A Vision-Based Strategy for Autonomous Aerial Refuelling Tasks

Carol Martínez[a,*], Thomas Richardson[b], Peter Thomas[b], Jonathan Luke du Bois[b], Pascual Campoy[a]

[a] *Computer Vision Group, Centro de Automática y Robótica CAR-UPM-CSIC,*
Universidad Politécnica de Madrid, C. José Gutiérrez Abascal 2. 28006 Madrid, Spain
[b] *Department of Aerospace Engineering, Queens Building*
University of Bristol, BS8 1TR, UK

Abstract

Autonomous aerial refuelling is a key enabling technology for both manned and unmanned aircraft where extended flight duration or range are required. Generated as part of the UK ASTRAEA programme, the results presented within this paper offer one potential vision based sensing solution, together with a unique test environment. A hierarchical visual tracking algorithm based on direct methods is proposed and developed for the purposes of tracking a drogue during the capture stage of autonomous aerial refuelling, and of estimating its 3D position. Intended to be applied in real time to a video stream from a single monocular camera mounted on the receiver aircraft, the algorithm is shown to be highly robust, and capable of tracking large, rapid drogue motions within the frame of reference. The proposed strategy has been tested using a complex robotic testbed and with actual flight hardware consisting of a full size probe and drogue. Results show that the vision tracking algorithm can detect and track the drogue at real-time frame rates of more than thirty frames per second, obtaining a robust position estimation even with strong motions and multiple occlusions of the drogue.

Keywords: Autonomous Aerial Refuelling, Probe and Drogue Refuelling, Unmanned Aerial Vehicles, Visual Tracking, Computer Vision, Direct Methods

[*] Corresponding author
Email address: `carolviviana.martinez@upm.es` (Carol Martínez)
URL: `www.vision4uavs.com` (Carol Martínez)

Preprint submitted to Robotics and Autonomous Systems *September 14, 2017*

1. Introduction

Aerial refuelling, also referred to as air-to-air refuelling (AAR), was first developed in the 1920s and has since evolved into an established means for extending the range, payload, and endurance of manned aircraft in military operations [1].

There are two primary methods for carrying out AAR: the probe and drogue method pioneered by Flight Refuelling Ltd. [2], shown in Figure 1; and the Flying Boom method developed by Boeing [3]. In the latter, a retractable boom is extended from the tanker aircraft, and steered by means of two "ruddervators", aerodynamic control surfaces attached to the boom. An operator in the tanker aircraft steers the tip of the boom to a coupling on the receiver aircraft, which holds a formation position below and to the aft of the tanker. For probe and drogue refuelling (shown in Figure 1), the tanker trails a flexible hose terminating in a drogue assembly, comprised of a canopy to provide stability and a coupling for the fuel transfer. The receiver aircraft is equipped with a probe which is maneuvered into the drogue by the pilot. For clarity, this paper focuses solely on the automation of probe and drogue refuelling.

In unmanned aerial vehicles (UAVs), where endurance is no longer limited by pilot fatigue, aerial refuelling capabilities offer significant benefits. Refuelling operations have historically been conducted as a piloted operation demanding a high level of training and fast reactions, and as such are not appropriate for remotely piloted aircraft controlled over relatively slow data links. The recent proliferation of UAVs has therefore offered a potential new market for autonomous air-to-air refuelling (AAAR) capabilities. Development of this capability relies on two key technologies: position sensing and tracking, in order to allow the receiver aircraft to determine the relative position of the refuelling drogue; and control strategies, to enable a robust and safe approach and coupling.

There have been extensive works on appropriate control systems developed with numerical flight simulations, using for example traditional PID and LQR methods, gain scheduling [4], adaptive controllers such as neural networks [5, 6] and model reference adaptive control [7], differential game approaches [8, 9, 10], and feedback linearisation techniques [11]. Other work has investigated fault tolerance [12] and actuator failure cases [13]. Numeri-

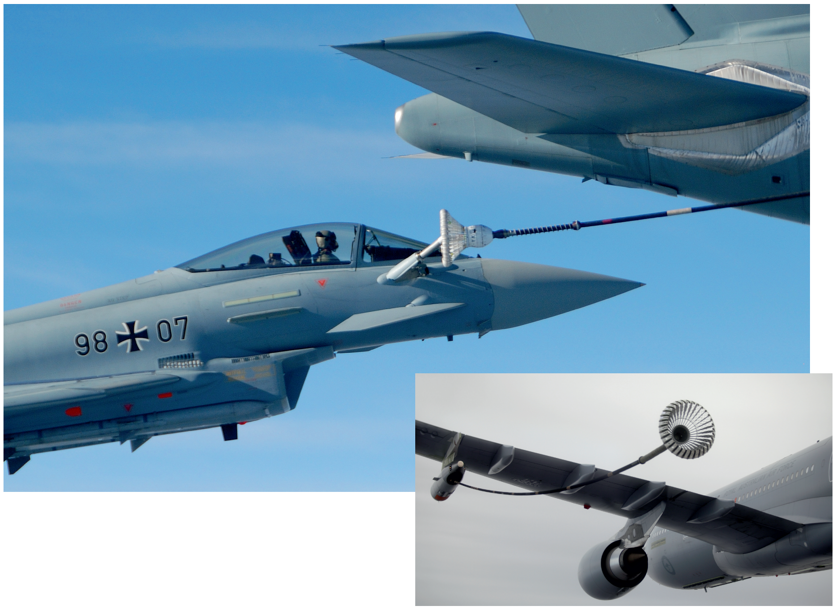


Figure 1: Probe-drogue refuelling: courtesy of Cobham Mission Equipment

cal simulations have been enhanced with the inclusion of turbulence models and the development of improved tanker wake models [14] and drogue modelling [15]. In addition to the simulation results of the above studies, actual flight tests have been successfully conducted demonstrating formation flying and moving between stations for both the boom [16, 17] and the probe and drogue [18, 19, 20] methods. The latter study also demonstrated full contact with the drogue on one flight, engaging successfully in two out of six attempts.

For position tracking in AAAR, a variety of sensing technologies have been employed, including inertial measurements [21], differential GPS (DGPS), and electo-optical systems. Often, these are employed in tandem using wireless telemetry [22] and sensor fusion methods, where in addition to improved accuracy, redundancy affords a level of fault tolerance. Williamson *et al.* [22], for example, used Kalman Filtering in their laboratory-based flying boom experiments. Similarly, the combination of GPS measurements with position estimates from vision systems has been explored in a number of publications [23, 24, 25, 26, 27, 28], where the principal approach is to use the GPS measurements predominantly at a distance, filtering in the machine vision data with increasing proximity to the target.

Machine vision systems in unmanned air vehicle operations is an increasingly popular method for collating position data. Research has been carried out in applying electro-optical sensors to the tasks of navigation [29], tracking [30], and collision avoidance[31]. By identifying key features of a target from an image, relative position and orientation can be inferred when the 3D location of these features on the target are known [32] (monocular system), or when the position of these features from different view points is known [33] (multi-camera or monocular systems). Advantages of using vision systems for AAAR include the potential for installation without modification being required to the target aircraft, and increasing precision with proximity to the target. Disadvantages of vision systems can include high processing requirements and susceptibility to environmental conditions such as cloud, fog, and variable lighting conditions.

Tackling the issue of processing power, Junkins *et al.* developed a system called VisNav [34] which has been used in several AAAR studies [35, 36, 37, 38]. In contrast to many vision systems, which analyze complete images, VisNav employs a lens and a position-sensing diode capable of detecting the line of sight of a light source based on the region of the diode the light is focused on, without digitizing an image. By employing sequenced illumination of beacons in known locations on the target, in conjunction with a communication link between the sensor and the beacons, the system can triangulate the position and orientation of the target with update rates of up to 100 Hz and relatively meagre processing requirements.

Also advocating the use of beacons, Pollini *et al.* [39] proposed placing light emitting diodes (LEDs) on the drogue and using an inexpensive CCD webcam with an infra-red (IR) filter to identify the LEDs. Images from the IR camera were fed into a modified Lu, Hager and Mjolsness (LHM) algorithm [40] in order to determine the relative position and attitude of the drogue. They conducted indoor tests and simulations with natural lighting conditions [41] and demonstrated that the algorithm was able to make reasonable estimates of the position of the target even with some markers unidentified. One disadvantage associated with the use of beacons in probe and drogue refuelling is that the hose to which the drogue is attached does not normally carry electrical power, and provision for such power can require non-trivial modifications to the tanker equipment.

Passive systems, on the other hand, do not require active cooperation from the target. Spencer [26] used a corner detection algorithm to extract both structural and painted features on a tanker. For each video frame, the detected features were compared to known features on the tanker in order to compute 3D pointing vectors which were used in a Kalman filter based navigation algorithm to determine the relative position of the tanker.

Saghafi and Zadeh[42] had success with a pattern recognition approach, although it was reliant on a radial basis neural network and was slow to converge on solutions. Generally the large computational overhead associated with pattern recognition techniques can lead to comparatively low update rates. Doebbler *et al* [43] demonstrated a deformable contour algorithm and integrated it into an automatic boom controller. The algorithm uses weighted colour statistics for the three image colour channels to converge on the outline of the docking markings around the refuelling port and estimate the position of the receiver with a 30 Hz refresh rate.

Vendra *et al* analyzed the performance of well-known corner detectors (SUSAN and Harris) for the use of a machine vision-based approach in the UAV Aerial Refueling problem. A camera was placed in the receiver aircraft looking upwards, capturing the tanker aircraft and the feature extractor algorithms detect corners of the tanker aircraft. These corners are matched with a set of known physical features on the tanker (2D-3D match) using a detection and labeling algorithm, and the positions of the matched corners are used by a pose estimation algorithm to evaluate the position and orientation of the receiver aircraft with respect to the tanker aircraft. The study analyzes the robustness of the corner detection algorithms in the event of image noise, variations in image contrast, motion blur; and also confirms the capabilities of the corner detection algorithms for interfacing with detection and labeling, and pose estimation algorithms in the AAAR problem.

Evidently, most of the existing vision-based approaches for autonomous aerial refuelling make use of features [44] such as corners, painted marks and LEDs. Not only do these methods often require the installation of specific hardware, but they are also susceptible to problems caused by the occlusion of one or more of these features. This paper proposes the use of direct methods [45] and hierarchical image registration techniques to solve the drogue tracking problem for automated aerial refuelling.

Direct methods have the advantage of solving –without intermediate steps– the motion of the camera, and the matching of the pixels using the intensity information of all the pixels in the template. Hierarchical methods [46] allow the detection of large image motion, with the additional advantage of increasing the robustness in the estimation of the motion model using a coarse to fine refinement of the parameters. This can be very important in the context of turbulence effects that will produce sudden, large motions in the image plane, and in an environment where finite processing power is likely to give a direct correlation between algorithm efficiency and position estimate refresh rate. High refresh rates are critical in a real time aircraft control system that needs to exhibit precise positioning in turbulent conditions.

In addition, whilst a limited set of studies have tested vision systems under realistic operating conditions, many have relied on simulated visual data to test the algorithms. This paper will focus specifically on the evaluation of a vision system for probe and drogue refuelling using a single camera in conjunction with real, full-scale aircraft hardware in a laboratory test environment. The real-time vision-based strategy is based on direct methods and image processing techniques and differs from previous approaches in that by using direct methods, installation of specialized hardware or software is not required, and that under partial occlusions of the drogue, the algorithm is able to continue with the tracking task. The test environment comprises a robotic cell that simulates the tanker and receiver aircraft in varying degrees of turbulence, linked to a drogue attached to the free end of one robot, and a refuelling probe attached to a second, track-mounted robot.

The paper is organized as follows: Section 2 describes the laboratory equipment used to recreate the relative motion of the probe and drogue hardware and Section 3 outlines the flight dynamics models and control algorithms from which the positional data is derived. The vision tracking algorithms are then presented in Section 4, followed by the experimental results in Section 5. Conclusions are drawn in Section 6.

2. Autonomous Air-to-Air Refuelling Testbed

The vision tracking algorithm presented in this paper was tested experimentally in a laboratory using real flight hardware on a bespoke AAAR testbed. This is a robotic cell which has been used to reproduce the relative motion of the probe and drogue, as pictured in Figure 2. The cell consists of two 6 degree of freedom (DOF) robotic arms, one fixed to the ground at its base and the other mounted on a linear track. Actual flight hardware is mounted on the end of the robot arms: a drogue is attached to the free end of the grounded robot (R1 in Figure 2) and a refuelling probe is fitted to the track-mounted device (R2 in Figure 2). The robot motion is then driven by data sets produced from a numerical simulation of a probe and drogue

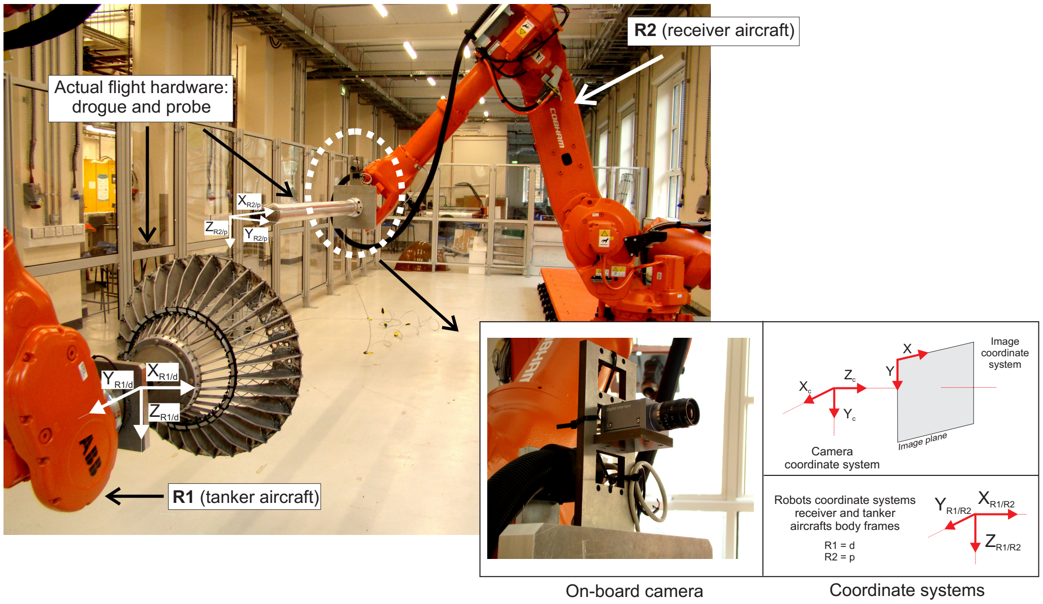


Figure 2: AAAR testbed: Probe and drogue mounted on two ABB IRB6640 robots

refuelling procedure, based on a closed loop F16 flight dynamics model. The model used to generate the motion data is outlined in Section 3. For the purposes of this paper, the robot motion is all open loop with predetermined flight paths. Future work will close the loop on the sensors for the purpose of control system development.

The robotic cell consists of two ABB IRB6640 robots, one fixed to the ground and one mounted on a 7.7 m IRBT6004 track. The cell is designed such that the track-mounted robot can place the probe anywhere in a working envelope defined by a cylinder 10m long and 2m in diameter. The fixed robot has a similar working cross section in which it can place the drogue, but it is limited in the longitudinal axis of the cylinder by the 2.55 m reach of the robot. Both the probe and the drogue can be positioned in any orientation and performance characteristics for the robots are given in Table 1.

