Vision based Station Keeping and Docking for an Aerial Blimp

Sjoerd van der Zwaan , Alexandre Bernardino, José Santos-Victor

Instituto de Sistemas e Robótica
Instituto Superior Técnico
Av. Rovisco Pais, Torre Norte 7.26, 1049-001 Lisboa, Portugal
{sjoerd,alex,jasv}@isr.ist.utl.pt

Abstract

This paper describes a method for station keeping and docking of a lighter-than-air vehicle based on visual input. Due to the motion disturbances in the environment (currents), these tasks are important to keep the vehicle stabilized relative to an external reference frame. The main difficulties to achieve station keeping and docking are related to the non-holonomic constraints of the blimp moving in 3D, having a limited number of controllable degrees of freedom. The relative position of the vehicle with respect to a docking station is tracked using vision. A planar surface is chosen as a reference plane which allows visual tracking of an environmental region, based on planar projective transformations. An image-based control law is proposed together with a dynamic model for the vehicle. Experiments and results are described and discussed.

1 Introduction

Recently, research on the utilization of unmanned aerial vehicles has grown with an increasing interest on robotic airships, also known as blimps or lighter-than-air vehicles. The motivation behind it is that airships outperform airplanes and helicopters in low-speed, low altitude applications, having an enormous potential for tasks like environmental and traffic monitoring, climate research, transportation, etc. Several references can be found in literature about airship modeling for autonomous navigation [3, 4].

This work is integrated in the NARVAL[1] project, where one of the goals consists in using visual feedback for controlling an underwater vehicle. Since the blimp kinematics and dynamics are a reasonable approximation of those of the submarine, we use the blimp as a test-bed for experiments in a laboratory environment. As water is more viscous than air, the control will be simpler in that case, when compared to the blimp.

The problem addressed is that of controlling a small-sized, indoor blimp in order to achieve station keeping and docking, based on visual input. Station keeping consists in stabilizing the vehicle against some external reference frame or environmental region, thus rejecting external disturbances like currents. The docking problem is intimately related to station keeping, and consists of controlling the vehicle in order to attain a desired position and orientation relative to a chosen coordinate frame. Assuming that the tasks of docking and station keeping are defined relative to short range regions in the environment, one can use vision in order to extract information about the vehicle's pose and use this information for visual servoing. In analogy to the approach adopted in [5], an image-based visual servo-law is proposed to close the control loop.

Assuming that an image patch is initially identified by some user, the temporal changes of this image patch induced by the vehicle's motion, need to be tracked. As the vehicle moves in 3D, such an image can be distorted according to a fairly general set of deformations when compared to those usually considered in robotics and computer vision (translational or affine motions). In this paper, the tracking system is based on full planar projective transformations in real-time, thus accommodating a wide set of possible image deformations.

Section 2 describes the method used for visual tracking of the identified image patch. Section 3 describes the airborne blimp together with its dynamic model. The control aspects are further detailed in Section 4 and experiments are described in Section 5. Finally, in Section 6, conclusions are drawn and future work is indicated.

[1]ESPRIT-LTR Project 30185, NARVAL - Navigation of Autonomous Robots via Active Environmental Perception

2 Tracking of Image Regions

Given a reference image I_0 and a target image I_1, the registration problem is that of computing an image transformation $(x', y') = \omega(x, y)$, that maps a region R of the reference image onto a specified region of the target image, such that $I_1(x, y) = I_0(\omega(x, y))$. Usually, these transformations, are parameterized as a function of a vector $\mathbf{q}$: $\omega(x, y) = \omega(x, y; \mathbf{q})$. In this work we are mostly interested in tracking planar regions. Planar motions cannot be adequately modeled by simple image transforms, like affine or translational. A projective planar transformation is the exact motion model when a camera rotates about its eyepoint or if the image surface is planar [6]. The 2D projective transformation depends on 8 parameters:

$$x' = \frac{q_1 x + q_2 y + q_3}{q_7 x + q_8 y + 1}, \quad y' = \frac{q_4 x + q_5 y + q_6}{q_7 x + q_8 y + 1} \tag{1}$$

