Visual Servoing under Changing Illumination Conditions

Geert De Cubber, Hichem Sahli
Vrije Universiteit Brussel (VUB)
Department ETRO - IRIS
Pleinlaan 2, B-1040 Brussels Belgium
{gdcubber, hsahli}@etro.vub.ac.be

Eric Colon, Yvan Baudoin
Royal Military Academy (RMA)
Department of Mechanics
Av. de la Renaissance 30, B-1000 Brussels Belgium
{eric.colon, yvan.baudoin}@mapp.rma.ac.be

Abstract

Visual servoing, or the control of motion on the basis of image analysis in a closed loop, is more and more recognized as an important tool in modern robotics. In this paper, we present a new model-driven approach to derive a description of the motion of a target object. This method can be subdivided into an illumination invariant target detection stage and a servoing process which uses an adaptive Kalman filter to update the model of the nonlinear system. This technique can be applied to any pan-tilt-zoom camera mounted on a mobile vehicle as well as to a static camera tracking moving environmental features.

1 Introduction

The implementation of a system capable of performing visual servoing in everyday environments requires careful consideration of the mechanical, control and vision issues involved in the closed-loop sensing system. The primary elements are the detection of objects of interest moving in the scene and their subsequent analysis during tracking over time. Mechanically, this requires a pan-tilt camera platform. The visual servoing approach is based on an information feedback loop, which determines an error vector defined in the vision space. This vector is updated after every image acquisition. In a target tracking scheme, the error vector is defined as a measure, at a given time, of the distance in image coordinates between the target position and the image center. This error serves to determine the control parameters of the pan-tilt platform (camera). Several research works have been done in this area, among which we can cite the work of Yoshimi and Allen [13] for target tracking and object alignment. In their approach the visual servoing is calibrated during operation using dedicated controlled motion of the robots end-effector. Zhang [14] and Corke [1] use calibrated cameras, with an initialization phase for the definition of a dynamic model leading to a predictive controller. In our approach, an online identification method is developed to estimate the target system dynamic model and this model is used in a Kalman filter for tracking.

The proposed scheme consists of a 2-phase process. The first phase deals with target detection. One of the major problems arising here is the effect of an ever-changing illumination, as a change in illumination will also change the perceived colors - or more generally the perceived image - of the environment. The presented classification algorithm doesn't take into account any other parameters (eg. shape or texture) than the color attributes like other authors have done [10], so illumination changes risk to be a problem. To counter this, we developed a color constancy approach to improve the classification capabilities of the color target-tracking algorithm. Color constancy, as defined in [8], is the ability to recover a surface description of color, independent of the illumination. Numerous attempts have already been made to solve this problem. Land and McCaan were the first to tackle the problem with their retinex theory [7]. Others relied on finite-dimensional linear models [2], while also neural nets have been proposed to solve the issue [3]. However, these techniques typically require hours of calculation time to process one non-synthetic image, making them totally unfit for real-time and real-world vision tasks, as in the field of robotics. Here, we propose a color constancy technique used for real-time target identification under varying illumination conditions. In the second phase, the one of the visual servoing, the motion model of the target object is retrieved. This movement is not known a-priori and the perspective projection relationship is a nonlinear one, so we have a nonlinear time-variant system. We approximate this system as a linear time variant one and use an observer-based full-state feedback control to implement the tracking function. From the online identification we can solve the system-modeling problem. The method of the observer-based full-state feedback control guarantees the system stability. The camera control exploits the detection in conjunction with an affine fit between two consecutive images. After the affine fit has been made, the camera

control parameters are estimated. To make the visual control loop compatible with the real-time constraint, a windowing technique was used for the image processing task: only a window around the detected object is processed. An Extended Kalman Filter is then used to predict the future size and position of the window in the image plane.
The rest of this paper is organized as follows: in section 2, the color-based classification process is described, while section 3 explains the used camera control strategy. The paper ends by showing some experimental results and drawing some conclusions about the presented work.

2 Illumination invariant classification

2.1 Modellization

2.1.1 The color reflection model

The main problem for the correct interpretation of a camera image is that the measured intensities are function of a large number of parameters and most of them cannot be retrieved due to their strong interconnectivity. The color of an object in the image must be considered as an appearance rather than as a real material property. Nevertheless, color can be used to identify objects as long as the parameters which influence the formation of the perceived color are taken into account. To do so, we make use of the dichromatic reflection model, which was first introduced by Shafer in [11]:

$$\begin{aligned}\rho_c &= k_b . \int_\lambda e(\lambda).f_c(\lambda).r_b(\lambda).d\lambda \\ &\quad + k_s . \int_\lambda e(\lambda).f_c(\lambda).r_s(\lambda).d\lambda \end{aligned} \quad (1)$$

Where ρ_c is the measured intensity of channel c, $e(\lambda)$ is the normalized light spectrum, $f_c(\lambda)$ is the c^{th} channel sensor response function, $r_b(\lambda)$ is the body reflectance function, $r_s(\lambda)$ is the surface reflectance function, k_b is the attenuation factor for the body reflectance and k_s is the surface reflectance attenuation factor.