The robots are driven by a proprietary ABB controller, which performs the coordinate transformations, kinematic computations, and regulates the current driving the six electric motors. Robot motions are prescribed using RAPID code, a purpose-made high level scripting language. For the tests described herein, a script reads position and orientation data from an ASCII file and executes the corresponding motion instructions at a rate of

Maximum acceleration	$\sim$2 g
Maximum relative velocity	$\sim$6 ms^{-1}
Operational envelope	$10\ \mathrm{m}\times\varnothing\ 2\ \mathrm{m}$
Pose accuracy	0.16 mm
Pose repeatability	0.07 mm
Pose stabilization time	0.36 s
Track length	7700 mm
Track maximum velocity	1.6 ms^{-1}
Track pose repeatability	0.08 mm

Table 1: Performance characteristics for the IRB6640 robots and the IRBT6004 track.

50 Hz. Generation of representative motion data to populate the ASCII file is described in Section 3. The target points provided in the data file define linear path segments, and the robot controller interpolates between the path segments in order to create a smooth path. Measurements of the *actual* robot positions along with time stamps provided by the robot controller are streamed over a TCP/IP connection at a rate of 50 Hz, and recorded using an external computer. It is these measurements that have been used in the comparative analysis in Section 5,5.2. The purpose of this paper is to evaluate the performance of the vision tracker, and hence the existing ABB RAPID interface was found to be sufficient, however for future, closed loop trials, an alternative real-time interface with reduced latency was found to be necessary.

A grey scale Sony XCD-V60 FireWire camera with an 8 mm focal length lens is mounted on robot R2, at the base of the probe, as shown in Figure 2. This camera is capable of capturing images at 60 Hz for both on-line and off-line tests. In the off-line tests, image data was stored at 30 Hz in order to reduce storage requirements; whilst the on-line tests utilised the full 60 Hz capabilities. The Sony camera was connected to a 2.7 GHz Intel core I7 MacBook Pro laptop, where the visual system ran using Ubuntu 11.04 as the operating system.

3. Simulation Environment

The motion data sets for the tracking experiments are generated using a simulation of the tanker aircraft, the receiver aircraft, and their surrounding environment, constructed using Mathworks' MATLAB and Simulink software. This simulation environment includes the aircraft flight dynamics models, control systems, and atmospheric turbulence effects. The data from two simulated approaches to the drogue are used for the purposes of this paper, the first is for light turbulence applied to the receiver aircraft, and the second for moderate turbulence. Positions of the probe tip relative to the drogue (${}^{\mathrm{d}}\mathbf{t}_{\mathbf{r}\mathrm{p}}$) are recorded in off-line runs for each case, and used as the trajectory input for the robot controller in order to replicate a six meter pre-contact approach to the drogue.

3.1. Aircraft models

For the purposes of this paper, the tanker is considered to be a rectilinear moving point with the drogue offset. An open-loop, six degrees of freedom, nonlinear model of an F-16 multi-role fighter was used as the receiving aircraft. The inputs to the receiver aircraft are the four primary control surfaces, δ_e, δ_a, δ_r, δ_t, relating to the elevator, aileron, rudder, and throttle positions respectively. Leading edge flaps and differential tail inputs were not included in the simulation. Equations relating to the aerodynamics of an aircraft in all six degrees of freedom are mathematically represented in the form:

$$C_{(\cdot)} = C_{(\cdot)_S} + C_{(\cdot)_D} + \Delta\delta_{(\cdot)} \tag{1}$$

which is the summation of the static (S) and dynamic (D) coefficients, and the effect from control surface deflection ($\Delta\delta_{(\cdot)}$). The aerodynamic coefficients are tabulated in lookup tables as functions of the angle of attack α, sideslip angle β, angular rates, and in some cases the control surface deflections. Values for these tables come from Stevens and Lewis [47], who presented a reduced version of the full range nonlinear F-16 model originally published by Nguyen *et al.* [48]. The reduced model is valid for $\alpha \in [-10, 45]$ degrees and $\beta \in [-30, 30]$ degrees, and wholly encompasses the flight envelope required for aerial refuelling.

Propulsive thrust is calculated in the engine model, originating from [47]. It is generated from lookup tables as a function of the current engine power demand c_P, the altitude, h, and the Mach number M. It acts along the X axis only and so there are no induced moments. For simplicity, the gyroscopic effects of the engine have been omitted from the aircraft model. The response of the engine power, P, is modelled using a first order lag:

$$\dot{P} = \frac{1}{\tau(r_P, P)}\Big(r'_P(r_P, P) - P\Big) \tag{2}$$

where the apparent desired power r'_P is dependent on the desired demand, r_P, and the current engine power. The desired power level is generated through the throttle gearing, modelled using the conditional function

$$r_P = \begin{cases} 64.95\delta_t, & \text{for } \delta_t \leq 0.77 \\ 217.38\delta_t - 117.38, & \text{for } \delta_t > 0.77 \end{cases} \tag{3}$$

The engine time constant is τ, is also a function of the actual and desired power, details of which can be found in [47]. All commands to the three primary control surfaces (the elevator, ailerons, and rudder) are passed through a rate limiter, first order lag filter with a time constant of 0.0495, and saturation limits in accordance with the actuator models described in [48].

3.2. Turbulence

Air turbulence was implemented using the following NASA Dryden power spectral densities [49]:

$$\left.\begin{aligned} \phi_u(\omega) &= \frac{2\sigma_u^2 L_u}{\pi U_0}\frac{1}{1+\left(L_u\frac{\omega}{U_0}\right)^2} \\ \phi_v(\omega) &= \frac{\sigma_v^2 L_v}{\pi U_0}\frac{1+3\left(L_v\frac{\omega}{U_0}\right)^2}{\left[1+\left(L_v\frac{\omega}{U_0}\right)^2\right]^2} \\ \phi_w(\omega) &= \frac{\sigma_w^2 L_w}{\pi U_0}\frac{1+3\left(L_w\frac{\omega}{U_0}\right)^2}{\left[1+\left(L_w\frac{\omega}{U_0}\right)^2\right]^2} \end{aligned}\right\} \tag{4}$$

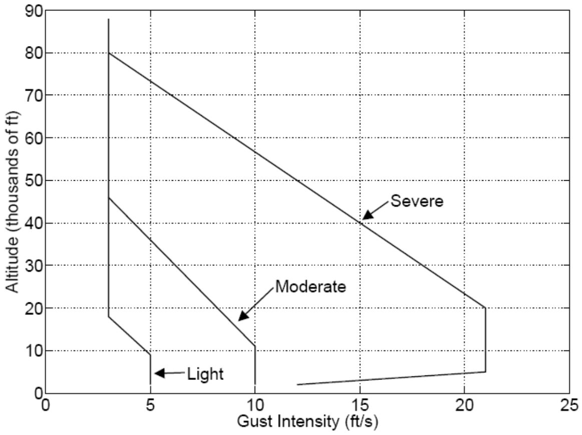


Figure 3: Gust intensity mapping for different aircraft altitudes and turbulence severities [50].

and

$$\left.\begin{aligned}
\phi_p(\omega) &= \frac{\sigma_w^2}{U_0 L_w} \frac{0.8\left(\frac{\pi L_w}{4b}\right)^{\frac{1}{3}}}{1+\left(\frac{4b\omega}{\pi U_0}\right)^2} \\
\phi_q(\omega) &= \frac{-\left(\frac{\omega}{U_0}\right)^2}{1+\left(\frac{3b\omega}{\pi U_0}\right)^2} \phi_v(\omega) \\
\phi_r(\omega) &= \frac{-\left(\frac{\omega}{U_0}\right)^2}{1+\left(\frac{4b\omega}{\pi U_0}\right)^2} \phi_w(\omega)
\end{aligned}\right\} \tag{5}$$

where $\sigma_{(\cdot)}$ are the gust intensities, $L_{(\cdot)}$ are the turbulence scales, U_0 is the still-air aircraft velocity, ω is the turbulence frequency and b is the wingspan.

Altitude and gust intensity were chosen in accordance with Figure 3 in order to satisfy the mathematical requirement for isotropic turbulence [50]. Equations (4) and (5) are solved in the time domain by transforming them into canonical state-space form so that the turbulent velocity components can be added to the aircraft's inertial velocity states prior to solving (1).

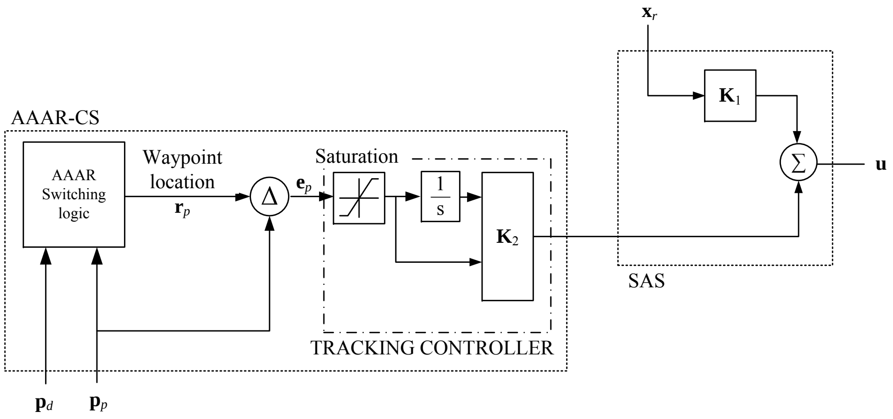


Figure 4: Control System (CS) implementation.

3.3. Control System

The receiver approaches the drogue using a control system which acts in order to minimise the position difference, $\mathbf{e}_p$, between the probe and a series of reference waypoints, $\mathbf{r}_p$, along the refuelling approach. The control structure used to achieve this is illustrated in Figure 4. AAAR switching logic dictates the active waypoint which the receiver follows on its approach to the drogue. Once selected position criteria have been satisfied, the next waypoint is activated. The series ends with the last waypoint being coincident with the position of the drogue; and the output from the control system provides the input demands $\mathbf{u}$ for the actuators and throttle command.

Controller gain matrices for both the Stability Augmentation System, SAS, and AAAR Control System ($\mathbf{K}_1$ and $\mathbf{K}_2$) were synthesised simultaneously using LQR. A standard cost function of the form

$$J = \frac{1}{2}\int_0^\infty (\mathbf{xQx}^\mathrm{T} + \mathbf{uRu}^\mathrm{T})\ \mathrm{d}t \tag{6}$$

was used with the state and input weightings, $\mathbf{Q}$ and $\mathbf{R}$, being used to optimise the controller gains. For the control system design, the F-16 model was linearised using MATLAB's `linmod` command at the desired operating point (V = 200 m/s and h = 8000 m, $\approx$ 26250 ft) in order to obtain the continuous time-invariant state space model of the receiver aircraft. State feedback was used in the trajectory generation with the assumption that perfect sensor data was available in order to describe the state of the receiver, probe and drogue positions. This simulation environment was then linked through a bespoke interface to the two RMR robots and used to move the flight hardware for the vision tracking tests, the results of which are given the following two sections.

4. Visual System

The objective of the visual system used in this work is to determine the relative three-dimensional position of the drogue with respect to the receiver aircraft. Due to the inherent difficulty of the task, fast, reliable and accurate relative position estimations are required in order to achieve a successful capture of the drogue.

The machine vision strategy proposed here encompasses a group of algorithms chosen under the criteria of robustness, efficiency, and simplicity. At the core of the vision system is the tracking algorithm, which is based on a hierarchical image registration based on direct methods. This algorithm provides a robust 2D position estimation of the drogue in the image plane using all the pixels related to the drogue. This estimation is considered to be robust under a large range of motion and with partial occlusions of the drogue.

In the implementation of the different techniques that were used, it is assumed that the camera is calibrated, the dimension of the drogue is known, and that the motion of the drogue in the image plane can be modeled using three parameters: the two-dimensional position coordinates in the image plane and a scale value. As an extension, one aspect of the test regime presented in this paper was to assess the behavior of the tracking algorithm when the motion of the drogue cannot be described by these three parameters.

The proposed algorithm was developed in C++ and the OpenCV libraries [51] were used for managing image data.

4.1. Strategy Overview

The proposed strategy contains four stages: detection, initialization, tracking, and 3D position estimation, as shown in Figure 5. The algorithm is initiated with a lost status L = 1 (i.e. no drogue has been detected). The detection stage is then used to find the region of interest (ROI) or image template ($\mathbf{T}$) corresponding to where the drogue is located in the first image $\mathbf{I}_0$. The coordinates that define this position in the image plane are found automatically using the detection strategy described in Section 4.3.

Once image $\mathbf{T}$ is found, the tracking algorithm is initialized (Section 4.4). In this initialization stage different components of the image registration algorithm are calculated and created (e.g. Hessian matrix; masks, Section 4.2). These steps are carried out every time the detection stage is activated (i.e. when L = 1).

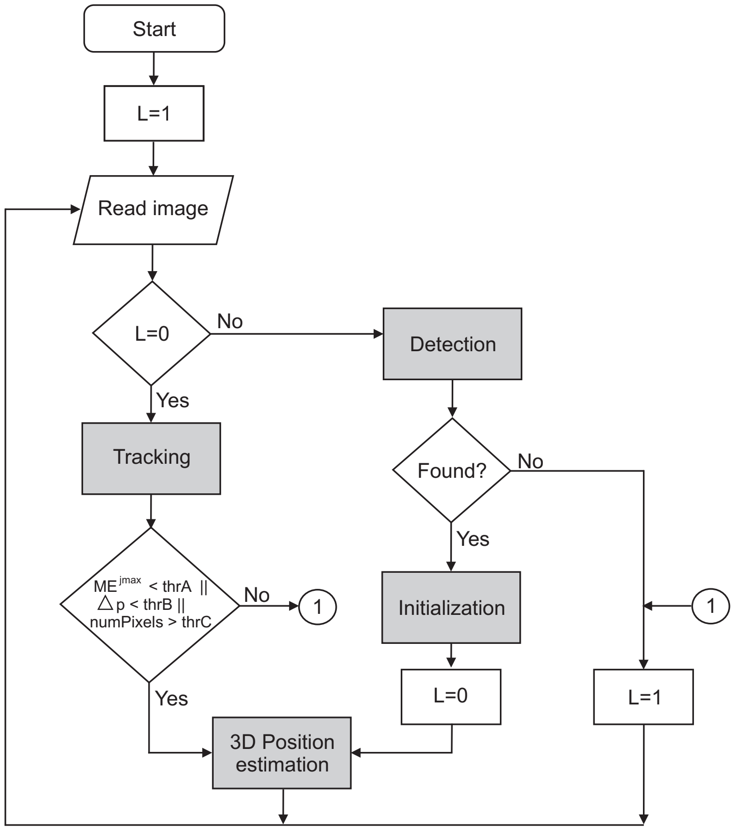


Figure 5: Proposed visual tracking system for AAAR tasks

When a new image is analyzed (e.g. $\mathbf{I}_1$), if L = 0, the tracking algorithm (described in Section 4.2) is used to estimate the transformation (or motion model) that describes the motion of the drogue in the image plane from image $\mathbf{I}_0$ to the image that is being analyzed (current image, e.g. $\mathbf{I}_1$). This motion model is also used to identify the location of the drogue in the current image.

