Visual servoing for a pan and tilt camera with upsampling control

Benoit Louvat Laurent Bonnaud Nicolas Marchand and Gerard Bouvier
INPG Gipsa-lab 961 rue de la houille blanche Saint Martin d’Heres France
Email: benoit.louvat@gipsa-lab.inpg.fr

Abstract—This paper deals with visual servoing for a pan and tilt camera embedded in a drone. Video is transmitted to the ground where images are processed on a PC, and turret controls are sent back to the drone. The objective is to track any fixed object on the ground without knowledge about shape or texture and to keep it centered in the image. In order to achieve this task an algorithm that combines feature-based and global motion estimation is proposed. This algorithm provides a good robustness to very strong video transmission noise and works at a frame rate close to 25 fps. The control of the system is based on a double closed loop, which achieves a fast convergence to the desired position. In order to improve the time response of the visual closed loop we propose an upsampling control law. Upsampling means that orders are sent to the pan and tilt camera before the convergence of the image analysis algorithm. Experimentation in real conditions shows the effectiveness of the proposed scheme.

I. INTRODUCTION

Today robots can perform various tasks. When tasks are simple robots take over from human workers with a high accuracy and productivity. Robots are more and more used in industrial systems, for example in assembly and material handling jobs. When tasks are complex and need an important adaptivity of the context, the lack of their sensor perception do not allow them to give good results. To address this deficiency of perception, considerable researches have been conducted on the link of visual perception and control.

The first research on the linking of visual perception and control are in an open-loop fashion, looking and moving [1], the accuracy of these methods depend strongly on the accuracy of the visual sensor and robot end-effector calibration. An alternative to increase the accuracy is to use direct visual feedback control loop. The methods that use this kind of loop are named visual servoing or visual feedback, the term was introduced by Hill [2] in 1979.

A classification of different visual feedback methods can be find in [3]. Weiss classify it in two major classes : position-based visual servoing and image-based visual servoing. Difference between position-based and image-based is the definition of the coordinate of the error signal. In position-based system the error signal is defined in the 3D cartesian space, whereas in image-based system error is defined directly in the image space. Each class of system (image-based and position-based) are divided into two subclasses : dynamic look-and-move system and direct visual servo system. The dynamic look-and-move system refers to architecture where the visual close loop provide set-points input to the joint-level controller, thus making use of joint feedback to internally stabilize the robot. This kind of system have a main advantage the sample rate of the system does not depend on the sample rate available from the camera. In contrast direct visual servo control system does not provide set-points input, the visual controller computes directly the joint-level input, thus using vision alone to stabilize the robot. The improvement of the power of calculator which allows fast image analysis, have generalized methods based on direct visual servo image-based or position-based visual servoing task. We can address more recent works which are not taken into account by Weiss, as visual 

\[ \frac{3D}{dt} \] [4] based on the motion field in the image or

\[ 2D/2 \] [5] based on an hybrid approach combining image-based visual servoing and position-based visual servoing.

The literature on robotic visual servoing has grown substantially in recent years from its beginnings over three decades ago, and the kinematics of visual control have been well covered. In many respects the performance of visually closed loop robot systems is similar to early force-controlled robot systems in the 1960s. the motion is slow and close to stability limits. Dynamic effects due to the manipulator and machine vision sensor which limit performance are not explicitly addressed. In visual servoing tasks, the control law scheme often neglect the dynamics of actuators and their transfer function are assumed to be a simple proportional gain [6]–[8]. In [9] Papanikolopoulos proposes various controller design, from a PI to a LQG; however he does not propose to take into account dynamics actuators and dynamics of the whole visual close loop.

To achieve such control it is necessary to have accurate dynamic models of the system to be controlled (the robot) and the sensor (the camera and vision system). More recent researches deal with the dynamic modeling of robots and visual servoing systems. The first work on this topics, was made by Corke [10], [11] which proposes a modeling of the system and some controller to achieve the task respecting the dynamics constraint. In [12], [13], Ganglof proposes a general modeling of dynamics with a GPC controller for a 6DOF robot in order to control it for positioning task or profile following in 3D [14]. Other researches address the control of mobile robot, for example the stabilization of an aircraft [15] or the stabilization of a pan and tilt camera for a submarine [16].