2.1.2 Color spaces

Among the different color spaces, our choice went out to the l_1-l_2-l_3-space, a color space which was originally introduced by Gevers and Smeulders in [4]. It poses an attractive alternative to the HSI space due to its computational simplicity. The space can be formulated as follows:

$$\begin{aligned} l_1 &= \frac{|R-G|}{|R-G|+|R-B|+|G-B|} \\ l_2 &= \frac{|R-B|}{|R-G|+|R-B|+|G-B|} \end{aligned} \quad (2)$$

$$l_3 = \frac{|G-B|}{|R-G|+|R-B|+|G-B|}$$

In [5], Gevers and Smeulders prove that according to the dichromatic reflection theory, this space is invariant to highlights, viewing direction, surface orientation and illumination direction. This means that we can work with a simplified form of equation 1:

$$H_{l_1-l_2-l_3}(x,t) = \int_\lambda e(\lambda,t).f_c(\lambda).r_b(\lambda,x).d\lambda \quad (3)$$

2.1.3 Discretization

Equation 3 can be discretized by sampling over a number of wavelength bands. We chose to use a finite dimensional linear model with a limited amount of parameters and using 10 basis functions:

$$\begin{aligned} e(\lambda,t) &= B_e.q_e \\ r_b(\lambda,x) &= B_r.q_r \end{aligned} \quad (4)$$

The columns of the $N \times N_e$ B_e matrix and those of the $N \times N_r$ B_r matrix represent the basis functions for the light and the reflectance spectrum respectively. The N_e element q_e vector and the N_r element q_r vector describe respectively the illuminant and the body reflectance spectrum. If $D(f_c)$ is the $N \times N$ diagonal matrix with f_c as diagonal elements, we get by inserting equations 4 in equation 3:

$$h_c = q_e^T.B_e^T.D(f_c).B_r.q_r \quad (5)$$

The problem with this representation is that the basis and sensor sensitivity functions are not well known. To avoid this difficulty, we use an approach similar to the one described in [12], which introduced a lighting and reflectance matrix, parameterized using $4 \times N_e$ variables in a manner independent of basis functions and sensitivity functions. The idea is to write the vector $B_e^T.D(f_c).B_r.q_r$ as σ_c, which is an alternative descriptive function for the body reflectance function and which can be used to discriminate between observed materials. This leads to a general equation:

$$h^T = q_e^T.\sigma \quad (6)$$

With h^T the color triplet in the l_1-l_2-l_3 color-space and σ an $N_e \times 3$ matrix holding all the reflection characteristics independently of the illumination. This matrix needs to be estimated and based upon this estimate the classification process can be performed.

2.2 Bayesian Color Classification

2.2.1 Learning

In a learning phase, the algorithm learns the reflection characteristics of the object to be tracked. Small patches of images are accumulated over time while the material in question is subjected to a varying illumination. All intensity

measurements h are combined in an $f \times 3.p$ color measurement matrix H, while p is the number of pixels in the scene patch and f the number of frames sampled. If we sample for long enough, then eventually f will grow larger than p and the light spectrum matrix Q and the reflection characteristics matrix S can be recovered by applying singular value decomposition on H, while $H = Q.S$.

$$H = \begin{pmatrix} h(x_1,t_1)^T & ... & h(x_p,t_1)^T \\ ... & ... & ... \\ h(x_1,t_f)^T & ... & h(x_p,t_f)^T \end{pmatrix}$$
$$Q = \begin{bmatrix} q(t_1)^T & ... & q(t_f)^T \end{bmatrix}^T \quad (7)$$
$$S = \begin{bmatrix} \sigma(x_1) & ... & \sigma(x_p) \end{bmatrix}$$