As can be seen in Figure 5, to switch between the detection and tracking stages, performance assessment criteria are proposed to monitor the behavior of the tracking algorithm. Therefore, the result of the tracking stage is evaluated according to selected criteria. If one of those criteria is not satisfied, the lost status is activated (L = 1), and the detection stage will then be used until the drogue is found again. Conversely, if the criteria are satisfied, the estimated motion model is used to locate the drogue in the current image, and the tracking algorithm continues on to locate the drogue in the subsequent images.

Most of the time the tracking stage operates in isolation, as this is the most accurate and computationally efficient approach. If the tracking stage is unable to determine the position of the drogue with a high degree of confidence, then the algorithm switches to the detection stage. The criteria used to switch between those stages are defined as follows:

- $\mathrm{MAE}^{j_{max}} > \mathrm{thrA}$: the mean absolute error (MAE) at the lowest resolution level ($j_{\max}$) of the hierarchical structure in the tracking stage (Section 4.2) is below a threshold. Where $\mathrm{MAE}^{j_{max}}$ is defined as:

$$\mathrm{MAE}^{j_{max}} = \frac{\sum_{\mathbf{x}} |T^{j_{\max}}(\mathbf{x}) - I^{j_{\max}}(\mathbf{W}^{j_{\max}}(\mathbf{x};\mathbf{p}))|}{n^{j_{\max}}} \tag{7}$$

  where $\mathbf{x}$ are the pixel coordinates in image $\mathbf{T}^{j_{\max}}$, $\mathbf{W}^{j_{\max}}(\mathbf{x};\mathbf{p})$ are the pixel coordinates in image $\mathbf{T}^{j_{\max}}$ transformed to image $\mathbf{I}^{j_{\max}}$, and $n^{j_{\max}}$ is the total number of pixels. It was found that by controlling the MAE at the lowest resolution level it is possible to identify the moment when the tracking algorithm fails.

- $\Delta\mathbf{pos} > \mathrm{thrB}$: the difference between the current and the previous position of the drogue in the image plane is below a given threshold. This position is controlled by analyzing the change of the values of the ROI, that defines the position of the drogue in the image plane.

- % of pixels $<$ thrC: the percentage of pixels that are used in the minimization process is below a given threshold. This condition is particularly useful when the drogue goes out of the FOV of the camera or when it is occluded.

The previously mentioned criteria are used in order to set the flag L = 1, and also are used as performance assessment criteria in order to monitor the behavior of the tracking algorithm. The different thresholds used in this paper have been found experimentally. Section 5.1.2 presents an analysis of the different criteria and the values of the different thresholds used for the AAAR application.

Finally, once the 2D position of the drogue in the image plane is known either by using the detection or via the tracking algorithms, a 3D position estimation stage is used to calculate the three-dimensional position of the drogue with respect to the probe coordinate system. This estimation is found assuming that the camera is calibrated and that the dimension of the drogue is known (see Section 4.5).

4.2. Tracking Stage

The tracking algorithm is based on a hierarchical implementation of an image registration technique. The process of registration [52] consists in aligning two images, a reference image (the image template **T**) and a current image (**I**) by finding the transformation (motion model **W**) that best aligns them. This transformation is normally found iteratively by minimizing the Sum of Squared Differences (SSD) between images **T** and **I** [52]. A widely used approach is a gradient based optimization of the SSD, using a first order Taylor series expansion.

In this application the reference image corresponds to the image of the drogue **T** supplied by the detection stage during the initialization of the tracking task (i.e. every time the detection stage finds a new template image), and **I** corresponds to the image that is being analyzed (i.e. the current image). The tracking task then consists of an incremental image registration task, where the 2D position of the drogue in **I** is found assuming that an initial position of the drogue in the previous frame is known (i.e. the motion model is propagated to the next frame, as an initial estimation for the image alignment algorithm), and assuming that the 3D motion of the drogue projected in the image plane can be modeled by a 2D transformation [53].

The tracking algorithm that is used is a Hierarchical Multi-Parametric and Multi-Resolution implementation of the Inverse Compositional Image Alignment technique HMPMR-ICIA.

The ICIA algorithm [54] permits an efficient identification of **W**. However, this iterative algorithm relies on a linearization stage which is only valid when the range of motion is small (so that the first-order approximation can be valid – i.e. close enough – to find a minimum). In the current application, this assumption is not always applicable as the medium turbulence can produce large and sudden motion from one image to the next. In addition, the effects of this motion in the image plane increase during the final stage of the refuelling task (i.e. the closer the probe and the drogue are, the greater the perceived motion in the image). For this reason, this ICIA algorithm is used in an HMPMR structure.

As described in Figure 6, the strategy makes use of two hierarchical structures: the Multi-Resolution (MR) and the Multi-Parametric (MP) ones. The

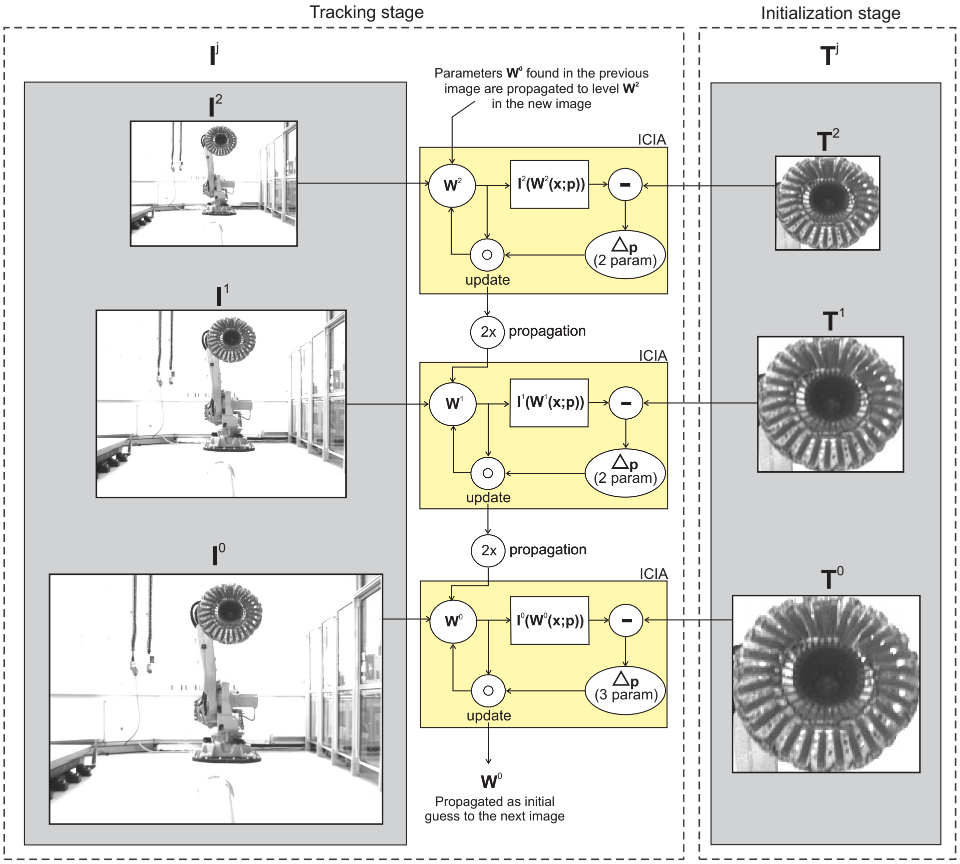

Tracking stage
Initialization stage
$\mathbf{I}^j$
$\mathbf{T}^j$
Parameters $\mathbf{W}^0$ found in the previous image are propagated to level $\mathbf{W}^2$ in the new image
$\mathbf{I}^2$
ICIA
$\mathbf{W}^2$
$\mathbf{I}^2(\mathbf{W}^2(\mathbf{x;p}))$
-
$\triangle\mathbf{p}$ (2 param)
update
$\mathbf{T}^2$
2x propagation
$\mathbf{I}^1$
ICIA
$\mathbf{W}^1$
$\mathbf{I}^1(\mathbf{W}^1(\mathbf{x;p}))$
-
$\triangle\mathbf{p}$ (2 param)
update
$\mathbf{T}^1$
2x propagation
$\mathbf{I}^0$
ICIA
$\mathbf{W}^0$
$\mathbf{I}^0(\mathbf{W}^0(\mathbf{x;p}))$
-
$\triangle\mathbf{p}$ (3 param)
update
$\mathbf{T}^0$
$\mathbf{W}^0$
Propagated as initial guess to the next image

Figure 6: HMPMR-ICIA. **I** and **T** are downsampled to create the MR structure. In each level, the ICIA algorithm is applied iteratively. The minimization is done with respect to the parameters defined at each level. At the highest resolution level the final estimation corresponds to the best transformation that locates the drogue in the current image.

general idea behind this strategy is that by estimating only a small number of parameters at the lowest resolution levels and smoothly increasing the complexity of the motion model through the MR pyramid, it is possible to obtain a robust estimation of the motion model.

The advantage of using the MR structure is that at low resolutions, the vector of motion is smaller and long displacements can be better approximated [46]. On the other hand, taking into account that at low resolutions, the quality and quantity of the available information does not allow a large number of parameters to be recovered (i.e. intensity values are smoothed due to the subsampling of the original image), by integrating the MP and

MR strategies the images with low resolution are used to estimate only small number of parameters (e.g. the translation motion model, 2 parameters), and the higher resolution images are used to refine these parameters and to estimate others (i.e. to estimate a more complex motion model).

The two hierarchical structures of the HMPMR strategy were created as follows. The MR structure is created by repeatedly downsampling the images by a factor of 2 [55, 56] in order to create the different levels (pL) of the pyramid. The number of levels pL has been defined as in [57], taking into account the size in pixels of the drogue (i.e. the size of image $\mathbf{T}$) when the visual system starts operating and the smallest dimension of $\mathbf{T}$ defined in the lowest resolution level. Therefore, the number of levels of the pyramid in our application are $pL = 3$, and so j is initialized as $j = \{2, 1, 0\}$.

As shown in Figure 6, the MR structure is accompanied by a concurrent MP analysis of the a motion model using the ICIA algorithm: for each level of the pyramid, a specific number of parameters to be estimate are defined.

The motion model chosen for the AAAR task has three parameters that define the positions x and y of the drogue in the image plane and the scale s. This motion model has been selected taking into account that during the tracking task the probe moves towards the drogue (i.e. there are scale changes), and that due to the drogue appearance (symmetric structure) rotations around the $\mathrm{Z_c}$ axis (roll motions) and small rotations around the other axes do not have a significant effect on the visual characteristics of the drogue in the image plane. This is why motion models that include rotational information have not been considered. It should also be noted that at this point, for the simulated trajectories, only the reference position of the drogue relative to the receiver aircraft has been used in within the control system structure. Orientation data has not been used.

The transformation that will map the pixels $\mathbf{x} = (x, y, 1)$ from image $\mathbf{T}$ to pixels $\mathbf{x}' = (x', y', 1)$ in image $\mathbf{I}$ can be defined as follows:

$$\mathbf{x}' = \mathbf{W}\,\mathbf{x} = \mathbf{W}(\mathbf{x}; \mathbf{p})$$

$$\mathbf{W} = \begin{bmatrix} 1 + p_1 & 0 & p_2 \\ 0 & 1 + p_1 & p_3 \\ 0 & 0 & 1 \end{bmatrix} \tag{8}$$

This 3×3 matrix (8) is parameterized by the vector of parameters $\mathbf{p} = (p_1, p_2, p_3)^{\mathrm{T}}$ in such a way that $\mathbf{W}$ is the identity matrix when the parameters are equal to zero. Therefore, p_2 and p_3 represent the translation in the X and

Y axes of the image coordinate frame shown in Figure 2, and p_1 represents the scale factor.

In the MP structure, the number of parameters increases with the resolution of the image. For this application therefore, the translation parameters (p_2 and p_3) defined in (8) will be estimated in the lowest resolution images ($j = 2$ and $j = 1$), and the whole motion model (p_1, p_2, and p_3) will be estimated using the highest resolution image at $j = 0$.

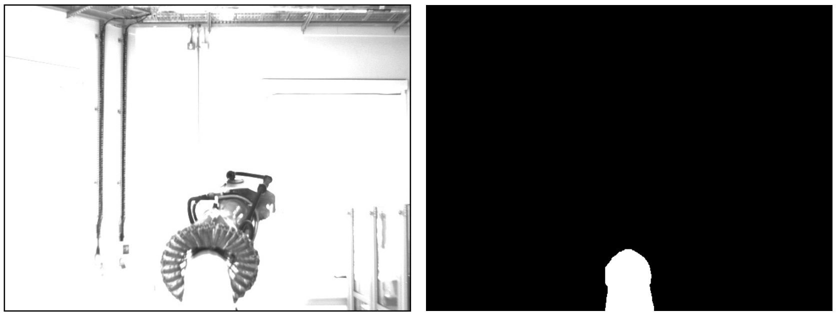
4.2.1. Hierarchical Multi-Parametric and Multi-Resolution ICIA algorithm

As shown in Figure 6, some information required by the tracking algorithm is calculated only once in the initialization stage (as explained in Section 4.4), such as the number of levels of the hierarchical structure, the MP structure, and the MR structure of $\mathbf{T}$, amongst others. After the initialization stage, a new image is acquired (i.e. $\mathbf{I}_1$). This image is then downsampled by a factor of 2 in order to create the MR structure (see Figure 6, tracking stage).

The tracking process starts at the lowest resolution level ($j = j_{\max} = 2$), as shown in Figure 6. At this stage, the motion model ($\mathbf{W}^{j_{\max}}$) in this level is initialized as the identity matrix (because this is the first frame to be analized). Then, as shown in Figure 6, the image coordinates $\mathbf{x}$ in $\mathbf{T}^2$ are transformed using $\mathbf{W}^2$ (the upper scripts represent the level), and the ICIA algorithm presented by Ishikawa, Matthews and Baker [58] adapted to the drogue tracking task is used in order to minimize:

$$\sum_{\forall \mathbf{x} \in \mathbf{T}^j : P^j(\mathbf{W}^j(\mathbf{x};\mathbf{p}))=0} M^j(\mathbf{x})\left[T^j(\mathbf{W}^j(\mathbf{x};\Delta\mathbf{p})) - I^j(\mathbf{W}^j(\mathbf{x};\mathbf{p}))\right]^2 \tag{9}$$

where $\mathbf{M}^j$ is a constant mask (as defined in the initialization stage Section 4.4), which is used to ensure that in the minimization process only the pixels in $\mathbf{T}^j$ that belong to the drogue are used (the drogue is circular, as a consequence not all the pixels in image $\mathbf{T}^j$ should be used); and $\mathbf{P}^j$ is another constant mask (see Figure 7), created manually when the visual system starts operating (for these tests, the camera is always located in a defined and fixed position with respect to the probe). This mask $\mathbf{P}^j$ is used to exclude the pixels of the probe when it is approaching the drogue, as shown in Figure 7. Therefore, $\mathbf{P}^j$ is used to select the pixels in the image $\mathbf{I}^j$ that are included in the minimization process: only those pixels $\mathbf{x}' = \mathbf{W}^j(\mathbf{x};\mathbf{p})$ whose intensity values in $P^j(\mathbf{W}^j(\mathbf{x};\mathbf{p}))$ are equal to 0 are considered. This mask was shown to play an important role in the stability of the tracking algorithm, because during the final stage of the refuelling approach the drogue will almost always be occluded by the probe (see Figure 7).


