75\ 0\ 0)$	$(-0\ -1.158\ 0.688)$
$\widehat{P}_4$	$(-62.03\ -29.02\ 11.59)$	$(0.652\ -0.921\ 0.579)$
$\widehat{P}_4$'	$(75.03\ -0.17\ 0.22)$	$(0.003\ -1.204\ 0.675)$

TABLE I

III. Pose using an exhaustive optimization

A. Method

As for the previous method, we determine the pose in two steps. The rotational motion is determined at first and the translational motion is then again obtained using (8). Let us suppose that the unknown position of the camera (i.e. the position referenced by the frame $\mathcal{F}_c$) is such that its image plane is parallel to the object. In that case, we would have only to apply a rotational motion by $\alpha^* - \alpha$ around the camera optical axis to cancel the rotation between $\mathcal{F}_c$ and $\mathcal{F}_p$. Thus, if the rotational motion between the camera in its unknown position and the parallel position is determined, the other parameters of the pose (i.e translation $(t_x,\, t_y,\, t_z)$ and the rotation around the optical axis) can be computed by analytical formulas, that are (8) and $u_{\theta_z} = \alpha^* - \alpha$.

Let $\mathbf{r} = (r_1,\, r_2,\, ...,\, r_n)$ be a set of invariants such those given by (7). The set $\mathbf{r}$ is invariant to translational motion and to rotational motion around the optical axis only when the object and the camera plane are parallel. Hence the rotational motion between the camera unknown position and the parallel position is such that the error between the invariant vector computed for the two positions vanishes. This rotational motion can be determined by an exhaustive search of the rotation couple $(\alpha,\, \beta)$ that minimizes:

$$e_{\mathbf{r}} = ||\mathbf{r}(\alpha, \beta) - \mathbf{r}^*||$$

where α and β, which represent the rotation angles around x and y axis, are varying from a minimal value to a maximal value with a given step. $\mathbf{r}^*$ is the invariant vector computed for the parallel position, and $\mathbf{r}(\alpha, \beta)$ is computed from the image acquired in the unknown position and the rotation couple (α, β) using (11).

Thanks to the moment invariants, the exhaustive optimization search concerns only two degrees of freedom (i.e rotational motions around x and y-axis), while the pose is represented by six parameters. This process is thus not time consuming. In the next paragraph, we validate this method using the same images used for the VVS method.

B. Results

Figure 4 gives the plot of the function $f(e_{\mathbf{r}}) = \frac{1}{c+e_{\mathbf{r}}}$ where $c = 0.1$ ($f(e_{\mathbf{r}})$ is maximal when $e_{\mathbf{r}}$ is minimal). This table is built by varying the rotation angles around x

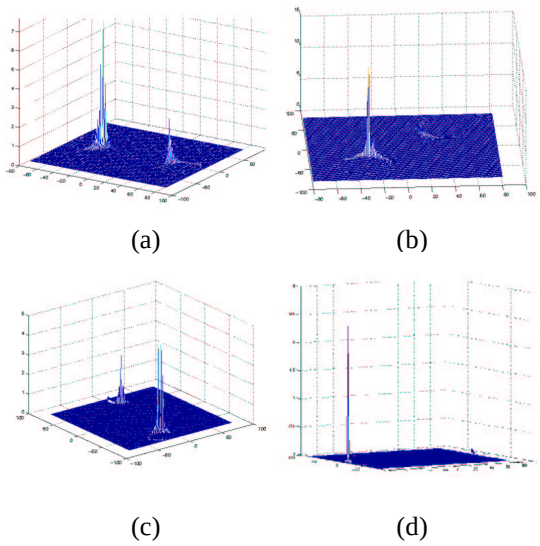

Fig. 4. Amplitude of $f(e_r)$ with respect to rotation : (a) obtained result for P_1, (b) for P_2, (c) for P_3, (d) and for P_4

and y axis between -80^o and 80^o by step equal to 1^o. From this figure, we note the existence of the big peak that corresponds to the global minimum. The other peaks correspond to local minima. Those local minima are potential problems of the iterative method. The estimated values of the pose computed from the global minimum are given by P_i' on Table I. It is clear that the estimated values are accurate (up to 1^o on each axis), even for the difficult pose P_4

For both previous methods, the determination of the rotational motion does not require the exact knowledge of the object model. Only an image such that the image and the object planes are parallel is required. In the next section, we present a new method to determine a partial pose estimation scheme (i.e. the rotational motion and the translational motion up to a scale factor) using two images acquired with any orientation. We will see that an image of the object in parallel position can be also derived.

IV. Free model partial pose estimation

The estimation of the motion between two camera poses with no accurate object model is a classical problem in computer vision. Several methods were proposed to solve the problem of "structure from motion" by linear algorithms. For uncalibrated systems, the approaches are generally based on the fundamental matrix [8]. Otherwise, for calibrated systems, they are based on the essential matrix [5]. The fundamental matrix must be of rank 2, while the essential matrix hold the Huang-Faugeras conditions [6]. Those constraints are non linear and they are introduced after a linear estimation of the fundamental or essential matrices. The estimated value using linear methods can be improved using a non linear method [7]. Unfortunately, this kind of methods are very sensitive to initialization. From the essential matrix, it is possible to estimate the rotational motion of the camera and the translational motion with a scale factor.