At this moment, $p(q_e|l)$ or the light spectrum distribution if the illuminant l is known, can be calculated. This can be done because Q is independent of the material. We use an Expectation Maximization (EM) clustering method to derive the reflection distributions. This algorithm applies multivariate Gaussian mixture modeling with an unknown number of mixture components, which makes the classification very flexible. The EM algorithm adds clusters until the added model compliance becomes lower than a preset treshold. As the log-likelihood for adding a new cluster decreases drastically once there are more clusters than illumination conditions, the algorithm generally succeeds in discriminating the illumination conditions and creates clusters accordingly. The result of this calculation is an $N_{LS} \times N_e$ light spectrum matrix L, with N_{LS} the number of illuminant spectra distinguished:

$$L = \begin{bmatrix} q_e^T(1) & ... & q_e^T(n) & ... & q_e^T(N_{LS}) \end{bmatrix}^T \quad (8)$$

Together with the calculation of L, the nominal color for each of the clustered lighting conditions is calculated and stored in an $N_{LS} \times 3$ color measurement matrix H_N. With this knowledge, we can calculate the inverse of the $N_e \times 3$ reflectance spectrum matrix R:

$$R^{-1} = H_N^{-1}.L \quad (9)$$

2.2.2 Pixel Classification

Having the reflectance spectrum of the target object and the obtained illuminant spectra corresponding to different lighting conditions, we can correctly classify newly presented pixels as belonging to the target object or not, while keeping track of newly arising lighting conditions. We present a Bayesian solution to solve these problems. New scene properties are brought into the model based upon the Maximum A Posteriori (MAP) estimate of these parameters given the color measurements. When applying this classification, we search for the conditions that maximize $p(o = o_{Target}, l, q_e, \sigma|h)$ for any values of the lighting condition l, the illuminant spectrum q_e and the reflectance spectrum of the target object σ, given the color measurement triplet h.

$$[\hat{o}, \hat{l}, \hat{q_e}] = \underset{[l,q_e]}{argmax}\, p(o, l, q_e, \sigma|\hat{h}) \quad (10)$$

Using Bayes' rule, it can be shown that:

$$p(o, l, q_e, \sigma|\hat{h}) \propto p(\hat{h}|q_e, \sigma).p(q_e|l).p(l).p(o) \quad (11)$$

To calculate $p(\hat{h}|q_e, \sigma)$, we suppose that the measurements are corrupted by Gaussian noise:

$$p(\widehat{h}|q_e, \sigma) = \left(\frac{2.\pi}{|\Sigma_h|}\right)^{-\frac{3}{2}} .e^{-\left\|\widehat{h}^T - q_e^T.\sigma\right\|_{\Sigma_h}} \quad (12)$$

Where Σ_h is the measurement covariance matrix, $|\quad|$ denotes the determinant and $\|\quad\|_{\Sigma_h}$ is the Mahalanobis distance: $\|a\|_\Sigma = a^T.\Sigma^{-1}.a$

The measurement covariance matrix is calculated together with the color measurement itself. To calculate the factor in the exponent, we record the nominal color values h_N of the perceived illuminants and these values are used to calculate the Mahalanobis distance to the current color triplet.

The second factor in equation 11 represents the prior probability density of observing a certain illuminant spectrum q_e, given the lighting condition l. This is calculated during the Expectation Maximization phase of the learning process.

The third factor in equation 11 $p(l)$ describes the prior probability of observing a certain illumination condition on a given point in the scene. There is no a priori knowledge about this, yet over time, it is possible to build up some knowledge about the different lighting situations at different points in the scene and this information can be used to derive a probability for the occurrence of lighting conditions in novel scenes. To do this, an illumination map of the surroundings of the target object is recorded. The values recorded in this map represent for each of the different possible illumination conditions, the probability that they would occur. These probabilities are calculated during the classification process using a voting system: a positive classification for a pixel given a lighting condition increases the probability for this lighting condition at this pixel position, while decreasing all other probabilities.

The last factor in equation 11, $p(o)$, representing the prior probability of observing the target object in the scene, is estimated by dividing the number of pixels belonging to the target object, estimated at the previous time-instance, by the total number of pixels in the image window.

