DESIGN AND IMPLEMENTATION OF VISUAL FEEDBACK FOR AN ACTIVE TRACKING

Tomasz Żabiński, Tomasz Grygiel, Bogdan Kwolek
Rzeszów University of Technology, W. Pola 2, 35-959 Rzeszów, Poland
tomz, bkwolek@prz-rzeszow.pl

Abstract Active visual tracking is used to direct the attention of the camera to an object and maintain it in the camera's field of view. A steered camera is used to decrease the relative motion of the target in the image plane. This leads to better performance of the mean-shift based tracking algorithm, which requires an overlap between the object undergoing tracking in current and previous frame.

Keywords: Active vision, vision-based tracking, visual-servoing

1. Introduction

Active camera-based systems are often used to track a target. A closed-loop control of the camera can increase the performance of the tracking system. The aim of the camera in many active tracking systems is to keep the target in its field of view. In more sophisticated systems the goal of the camera is to keep the tracked object at a specific position in the image. A steered camera can then be used to decrease the relative motion of the target in the image plane. This in turn can lead to better performance of deterministic search-based tracking algorithms, which require an overlap between the object undergoing tracking in current and previous frame.

There are several limitations for active tracking systems. The first is motor speed bandwidth because for fast moving targets the motors may not have the speed and power capabilities to maintain the target at desired location in the image. The second is the visual controller. That is, a controller that is not properly chosen and tuned may produce steady-state errors, which for the small height and width in the image of object undergoing tracking can lead to tracking failure. This paper describes a design of a visual feedback controller as well as servo motor that address the mentioned limitations in order to track object with active camera. A simple robot is used to set the desired location of the target with respect to the camera. The tracking is done using the mean-shift algorithm [1].

2. Mean-shift based tracking

In a mean shift based tracking algorithm we can distinguish three stages [3]. Let $\{q_u\}_{u=1}^m$ and $\{p_u(y)\}_{u=1}^m$ be weighted color histograms of the target to be tracked and target candidate. They are calculated in the first stage as follows:

$$q_u = C_q \sum_{j=1}^{n_q} k\left(\|\frac{x_j}{h_q}\|^2\right) \delta(c(x_j) - u) \tag{1a}$$

$$p_u(y) = C_p \sum_{j=1}^{n_p} k\left(\|\frac{y - x_j}{h_p}\|^2\right) \delta(c(x_j) - u) \tag{1b}$$

where the function $c : \Re^2 \rightarrow \{1, ..., m\}$ associates the value of pixel at location x_j to the bin number, h_q and h_p are kernel bandwidths, k is a kernel profile and C_q and C_p are normalization factors. In the second stage a Bhattacharyya coefficient based similarity measure is calculated. It is defined as follows: $\rho(y) = \rho[p_u(y), q_u] = \sum_{u=1}^m \sqrt{p_u(y), q_u}$. Using Taylor expansion it can be expressed as follows:

$$\rho[p_y(y), q_u] = 0.5 \sum_{u=1}^{m} \sqrt{p_u(y_0) q_u} + 0.5\, C_p \sum_{i=1}^{n_p} w_i k\left(\|\frac{y - x_i}{h_p}\|^2\right) \tag{2}$$

where $w_i = \sum_{u=1}^m [\delta(c(x_i) - u] \sqrt{q_u / p_u(y_0)}$. In the third stage an iterative procedure is used to maximize the similarity measure starting from the previous target position y_0. A new location y_1 is calculated according to:

$$y_1 = \frac{\sum_{i=1}^{n_p} x_i w_i g(\|\frac{y_0 - x_i}{h_p}\|^2)}{\sum_{i=1}^{n_p} w_i g(\|\frac{y_0 - x_i}{h_p}\|^2)} \tag{3}$$

and converges in a few iterations [2].