Our context deals with visual servoing task for a pan and tilt camera turret embedded in a drone. The visual servoing task which has to be achieve can be resume as follow : Video is
transmitted to the ground where images are processed on a PC, and turret controls are sent back to the drone. The objective is to track any fixed object on the ground. In this context, motion object on the ground can be very fast. To keep it in the image, the system have to be very reactive and neglecting the actuator's dynamics is not suitable. In order to respect this constraint, we design an inner-loop to take into account the actuator's dynamics and to increase the turret's constant time. The most computational cost in visual servoing is due to the image analysis algorithm. An accurate and robust estimation of the position of the target involve an important computational cost. It is in contradiction with the necessity of reactivity of the visual system. All methods presented above have a similarity, they all wait the convergence of the image analysis algorithm estimation to provide set-point to the joint-controller in the case of dynamic look-and-move or to compute the joint-controller level in case of direct visual servo. The constant time of the whole visual servoing system is limited by the image analysis. In order to increase the global reactivity of the system, we present in this paper an other approach to improve the time constant of the visual close loop. Lots of control system use an observer to estimate a state of the system. The estimation provided by the observer is used directly in the control law as soon as the first iteration is done. Respect to this, we use the image analysis algorithm in the visual control law as an observer in classical control theory. We do not wait the convergence of the image analysis algorithm to send orders to the pan and tilt turret. Orders control are sent during the estimation of the target position. Orders are compute by the visual controller using intermediate estimation provided by the algorithm. Theoretically, orders can be sent at each algorithm's iterations.

The remainder of the paper presents in the first section the image analysis algorithm. It is based on global local motion estimation, a description of this algorithm is set out in details in [17]. Then, a presentation of the control law based on two closed-loops, in order to take account of the turret’s dynamics. The next section presents the upsampling control law scheme. Afterwards, results obtained by upsampling control and classical control will be shown. Finally, we will conclude and present some future improvements.

II. TARGET ESTIMATION ALGORITHM

The inter-frame motion has to be estimated in order to find the position of the desired object in the image. Considering that the ground filmed by our drone is a planar surface, two characteristics have to be taken into consideration: the quality of images and the size of objects. So the algorithm needs a high degree of accuracy and a good robustness. For that, we combine the two algorithms, KLT [18] for the feature points extraction and tracking, and RMRm [19] for the estimation of the global motion model. The algorithm breaks down into several steps. A first step of initialization is necessary. It includes the extraction of feature points from the part of image which contains the object selected by the user and the initialization of the support to estimate the global motion (i.e. the window where the parametric model is calculated). This support is smaller than the whole image for computing time reasons. On the other hand, it must be large enough in order to contain a sufficient number of pixels even in case of partial image transmission. In practice, estimation support is a window of 200x200 for an image of 320x240. This size also makes possible to limit the effect of edges. The second step which corresponds to the tracking itself is divided into two parts. The first part is the global motion estimation using RMRm. In [20] a quadratic model is advocated to describe the motion. Actually estimating a quadratic model is too long in computational time to be integrated in a closed control loop. Consequently we choose to consider an affine model to describe the global motion. In fact, the target is fixed, so the global motion can be roughly approximated to the target’s motion which allows a prediction of the position of feature points.

The second part is the tracking of feature points by using KLT. A multiresolution strategy is used to solve this equation. A pyramid of images is built, the original image is divided into several images with different levels of resolution. Each level represents the original image with a resolution divided by 2. The initialization is performed at the lower level. The value of the displacement d of the feature is initialized by the amount of the global motion previously estimated. All the feature points are moved by this amount which allows a refining the research of the target. This process can be assimilated to a feature points prediction. When the minimization is done at this level, the position of the feature point is projected to the upper level and the same scheme is done until the upper resolution level is reached. The great advantage of this initialization by RMRm is the increasing of robustness.