Using these considerations, the pixel classification procedure calculates the probability for each pixel and labels the pixel as belonging to the target object or not based upon the

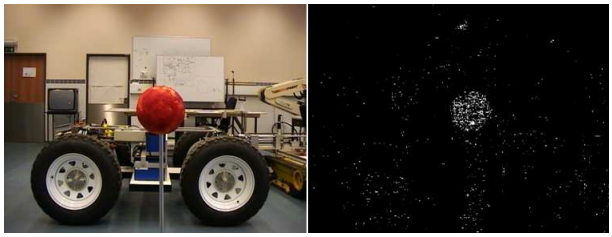

Figure 1. Image & Probability Distribution

result. Figure 1 shows an example of a probability distribution for object presence calculated during the pixel classification process. The circular target object can clearly be identified when observing this distribution, while the outlier pixels can be considered as false classifications. Using this classification theorem, the pixel classification is no longer performed directly based upon the pixels color value, as is classically done, but based upon the derived reflection characteristics, which makes the detection process very robust. This can also be observed by analyzing figure 2 which represents the unclassified pixels in grey and the classified pix-

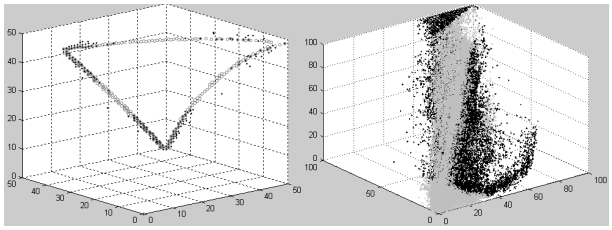

Figure 2. Classification in $l_1, l_2, l_3 \& RGB$ space

els in black, both in the $l_1 - l_2 - l_3$ and in the RGB-space. This figure shows that the applied classification strategy allows a large flexibility in the definition of the target objects color domain, as the classified pixels account for a considerable volume in both of the color spaces.

2.2.3 Model Updating

During the actual tracking phase, the illumination model is continually updated using Bayesian reasoning. In this model updating stage, estimates for new lighting conditions and their corresponding illuminant spectra are calculated. It is this procedure that ensures the adaptive nature of the pixel classification process within the general target-tracking program. The philosophy of this procedure is that we take a small patch from the target object (shown on figure 9 as the small square), try to recover the spectrum of the illuminant shining on this part of the target object and update our model if necessary. So, the first step in this process is to obtain a patch from the target object. For this, we cannot rely on the pixel classification process to tell us where the ball is, as in this case no new information would be added to the existing illumination model. The strategy here is to apply a circle or ellipse fitting upon the classified pixels and then to randomly select a patch within this circle or ellipse. The model updating algorithm doesn't need to run completely at every iteration, since there won't be a new illumination condition with every new frame and only noteworthy changes in illumination will result in the model being updated. These exit conditions test the physical possibility of the proposed model update considering the reflection characteristics of the target object, the change in illumination and the covariance on the measurements. The calculation of the new illumination condition itself can happen very rapidly, since we already know the reflectance spectrum matrix. After acquiring a nominal color triplet measurement h_N, we can write:

$$q_e(N_{new}) = h_N.R^{-1} \tag{13}$$

Where R^{-1} is the pseudo-inverse of the reflectance spectrum matrix acquired during learning and N_{new} is the index of the rarest illumination condition within the L matrix, which will thus be replaced by the new lighting condition.
The performance of this model updating process is illustrated in figure 3. Figure $3a$ shows the initial probability distribution for target object presence, while figure $3b$ shows the same distribution at a later time instance. This illustrates how the update step improves the Bayesian reflection model, such that the target object can be classified more clearly. In figures $3c$ and $3e$, the initially classified pixels are represented in black and the unclassified pixels in grey, respectively in the $l_1 - l_2 - l_3$ and the RGB-space, while figures $3d$ and $3f$ show the same at a later time. These two time instances are separated by a change in illumination conditions and as one can observe, the cluster of classified pixels has moved too in the color space.
The preceding discussion shows how we can acquire a description for the color of an object which is quite independent of the illumination conditions. Now, the object can be identified reliably and tracked in a following stage, as we'll explain in the next paragraph.

3 Camera Control for Target Tracking

We use the pinhole camera model and map the 3D world coordinates onto the image plane using the perspective projection. Now, let us consider a point P in the world coordinate system and its projection in the image p, as shown in figure 4. The point p is given by (u, v). The reciprocal values of pixel size (d_x, d_y), the camera focal length f and the principal point (o'_u, o'_v) are known from the camera calibration step.