3. Object tracking with pan-tilt camera

In this section we present our visual controllers which produce the reference values for servo motors. The goal is to find controls to nullify the object position error to the desired position in the image. A PID controller was used for the purpose of obtaining the strong convergence properties of such an error. In our approach this error tends during tracking to be in an arbitrary small neighborhood of the desired position. The proposed control is continuous. The PID controller with a filtering of differential part is described as follows [4]: $K(s) = k_p \left(1 + \frac{1}{T_i s} + \frac{T_d s}{\frac{T_d}{D} s + 1}\right)$. The proportional term of the PID controller provides an output that is a function of the immediate position error. The integral term of the PID controller accumulates successive position errors determined during each control iteration and improves the low frequency open-loop

gain of the control system. The effect of the integral term is to reduce small steady-state position errors. The differential term of the PID controller is a function of the difference in error between the current controller update period and the previous one. The differential term improves the high frequency open-loop response of the control system.

The camera that is steered by a visual controller can be modeled as an inertial object with delay: $\frac{k_c}{T_c s+1} \exp(-\tau_c s)$. The parameters of this transfer function were identified through a registration of location in the image of the object, which had been observed by rotating camera in the response for a forced angle. The following values were obtained: $T_c = 0.17$ sec., $\tau_c = 0.1$ sec., $k_c = 3.5$ and $k_c = 3.2$ for the pan and tilt axis, respectively. For the assumed settling time $t_r = 1$ sec., phase margin $= 64.6^o$, and $D = 2$ the following PID parameters were obtained in the process of controller tunning [5][6]: $T_i = 0.14$, $T_d = 0.03$, $k_p = 0.16$ and $k_p = 0.18$ for pan and tilt, respectively.

4. Servo motor of the camera's pan and tilt

The design of the servo motor and camera's motion control has been done using Matlab/Simulink. The real-time control of the motors has been realized on the basis of RT-CON software [8], which provides Real-Time Workshop toolbox (RTW) [7] with additional functions. On the basis of Simulink-based control block diagram, RTW generates the source code in C, which is then linked with RT-CON functions to generate the executable code for the designed real-time application. When the application runs and controls the motion of the motors via USB-based card, the user can visualize as well as modify the parameters of the controllers without a recompilation of the application.

The camera is driven by two DC motors. Each motor is equipped with an encoder that provides the position signal for the feedback. The method called PWM (Pulse Width Modulation) is utilized to provide the drive signal for the tilt motor. In this method, the current of the motor is converted into a controlled pulse width that is proportional to the amplitude of the sine wave. The motor controlled in such a way can be modeled with the following transfer function: $G_t(s) = \frac{k}{s(Ts+1)}$, where k is gain, and T is time constant. They were determined on the basis of step response by doing the open-loop experiment [5]. The current-based motor control has been used for steering the camera's pan. The object was modeled by double integral transfer function $G_p(s) = \frac{k}{s^2}$, where the gain k has been identified using an open-loop step response.

A discrete PID algorithm is used to control the motors since it is widely used in industrial applications and is easy to implement. The discrete PID is described as: $H(z) = k_p + k_i \frac{z\Delta}{z-1} + k_d \frac{z-1}{z\Delta}$. The parameters of the controllers were determined on the basis of a method described in [6]. For assumed settling time t_r and Δ, the following settings for the pan axis were determined in process of controller tuning: $k_p = \frac{216}{k \cdot t_r^2}, k_i = \frac{432}{k \cdot t_r^3}, k_d = \frac{27}{k \cdot t_r}$, where k denotes

the object gain. The parameters have been derived for a continuous system and they are employed in our discrete controller since $t_r > 200 * \Delta$. The step response of the pan axis for $t_r = 1$ sec., $\Delta = 4$ ms is depicted in Fig. 1a. The controller parameters for tilt axis have been determined from the following formulas: $k_p = \frac{12}{k}\frac{t_r+3T}{t_r^2}, k_i = \frac{36}{k \cdot t_r^2}, k_d = \frac{12T}{k \cdot t_r}$. The step response of the tilt axis for $t_r = 0.3$ sec and $\Delta = 4$ ms is depicted in Fig. 1b. The depicted step responses were obtained using set-point filters: $\frac{1-\alpha}{z-\alpha}$, where $\alpha = \exp(-\frac{4}{t_r}\Delta)$ for pan and $\alpha = \exp(-\frac{3}{t_r}\Delta)$ for tilt, respectively. Such filters are used to eliminate the overshoot without extension of the settling time.