To register the two image regions, the best possible match can be obtained through the minimization of an error function, using an appropriate norm, such as the sum-of-squared-differences ($L2$-error criterion). We assume that at each time instant we have a prediction of the current transformation parameters, $\mathbf{q_0}$. The most simple prediction may be the parameters of the previous step. In our case we compute the optic flow in the region and adjust an affine model to the computed flow, providing a first step towards the solution so only small adjustments remain to be made. Considering images as column vectors, the $L2$ error function is given by :

$$e(\Delta\mathbf{q}) = \frac{1}{2} \parallel I_1 - \bar{I}_0(\Delta\mathbf{q}) \parallel^2 \tag{2}$$

where $\bar{I}_0(\Delta\mathbf{q}) = I_0(\omega(x, y; \mathbf{q_0} + \Delta\mathbf{q}))$ and $\mathbf{q}_0$ is the initial prediction of the transform. To minimize this error function, usual gradient descent methods use the partial derivatives of $\bar{I}_0(\Delta\mathbf{q})$ whose discrete approximations can be computed as $\bar{I}_0(\delta\mathbf{e}_i) - I_0$, where $\mathbf{e}_i$ is the i'th basis vector and δ is an adequate value for discretization, depending on the shape of the error function. However, for an 8 dimensional space this method has a weak interpolation capability, i.e. it only searches the space effectively in the coordinate directions, and within a range determined by the parameter δ. Furthermore, minimizing eq. (2) using an exhaustive search on the parameter space would be impractical. Instead we assume a set of vectors $\{\Delta\mathbf{q}_i : i \in (1 \ldots m)\}$ that sample the parameter space more densely for the expected image deformations and range. The parameter vector $\Delta\mathbf{q}$ can be expressed as a linear combination of the various $\Delta\mathbf{q}_i$:

$$\Delta\mathbf{q} = \sum_{i=1}^{m} k_i \Delta\mathbf{q}_i \tag{3}$$

Now the image space can be considered as a function of the parameter vector $\mathbf{k} = [k_1 \ldots k_m]^T$. The new parameterization is given by: $\hat{I}_0(\mathbf{k}) = \bar{I}_0(\sum_{i=1}^{m} k_i \Delta\mathbf{q}_i)$. For small deviations about $\mathbf{k} = \mathbf{0}$ we have the first order approximation: $\hat{I}_0(\mathbf{k}) \approx I_0 + \sum_{i=1}^{m} \frac{\partial \hat{I}_0}{\partial k_i} k_i$. A suitable discrete approximation of each partial derivative in the previous equation is given by:

$$\frac{\partial \hat{I}_0}{\partial k_i} = \hat{I}_0(\mathbf{e}_i) - I_0 = \bar{I}_0(\Delta\mathbf{q}_i) - I_0 = B_i \tag{4}$$

In [2], the set of vectors B_i are denoted "Difference Templates" and are also used for image registration, but they are justified in a different form. The error function now takes the form:

$$e(\mathbf{k}) = \frac{1}{2} \sum_{i=1}^{m} \parallel D - B\mathbf{k} \parallel^2 \tag{5}$$

where D is the difference image $I_1 - I_0$ and B is the partial derivatives matrix: $B = [B_1 \ldots B_m]$. Vector k can be determined minimizing:

$$min_k \parallel D - B\mathbf{k} \parallel^2 \quad \Rightarrow \quad B^T D - B^T B\mathbf{k} = 0 \tag{6}$$

After determining $\mathbf{k}$, the solution for $\Delta\mathbf{q}$ can be calculated using the linear combination in eq. (3). Using the updated estimates of the parameter vector $\Delta\mathbf{q}$, this process can proceed in subsequent iterations and at different resolutions of the image pyramid.

3 Airship Modeling

The small-size indoor blimp used, equipped with an on-board camera, radio control and video link is illustrated in Figure 1.

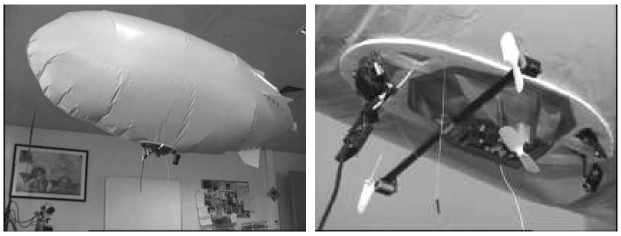

Figure 1: Radio controlled indoor blimp with on-board camera.