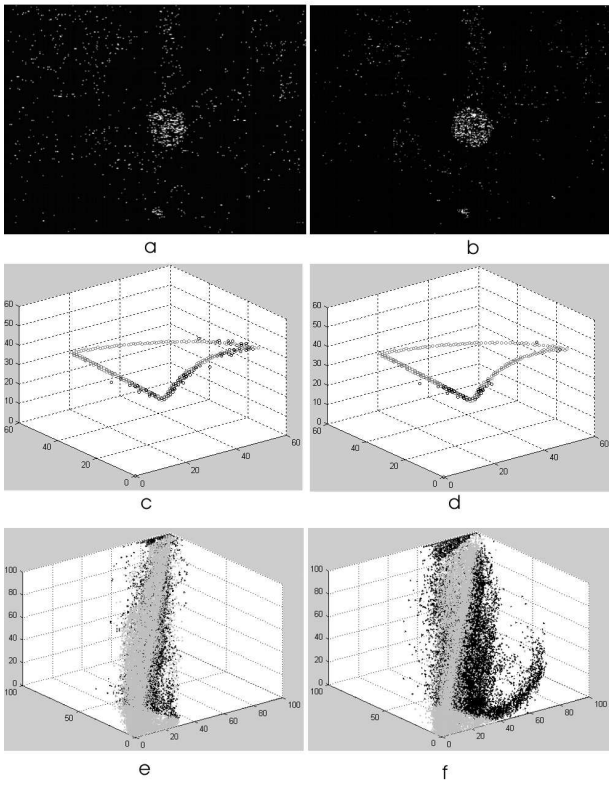

Figure 3. Effects of Model Updating

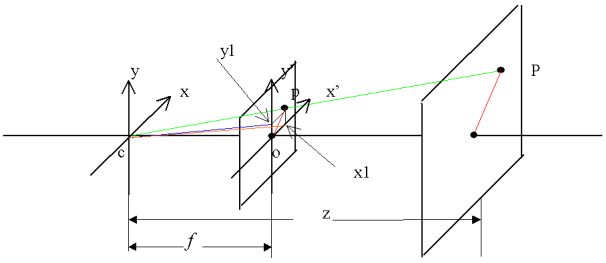

Figure 4. Definition of parameters

In figure 4 we define two angles:

$$\alpha = \angle ocx_1 \tag{14}$$

$$\beta = \angle ocy_1 \tag{15}$$

α and β represent the difference in orientation between the optical axis and the line cpP and can be calculated by:

$$\alpha = \tan^{-1}\left(\frac{u - o'_u}{f \cdot d_x}\right) \tag{16}$$

$$\beta = \tan^{-1}\left(\frac{o'_v - v}{f \cdot d_y}\right) \tag{17}$$

Our aim is to keep the target center coincident with the image center, thus α and β will define the pan/tilt control parameters of the camera. We define the servomotor-target-camera system as our plant. The above defined angles are used for camera control and target tracking. The plant is considered as a time variant system due to the unknown motion of the target. The target movement is estimated in real-time and considered in our system as the plant state transition of free response. In order to meet the system dynamic characteristic requirements, a two phase control strategy was implemented with a separate initialization phase and an observer based full state feedback control phase. During the system initialization phase a Proportional and Integral regulator is used to track the target. At the same time, the plant input and output data are collected to identify the plant model and to train the state observer and the adaptive filters used in the system. The plant model is used in state observation and state feedback control. After a certain period of time, the system control strategy is switched from phase one into phase two: the full state feedback control state.

3.1 Target Tracking during Initialization

During the initialization phase, the system (camera) is controlled by a PI regulator designed for target tracking. The system is considered as a time invariant one and the target movement is considered as an environment disturbance to the system. The block diagram of the control system for this phase is given in figure 5. We represent the image center as o, v is the target movement, n is the noise caused by the target movement in perspective view angle, $F(v)$ is the transfer function, representing the relationship between v and n, e is the signal error, u is the output of the regulator, m is the camera optical axis movement and y is the camera's output, i.e. the new target image center. Using this

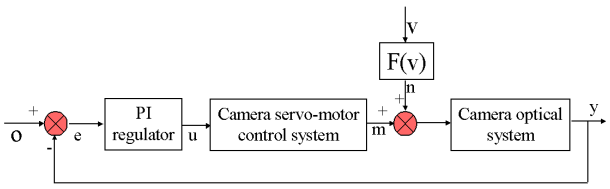

Figure 5. Initialization System Block Diagram

control method, the camera can start tracking right away, while the model is being built up from zero.