Finally, a last step consists in testing the relevance of the feature points found before updating the control. All the tracked feature points are tested. If the position of a feature is too far from the center of gravity of the group than a threshold, the feature is rejected and the target’s position is recomputed. During the second step, it happens that KLT loses feature points on account of the unconvergence of Newton-Raphston algorithm. In order to solve this problem we fix a threshold which corresponds to the minimum number of feature points to continue the tracking. When this threshold is crossed, new feature points are extracted around the target position from the current image. This process allows tracking target for a longer time. Finally, the adjusted position is sent to the control. The algorithm is summarized on figure 1.

A film taken during a flight [21], shows the behavior of this algorithm embedded in our visual closed loop. It can be seen that the proposed algorithm is robust to bad video transmission. When the image is damaged, features prediction provided by global motion estimation allows the visual control system to continue tracking the target. The processing rate is 20 frames per second with RMRm set to 4 iterations and KLT set to 10 iterations maximum. The PC for vision processing is a Pentium 4 3.2GHz.
The control law of our drone system is designed in order to keep a target in the center of the image (the center of gravity of the part of image) selected by a user. In the present context, the motion of the target in the image can be fast and neglecting the actuators dynamics is not suitable. Hence the control structure is based on two closed loops. The inner loops or dynamics actuators loops are designed to improve the angular speed of the turret. Proportional Integral controllers (PI) provide this angular speed control. Note that, for simplicity, the inner loops are schematized as a single control loop. The outer control loop or visual closed loop is designed using the visual information. It provides the setting points in speed to the inner loops using the error between the desired (centered) and the current position of the target in the image. The delays in the loop due to electronics, transmissions and image processing make the system unstable. In order to solve this problem, we introduce a compensation’s term based on Pade approximation in the outer closed loop. Figure 2 illustrates the control structure proposed to achieve the task. We now present the configuration and design of the two closed loops.

A. Inner-loop

1) Turret’s identification: Camera system provides pan and tilt control through two motors and belt system. In first the definition of a model for pan and tilt motor is necessary to take into account dynamic constraints. Our system does not provide a measure of the output angular speed of motors. It only provides a measure of the angular position using potentiometers. We decide to identify the model using the motor in open loop with order in speed and output measurement (potentiometer) in position. Orders are steps’ speed center on zero speed with a range larger and larger. In order to estimate only the motor’s model without potentiometer, we need the output speed of the motor. Potentiometers are simply an integrator without input control. Hence to find the motor’s speed corresponding to the position measured by potentiometer, we design an observer to reconstruct the motor’s speed. We use the speed observed to estimated the model of motors.

The model motor can be approximated at a three order model, a mechanic time constant (we neglect the inductance of the motor) and two frictions time constant due to belts. Figures 4 and 3 show the results of the identification process with a model of third degrees and a model of first degree. The difference of accuracy between the two model is not really significant. Therefore results allow us to choose a first order model with a static gain of 2. In fact, the the two times constant due to belts can be neglect and the motors’ transfer function can be identified to a simple first order. Then we equate model motor with:

\[
\frac{u}{\omega} = \frac{K_p}{1 + \tau_p}
\]

Where \( u \) is the command tension and \( \omega \) is the angular speed of the motor. Time constant \( \tau \) are estimated to 0.0204 s for pan motor and 0.0272 for tilt motor. \( K_p \) is estimated to 2 for both.

2) Turret’s Dynamic control: Since only the angular positions are measured by potentiometers, observers were developed in order to estimate the current angular speed in pan and tilt. In order to develop an observer to the inner loop, we rewrite the motor model previously estimated in the shape of state space.

\[
\dot{X} = AX + Bu \\
Y = CX
\]
With the state vector $X = [\alpha \ \omega]^T$, $\alpha$ is the angular position and $\omega$ is the angular speed. Matrix $A$, $B$, $C$ are settled to:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & \frac{1}{\tau} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \frac{K}{\tau} \end{bmatrix}, \quad and \quad C = [1 \ 0]^T$$

The speed observator can be written as follow:

$$\dot{\hat{X}} = AX + Bu + MC(\tilde{X} - X)$$

Where $\tilde{X}$ is the estimation of the state vector $X$ and $M$ the matrix placement observator. In order to validate our observator we tested the value of the speed estimated during a position’s regulation of the system. The test are made on a closed loop regulation’s position for drift of operating point reason. Indeed variation around zero speed carry quickly motors along the stop. Results on Figures 5 and 6 show the estimate speed during two steps’ position order. The speed is maximum when the system pursuit the desired position and null when the position is reached.