In order to derive a mathematical model, it is useful to first review some physical principles of airship operation [4]. First of all, an important characteristic is the aerostatic lift, which, unlike the lift forces generated over a wing surface, is independent of flight speed. The aerostatic lift force comes from Archimedes' Principle and is equal to the mass of the volume of air displaced by the airship's envelope. The aerostatic lift is also known as the buoyant force. An upward lift is obtained when the airship's envelope contains a gas with a density lower then air. Helium is the most commonly used lifting gas.

Second, a buoyant body in motion in a fluid displaces each fluid particle in the direction of motion of the body. The fluid therefore gains kinetic energy and the body experiences a resistance to its motion. This effect can be taken into account by considering added mass and inertia terms. As the blimp displaces a relative large volume of air during flight, these properties become significant and the body behaves as if it had a mass and moments of inertia substantially higher than those indicated by conventional physical methods.

Finally, because the airship center-of-mass is difficult to locate and time-variant during flight, motion has to be referenced to by a system of orthogonal body axes, $\{LTA\}$, placed at the geometric center of the envelope volume (CV), as shown in Figure 2.

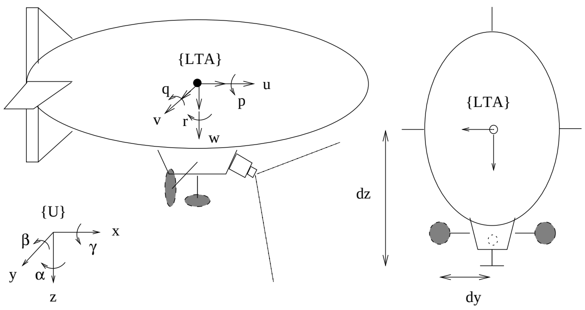

Figure 2: Definition of reference frames.

The centre of volume can be assumed to coincide with the center of buoyancy.

3.1 Dynamics and kinematics

Considering the vehicle as a rigid body (not taking into account its elasticity) the dynamic model can be obtained by writing down the Newton-Euler equations of motion resolved into the body-fixed reference frame, which can be stated as [7, 4]:

$$M\dot{x} + C(x) + D(x)x + G(\alpha, \beta, \gamma) = \tau, \qquad (7)$$

where:

- $x = [u \quad v \quad w \quad p \quad q \quad r]^T$, is the 6×1 velocity vector containing the three components of linear velocity u, v, w and the three components of angular velocity p, q, r, measured in some earth-fixed inertial reference frame $\{U\}$ and expressed in the body-fixed reference frame $\{LTA\}$.

- $M \equiv M_{RB} + M_A$, is the 6×6 mass matrix containing all masses and inertias of the rigid body (M_{RB}) and the added mass and inertia terms (M_A). Due to the symmetry of the vehicle, cross-coupling inertial terms in M_{RB} can be neglected. The following model for the added mass and inertia terms is used: $M_A = diag\{A_{11}, A_{22}, A_{33}, A_{44}, A_{55}, A_{66}\}$. The elements $A_{ii}, i \in (1, 2, .., 6)$ can be estimated from the dimensions of the airship's hull [7].

- $C \equiv C_{RB} + C_A$, is the 6×1 dynamic forces vector containing all Coriolis and centrifugal terms of the dynamic model [4, 7] .

- $D(x) \equiv D_S(x) + D_M(x)$, is the 6×6 aerodynamic damping matrix. Linear and quadratic skin friction drag are modeled with $D_S(x)$ and arise due to laminar and turbulent boundary layers. Damping due to vortex shedding, which depends on the streamline of the airship's hull, can be modeled with $D_M(x)$. For small indoor blimps moving at low speed, laminar boundary layer conditions can be assumed, considering only linear skin friction coefficients. For this case, the following model is proposed: $D = diag\{X_u, Y_v, Z_w, K_p, M_q, N_r\}$. The elements of D can be estimated either from wind-tunnel testing or system identification tools.

- $G(\alpha, \beta, \gamma)$, is the 6×1 restoring forces vector containing the gravity and buoyancy forces resolved into body-fixed axes by a rotation matrix, parameterized by the standard $roll(\gamma)$, $pitch(\beta)$ and $yaw(\alpha)$ angles representation.