Probe occluding part of the drogue

Mask **P**

Figure 7: Probe mask **P**. This constant mask excludes the pixels that belong to the probe.

As can be seen in Figure 6, in the lowest resolution level ($j = 2$), the ICIA algorithm is applied: the error between $T^2(\mathbf{x})$ and $I^2(\mathbf{W}^2(\mathbf{x};\mathbf{p}))$ is calculated, the increment of the parameters is found after a first-order Taylor series expansion of (9), and the motion model is updated as follows: $\mathbf{W}(\mathbf{x};\mathbf{p}) \leftarrow \mathbf{W}(\mathbf{x};\mathbf{p}) \circ \mathbf{W}(\mathbf{x};\Delta\mathbf{p})^{-1}$

The ICIA algorithm iteratively updates the parameters, until stopping criteria are reached denoting the best local alignment solution. In this application, three criteria were used: the minimum is reached if the increment of the parameters is below a threshold ($||\Delta\mathbf{p}|| \leq 10^{-5}$), if the MAE does not decrease after a defined number of iterations (10 iterations), or if the maximum number of iterations have been reached (100 iterations).

When the stopping conditions of the ICIA have been reached, the parameters are propagated to the next level of the pyramid ($j = 1$) taking into account that the images have been downsampled by a factor of 2, as follows:

$$\begin{aligned} p_1^{j-1} &= p_1^j \quad for \qquad i = 1 \\ p_i^{j-1} &= 2p_i^j \quad for \quad i = \{2,3\} \end{aligned} \tag{10}$$

where the subscript i represents the parameters of the motion model defined in (8). The process is then repeated at each level of the MR structure minimizing only with respect to the parameters defined for that level. At the lowest level of the pyramid $j = 0$ (i.e. the highest resolution level), the motion model (8) is estimated. The result at this level corresponds to the best transformation that can be applied in order to locate the drogue ($\mathbf{T}$) in the current image ($\mathbf{I}$).

As can be seen in Figure 6, when another image is acquired (i.e. $\mathbf{I}_2$), the estimated motion model ($\mathbf{W}^{j_{\max}}$) in the previous image (i.e. $\mathbf{I}_1$) is propagated as an initial guess of the motion model at the lowest resolution level ($j = 2$) of the new image $\mathbf{I}_2$. This propagation of the parameters from the lowest level of the pyramid in the previous image ($\mathbf{I_1}$) to the highest level of the pyramid in the new image ($\mathbf{I_2}$) normally makes $\mathbf{T}$ and $\mathbf{I}$ in the new image close enough to find a minimum.

4.3. Detection Stage

The detection stage is used to automatically detect the drogue every time the tracking algorithm is initialized (i.e. every time L = 1), either at the start of the run or when the drogue has gone out of the FOV of the camera; or because the tracking algorithm has failed to track the drogue.

The detection stage is composed of two algorithms, one based on a basic template matching (TM) algorithm (see Figure 8), and the second one based on image segmentation (see Figures 9(a) and 9(b)). When using these algorithms, it is assumed that the center of the drogue is within the field of view of the camera, so that the visual characteristics can be used to define criteria for verifying that the object that has been detected is the drogue.

The template matching algorithm that is used is the one implemented by Bradski and Kaehler [51]. This TM algorithm is applied over edge images in order to improve the matching stage, taking into account that the structure of the drogue contains important edge information that can be exploited to avoid mismatches. Thus, before applying the TM algorithm edges are found in the images using the sobel operator [59], as shown in Figure 8. Additionally, taking into account that the TM method is computationally expensive, low resolution images are used ($\frac{ImgSize}{2}$) in order to alleviate this problem.

Therefore, the TM algorithm consists in sliding a reference image $\mathbf{I}^k_{\text{e\_ref}}$ over an input image $\mathbf{I}_\text{e}$, and calculating the quality of the match according to the normalized cross correlation (NCC) method (using a normalized method variations in lighting levels are accounted for), as follows:

$$R^k_{\text{ccorr}}(u,v) = \frac{\sum_{x,y}[I^k_{\text{e\_ref}}(x,y)I_\text{e}(x+u,y+v)]^2}{\sqrt{\sum_{x,y} I^k_{\text{e\_ref}}(x,y)^2 \sum_{x,y} I_\text{e}(x+u,y+v)^2}} \tag{11}$$

Where $I^k_{\text{e\_ref}}(x,y)$ and $I_{\text{e}}(x+u,y+v)$ represent the intensity values of the edge images at positions (x,y) and $(x+u,y+v)$, and k represents the reference image that is analyzed. The reference images correspond to different images of the drogue that have been captured, cropped manually, and stored off-line (10 images, $k = \{1...10\}$). These images contain the drogue with different variations, such as scale, illumination, and position, as shown in Figure 8, images $\mathbf{I}^{\text{k}}_{\text{ref}}$. On the other hand, $\mathbf{R}^k_{\text{ccorr}}$ is the NCC image (see Figure 8) that contains the correlation coefficients $R^k_{\text{ccorr}}(u,v)$ of each shift in position. Thus, u and v define the offset of the reference image k relative to the image origin of $\mathbf{I}$ (each $\mathbf{I}^k_{\text{e\_ref}}$ is slid over $\mathbf{I}_{\text{e}}$).

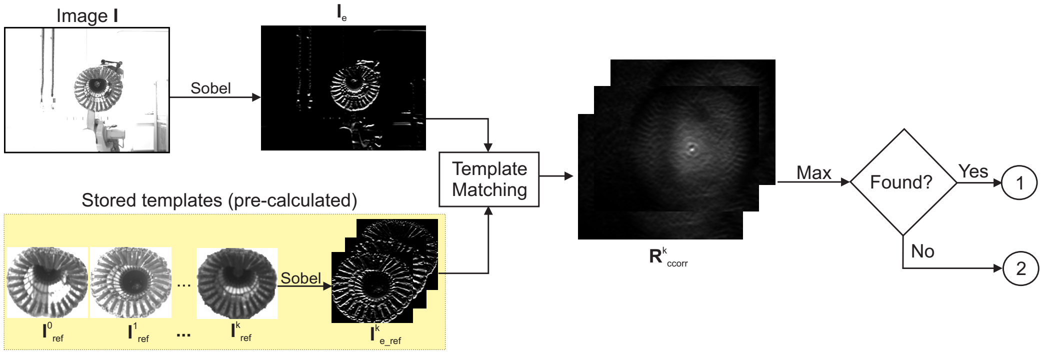
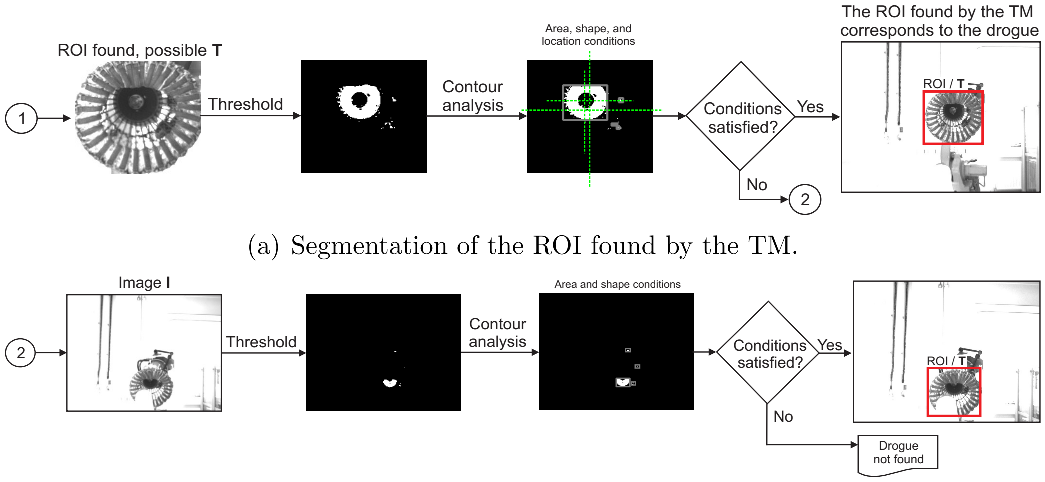


Figure 8: Template matching (TM). The TM consists in sliding reference images $\mathbf{I}^k_{\text{e\_ref}}$ over the current image $\mathbf{I}_{\text{e}}$, and calculating the match using NCC. The results are stored in different $\mathbf{R}^k_{\text{ccorr}}$ images. If a high NCC coefficient is found, different strategies are used to find the position of the drogue (see Figure 9).

The $\mathbf{R}^k_{\text{ccorr}}$ images are used to determine if a drogue has been found in $\mathbf{I}$. A found status is determined if there is a $\mathbf{R}^k_{\text{ccorr}}$ image that contains a $R^k_{\text{ccorr}}(u,v) > ccThr$, where $ccThr$ is a threshold that has been found experimentally. Subsequently, from all the images that satisfy this condition, the one that contains the highest NCC coefficient is used to extract the coordinates of the drogue.

Depending on whether the $R^k_{\text{ccorr}}(u,v) > ccThr$ condition is satisfied or not, different strategies are used to determine the position of the drogue, as shown in Figure 9. If the $R^k_{\text{ccorr}}(u,v) > ccThr$ condition is satisfied, a second algorithm is applied (see Figure 9(a)). This algorithm uses as an input the region of interest (ROI) found by the TM algorithm within which the drogue may lie. The main objective of this second algorithm is to verify that the object found is the drogue, by analyzing the ROI with respect to the characteristic features of the drogue. The key characteristic used in this study is the dark circular zone, which should be located approximately in the center of the region found by the TM algorithm, as shown in Figure 9(a).



(a) Segmentation of the ROI found by the TM.

(b) Segmentation of the entire image.

Figure 9: Segmentation stage. If a maximum in the TM strategy is found, the center of the drogue is segmented (Figure 9(a)) and analyzed to ensure that the detected area is the drogue. If the TM does not find a maximum or the area found does not correspond to the drogue, the whole image is segmented and analyzed (Figure 9(b)).

The inner part of the drogue is imaged as the darkest area of the drogue, as can be seen in Figure 9(a). Using this, the center of the drogue is segmented applying a fixed threshold, found experimentally, over the ROI image found by the TM method. Then, the contours of the objects in the segmented image are extracted [51] and analyzed using the following conditions:

- Area condition: if the area of the inner part of the drogue is within a range found experimentally.
- Shape condition: if the found contour corresponds to a circle. This condition is evaluated analyzing the shape of the fitted rectangle.
- Location condition: if the object found is the drogue, its darkest area should be located in the center of the ROI found by the TM algorithm.

If all of these conditions are satisfied, the ROI found by the TM algorithm is considered the one that corresponds to the location of the drogue in the image (see Figure 9(a)), and will define the template image $\mathbf{T}$ used in the tracking stage (Section 4.2).

There are cases, however, where the template matching algorithm can fail. For example, because it does not find an area with a high correlation score (Figure 8), or because the location found does not correspond to the drogue (Figure 9(a)). Under these circumstances, a second strategy is used to find the drogue. The strategy, as can be seen in Figure 9(b), follows the same idea of the segmentation strategy explained previously. The differences in this case, however, are that the threshold is applied to the entire image, and that the location condition is not applied. Additionally, the previous segmentation strategy was only used to ensure that the area found by the TM algorithm was the area where the drogue was located, whereas in this second algorithm, the area that describes the drogue is defined using the information of the segmented dark area (see Figure 9(b)).

It is important to highlight that all the methods described in this section comprise an important part of the overall tracking strategy. Their interaction allows the identification of the drogue when: the tracking algorithm fails, the drogue is out the FOV (Field of View) of the camera, or when the tracking task is first initialised. Figures 8 and 9 describe the way these strategies interact, and the resulting images at each stage.

4.4. Initialization Stage

This stage encompasses the initialization of the different variables required by the tracking algorithm which is the core of the visual system. When the template image $\mathbf{T}$ is detected for the first time or is changed, some elements of the tracking algorithm must therefore be recalculated, the mask $\mathbf{M}$ in (9) must be created, and the MR structures of $\mathbf{T}$ and $\mathbf{M}$ are to be created. The MR structures of these images are created by downsampling the images by a factor of 2 according to the different pyramid levels.

On the other hand, as shown in Figure 10, due to the circular shape of the drogue, $\mathbf{T}$ contains background information that should not be considered during the tracking task, because the information can affect the behavior of the tracking algorithm (values may change during the task). For this reason, every time a new image $\mathbf{T}$ is found, the mask $\mathbf{M}$ must be initialized.

The mask $\mathbf{M}$ is used in the tracking stage to define the pixels of the template that are going to be used in the in the minimization of (9). Two options were analyzed in order to create the circular mask $\mathbf{M}$:

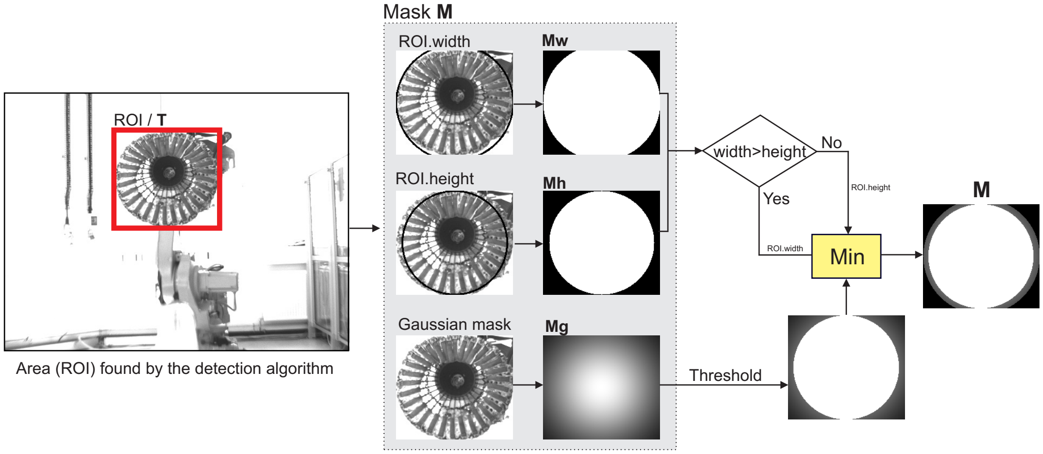


Figure 10: Drogue mask **M**. Background information is excluded using this mask. **M** is created by combining a Gaussian mask with the mask obtained using the ROI found in the detection stage.

- Based on the detected ROI: using the width or height information of the ROI where the drogue is located. If the smallest value is used, then important information could be lost (see Figure 10, images ROI.height and **Mh**). If the largest value is used, then background pixels could be included in the minimization of (9). See Figure 10, images ROI.width and **Mw**.

- Based on a gaussian mask: the idea of this mask is to give less weight to the pixels located at the corners of the template (background pixels), and higher weight to the ones located in the center. However, by only using this mask, less weight would be given to pixels that are known to correspond to the drogue (see Figure 10, image Gaussian mask and **Mg**).

Taking into account that there can be occlusions by the probe or situations where part of the drogue is out of the FOV of the camera, it is considered to be important to use as many pixels that belong to the drogue as possible in the tracking stage, without compromising the frame rate of the algorithm. This is why, in order to create **M**, both previously mentioned options have been combined.