The observator is validated by the results showed on figures 5 and 6. The last step of the design of the inner-loop is to tune PIs controller to have a closed loop time constant of 10ms. We choose this tuning in order to have a faster convergence to the desired speed and insure the stability of the system. We will see in section 4 that we need a fast system to used upsampling control. The whole inner-loop is embedded in the drone on the turret’s microchip.

### B. Outer-loop

The outer-loop is a classical image-based visual closed loop. It permits to center and keep centered the target in the image. Thanks the current position of the target is estimated by the motion estimation algorithm. We have design a standard control law to realize the regulation of the target to the center of the image. The control scheme used is the image-based visual servoing control law presented in [22]. It can be resume by the regulation to zero of the error $e$ between the position of the target $s = (x, y)^T$ and the desired position $s^* = (0, 0)^T$. Due to the fact that $s^*$ represents the center of image, $s$ can be viewed directly at the error vector. The visual servoing goal is then to bring and maintain this error to zero by controlling the camera pan and tilt. To design the control law, we use
the relation between the temporal variation \( s \) and the camera motion. As this motion is restricted to rotational velocity \( \Omega_x \) (pan) and \( \Omega_y \) (tilt) and the image feature is a point representing the center of gravity, we can write:

\[ s = L \begin{pmatrix} \Omega_x \\ \Omega_y \end{pmatrix} \]

with

\[ L = \begin{bmatrix} xy & (1 - x^2) \\ (1 + y^2) & -xy \end{bmatrix} \]

where \( L \) is the image Jacobian matrix. Specifying exponential decay of the error with gain \( \lambda (\dot{s} = \lambda s) \), the control law is given by

\[ \begin{pmatrix} \Omega_x \\ \Omega_y \end{pmatrix} = \lambda L^{-1} s \]

A simple proportional gain is not sufficient to insure convergence because of the delays in the loop due to electronics, transmissions and image processing. A prediction term based on a Padé approximation of the delay is added to remove the oscillations introduced by the delays. Let be \( \tau \) the delay and \( e(t) \) the error at time \( t \), by a Padé approximation we can predicted the error at time \( t + \tau \) as follow:

\[ e(t + \tau) = e(t) + \frac{\partial e}{\partial t} \tau + \frac{\partial^2 e}{\partial t^2} \tau^2 \]

In our case \( e \) is equal to \( s \) and the prediction is directly use in the control law. Note that contrary to classical automatic control theory, delays and time periods are linked to transmission delay and the computational cost of the image analysis and control law. Hence delays and time periods are not constant. These variations are explicitly taken into account in the control law. The outer-loop process on the ground station. It is implemented on the same PC where the image analysis is done.

IV. UPSAMPLING CONTROL

The idea of the upsampling control is to send order to the inner-loop before the image analysis algorithm reach the final estimation of the position’s target in order to reduce the visual closed loop time constant. The dynamic of the turrets is settled by the inner loop at a delay of 10ms, the delay of the whole visual servoing system, impose by the outer-loop is 50ms (due to image analysis and electronic transmissions), if we consider that the algorithm has an exponential convergence to the position’s target, we can send control to the turret five times more often than the classical control law without upsampling. The image algorithm process on a pyramidal structure to estimate the position of the turret. We decide to send order to the inner-loop every times that the algorithm finished its estimation on a pyramid level. Due to technical limitations, saturation of the buffer of the onboard turret’s microchip, the upsampling control law are tuned to send order only three times during an image processing. There are also the limitation of the dynamic response of our turret, we have tuned it to have a closed loop time constant of 10ms, so we cannot control the turret faster without unstabilize the whole system. The Intervals of upsampling are f to 20 ms.

A resume of the upsampling control law is illustrated on the figure 7. Figure shows the pyramidal structure for an image proceeded by the algorithm presented section 2 and the
moment where outer-loop set-point to inner-loop. Pyramidal structure have 4 levels for global movement estimation and 2 level for local refinement. During the global estimation orders are sent two times: once after the first initialization of the constant model and once after the third pyramid level. Then a final order is sent after the local refinement of the target’s position. Actually control during the convergence of the global estimation of the image analysis algorithm is made in open loop. Thanks the control law uses an image-based approach, error are stabilized only in the image, there is no feedback measure. However, we can assume that the displacement of the turret and consequently the displacement of the target in the image is not large. So estimation of error and delay taken previously stay correct. The last order sent to the inner-loop is the classical order in closed loop.