- $\tau = \tau_A + \tau_P$, is the 6×1 applied forces vector containing all forces and moments acting on the rigid body due to the aerodynamic control surfaces (τ_A) and the propulsive units (τ_P). For an indoor blimp moving at low speed under laminar boundary layer conditions, aerodynamic control surfaces will have no effect and thus are not included. In this case, the vector τ will be a function of the geometrical arrangement of the propulsive units around the body axes: $\tau = [T_{cmn} \quad 0 \quad T_v \quad 0 \quad (d_z T_{cmn}) \quad (d_y T_{diff})]^T$, where $T_{cmn} = (T_s + T_p)$ is the common mode component of thrust coming from the starboard and port side propellers, T_v is the thrust resulting from the vertical propeller, $T_{diff} = T_s - T_p$ is the difference between starboard and port side propeller thrust, d_y is the horizontal offset from the center of volume of the horizontal propellers and d_z is the vertical offset of the vertical propeller (see Figure 2).

For control and navigation purposes, the velocity vector in eq.(7) must be transformed to the earth-fixed inertial frame, leading to the kinematic relations. The relative position and orientation of $\{LTA\}$ with respect to $\{U\}$ will be denoted by the vector $\eta = [x \quad y \quad z \quad \alpha \quad \beta \quad \gamma]^T$, where a standard *roll, pitch* and *yaw* angles representation is assumed for the orientation. The kinematic equations can then be written as:

$$\dot{\eta} = \begin{bmatrix} J_1(\alpha,\beta,\gamma) & 0_{3\times 3} \\ 0_{3\times 3} & J_2(\alpha,\beta,\gamma) \end{bmatrix} x, \qquad (8)$$

where J_1 is the corresponding rotation matrix and J_2 a Jacobian matrix relating the angular velocity vector to the time derivatives of the attitude parameters. The Jacobian J_2 can be easily calculated from the rotation matrix (see [1]).

3.2 Linearized equations of motion

Assuming airship motion to be constrained to small pertubations about some equilibrium condition, a considerably simplified linear model can be obtained. Writing eq.(7) as $M\dot{x}(t) = f(x(t), \tau(t))$ with $x(t) = \tilde{x}(t) + x_\delta(t)$ and $\tau(t) = \tilde{\tau}(t) + \tau_\delta(t)$ written as small pertubations x_δ, τ_δ around some equilibrium condition $\tilde{x}, \tilde{\tau}$, the linearized system around the equilibrium condition is given by:

$$\dot{x}_\delta(t) = Ax_\delta(t) + B\tau_\delta(t), \qquad (9)$$

where $A = \frac{\delta f}{\delta x}(\tilde{x}(t), \tilde{\tau}(t))$ and $B = \frac{\delta f}{\delta \tau}(\tilde{x}(t), \tilde{\tau}(t))$ are Jacobian matrices obtained from a first order Taylor expansion of $f(x(t), \tau(t))$ about the equilibrium. In particular, the products and squares of small pertubation variables become negligibly small in a linearized model so all Coriolis and centrifugal terms in eq.(7) can be neglected. With $C(x) = 0$, the dynamic system becomes decoupled in longitudinal motion (vertical X-Z-plane) and lateral motion (horizontal X-Y-plane).

Linearizing about $\beta = \gamma = 0$ corresponds to small pitch and roll angles, which is a reasonable condition for airship operation. The decoupled linearized systems for this equilibrium condition are given by the longitudinal and lateral models as given in the sequence.

3.3 Longitudinal and lateral models

The longitudinal or vertical model describes the airship motion in the X-Z-plane of the body-fixed reference frame. The corresponding state and input vectors are given by:

$$\begin{aligned} x &= [\delta u \quad \delta w \quad \delta q \quad \delta\beta]^T \\ \tau &= [\delta T_{cmn} \quad \delta T_v]^T \end{aligned}$$

where the small perturbation of the pitch angle, $\delta\beta$, is included in the state vector to accommodate the gravitation and buoyancy forces so as to obtain squared matrices. The dynamics of the pitch angle is given by the kinematic relations derived in eq.(8). Writing down the dynamic equations from eq.(7) corresponding to each state variable, the linearized longitudinal model about $\beta = \gamma = 0$ is given by:

$$\begin{aligned} M\dot{x} &= Ax + B\tau, \qquad (10) \\ M &= \begin{bmatrix} m + A_{11} & 0 & ma_z & 0 \\ 0 & m + A_{33} & -ma_x & 0 \\ ma_z & -ma_x & I_{yy} + A_{55} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\ A &= \begin{bmatrix} -X_u & 0 & 0 & -(mg - f_b) \\ 0 & -Z_w & 0 & 0 \\ 0 & 0 & -M_q & -a_z mg \\ 0 & 0 & 1 & 0 \end{bmatrix}, \\ B &= \begin{bmatrix} 1 & 0 & d_z & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}^T \end{aligned}$$