As shown in Figure 10, by combining both options it is possible to obtain the mask **M**, that includes most of the pixels of the drogue whilst giving less weight to those pixels located in the boundaries. It should be noted that the mask based on the ROI that is chosen to create **M** is the one that uses the maximum value between width and height values of the ROI. As can be seen in Figure 10, the **Mw** mask is the one that contains the largest number of drogue pixels. The few background pixels that it includes will receive less weight when combining this mask with the Gaussian mask.

4.5. 3D Position Estimation Stage

The 2D positions of the drogue in the image plane (see Figure 11) obtained either in the visual tracking or the detection stages are transformed into 3D positions assuming that the dimension of the drogue is known (the diameter is known), that the 3D points of the drogue lie on a plane as shown in Figure 11, and that the camera calibration parameters [60] (optical center, focal length, etc.) are also known.

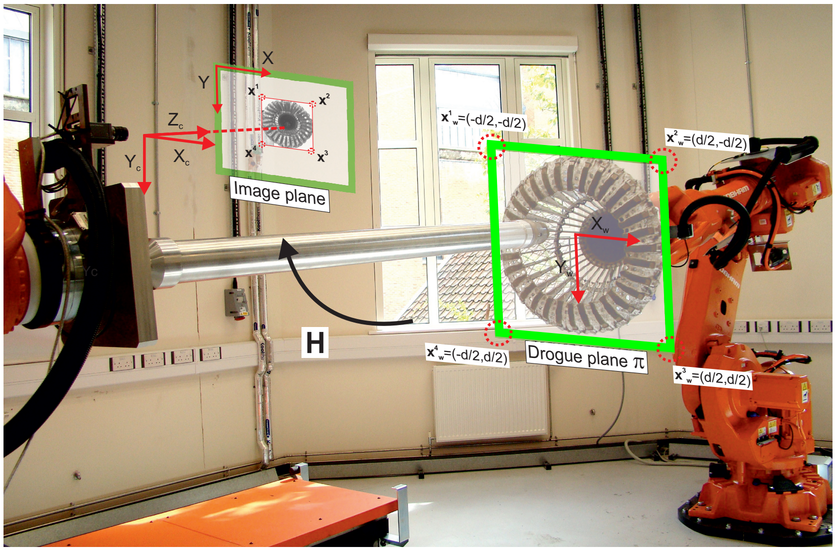


Figure 11: Position estimation strategy. The 2D positions of the drogue in the image plane are transformed into 3D positions assuming that diameter of the drogue is known, the known 3D points lie on a plane, and the camera calibration parameters are known.

Using the pinhole camera model [53], 3D coordinates can be related to the 2D image coordinates, as follows:

$$\mathbf{x} = \lambda \mathbf{K}[\mathbf{R} \mid \mathbf{t}]\mathbf{x}_{\mathrm{w}} \tag{12}$$

Where $\mathbf{x} = (x, y, 1)$ are the 2D image coordinates of a point, $\mathbf{x}_\mathrm{w} = (x_\mathrm{w}, y_\mathrm{w}, z_\mathrm{w}, 1)$ are 3D world coordinates of the same point, λ is a scale factor, $\mathbf{R}$ and $\mathbf{t}$ are the orientation and position of the world reference frame in the camera coordinate system, and $\mathbf{K}$ is the camera calibration matrix found by an off-line calibration process [60] using the camera calibration toolbox for Matlab[61].

As shown in Figure 11, the known diameter of the drogue can be used to define a plane where the world coordinate system is defined, and so four 3D points that lie on this plane can also be defined. With these four 3D points and the four 2D points of the ROI that describe the drogue in the image plane, expression (12) is simplified for the planar case (being $z_\mathrm{w}^i = 0$), as follows:

$$\begin{gathered} \begin{pmatrix} x^i \\ y^i \\ 1 \end{pmatrix} = \lambda \mathbf{K}[\mathbf{r_1}\ \mathbf{r_2} \mid \mathbf{t}] \begin{pmatrix} x_\mathrm{w}^i \\ y_\mathrm{w}^i \\ 1 \end{pmatrix} \\ \mathbf{x}^i = \mathbf{H}\mathbf{x}_\mathrm{w}^i \end{gathered} \tag{13}$$

where $\mathbf{x}_\mathrm{w}^i$ contains the coordinates of one of the four points that lie on the plane π, the index i represents each corner of the ROI that inscribes the drogue in the image and in the world planes ($i = \{1, 2, 3, 4\}$), and $\mathbf{H}$ is the planar homography (a 3×3 matrix) that transforms points in the world plane into points in the image plane, as shown in Figure 11.

Therefore, with the point-to-point (2D-3D) correspondence of the four corners of the drogue and reorganizing (13), a system of equations of the form $\mathbf{Ah} = \mathbf{b}$ can be created, where $\mathbf{h}$ corresponds to the components of $\mathbf{H}$ stacked into a vector. Therefore, $\mathbf{H}$ can be estimated.

Once $\mathbf{H}$ is estimated, the translation vector $\mathbf{t}$ is found based on (13) and taking into account that $\|\mathbf{r_1}\| = \|\mathbf{r_2}\| = 1$, as follows:

$$\begin{gathered} \mathbf{H} = [\mathbf{h_1}\ \mathbf{h_2}\ \mathbf{h_3}] = \lambda \mathbf{K}[\mathbf{r_1}\ \mathbf{r_2}\ \mathbf{t}] \\ \lambda = \|\mathbf{K}^{-1}\mathbf{h_1}\| = \|\mathbf{K}^{-1}\mathbf{h_2}\| \\ \mathbf{t} = \tfrac{1}{\lambda}\mathbf{K}^{-1}\mathbf{h_3} \end{gathered} \tag{14}$$

Using (14), the 3D position of the drogue (subscript d), or in our testbed, the position of R1 (subscript R1) with respect to the camera coordinate system (superscript c), is found $\mathbf{t} = {}^\mathrm{c}\mathbf{t}_{\mathbf{v}\mathrm{d}} = {}^\mathrm{c}\mathbf{t}_{\mathbf{v}\mathrm{R1}}$, where $\mathbf{t_v}$ refers to a vision-based estimation.

5. Results and Discussion

5.1. Tracking Evaluation

In this section, the performance of the tracking algorithm described in Section 4 is analyzed in three different conditions. In the first test, the advantages of including the masks $\mathbf{M}$ and $\mathbf{P}$ in the tracking algorithm are shown. A second test is conducted in order to define the limits of the tracking algorithm in terms of speeds and perturbations that under which it can continue to function correctly. Additionally, in a third test an analysis of the selection of the different proposed performance assessment criteria is presented.

5.1.1. Test 1: advantages of using the masks $\mathbf{M}$ *and* $\mathbf{P}$

In this test, an image sequence that contains different movements of the drogue and the probe is used to analyze the advantages of including the masks $\mathbf{M}$ and $\mathbf{P}$ presented in Section 4. The analysis of the results is done visually, by analyzing the ROI (red/dark square) found by the tracking algorithm.

The image sequence includes: basic motions of the drogue (moving left, right, up and down) where the background information changes (e.g. Figure 12, frames 236 and 2504); spiral movements of the probe that cause, in some situations, occlusions of the drogue by the probe (e.g. Figure 12, frames 2716 and 2720); and inclinations of the drogue that produce changes in the appearance of the drogue (e.g. Figure 12, frame 2782, due to the inclination, the black circular center of the drogue is not well perceived).

The first row of Figure 12 presents the results of the tracking task when the masks are not included in the tracking algorithm, and the second row shows the results when these masks are included. As can be seen in these images, by including the masks ($\mathbf{M}$ and $\mathbf{P}$) the performance of the tracking algorithm is markedly improved. In all the situations identified here where the algorithm without the mask failed, the algorithm that included the masks was able to successfully track the drogue.

With the mask $\mathbf{M}$ the background information that surrounds the drogue is excluded, which is why changes in the background information do not affect the performance of the algorithm (see e.g. Figure 12, frames 236, 2504, and 2782).

On the other hand, by including the mask $\mathbf{P}$, the pixels that belong to the probe are also excluded from the minimization process. Therefore, the

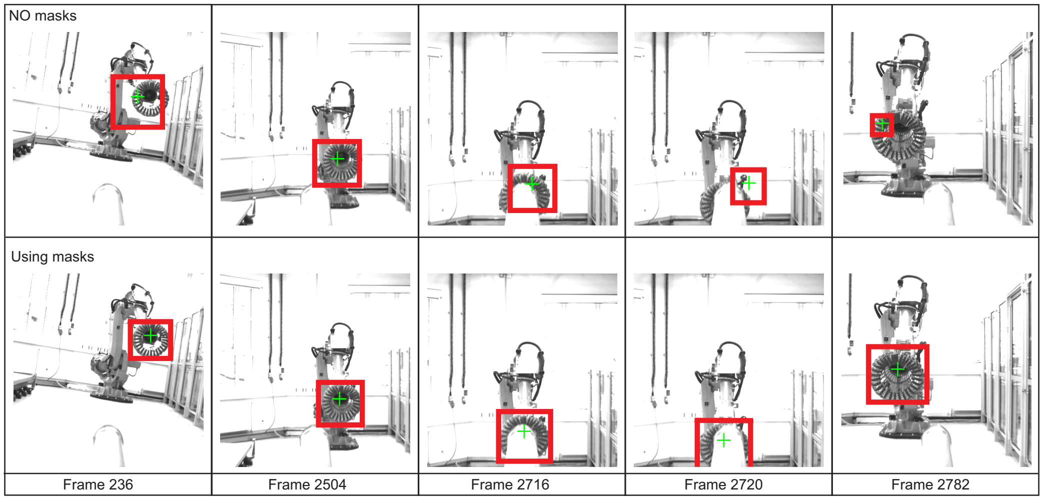


Figure 12: Tracking results including masks **M** and **P**. The first row shows the results when the masks are not included. The second row shows that the performance of the algorithm, tracking the drogue, markedly improves when using the masks defined in Section 4.

algorithm is more able to tolerate this kind of occlusion, as can be seen in Figure 12, frames 2716 and 2720.

5.1.2. Test 2: limits of the tracking algorithm

This second test analyzes the limits of the tracking algorithm using specific motions of the probe and drogue performed at a range of different speeds and with varying extent. The five motions are: drogue movements in the X_{R1}, Y_{R1}, and Z_{R1} directions at three different speeds: 400 mm/s, 1000 mm/s, and 2000 mm/s; a wave motion, in the form of combined sinusoidal variations of the vertical translation and the pitchwise orientation using three different angle ranges: $\pm 5°$, $\pm 10°$, and $\pm 20°$; and a spiral movement of the probe in the transverse plane combined with a steady approach towards the stationary drogue.

The first three sets of tests, in the X_{R1}, Y_{R1}, and Z_{R1} directions, are intended to identify the extent of motion in each direction that can be accommodated by the algorithm between two image frames before the accuracy of the tracking is degraded. In the fourth test, the changing inclination of the drogue is used to examine the response of the tracking system to a motion that violates the assumptions made in the development of the algorithms: that the motion of the drogue can be described by two translational parameters and a scale factor that represents the longitudinal translation (8). Finally, the spiral approach exhibits representative occlusions of the drogue by the probe as are likely to be seen in a real refuelling exercise.

For each of the previously mentioned situation, the motion was repeated three times, and the tests were performed off-line with the tracking algorithm running continuously throughout each test. Figure 13, presents examples of the tracking task performance for the different motion cases and the limits found of the tracking algorithm.

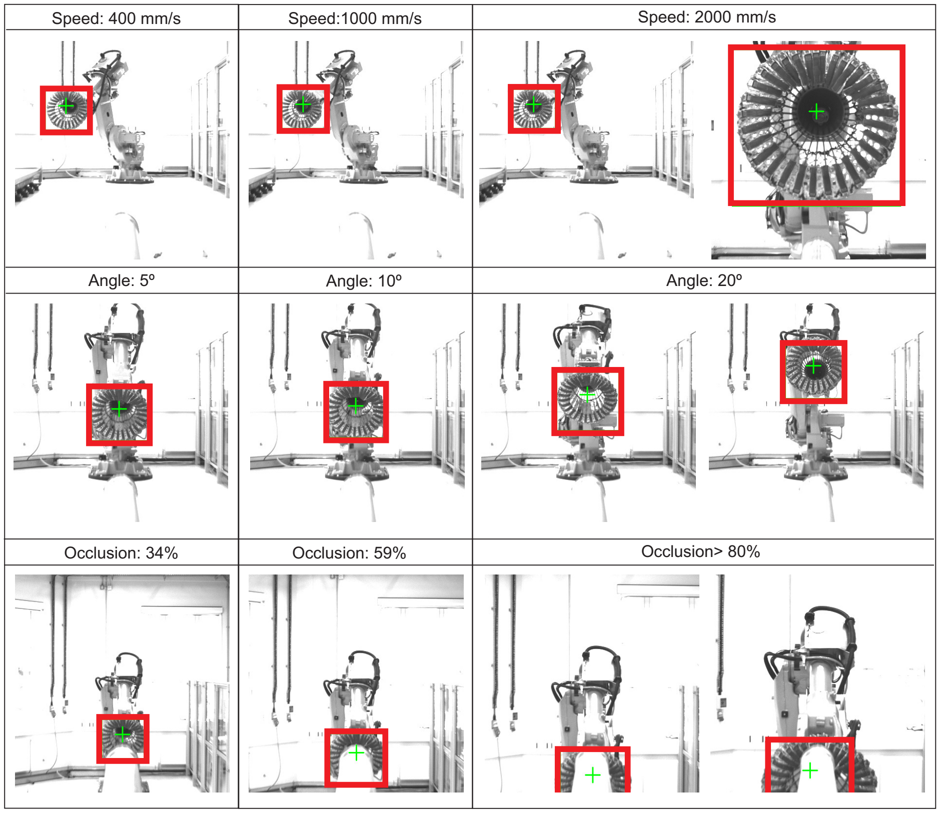


Figure 13: Limits Description. Different motions were tested: different 3D movements at different speeds (first row), wave movements with different angles (second row), and spiral movements that produced different type of occlusions (third row).

In the case of changes in position, as can be seen in Figure 13 first row, the algorithm presented an unstable behavior (the red/dark box is shifted to the right) in some parts of the sequence when the speed changed from 400 to 2000 mm/s (when moving in the plane Y_{R1}, Z_{R1}). This instability was detected always at the same point of the sequence (drogue moving to the left in the image plane). Because of this, it is suggested that this is due to a change in the appearance of the drogue at those points rather than because of a change in the speed. Nonetheless, in spite of these minor instabilities, the algorithm tracked the drogue throughout the whole sequence when the changes in position were applied in the $\mathrm{Y_{R1}}$ and $\mathrm{Z_{R1}}$ axes, and also when significant changes in scale were applied at the three different speeds (in the longitudinal $\mathrm{X_{R1}}$ axis, the position changes from 0 m to 5 m). This range of speeds was chosen as being representative of the range of speeds that the camera image would be subjected to in an actual refuelling scenario.

When the wave movements were conducted (see Figure 13, second row), the performance of the algorithm was affected when the inclination of the drogue increased from 10° to 20°. At these angles, the appearance of the drogue changes drastically, and the exact position of the drogue cannot be found using the existing methods. It is important to note however for rotations of less than 10°, the performance of the tracking algorithm was found to be robust and accurate. Whilst further work could extend the algorithms used to allow for tracking through greater rotations of the drogue, these are not expected to be present in a typical refuelling scenario.