To resume, the upsampling control law works two times in open loop and once in closed loop between two images. It can be assimilate to an increasing of the global sample rate of the visual loop (i.e. without increasing the image analysis algorithm frame rate).

In order to offset the delay at each time that orders are sent, we have an adaptive delay compensation. Delay are measured dynamically between each order and add to the static transmission delay. After, we estimate the position of the target with the pade approximation presented previously in section 3.

V. RESULTS

During a fly, tests are not reproducible. Due to this, control law with and without upsampling are not comparable using the real visual servoing task. That is why all validation tests are made in laboratory, without the turret’s is embedded on the drone. The camera and the turret were dismantled,
and to reproduce the real conditions, we used wireless communication to receive images and to send orders to the turret. Furthermore, difference between the two control laws is difficult to be shown on video sequence film by our drone. The only feedback measure of the target’s position and therefore the turret’s displacement is provided by the image analysis algorithm. Due to this, we use very simple kind of image to make tests in order to compare upsampling control law and classical control law. Figure 9 shows the image test, a checkerboard. This simplicity permits to insure a good estimation of the target’s position by the image analysis algorithm. We can focus only on the behavior of the control scheme proposed. To compare the two different control law, we draw the response of the system to a step with respect to pan and tilt axis. The step is equal to the size of a square’s side. In order to make a step in pan and tilt axis, we click on the upper or lower corner of a square to center it in the image, then we click a second time on the opposite corner (upper or lower depends on the first click). Figures 9 and 10 resume the testing process. All results presented in this section are made with this process. All displacements of the turret are measured in the image, it means that the ordinate axis is in pixels, the x axis is time (second). In order to avoid outliers response and to represent realistic behavior of our system, all results presented above are made on a mean of five identical steps displacement. The same gain are tuned to the proportional controller of the visual closed loop. Figure 11 shows the result for a step of 55 pixels (more or less 2 pixels depending on the first click) in pan and tilt axis. The propose upsampling control law is faster than the classical control law. The improvement of the time constant of the system is close to 150ms. As we presented in section 3, we can see the delay of the system on each figure, it corresponds to the flat part at the beginning of each draw. Due to the compensation of the delay, prediction of the target’s position is above the real target’s position hence the control law order the turret to the opposite way and the final target’s position in the image not correspond to the real position. Steady-state errors can be shown on all results. These errors between the desired position and the real position measured after the stabilization of the system is due to turret’s motors frictions. Motors system have important frictions at low speed. Consequently when orders are in a range corresponding to a low speed, friction prevent turret’s displacement.

Figure 12 and 13 show respectively the result for a step only in pan axis and for a step only in tilt axis. we can see the same behavior of the system. The upsampling control law is faster than the classical one. Figure 8 shows the response of the system to a step of 60 pixels with and without upsampling for different video rate of the image analysis algorithm. Usampling is always faster than the classical control law. This is an important characteristic, it can be used to spend more time in image analysis, to improve the robustness or the accuracy of the system without penalize the time constant of the visual closed loop.

VI. Conclusion

The approach proposed in this article improves the performance of our visual servoing task. The upsampling control law scheme improves the time constant visual closed loop. This improvement allows our visual task to track fast motion target and also allows to use more complex image analysis algorithm without loosing in global system time constant. This control law permit also to increase the global sample rate of the visual close-loop without increasing the image analysis algorithm frame rate.

Future works may concern drone’s dynamics, in order to taking into account the whole dynamic of the visual servoing task. We can also develop a better estimation of the delay, to avoid the phenomenon of rebond. Another important improvement concern the compensation of friction in order to discard the steady-state errors. Finally, we can improve the implementation of our upsampling control law by taking into account the target’s displacement between two upsampled orders. For example an estimation of this displacement can be done before setting-point to the inner-loop.
Fig. 13. Response to pan step

REFERENCES