The lateral or horizontal model describes the dynamics of the airship orientation in the X-Y-plane and includes the following state and input variables:

$$\begin{aligned} x &= [\delta v \quad \delta p \quad \delta r \quad \delta\gamma]^T \\ \tau &= \delta T_{diff} \end{aligned}$$

Here the roll angle, γ, is included and the linearized model is given by:

$$\begin{aligned} M\dot{x} &= Ax + B\tau, \qquad (11) \\ M &= \begin{bmatrix} m + A_{22} & -ma_z & ma_x & 0 \\ -ma_z & I_{xx} + A_{44} & 0 & 0 \\ ma_x & 0 & I_{zz} + A_{66} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\ A &= \begin{bmatrix} -Y_v & 0 & 0 & mg - f_b \\ 0 & -K_p & 0 & -a_z mg \\ 0 & 0 & -N_r & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \\ B &= [0 \quad 0 \quad d_y \quad 0]^T \end{aligned}$$

In these models, m is the airship mass, g is the gravitational acceleration, f_b is the buoyancy force, $(a_x \quad a_z)$ are the coordinates of the center off mass, d_z is the coordinate of the vertical propeller, d_y is the (symmetric) coordinate of the horizontal propellers, all expressed in the body fixed reference frame, I_{ii} are inertia terms, A_{ii} are added mass and inertia terms and $X_u, Y_v, Z_w, K_p, M_q, N_r$ the linear friction terms.

Note that although the linearized models are valid only for small perturbations of the roll and pitch angles about $\gamma = \beta = 0$, no such assumptions exist for the yaw (α) angle.

4 Controller Design

As a first approach for station keeping and docking, we use a control strategy based on measurements in the image-plane. This strategy uses experimentally tuned PID-controllers for each controllable degree of freedom. This demonstrates the possibility of controlling the system with simple controllers. Additionally it has allowed the identification of some model parameters, required for more sophisticated controllers based on the linear models presented in section 3.3.

Consider the coordinates of the centroid of the tracked window as an error signal to control the blimp. Hence, the desired behaviour for the closed loop system is to bring the tracked window back to the image center. The error in the vertical direction of the image coordinates, e_v, is used to control the blimp's forward motion:

$$T_{cmn}(t) = k_p.e_v(t) + k_d.\frac{d}{dt}e_v(t)$$

Note that there is no need for including the integral term of the PID-controller due to the existence of integration terms in the open-loop system. Similarly, the lateral deviation of the target from the image center, e_l, is used to control the blimp lateral motion:

$$T_{diff}(t) = k_p^{'}.e_l(t) + k_d^{'}.\frac{d}{dt}e_l(t)$$

The altitude of the blimp can be controlled by considering the area of the tracked window and taking the area of the initial user-selected window as a reference. The desired behavior of the closed-loop system is to maintain the area of the tracked window close to its reference, thus trying to maintain the blimp at some constant height. Defining the error between the area of the tracked window and its reference by e_A, the altitude control is given by:

$$T_v(t) = k_p^{''}.e_A(t) + k_d^{''}.\frac{d}{dt}.e_A(t)$$

Again, no integral term is considered for the controller. Plugging the error signals into the linearized models by substituting the input signals, provides a means for analytic controller synthesis. It should be noted however that for this purpose, the error signals need to be transformed from the image-plane to Euclidean-space, involving a non-linearity of the camera Jacobian.

5 Results

A set of real experiments were done using the blimp with the on-board camera. Figure 3 shows the results of tracking an image window which defines the coordinate frame used for station keeping. In spite of the distortion introduced by the blimp's camera, the implemented tracker successfully follows the specified image patch while the blimp is moving in 3D. Feature tracking is realized at a sampling rate of 12Hz.

Figure 3: Tracking an image window from the flying blimp.