Finally, Figure 13 third row, shows the results of the algorithm during spiral movements. As can be seen, one of the advantages of the proposed strategy based on direct methods, is that by using the information of all the pixels that belong to the drogue, the algorithm is able to continue tracking the drogue during partial occlusions, or when part of the drogue is out the FOV of the camera. Nonetheless, when the percentage of pixels occluded in $\mathbf{T}$ was greater than 80%, or in other words, when the percentage of pixels in $\mathbf{T}$ that are used in the minimization process of the tracking algorithm was lesser than 20%, the drogue was not tracked correctly (see Figure 13, third row). These percentages were calculated taking into account the number of pixels in $\mathbf{T}$ that were excluded in the minimization process, hence the pixels occluded by the probe and the pixels out of the FOV of the camera.

5.1.3. Test 3: performance assessment and switching criteria

As mentioned in Section 4, three performance assessment criteria are proposed in order to evaluate the performance of the tracking algorithm, that are also used to switch between the tracking and detection stages. The criteria have been selected experimentally, and some examples of the different situations analyzed are given in Figure 14.

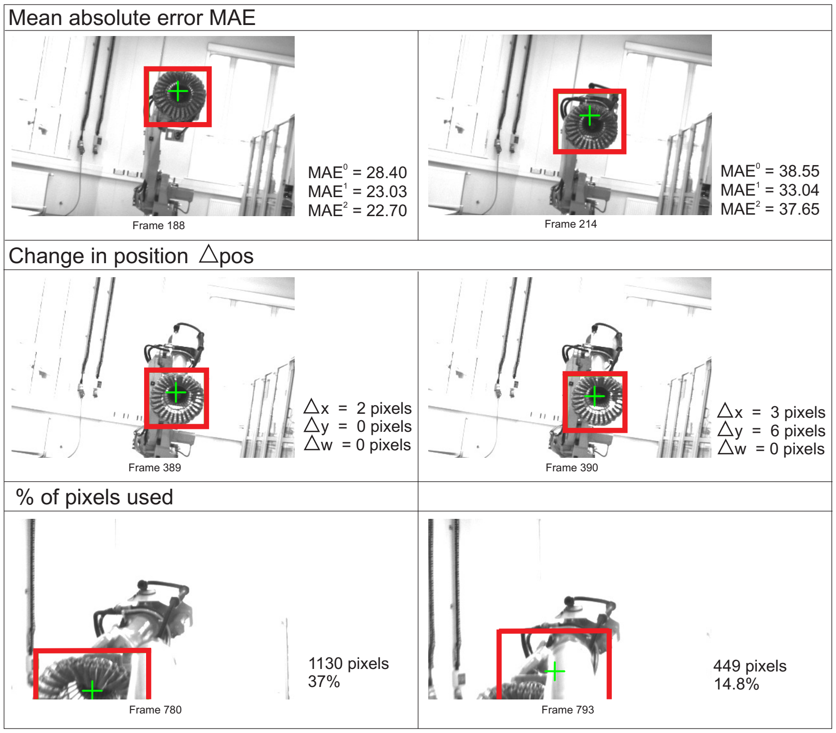


Figure 14: Performance assessment and switching criteria. These three criteria are used to detect when the tracking algorithm fails. They are used to evaluate the tracking algorithm, and are also used as switching criteria between the tracking and detection stages.

Under ideal conditions (e.g. no appearance or illumination changes), the MAE (Mean Absolute Error) can be a good measurement of the performance of the tracking algorithm. However, it was found that there are some situations where the distribution of intensity values in parts of the image (e.g. Figure 14, first row, frame 214) are similar to the ones of the drogue (**T**), whereupon the MAE(s) are low (frame 188 has similar MAE as frame 214) but the object that is tracked does not correspond to the object of interest (e.g. in frame 214 part of the background is identified as part of the drogue), e.g. when part of the tanker aircraft is imaged as part of the background of the drogue. This situation led to the use of additional criteria in order to evaluate the performance of the tracking algorithm.

The second criterion considered is the change in position of the ROI that inscribes the drogue. During the refuelling approach, the image of the drogue will smoothly increase in size (changes in scale), and the position of the drogue in the image plane will have drastic changes only due to turbulence effects. Nonetheless, using the HMPMR approach and considering that the algorithm runs at a frequency > 30 fps (frames per second), it is possible to see relatively large changes in position as smooth variations in the image plane. Taking this into account, it was found that incorrect results generated by the tracking algorithm can be detected by tracking incremental changes in position in the image plane.

In the images of the second row of Figure 14 it can be seen that during normal operation of the tracking algorithm, the increments in position are small (frame 389, $\Delta x = 2$). Nonetheless, in frame 390 of Figure 14 (second row, right image) it can be seen that the increment in the Y axis gives an indication of incorrect results generated by the tracking algorithm.

A third criterion was selected to deal with situations when the drogue leaves the FOV of the camera (e.g. under strong turbulence effects), and also when it is occluded by the probe. As can be seen in the third row of Figure 14, when only a few transformed pixels of $\mathbf{T}$ lie on $\mathbf{I}$, the lack of information causes the tracking algorithm to fail. In the previous test (test 2: limits of the tracking algorithm), we could also see that when the percentage of pixels occluded was greater than 80% the drogue was not tracked correctly (see Figure 13, third row).

All the different thresholds mentioned in Section 4, that define the different criteria described above have been found experimentally. For the $\text{MAE}^{j_{\max}}$, thrA $= 50$. For the change in position in the X and Y axis thrB $= 8$ pixels, and for the scale thrB $= 5$ pixels (scale is controlled using the width of the ROI). Finally, thrC that represents the percentage of the total number of pixels in $\mathbf{T}$ that after being transformed ($\mathbf{x}' = \mathbf{W}(\mathbf{x}; \mathbf{p})$) lie inside image $\mathbf{I}$, was selected to be thrC $= 15$.

5.2. Aerial Refuelling Tracjectory Tests

The different components of the drogue tracking strategy presented in Section 4 are tested using motions representative of an aerial refuelling task using the AAAR testbed presented in Section 2. The simulation environment presented in Section 3 provides the motion data for the task, using two scenarios: light turbulence and moderate turbulence. The drogue is kept stationary for these tests, and motion of the probe is used exclusively to reproduce the relative motion of the two bodies.

During the tests adverse conditions are presented in the image sequences, including the drogue being out of the FOV, changes in appearance (e.g. pitch and roll effects cause the aspect of the drogue to change in the image plane), occlusions (drogue occluded by the probe), and sudden, rapid motions. The evaluation of the visual system during the refuelling tasks is based on the analysis of the performance assessment criteria defined in Section 4 and Section 5, based on a visual examination of the tracking results, and also based on a comparison of the estimated motion from the vision algorithm with the recorded position data from the robots. This later comparison is evaluated using the RMSE (Root Mean Square Error) between the data.

On-line and off-line tests of the visual system during AAAR tasks have been conducted using the AAAR testbed (Section 2). Nonetheless, for the analysis of the visual system presented in the following paragraphs, the image data and the robots data were recorded and processed off-line.

5.2.1. Light turbulence test

Figure 15 presents a collection of images illustrating the performance of the tracking task with light turbulence effects with the red/dark box indicating the results of the visual system. The drogue was detected automatically by the detection algorithm in frame 1, shown in Figure 15, and tracked during the entire refuelling task, in spite of the changes in scale (see Figure 15, frames: 1-1632), occlusions (see Figure 15, frames: 665, 903, 931, etc), and periods where part of the drogue was out of the FOV of the camera (see Figure 15, frames: 931, 971, 1613, and 1632).

As can be seen in Figure 16, regardless of the different motions during the task, the tracking strategy did not require an update of the template. The different control parameters were always under the thresholds defined in Section 5, so that the lost status (L = 1) was never reached (the red/dashed line was always zero) .

Figure 17 compares the measured positions of the probe computed from the robot joint positions and the positions of the probe estimated by the visual system. The visual system estimates ${}^{\mathrm{c}}\mathbf{t}_{\mathbf{v}\mathrm{d}}/{}^{\mathrm{c}}\mathbf{t}_{\mathbf{v}\mathrm{R1}}$: the position of the drogue (or robot R1) with respect to the camera coordinate system (14). Nonetheless, because during the task the drogue (robot R1) is kept stationary, the changes in position of the drogue in the image plane are due to the motion of the probe. Therefore, in order to compare the data, the motion estimation ${}^{\mathrm{c}}\mathbf{t}_{\mathbf{v}\mathrm{d}}/{}^{\mathrm{c}}\mathbf{t}_{\mathbf{v}\mathrm{R1}}$ can be used directly in order to obtain the relative motion of the probe.

For the purposed of comparison, the position of the probe with respect to the drogue coordinate system (${}^{\mathrm{d}}\mathbf{t}_{\mathbf{r}\mathrm{p}}/{}^{\mathrm{R1}}\mathbf{t}_{\mathbf{r}\mathrm{R2}}$) recorded by the robot controller

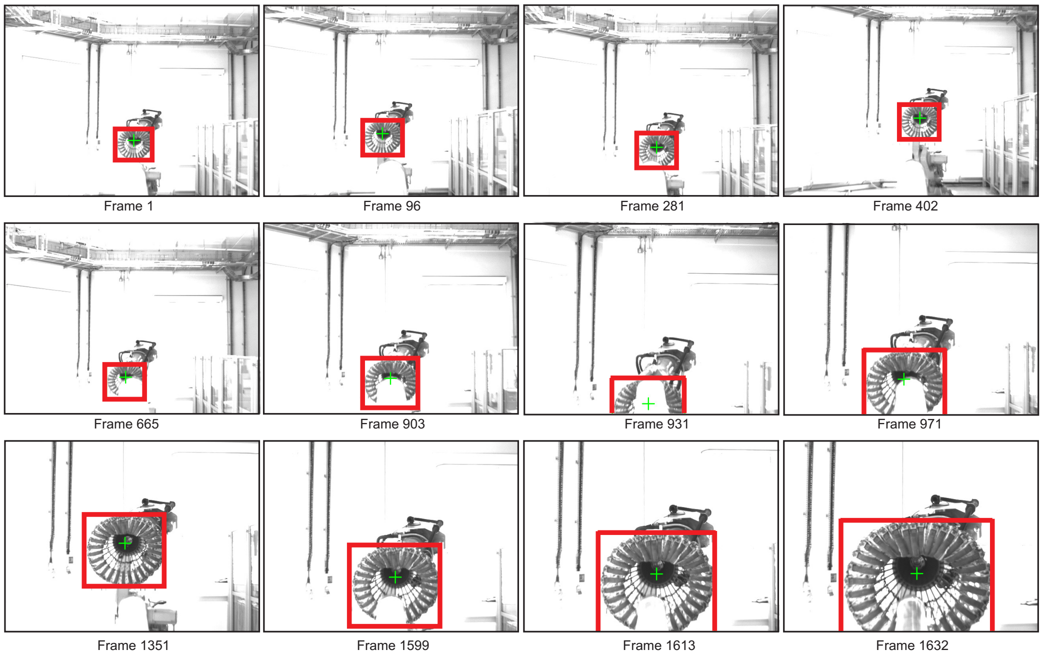


Figure 15: Tracking results: light turbulence. The green crosshair and red box indicate the estimated position and extent of the drogue.

($\mathbf{t_r}$) is transformed into relative positions of the probe ${}^{\mathrm{p}}\mathbf{t_r}$.

Because the visual information does not recover orientation, the image points must be compensated for rotation before the position estimation algorithm acts. This compensation is done using the known orientation data of the probe recorded by the robot controller: relative rotation angles of the probe from the starting point of the tests. Therefore, the visual data ${}^{\mathrm{c}}\mathbf{t}_{\mathbf{v}\mathrm{d}}/{}^{\mathrm{c}}\mathbf{t}_{\mathbf{v}\mathrm{R1}}$ found in Section 4, is transformed into the probe coordinate system ${}^{\mathrm{p}}\mathbf{t}_{\mathbf{r}\mathrm{d}}/{}^{\mathrm{R2}}\mathbf{t}_{\mathbf{r}\mathrm{R1}}$, using the known fixed rotation between both coordinate systems (${}^{\mathrm{p}}\mathbf{R}_{\mathrm{c}}$), and is used to determine the relative motion of the probe ${}^{\mathrm{p}}\mathbf{t_v}$ estimated by the visual system. This transformation is considered valid, as the orientation of the receiver aircraft is expected to be known in a typical refuelling scenario. From Figure 2, ${}^{\mathrm{p}}\mathbf{R}_{\mathrm{c}}$ is defined as a rotation of 90° in the $\mathrm{Y_c}$ axis, followed by a rotation of 90° in the rotated $\mathrm{Z_c}$ axis.

As can be seen Figure 17, the motion of the receiver aircraft (R2/p) inferred using the visual estimation obtained in the 3D position estimation stage (red/dashed line), corresponds well with the motion recorded from the robots (green/solid line). The RMSE(s) reached by the visual system

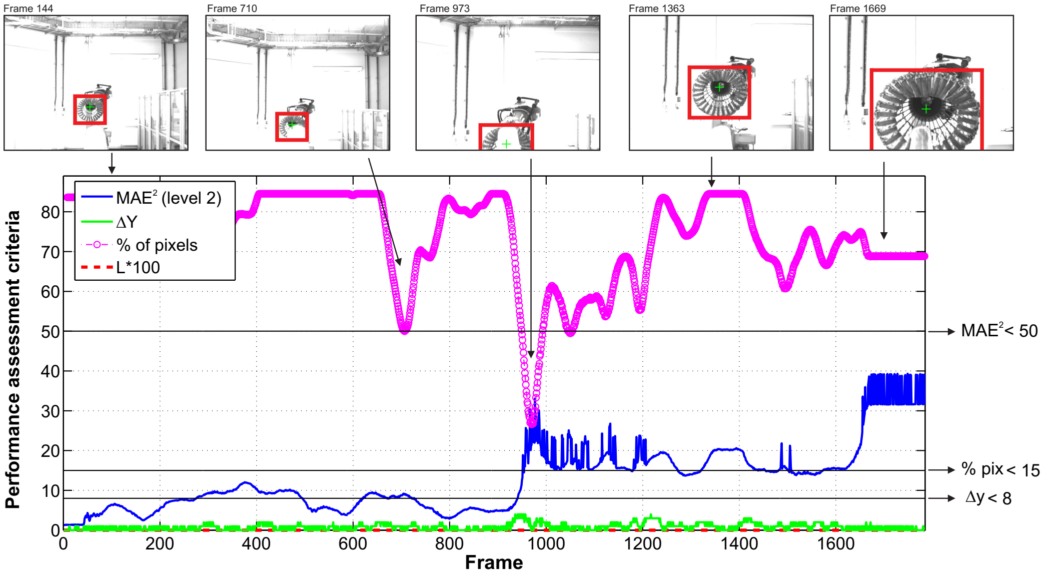


Figure 16: Performance assessment criteria: light turbulence. During the test, the lost status (L = 1) was never reached, the red/dashed line was always zero.

estimating the position of the probe were in the range of 5.5 cm in the ${}^{P}X$ axis, 1.3 cm in the ${}^{P}Y$ axis, and 1.6 cm in the ${}^{P}Z$ axis. These are considered to be relatively low, in particular for a monocular based system.