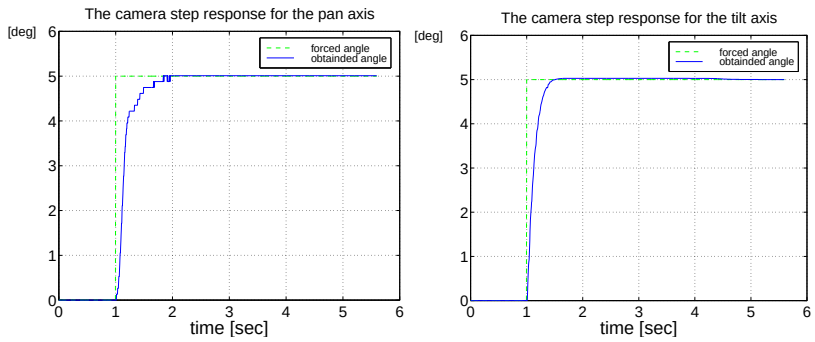

Figure 1. The camera step response, pan axis (left), tilt axis (right)

5. Experiments

All experiments were carried out in an experimental setup that is graphically shown in Fig. 2. It consists of hand-made robot as well as camera head. The robot as well as the head are equipped with DC motors with digital encoders. An USB card that is supplied with RT-CON software steers the motors as well as acquires the signals from the encoders. The pose of both the robot and the camera were utilized in estimation of tracking errors and evaluation of tracking performance. The tracking algorithm runs at 320x240 image resolution at frame rates of 10 Hz. It was implemented using C/C++ language. The kernel radius is 10 pix.

The performance of the proposed active tracking has been evaluated through controlling the robot with a target in the gripper and simultaneously conducting the active tracking. The aim of the camera was to keep the target at specific location at the image. The active tracking error is a difference between the position value estimated by the tracker and a desired target location at the image. At the beginning of experiments we tested the PID visual controller in a system configuration that is described in Section 3. The active tracking error is depicted in Fig. 3a. The dashed curve represents the target location, which was determined for the horizontal axis by the tracking algorithm operating on

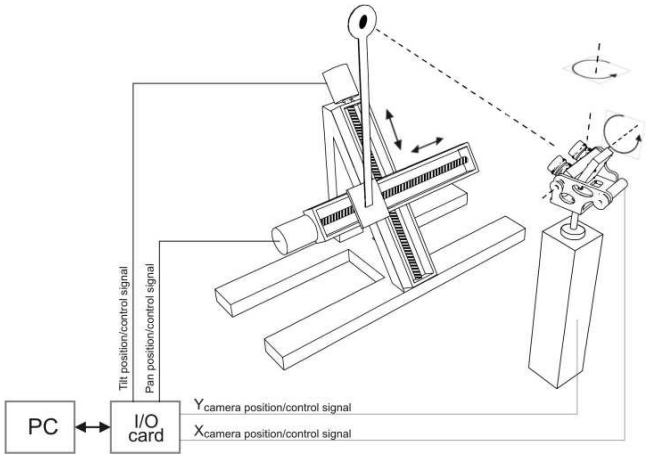

Figure 2. The setup for performance evaluation of active tracking

images acquired from the fixed camera. In the discussed experiment the gripper moved with similar motion in the vertical axis. The ability of the visual controller to nullify active tracking errors for the tilt axis is comparable.