The partial motion of the camera can be also determined through an homography matrix (see [10] for instance). If the object is planar, the homography matrix is called collineation matrix, and it is defined by:

$$\mathbf{m}' = \beta\,\mathbf{H}\mathbf{m} \quad \text{with} \quad \mathbf{H} = \mathbf{R} + \mathbf{t}\mathbf{n}^\top/d \qquad (12)$$

where β is an unknown scalar factor, $\mathbf{n}$ is the normal vector to the object plane expressed in $\mathcal{F}_{c_1}$, $\mathbf{m} = (x,\, y,\, 1)$ and $\mathbf{m}' = (x',\, y',\, 1)$ are the homogeneous coordinates of the image points expressed respectively in $\mathcal{F}_{c_1}$ and $\mathcal{F}_{c_2}$ (see Fig. 5). From the value of $\mathbf{H}$, it is possible to determine $\mathbf{R}$, and if $\mathbf{t} \neq \mathbf{0}$, $\mathbf{n}$ and $\mathbf{t}/||\mathbf{t}||$ [10]. In the next paragraph, a new method to determine the partial pose using moment invariants and without computation of the homography matrix is proposed.

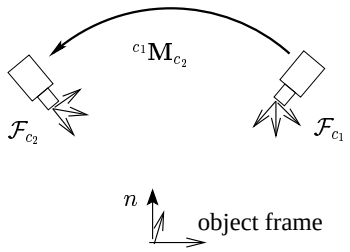

Fig. 5. Partial motion determination using two different images

1) Method: Our goal is to determine the rotation between the two camera poses, as well as the object plane orientation. The idea of the method is similar to that using the exhaustive optimization algorithm. Indeed, we determine the two rotational motions to apply to the camera in the two positions that minimize the error:

$$e_r = ||\mathbf{r}(\alpha, \beta) - \mathbf{r}^*(\alpha', \beta')||$$

where $\mathbf{r}(\alpha, \beta)$ and $\mathbf{r}^*(\alpha', \beta')$ are computed from both images and varying rotational motions. Since $\mathbf{r}$ is invariant to translational motion only when the object is parallel to the image plane, if the translation considered between the two camera poses is not null (i.e. $\mathbf{t} \neq \mathbf{0}$), the obtained rotational motions allow to determine the rotations from the two unknown positions to positions where the camera and object planes are parallel. Otherwise, if $\mathbf{t} = \mathbf{0}$, the object plane orientation cannot be determined. However, the rotation between the two camera poses can be determined from (α, β) and (α', β') by ${}^1\mathbf{R}_2 = {}^1\mathbf{R}_{p(\alpha,\beta)}\,{}^2\mathbf{R}^\top_{p(\alpha',\beta')}$

2) Experimental Results: The images used in the first experiment are given on Figure 6. In that case, the estimated value of the rotational motions necessary to move the camera from its position to a position where the object and the camera planes are parallel corresponds exactly to the real one, which validates our approach. For the images given on Fig. 7, the estimated rotation is given by $\widehat{\mathbf{u}\theta} = (-19.50, -0.23, 0.08)$ while the real one is a rotation of 20^o around x-axis, which proves also the validity of our method.

V. Generalization to the discrete case

The whole results obtained in this work when the object is defined by a dense distribution (4) can be generalized to objects defined by a set of discrete points

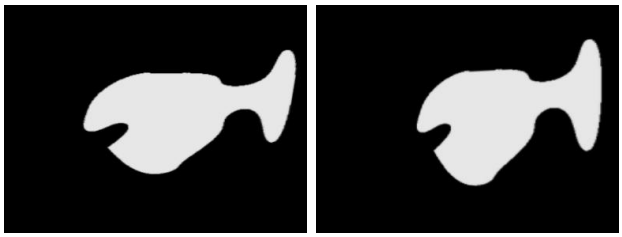

Fig. 6. Images used in free model pose estimation: (a) image obtained after a rotation of 30^o around x-axis with respect to a parallel position, (b) and after a rotation of 30^o around y-axis.

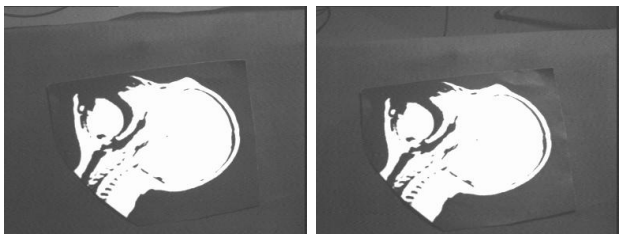

Fig. 7. Images used in free model pose estimation using "brain".

(for example, a set of points of interest extracted using the Harris's detector [4] and tracked using a SSD algorithm [3]). The use of moments instead of points coordinates may solve several problems, such as for instance local minima. Furthermore, the use of moments requires only that the set of points considered in the desired image is the same as in the initial and tracked images (no matching between these points is necessary to compute the moments).