5.2.2. Moderate turbulence test

The second test was conducted using data from the simulation environment operating in moderate turbulence conditions. Figure 18 shows a collection of images illustrating the performance of the tracking task. The drogue was again detected automatically in the first image using the detection algorithm, and tracked during much of the refuelling task. The moderate turbulence conditions represent a more challenging scenario for the tracking algorithm, with motions that are relatively sudden and faster than those exhibited in the light turbulence case. Specific difficulties are posed by: the occlusion of the drogue by the probe (Figure 18, frames: 670, 694, and 1445, among others), large changes in scale (Figure 18, frames: 1582-1608), changes in orientation, including roll, pitch and yaw (Figure 18, frame: 670), and periods where the drogue is outside the FOV of the camera (Figure 18, frame: 931).

A significant feature of this test is the disappearance of the drogue from

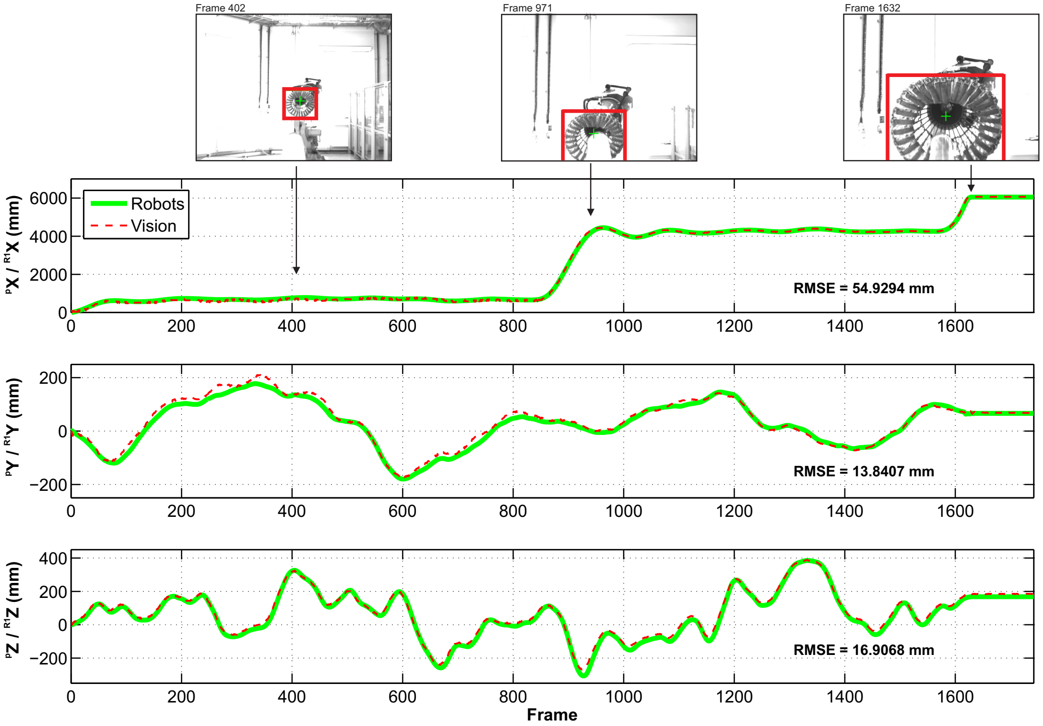


Figure 17: Position estimation: light turbulence. The motion estimated by the tracking algorithm (red/dashed line) is compared with the motion of the R2 robot (green/solid line).

the FOV for approximately 200 frames. When the drogue goes out of the FOV of the camera at frame 891, the detection algorithm is activated. The drogue remains outside the FOV until frame 1105, where it begins to reappear in the image. At this stage the position is not immediately recovered: the segmentation scheme used to find and verify the location of the drogue relies on an unobstructed view of the central region of the drogue. When this region is partially occluded, as seen in Figure 18, frame 1111, the drogue position and size can be misidentified. When the drogue moves further from the occluded zone the detection algorithm is able to recover the position and size effectively, as in Figure 18, frame 1175. Throughout this period the tracking algorithm performs well, but it is clear that there is scope for a more timely recovery of the drogue position through the implementation of detection methods which are more robust to occlusions.

Promisingly, the tracking was robust with respect to the large changes in

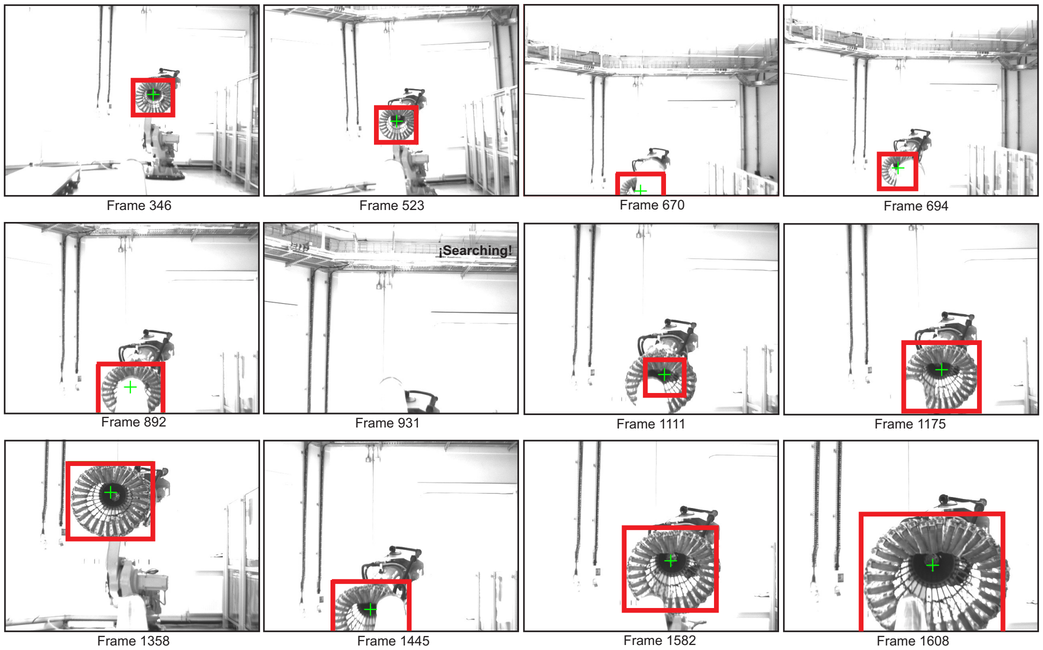


Figure 18: Tracking results: medium turbulence. The green crosshair and red box indicate the estimated position and extent of the drogue.

scale and orientation of the drogue. The latter is particularly important, as changes in orientation are not modeled by the transformation applied in the algorithm. The success of the scheme in these circumstances is attributed to the nature of the drogue's appearance: symmetry means that roll does not have a significant effect on the patterns being searched, and small pitch and yaw motions do not greatly alter the visual characteristics.

In Figure 19, the performance assessment criteria for the tracking task are plotted. In this figure it is possible to see the different times the template was updated by the detection stage, and the reason for each update. In Figure 19, frame 892 (in the dashed circular area marked on the plot), it can be seen that when the drogue was going out of the FOV of the camera, there is an error in the tracking algorithm that the Δy criterion detected, and therefore the lost status ($L = 1$) was activated. From frames $891 - 1105$ the lost status remains activated (red/dashed line). After frame 1105, when the drogue reappears in the FOV of the camera, the detection algorithm recovers the position of the drogue. When the drogue does reappear, because the centre remains obstructed the position of the drogue was misidentified (e.g. frame

1108).

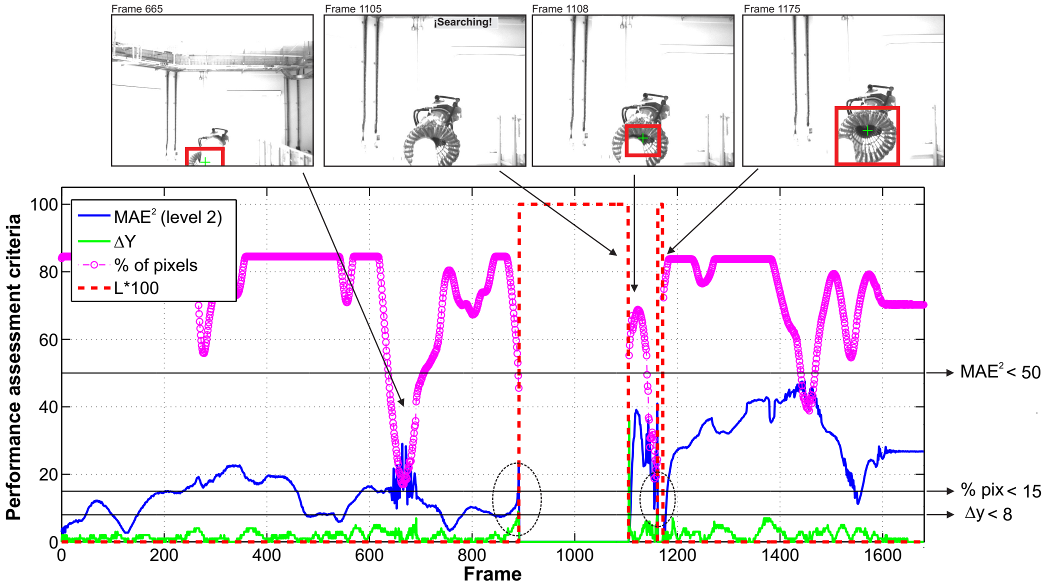


Figure 19: Performance assessment criteria: medium turbulence. The control parameters activated the lost status (red/dashed line) twice. In each situation, the detection algorithm was used to recover the the drogue in order to continue with the tracking task.

Nevertheless, Figure 19 shows that in frame 1161 the lost status was activated for a second time (red/dashed line). In this situation, the % of pixels (magenta/dashed-circle line) activated this status, and the other performance criteria Δy and MAE^2, can be seen to be close to the trigger level at this point as well.

It is interesting to see that when the template was incorrectly updated, the different criteria (Figure 19, frames 1105-1161) rapidly detected a failure of the tracking algorithm. This occurs due to the fact that when the detection algorithm misidentifies the drogue, the incorrect template encompasses most of the centre of the drogue. The pixels in that area have low gradient information, whereupon, the HMPMR-ICIA algorithm is not able to find a good transformation of the parameters, so that high MAE(s) and unstable positions are obtained, and hence the lost status is activated again.

At this point, the detection stage was used for approximately 10 frames (Figure 18, frames 1161-1172). After frame 1172, a new template was correctly found (see Figure 18, frame:1175), and the tracking task continued. In this sequence due to the different motions, the update of the template was required in different frames. Nonetheless, it is possible to see that the detection algorithm was able to correctly detect the drogue when required, allowing the tracking task to continue.

The comparison of the position data from the vision estimation (${}^{P}\mathbf{t_v}/{}^{R_2}\mathbf{t_v}$) and the robot joint measurements (${}^{P}\mathbf{t_R}/{}^{r_2}\mathbf{t_R}$) is shown in Figure 20. As mentioned in the light turbulence test, the image coordinates of the drogue found using the tracking algorithm are compensated for rotation, and then the position data estimated in Section 4 is used to determine the relative motion of the probe ${}^{P}\mathbf{t_v}$.

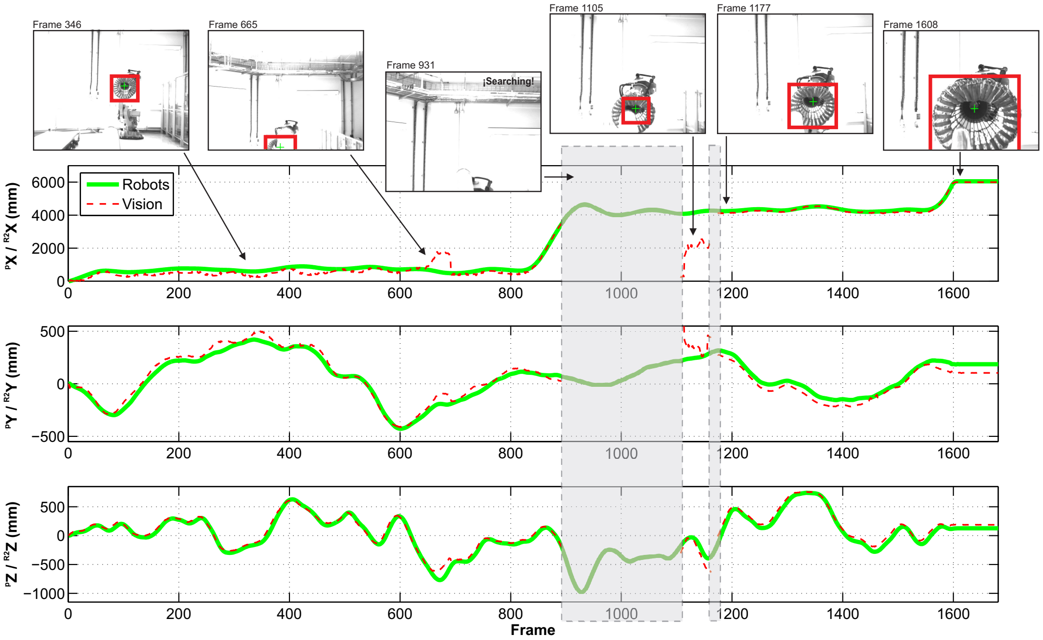


Figure 20: Position estimation: medium turbulence. The motion estimated by the tracking algorithm (red/dashed line) is compared with the motion of the R2 robot (green/solid line). The shaded areas represent the moments when the lost status was activated, and the detection algorithm was operating.

The shaded areas in the plot represent the moments when the lost status was activated, and the detection algorithm was operating. In those frames (frames 891-1105, and frames 1161-1172), position data is not estimated. For the purposes of analyzing the results, the RMSE(s) are calculated separately for the initial and final part of the test.

In the first half of the test, from frames 1-891, the RMSE(s) are 29 cm in the ^{P}X axis, 4.3 cm in the ^{P}Y axis, and 5.1 cm in the ^{P}Z axis. Although the error in the ^{p}X axis is not high for a monocular-based position estimation, the value is higher than in the other axes. The reason for this value is due to a small tracking error in frames 650-690.

In Figure 20, the tracking error in the estimation in frames 650-690 in the ^{p}X and ^{p}Y axes can be seen. This discrepancy is attributed to a small error of the tracking algorithm caused by the turbulence effects, when part of the drogue goes outside the FOV of the camera and the drogue is also occluded by the probe (see Figure 20, frame 665). This loss of information is also indicated by the performance assessment criteria. In Figure 19 it can be seen that the % of pixels (magenta/dashed-circle line) is close to the trigger level, producing an unstable behavior of the tracking algorithm that is also reflected in the unstable behavior of criterion MAE^2 (blue/dark line).

In the final part of the test, from frames 1172-1680, the obtained RMSE(s) are 8.6 cm in the ^{p}X axis, 5.7 cm in the ^{p}Y axis, and 6.1 cm in the ^{p}Z axis.

For this test it can be seen that the motion estimated with the vision system closely matches the positions computed within the robot controller. The detection scheme operated as intended, successfully locating the drogue when it reentered the field of view of the camera. The scheme relies heavily on the visibility of the central portion of the drogue, however if required, further work would allow improvements to be made both to the accuracy and the recovery time in the presence of partial occlusions.