3.2 Plant Model Identification

The dynamic properties of our system can be described by the following set of nonlinear differential equations [9]:

$$\dot{\vec{x}}(t) = f\left(\vec{x}(t), \vec{u}(t), t\right) \tag{18}$$

Where $\vec{x}(t) \in \Re^n$ is the state vector, $\vec{u}(t) \in \Re^m$ is the input vector and f is a mapping $\Re^n \times \Re^m \rightarrow \Re^n$ defined as:

$$f(\vec{x}(t), \vec{u}(t), t) = \begin{bmatrix} f_1(\vec{x}(t), \vec{u}(t), t) \\ \vdots \\ f_n(\vec{x}(t), \vec{u}(t), t) \end{bmatrix} \quad (19)$$

To establish a practically useful plant model we must apply a linearization around the equilibrium point $(\vec{x}_0, \vec{u}_0)$ where both $\vec{x}_0$ and $\vec{u}_0$ are zero. The control strategy consists of keeping the target center and the image center coincident, so we can always linearize the nonlinear dynamic system around this equilibrium point and use a linear model to approximate the plant dynamics. For a discrete time system, the corresponding function can be written as:

$$\vec{x}(k+1) \approx \mathrm{A} \cdot \vec{x}(k) + \mathrm{B} \cdot \vec{u}(k) \quad (20)$$

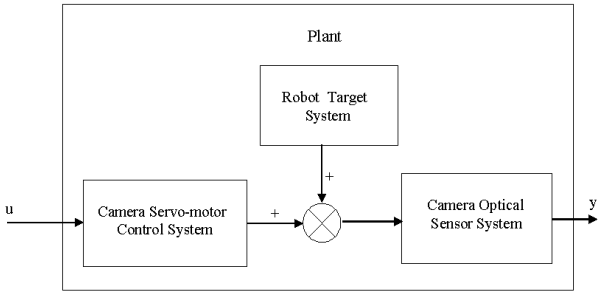

Figure 6. Dynamic System Model

$$X(k+1) = A(k).X(k) + B(k).u(k) + W(k) \quad (21)$$

$$y(k) = C(k).X(k) + v(k) \quad (22)$$

This can be observed on figure 6 and equations 21 and 22 where $X(k)$ represents the system state vector consisting of the angular position and angular speed of the target image, $y(k)$ is the system output representing the difference between the camera principal point and the target image position, $A(k)$ is the plant system matrix, $B(k)$ is the plant input matrix, $C(k)$ is the plant output matrix, $W(k)$ is the model noise vector, $v(k)$ is the measurement noise variable and $u(k)$ is the system input, which is in our case the center of the image plane. The matrices A and B are time-dependent, so the corresponding linear system is a time variant one.
We simplified the plant model by considering a second order difference model in order to reduce the calculation burden. The parameters of the plant state space function and the plant output function can then be written as:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -a_0 & -a_1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k) \quad (23)$$

$$y(k) = \begin{bmatrix} b_0 & b_1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} \quad (24)$$

(x_1, x_2) is the state vector corresponding to one of the camera angles (pan or tilt) and angular velocities. (a_0, a_1, b_0, b_1) are the system parameters to be estimated.
We use a Least-Mean-Square (LMS) 2^{nd}-order adaptive filter as plant parameter estimator [6]. The same structure for the LMS filter is used for both pan and tilt plant parameter estimation. The estimator works in two steps. First, it uses the updated input data, output data and filter's tap weights to estimate the system current output value. In the second step, it uses the updated input data, output data and the error between the estimated current output and the real output of the system to modify the tap weights $\vec{w}(k)$ of the filter. The updated tap weights are our plant parameters' estimates:

$$\vec{w}(k) = \begin{bmatrix} -\hat{a}_1(k) & -\hat{a}_0(k) & \hat{b}_1(k) & \hat{b}_0(k) \end{bmatrix}^{\mathrm{T}} \quad (25)$$

After estimating the plant parameters, the matrices of the state space model from instance k to $k+1$ can be estimated:

$$\mathrm{A}(k+1,k) = \begin{bmatrix} 0 & 1 \\ -\hat{a}_0(k) & -\hat{a}_1(k) \end{bmatrix} \quad (26)$$

$$\mathrm{B}(k+1,k) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (27)$$

$$C(k+1,k) = \begin{bmatrix} \hat{b}_0(k) & \hat{b}_1(k) \end{bmatrix} \quad (28)$$

3.3 Full State Feedback Control

The second phase control strategy consists of an observer-based full-state-feedback control strategy. An online identification method identifis in real-time the plant model and applies the identified model in the Kalman observer to emphasize the influence of the change of plant model on the plant state estimation. At the same time, the estimated state models are used for the state feedback strategy calculation to emphasize the time variant property of the control system. The main tasks of this phase are observing the plant states, calculating the feedback control value and identifying the plant model, as shown in figure 7.
Now that the plant model has been identified, its state vector is estimated using Kalman filtering [6]. The Kalman filter works as a current observer, taking into account the dynamics of the target's movement by using the time variant plant model. It reduces the influence of noise coming from the measurement inaccuracy and the model inaccuracy. From figure 8, we can see that the observer is a dynamic system. It takes the plant input and output as its input and the estimated plant states as its output. The observer is a negative feedback subsystem and it guarantees the convergence of the observation.