The moments in the discrete case are defined by:

$$m_{ij} = \sum_{h=0}^{N} x_h^i \; y_h^j \tag{13}$$

The definition of the moments in that case is different of(4). However, all obtained results about moment invariants and pose estimation remain true [16], [15]. In the following, we consider the free model pose estimation. The images used are given on Figure 8. The real value of rotation between the two poses of the camera is given on Table II. The estimated value using our method is given on the same table by $\widehat{\mathbf{u}\theta}$. We note that the estimated value is very accurate. Furthermore, the value obtained using an approximation of the homography matrix [10] is given on the same table by $\widehat{\mathbf{u}\theta}'$. This method consists in determining first the homography matrix given by (12). From that matrix, rotation matrix $\mathbf{R}$ is determined. We note that the obtained result is less accurate than that obtained using our approach. This is due to the fact that the orthogonality condition on rotation matrix (i.e. $\mathbf{R}\mathbf{R}^\top = \mathbf{I}$) is not introduced during the estimation of the homography matrix.

$\mathbf{u}\theta$	(-10,	0,	0)
$\widehat{\mathbf{u}\theta}$	(-9.99,	-0.2,	0)
$\widehat{\mathbf{u}\theta}'$	(-12.7,	1.8,	-2.4)

TABLE II

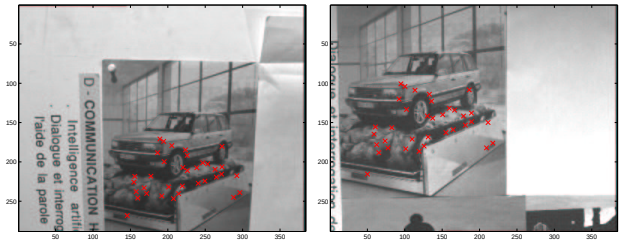

Fig. 8. Images used in free model pose estimation.

VI. Conclusion

In this paper, moment invariants have been used to solve the pose estimation problem of planar objects. A generalization to free model pose estimation has also been given. Moment invariants have been used to decouple the camera dof, which allows the system to have a large convergence domain and to avoid local minima. The experimental results show the validity of the approach. Future works will be devoted to develop similar methods for non planar objects.

References

[1] D.F. Dementhon and L.S Davis. Model-based object pose in 25 lines of code. *Int. Journal of Computer Vision*, 15(1-2):123–141, June 1995.

[2] T. Drummond and R. Cipolla. Real-time tracking of complex structures for visual servoing. In *BMCV*, 1999.

[3] G. Hager and K. Toyoma. The X-vision system: A general-purpose substrate for portable real-time vision applications. *Computer Vision and Image Understanding*, 69(1):23-37, 1997.

[4] C. Harris and M. Stephens. A combined corner and edge detector. *4th Alvey Vision Conf.*, pp. 147-151, 1988.

[5] R. Hartley. In defense of the eight-point algorithm. *IEEE on PAMI*, 19(6):580-593, 1997.

[6] T. S. Huang and O.D Faugeras. Some properties of the E matrix in two-view motion estimation. *IEEE Trans. on PAMI*, 11(12):1310-1312, 1989.

[7] C.P. Jerian and R. Jain. Structure from motion-a critical analysis of methods. *IEEE Trans. on SMC*, 21:572-588, 1991.

[8] Q.-T. Luong and O.D. Faugeras. The fundamental matrix: theory, algorithms, and stability analysis. *IJCV*, 17(1):43-76, 1996.

[9] E. Malis. Improving vision-based control using efficient second-order minimization techniques. *ICRA'04*, New Orleans, April 2004.

[10] E. Malis, F. Chaumette, and S. Boudet. 2 1/2 d visual servoing. *IEEE Trans. on Robotics and Automation*, 15(2):238-250, Apr. 1999.

[11] E. Marchand and F. Chaumette. Virtual visual servoing: a framework for real-time augmented reality. *Eurographics'02*, Saarebrücken, Germany, Sept. 2002.

[12] R. Mukundan. Estimation of quaternion parameters from two dimensional image moment. *Graphical Model and Image Processing*, 54(4):345-350, July 1992.

[13] R. Mukundan and K. R. Ramakrishnan. An iterative solution for object pose parameters using image moments. *Pattern Recognition Letters*, 17:1279-1284, 1996.

[14] R.J. Prokop and A. P. Reeves. A survey of moments based techniques for unoccluded object representation. *Graphical models and Image Processing*, 54(5):438-460, Sep. 1992.

[15] O. Tahri. *Application des moments à l'asservissement visuel et au calcul de pose*. PhD thesis, University of Rennes 1, Rennes, March 2004.

[16] O. Tahri and F. Chaumette. Image moments: Generic descriptors for decoupled image-based visual servo. In *IEEE Int. Conf. on Robotics and Automation, ICRA'04*, volume 2, pages 1185-1190, New Orleans, USA, April 2004.