6. Conclusions

Previous work on machine vision systems applied to aerial refuelling tasks have predominantly employed feature-based methods to detect and track the target entity, often requiring the placement of beacons or painted features. Additionally, many of the tests found in the literature have been conducted using only simulated visual information.

This paper presents a drogue tracking strategy for use in probe and drogue refuelling tasks based on direct methods that allows position tracking of the drogue without the need for modifications to the tanker hardware.

The vision strategy is comprised of three main parts: an efficient, robust tracking stage based on the Hierarchical Multi-Parametric Multi-Resolution Inverse Compositional Image Alignment method; a detection stage based on template matching and image segmentation methods; and a 3D position estimation stage based on the known dimensions of the drogue.

The strategy has been tested in a robotic laboratory facility, using unmodified flight refuelling hardware and simulated aircraft motion data, recreating an automated refuelling approach conducted in both light and moderate turbulence modes.

During these tests, the proposed visual system proved to be robust under adverse conditions including rapid motions, large changes in proximity and scale, small changes in orientation of the drogue, and significant occlusions of the drogue by the probe. It was found that the drogue's position was only lost when it left the field of view of the camera. The average accuracy of the position estimation was found to be within 2 cm for the light turbulence conditions and 10 cm for the moderate turbulence test whilst running at real-time frame rates of > 30 fps.

The performance of the proposed vision strategy has been shown to be of a standard appropriate to the probe and drogue autonomous aerial refuelling problem. It has low computational overheads for a vision system, and requires no modification of the target. It is particularly relevant to the final approach and contact stage, where changes in orientation are small and the algorithm's strengths in the presence of occlusions and restricted field of view can be exploited. It is anticipated that in future work this technology will be augmented by other sensing technologies and incorporated into a control algorithm using sensor fusion techniques.

7. Acknowledgements

This work is jointly funded by a Ph.D. Scholarship from the Universidad Politécnica de Madrid, the Spanish Ministry of Science MICYT DPI2010-20751-C02-01, and by Cobham Mission Equipment as part of the ASTRAEA Programme. The ASTRAEA programme is co-funded by AOS, BAE Systems, Cobham, EADS Cassidian, QinetiQ, Rolls-Royce, Thales, the Technology Strategy Board, the Welsh Assembly Government and Scottish Enterprise. Website: http://www.astraea.aero/

[1] Atp-56(b) air-to-air refuelling, NATO (2010).

[2] C. H. Latimer-Needham, Apparatus for aircraft-refuelling in flight and aircraft towing (Aug. 30 1955).

[3] C. J. Leisy, Aircraft interconnecting mechanism (1953).

[4] E. Kim, A. Dogan, W. Blake, Control of a receiver aircraft relative to the tanker in racetrack maneuver, in: AIAA Guidance, Navigation and Control Conference and Exhibit, Keystone, Colorado, 2006.

[5] J. Wang, V. Patel, C. Cao, N. Hovakimyan, E. Lavretsky, $\mathcal{L}_1$ adaptive neural network controller for autonomous aerial refueling with guaranteed transient performance, in: AIAA Guidance, Navigation and Control Conference and Exhibit, Keystone, Colorado, 2006.

[6] J. Wang, V. Patel, C. Cao, N. Hovakimyan, E. Lavretsky, Novel $\mathcal{L}_1$ adaptive control methodology for aerial refueling with guaranteed transient performance, Journal of Guidance, Control, and Dynamics 31 (1).

[7] J. Wang, V. V. Patel, C. Cao, N. Hovakimyan, E. Lavretsky, Verifiable $\mathcal{L}_1$ adaptive controller for aerial refueling, in: AIAA Guidance, Navigation and Control Conference and Exhibit, Hilton Head, South Carolina, 2007.

[8] V. Stephanyan, E. Lavretsky, N. Hovakimyan, A differential game approach to aerial refuelling autopilot design, in: IEEE International Conference on Decision and Control, Maui, Hawaii, 2003.

[9] V. Stephanyan, E. Lavretsky, N. Hovakimyan, Aerial refuelling autopilot design methodology: Application to f-16 aircraft model, in: AIAA Guidance, Navigation and Control Conference and Exhibit, Providence, Rhode Island, 2004.

[10] J. Ding, J. Sprinkle, S. S. Sastry, C. J. Tomlin, Reachability calculations for automated aerial refuelling, in: Procedings of the IEEE Conference on Decision and Control, Cancun, Mexico, 2008.

[11] C. M. Elliot, A. Dogan, Investigating nonlinear control architecture options for aerial refueling, in: AIAA Guidance, Navigation and Control Conference and Exhibition, Toronto, Canada, 2010.

[12] A. D. Marwaha, J. Valasek, A. Narang, Fault tolerant sami for vision-based probe and drogue autonomous aerial refueling, in: AIAA Infortech@Aerospace and AIAA Unmanned... Unlimited Conference, Seattle, Washington, 2009.

[13] J. Wang, N. Hovakimyan, C. Cao, $\mathcal{L}_1$ adaptive augmentation of gain-scheduled controller for racetrack maneuver in aerial refueling, in: AIAA Guidance, Navigation and Control Conference and Exhibit, Chicago, Illinois, 2009.

[14] A. Dogan, S. Sato, W. Blake, Flight control and simulation for aerial refueling, in: 2005 AIAA Guidance, Navigation, and Control Conference and Exhibit, San Francisco, California, 2005, pp. 1–15.

[15] K. RO, J. KAMMAN, Modeling and simulation of hose-paradrogue aerial refueling systems, Journal of guidance, control, and dynamics 33 (1) (2010) 53–63.

[16] S. Ross, M. D. Menza, E. T. Waddell, Jr., A. P. Mainstone, J. Velez, Demonstration of a control algorithm for autonomous aerial refueling (project "no gyro"), Tech. Rep. AFFTC-TIM-05-10, AIR FORCE FLIGHT TEST CENTER EDWARDS AFB CA (December 2005).

[17] S. Ross, M. Pachter, D. Jacques, B. Kish, D. Millman, Autonomous aerial refueling based on the tanker reference frame, in: Aerospace Conference, 2006 IEEE, IEEE, 2006, pp. 22–pp.

[18] J. Hansen, J. Murray, N. Campos, The nasa dryden aar project- a flight test approach to an aerial refueling system, in: AIAA Atmospheric Flight Mechanics Conference and Exhibit, 2004.

[19] J. Hansen, J. Murray, N. Campos, The nasa dryden flight test approach to an aerial refueling system, Tech. Rep. NASA/TM-2005-212859, National Aeronautics and Space Administration (February 2005).

[20] R. Dibley, M. Allen, N. Nabaa, N. Sparks, Autonomous airborne refueling demonstration, phase i flight-test results, Tech. Rep. NASA/TM-2007-214632, National Aeronautics and Space Administration (December 2007).

[21] A. D. Weaver, Using predictive rendering as a vision-aided technique for autonomous aerial refueling, MS Thesis, Air Force Institute of Technology (March 2009).

[22] W. R. Williamson, G. J. Glenn, V. T. Dang, J. L. Speyer, S. M. Stecko, J. M. Takacs, Sensor fusion applied to autonomous aerial refueling, Journal of Guidance, Control, and Dynamics 32 (1).

[23] G. Campa, M. L. Fravolini, A. Ficola, M. R. Napolitano, B. Seanor, M. G. Perhinschi, Autonomous aerial refueling for uavs using a combined gps-machine vision guidance, in: AIAA Guidance, Navigation, and Control Conference and Exhibit, Providence, Rhode Island, 2004.

[24] M. L. Fravolini, A. Ficola, G. Campa, M. R. Napolitano, B. Seanor, Modeling and control issues for autonomous aerial refueling for uavs using a probe-drogue refueling system, Aerospace Science and Technology (8) (2004) 611–618.

[25] R. Mati, L. Pollini, A. Lunghi, M. Innocenti, G. Campa, Vision based autonomos probe and drogue refueling, in: 14th Mediterranean Conference on Control and Automation, 2006.

[26] J. H. Spencer, Optical tracking for relative positioning in automated aerial refueling, MS Thesis, Air Force Institute of Technology (March 2007).

[27] M. Mammarella, G. Campa, M. R. Napolitano, M. L. Fravolini, Y. Gu, M. G. Perhinschi, Machine vision/gps integration using ekf for the uav aerial refueling problem, in: IEEE Transactions on Systems, Man, and Cybernetics - part C: Applications and Reviews, Vol. 38, 2008.

[28] M. Mammarella, G. Campa, M. R. Napolitano, B. Seanor, M. L. Fravolini, L. Pollini, Gps/mv based aerial refueling for uavs, in: AIAA Guidance, Navigation and Control Conference and Exhibit, Honolulu, Hawaii, 2008.

[29] G. Conte, P. Doherty, Vision-based unmanned aerial vehicle navigation using geo-referenced information, EURASIP J. Adv. Signal Process 2009 (2009) 10:1–10:18. doi:http://dx.doi.org/10.1155/2009/387308.
URL http://dx.doi.org/10.1155/2009/387308

[30] P. Campoy, J. Correa, I. Mondragón, C. Martínez, M. Olivares, L. Mejias, J. Artieda, Computer Vision Onboard UAVs for Civilian Tasks, Journal of Intelligent and Robotic Systems 54 (2009) 105–135,

10.1007/s10846-008-9256-z.
URL http://dx.doi.org/10.1007/s10846-008-9256-z

[31] L. Mejias, S. McNamara, J. Lai, J. J. Ford, Vision-based detection and tracking of aerial targets for UAV collision avoidance, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2010.

[32] S. Saripalli, J. F. Montgomery, G. S. Sukhatme, Vision-based Autonomous Landing of an Unmanned Aerial Vehicle, in: IEEE International Conference on Robotics and Automation (ICRA, 2002, pp. 2799–2804.

[33] C. Martínez, P. Campoy, I. Mondragón, M. A. Olivares-Méndez, Trinocular ground system to control UAVs, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), St. Louis, MO, USA, 2009.

[34] J. L. Junkins, H. Schaub, D. Hughes, Non contact position and orientation measurement system and method, US Patent 6,266,142 (2011).

[35] J. Valasek, K. Gunman, J. Kimmett, M. Tandale, L. Junkins, D. Hughes, Vision-based sensor and navigation system for autonomous air refueling, in: Proceedings of the 1st AIAA Unmanned Aerospace Vehicles, Systems, Technologies, and Operations Conference and Exhibit, Vancouver, 2002.

[36] J. Kimmett, J. Valasek, J. L. Junkins, Autonomous aerial refueling utilising a vision based navigation system, in: AIAA GUidanc, Navigation, and Control Conference and Exhibit, Monterey, California, 2002.

[37] J. Kimmett, J. Valasek, J. L. Junkins, Vision based controller for autonomous aerial refueling, in: Proceedings of the 2002 IEEE International Conference on Control Applications, Glasgow, Scotland, 2002.

[38] M. D. Tandale, R. Bowers, J. L. Valasek, Robust trajectory tracking controller for vision based probe and drogue autonomous aerial refueling, in: AIAA Guidance, Navigation, and Control Conference and Exhibit, San Francisco, California, 2005.

[39] L. Pollini, R. Mati, M. Innocenti, Experimental evaluation of vision algorithms for formation flight and aerial refueling, in: AIAA Modeling and SImulation Technologies Conference and Exhibit, Providence, Rhode Island, 2004.

[40] C. Lu, G. Hager, E. Mjolsness, Fast and globally convergent pose estimation from video images, Pattern Analysis and Machine Intelligence, IEEE Transactions on 22 (6) (2000) 610–622.

[41] L. Pollini, M. Innocenti, R. Mati, Vision algorithms for formation flight and aerial refueling with optimal marker labeling, in: AIAA Modeling and SImulation Technologies Conference and Exhibit, San Francisco, California, 2005.

[42] F. Saghafi, S. M. K. Zadeh, Vision-based trajectory tracking controller for autonomous close proximity operations, in: IEEE Aerospace Conference, 2008.

[43] J. Doebbler, J. Valasek, M. J. Monda, H. Schaub, Boom and receptacle autonomous aerial refueling using a visual pressure snake optical sensor, in: AIAA Guidance, Navigation, and Control Conference and Exhibit, Keystone, Colorado, 2006.

[44] P. H. S. Torr, A. Zisserman, Feature based methods for structure and motion estimation, in: ICCV '99: Proceedings of the International Workshop on Vision Algorithms, Springer-Verlag, London, UK, 2000, pp. 278–294.

[45] M. Irani, P. Anandan, About direct methods, in: B. Triggs, A. Zisserman, R. Szeliski (Eds.), Vision Algorithms: Theory and Practice, Vol. 1883 of Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2000, pp. 267–277.

[46] J. R. Bergen, P. Anandan, T. J. Hanna, R. Hingorani, Hierarchical model-based motion estimation, Springer-Verlag, 1992, pp. 237–252.

[47] B. L. Stevens, F. L. Lewis, Aircraft Control and Simulation, 2nd Edition, John Wiley & Sons Inc., 2003.

[48] L. Nguyen, E. Ogburn, W. Gilbert, K. Kibler, P. Brown, P. Deal, Simulator study of stall/post-stall characteristics of a fighter airplane with

relaxed longitudinal static stability, NASA Technical Paper 1538 (December 1979).

[49] MIL-HDBK-1797, Military specification, flying qualities of piloted aircraft, Department of Defence, USA (December 1997).

[50] MIL-F-8785C, Flying qualitites of piloted aircraft, Department of Defense, USA (November 1980).

[51] G. Bradski, A. Kaehler, Learning OpenCV: Computer Vision with the OpenCV Library, O'Reilly, 2008.

[52] R. Szeliski, Image alignment and stitching: a tutorial, Found. Trends. Comput. Graph. Vis. 2 (1) (2006) 1–104. doi:http://dx.doi.org/10.1561/0600000009.

[53] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, New York, NY, USA, 2003.

[54] S. Baker, I. Matthews, Equivalence and efficiency of image alignment algorithms, in: Proceedings of the 2001 IEEE Conference on Computer Vision and Pattern Recognition, Vol. 1, 2001, pp. 1090 – 1097.

[55] C. H. Anderson, J. R. Bergen, P. J. Burt, J. M. Ogden, Pyramid methods in image processing (1984).

[56] P. Burt, E. Adelson, The laplacian pyramid as a compact image code, Communications, IEEE Transactions on 31 (4) (1983) 532 – 540. doi:10.1109/TCOM.1983.1095851.

[57] C. Martínez, P. Campoy, I. Mondragón, M. A. Olivares-Méndez, Hierarchical Multi-Parametric and Multi-Resolution Strategy (HMPMR) for Improving Real-Time Tracking, in: submitted for publication.

[58] T. Ishikawa, I. Matthews, S. Baker, Efficient image alignment with outlier rejection, Tech. Rep. CMU-RI-TR-02-27, Robotics Institute, Pittsburgh, PA (October 2002).

[59] S. I., F. G., A 3x3 Isotropic Gradient Operator for Image Processing, Presented at a talk at the Stanford Artificial Project in 1968, unpublished.

[60] Z. Zhang, Flexible camera calibration by viewing a plane from unknown orientations, Computer Vision, IEEE International Conference on 1 (1999) 666. doi:http://doi.ieeecomputersociety.org/10.1109/ICCV.1999.791289.

[61] J. Y. Bouguet, Camera calibration toolbox for Matlab.