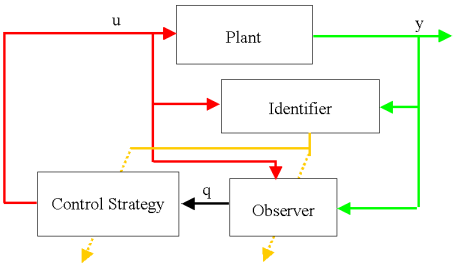

Figure 7. Observer Based Full State Feedback

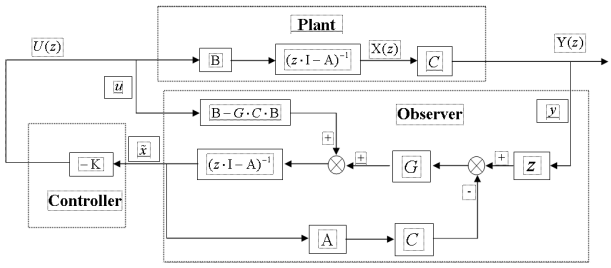

Figure 8. Full State Feedback Control System

$$\vec{x}(k+1) = \mathrm{A}(k+1,k)\cdot\vec{x}(k) + \mathrm{B}(k+1,k)\cdot\vec{u}(k) + \vec{v}_1(k) \quad (29)$$

$$\vec{y}(k) = C(k+1,k)\cdot\vec{x}(k) + \vec{v}_2(k) \quad (30)$$

In figure 8 and equations 29 and 30, u represents the plant input signal (pan or tilt control signal), y is the plant output signal (the angle estimated from the image), $\tilde{x}$ is the estimated plant state vector, $A(k+1,k)$ is the plant system transition matrix from instant k to instant $k+1$, $B(k+1,k)$ is the plant control input matrix from instant k to instant $k+1$, $C(k+1,k)$ is the plant output matrix from instant k to instant $k+1$, $\vec{v}_1(k)$ is the system process noise and $\vec{v}_2(k)$ the observation noise vector at instant k. The pole assignment method is used to design the state feedback controller.

3.4 Windowed Tracking

In order to increase the tracking sampling rate and the signal-to-noise ratio of the camera control, a bounding box (search window/region of interest) around the target image is defined. An LMS filter is used to estimate and to predict the position ($\bar{x}$,$\bar{y}$) and size (l ,h) of the defined search window, taking into account the activity of the camera. The window size is calculated by using the second order moments of the detected target boundary (μ_x^2 ,μ_y^2), following the equations 31 and 32:

$$l = C_1 \cdot \mu_x^2 + 2 \cdot \varepsilon \quad (31)$$

$$h = C_2 \cdot \mu_y^2 + 2 \cdot \varepsilon \quad (32)$$

where C_1 and C_2 are scale factors and ε is a tolerance.
The prediction of the search window position and size are made during the tracking process. Therefore, the time-variant characteristics of the system and the camera activity are taken into account. The structures of the adaptive LMS filters which are used for this purpose for the prediction of the search window position and size are identical. The difference between the filters lies in the fact that the first one uses the window position and the camera control signal as inputs to return a new window position estimate, while the second one uses the second order moments as inputs to calculate the window size. The predictor works in two steps. First, it uses the old input data and the current desired output data to train the filter; that is, to update the filter tap weights. In the second step, it uses the updated input data and tap weights to estimate a prediction for the real output. We define the prediction error as:

$$e(k) = d(k) - y(k) \quad (33)$$

where $d(k)$ is the desired output at the instant k and $y(k)$ is the predicted output at the instant k. The cost function is defined as:

$$J(k) = \frac{1}{2}\cdot \mathrm{E}\left[|e(k)|^2\right] \quad (34)$$

The filter minimizes $J(k)$, thereby estimating the search window parameters. Experimentally a 2^{nd} order filter was selected, because it proved to allow fast and stable tracking.

4 Experimental Results

Figure 9 shows the strength of the color constancy algorithm by comparing two pictures shot during the same indoor testing sequence, but with a difference in illumination conditions (lights turned off). As you can observe by noticing the whitened pixels which mean that a target has been found here, the algorithm succeeds in recognizing and classifying the searched object.
Figure 10 shows the tracking error in the X direction and demonstrates the tracking ability of this system. Notice the little increment when the target moves closer to the camera; it decreases when the target moves away from the camera.
Concerning the real-time capabilities, the target tracking program is able to run at about 10fps on a PC equipped with an 1.7GHz PIV processor, which is adequate for most every-day target tracking tasks.