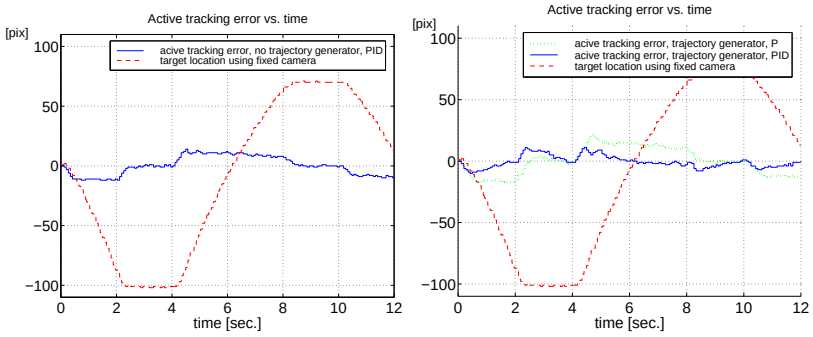

Figure 3. Active tracking errors, no trajectory generator (a), with trajectory generator (b)

In order to decrease the tracking error of the visual controller an additional trajectory generator was added to the system. That means that the camera's transfer function is as follows: $\frac{k_c}{s(T_c s+1)} \exp(-\tau_c s)$, where the parameters k_c, T_c and τ_c are identical with the parameters given in Section 3. Because the object has an integral part that is responsible for nullifying the steady-state error, the P and PD controllers can be utilized to steer the camera apart from the PID controller. The active tracking error is depicted in Fig. 3b. The best fixation of the target at desired location was achieved in a configuration with the trajectory generator and the PID visual controller. The active tracking errors arise mainly due to the delayed output in the control system.

The visual controller as well as servo motor were designed and then validated in Matlab/Simulink/RT-CON environment. The parameters we obtained in rapid prototyping will be used in tuning and then implementing the servo motor in a DSP processor. In such an implementation a shorter sampling time should lead to better tracking results. In current implementation the Matlab/Simulink and the tracker run as concurrent processes that communicate via shared memory. The operating system Windows XP running the prepared software is not true real-time system. This has also an influence on the tracking errors because the active camera based vision system consists of four controllers that run concurrently and the only support for real-time provides Simulink.

Experimental results, where a person moves facing the camera and active camera keeps his/her face at desired position at the image demonstrated the feasibility of active tracking. Thanks to active tracking the person remains in the camera's field of view in a larger area in comparison to a fixed camera.

6. Conclusions

The robustness of the developed visual controller, which results in accurate tracking, was confirmed by experiments. It is found that such visual controller can be designed easily using Matlab/Simulink and RTW. The system keeps the target at the desired location in the image. When the track is lost the tracking can be reinitialized by a searching for the target in a small neighborhood of the desired location. A visual feedback structure consisting of trajectory generator and a PID controller guaranties the best active tracking performance.

References

[1] Comaniciu D., V. Ramesh and P. Meer (2000). Real-time Tracking of Non-rigid Objects Using Mean Shift, In Proc. of IEEE Conf. on Comp. Vision and Pat. Rec., 142–149.

[2] Comaniciu D. and V. Ramesh (2000). Robust Detection and Tracking of Human Faces with an Active Camera, IEEE Int. Workshop on Visual Surveillance, Dublin, 11–18.

[3] Fritsch J., J. Schmidt and B. Kwolek (2006). Kernel Particle Filter for Real-Time 3D Body Tracking in Monocular Color Images, IEEE Int. Conf. on Face and Gesture Rec., Southampton, UK, IEEE Computer Society Press, 567–572.

[4] Lewis R. W. (1995). Programming Industrial Control Systems Using IEC 1131-3, IEE London.

[5] Trybus L. (2002). Root Locus and Frequency Designs of Industrial PID - a tutorial, IBS PAN, Warsaw.

[6] Zabinski T. (2006). Control of Mechatronics Systems in Real-time, PhD thesis, Cracow.

[7] Matlab User Manual (2004). The MathWorks, Inc.

[8] RT-CON User Manual (2006). Inteco.