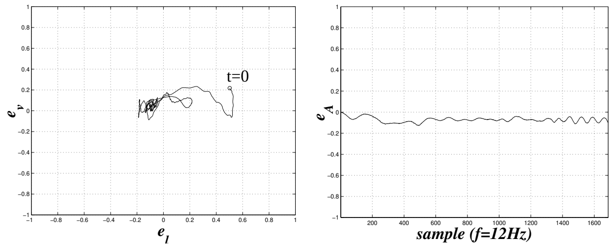

Figure 4: Docking and station keeping test with the blimp. Left: trajectory of the centroid of the tracked window (image errors in normalized pixel coordinates); Right: difference between the area of the image patch and the tracked window.

Figure 4 shows the temporal evolution of the error signals during a docking and station keeping experiment. At the left side, the image trajectory of the target point (centroid of the tracked window) is illustrated under closed-loop control. The control strategy aims at driving this point to the image center (docking) and keep it as close as possible to this center (station keeping). The right image shows the error between the areas of the reference window and the tracked window and indicates that the blimp is approximately maintained at a constant height.

Figure 5 shows the corresponding control signals (the blimp's linear and angular velocities) for the docking and station keeping experiment.

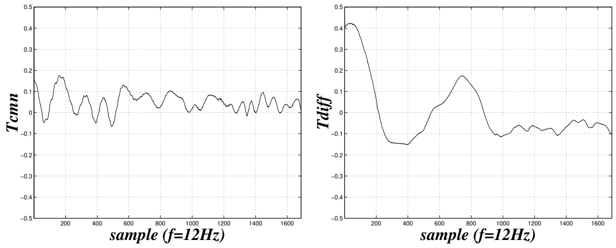

Figure 5: Docking and station keeping test with the blimp. Common mode forward velocity control (left) and differential angular velocity control (right).

Although preliminary, these approaches led to encouraging results in moving the error close to zero, and maintaining the target close to the image center. The main difficulties arise when the target moves laterally. In this case, since the blimp does not have lateral degrees of freedom, the only solution is to compensate this error by rotating the blimp. Then, as the rotation is not performed around the camera optical axis, it induces a translation motion in the image plane, which will generate errors for the forward motion control. This fact explains that the forward motion control, T_{cmn} in Figure 5, shows an oscillatory behavior near to its reference value (zero).

6 Conclusions

In this paper we have explained how the problems of docking and station keeping of an airship can be addressed using visual feedback, selecting an image patch on the docking station. The presented method tracks the selected region, thus following its temporal evolution in $3D$-space. This information is then used with a control strategy so as to perform station keeping and docking.

Results were obtained concerning the problem of automatically controlling the blimp so as to perform station keeping and docking. A first control strategy is proposed which succeeded in moving the control errors close to zero. The main difficulties arise in those situations for which positioning errors occur along the blimp y-axis. In this case, since the blimp is a non-holonomic vehicle and does not have lateral degrees of freedom, the only solution is to compensate this error by rotating the blimp. Work is being carried out to develop more robust and adequate control strategies. However, even with this simple control system it was possible to obtain encouraging results.

Acknowledgments

We would like to thank Matteo Perrone for helping in developing the tracking system. This work has been funded by ESPRIT-LTR Project 30185, NARVAL.

References

[1] John J. Craig. *Introduction to Robotics.* Addison Wesley, 2nd edition, 1986.

[2] M. Gleicher. Projective Registration with Difference Decomposition. *IEEE Conf. of Computer Vision and Pattern Recognition*, pages 331–337, jun 1997.

[3] A. Elfes and S. Sigueira Bueno and M. Bergerman and J.Jr.G Ramos. A Semi-Autonomous Robotic Airship for Environmental Monitoring Missions. *IEEE Int. Conf. on Robotics and Automation*, Leuven, Belgium, May 1998.

[4] S.B. Varella Gomes and J.Jr.G. Ramos. Airship Dynamic Modeling for Autonomous Operation. *IEEE Int. Conf. on Robotics and Automation*, Leuven, Belgium, May 1998.

[5] H. Zhang and J.P. Ostrowski. Visual Servoing with Dynamics: Control of an Unmanned Blimp. *IEEE Int. Conf. on Robotics & Automation*, Detroit, USA, May 1999.

[6] R. Szelinski. Image Mosaicing for Tele-Reality Applications. *IEEE Workshop on Applications of Computer Vision*, Sarasota, Florida, 1994.

[7] Thor I. Fossen. *Guidance and Control of Ocean Vehicles.* John-Wiley & Sons, 1995.