5 Conclusions

We have shown a powerful set of algorithms, which were combined to form a universally useable system for auto-

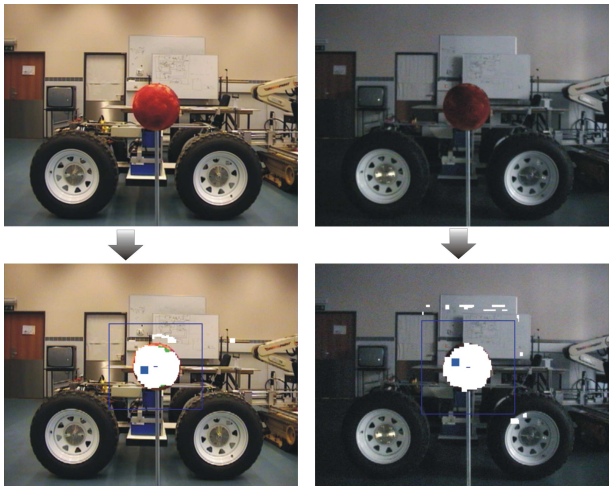

Figure 9. Color Constancy Results

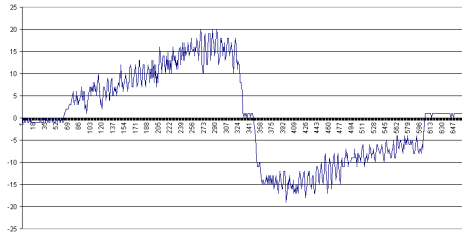

Figure 10. Tracking Error in Pixels

mated target detection, tracking and position estimation, using a single and fairly simple pan/tilt camera. The Bayesian-based color constancy approach which was used ensures that this system can keep working, even in harsh illumination conditions. This research was specifically aimed at applicability in the field of robotics, yet due to its general structure it can also be used for a range of applications.

6 Acknowledgements

This research has been conducted within the framework of the Inter-Universitary Attraction-Poles program number IAP 5/06 Advanced Mechatronic Systems, funded by the Belgian Federal Office for Scientific, Technical and Cultural Affairs.

References

[1] P. I. Corke and M. C. Good. Dynamic effects in visual closed-loop systems. *IEEE Transactions on Robotics and Automation*, 12(5):671–683, October 1996.

[2] M. D'Zmura and P. Lennie. Mechanisms of color constancy. *Journal of the Optical Society of America*, 3(10):1662–1672, 1986.

[3] B. Funt, V. Cardei, and K. Barnard. Learning color constancy. In *Fourth Color Imaging Conference: Color Science, Systems and Applications*, pages 58–60, 1996.

[4] T. Gevers and A.W.M. Smeulders. Color-based object recognition. *Pattern Recognition*, 32:453–464, 1999.

[5] T. Gevers and H. Stokman. Reflectance based edge classification. In *Vision Interface*, Trois-Rivires, Canada, May 1999.

[6] S. Haykin. *Adaptive Filter Technology*. Prentice-Hall, 1996.

[7] E. H. Land and J. J. McCaan. Lightness and the retinex theory. *Journal of the Optical Society of America*, 61:1–11, 1971.

[8] R.C. Love. *Surface Reflection Model Estimation from Naturally Illuminated Image Sequences*. PhD thesis, University of Leeds, 1997.

[9] T. P. McGarty. *Stochastical Systems and State Estimation*. John Wiley & Sons Inc, 1974.

[10] B. Schiele and J. Crowley. Probabilistic object recognition using multidimensional receptive field histograms. In *13th Intl. Conf. on Pattern Recognition, Volume B*, pages 50–54, August 1996.

[11] S.A. Shafer. Using color to separate reflection components. *COLOR research and application*, 10(4):pp. 210–218, 1985.

[12] Y. Tsin, R. Collins, V. Ramesh, and T. Kanade. Bayesian color constancy for outdoor object recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, December 2001.

[13] B. H. Yoshimi and P. K. Allen. Active, uncalibrated visual servoing. In *IEEE International Conference on Robotics and Automation*, pages 156–161, 1994.

[14] H. Zhang and J. P. Ostrowski. Visual servoing with dynamics: Control of an unmanned blimp. In *IEEE International Conference on Robotics and Automation*, pages 618–623, 1999.