

Towards Autonomous Air-to-Air Refuelling for UAVs Using Visual Information

Carol Martínez¹, Thomas Richardson², and Pascual Campoy¹

Abstract—For the purposes of achieving Autonomous Air-to-Air Refuelling (AAAR) in Unmanned Aerial Vehicles (UAVs), this paper presents the use of a visual tracking algorithm based on direct methods and image registration techniques, with the aim of solving the drogue tracking problem in order to obtain vision-based relative position estimations of the aircrafts for the probe and drogue refuelling method. Proposed vision-based AAAR approaches to date have explored the use of features (such as corners, painted marks, or LEDs) to detect and estimate the relative motion of either the receiver or the tanker aircraft, with the drawback that sometimes this requires the installation of specific hardware on-board. Conversely, the strategy we propose to use does not require the installation of additional hardware on-board. The strategy is based on a hierarchical implementation of an image registration technique: the Inverse Compositional Image Alignment ICIA. Real images and real flight hardware (probe and drogue) are used to test the algorithm using a robotic testbed that simulates the motion of the aircrafts (the tanker and the receiver) during the refuelling task. Results show that the tracking algorithm is robust to fast motion, changes in appearance, and situations where part of the drogue is occluded or is outside of the field of view of the camera. Additionally, results show that robust position estimations at real-time frame rates are obtained, proving that the technique is fast enough to form the basis for automated aerial refuelling sensing capabilities.

I. INTRODUCTION

This paper presents results of using visual information to estimate the relative motion of the aircrafts in Autonomous Air-to-Air Refuelling (AAAR) tasks for UAVs.

Two approaches are currently used for aerial refuelling [1]: the flying boom, where a retractable boom is extended to be connected to the fuel receptacle of the aircraft to be refueled; and the probe and drogue method, where a flexible hose with a drogue attached to its end is extended and the receiver aircraft maneuvers to insert the probe into the drogue.

In the literature, there have been a variety of computer vision solutions for the aerial refuelling problem. In [2], the drogue position is estimated using an infrared camera placed in the receiver aircraft and infrared leds placed on the drogue structure. The pose of the drogue is estimated matching the 2D position of the leds with their known 3D positions in the drogue. Synthetic images are used to test the algorithm. On the other hand, a set of beacons mounted on known positions in the drogue are used in [3]. These beacons are detected by a VisNav sensor placed in the receiver aircraft.

A communication link between the sensor and the beacons allows the receiver aircraft to triangulate the pose of the drogue with update rates of 100 Hz.

For the flying boom refuelling method, a vision system based on deformable contour algorithms was proposed in [4] to obtain relative 3D position estimations. A camera was placed on the tanker aircraft looking down towards the receiver aircraft, capturing images of a passive target painted near the refuelling receptacle. The HSV (hue, saturation, and value) color space was used to increase the robustness under variations in lighting conditions. Synthetic images were used in this case to test the performance of the algorithm.

On the other hand, in [5], the performance of well-known corner detectors (SUSAN and Harris) are analyzed for the docking maneuver involved in the flying boom method. A camera is placed in the receiver aircraft looking upwards, capturing the tanker aircraft. The feature extractor algorithms detect corners of the tanker aircraft, which are matched with a set of known physical features of the tanker. The positions of the matched corners (2D-3D match) are used to evaluate the relative position and orientation of the aircrafts.

As can be seen in the previously mentioned papers, features such as corners, painted marks, and LEDs have been used to estimate the relative position of either the receiver or the tanker aircraft. Some of these methods not only require the installation of specific hardware on-board, but they are also susceptible to problems caused by occlusions of one or more of these features.

Therefore, in this paper, we present and evaluate the use of a visual tracking algorithm, which is based on direct methods (i.e. all the pixels of the drogue are used), to solve the tracking problem in the probe and drogue refuelling method. We propose to use a Hierarchical Multi-Parametric and Multi-Resolution (HMPMR) implementation of the Inverse Compositional Image Alignment (ICIA) algorithm [6]. In the literature multi-resolution (MR) approaches are widely used for tracking, however to the authors' knowledge, HMPMR strategies have neither been applied in tracking nor for solving the tracking problem in AAAR tasks.

Some of the advantages of using direct methods include the fact that the detection of special sets of features is not required; and that because all the pixels of the drogue are considered in the motion estimation process, the algorithm is able to continue tracking the position of the drogue in situations where the center of the drogue is occluded (e.g. occlusions by the probe) or in situations where part of the drogue is outside the field of view (FOV) of the camera. Additionally, by using an HMPMR strategy instead of only

¹Computer Vision Group, Centro de Automática y Robótica CAR UPM-CSIC, Universidad Politécnica de Madrid, José Gutiérrez Abascal 2, 28006 Madrid, Spain. carolviviana.martinez@upm.es, pascual.campoy@upm.es

²Department of Aerospace Engineering, Queens Building, University of Bristol, UK. thomas.richardson@bristol.ac.uk

an MR one, the range of motion that the algorithm can tolerate is increased, which is crucial for this application.

Therefore, the main difference of the vision-based system we propose from previous AAAR approaches is that by using direct methods, modification of the refuelling hardware is not required. On the other hand, while many of the studies presented in the literature have been conducted on synthetic images to test the vision algorithms, in this paper, real vision hardware that captures real images from a drogue is used using a realistic laboratory environment with actual flight hardware (a probe and a drogue), as shown in Fig. 1.

The paper is organized as follows: Section II presents the problem statement for the probe and drogue method. In Section III, the hierarchical visual tracking algorithm is presented. The performance of the tracking strategy is analyzed in Section IV. Finally, conclusions and the direction of future work are presented in Section V.

II. PROBLEM STATEMENT

The probe and drogue method [1], used in air-to-air refuelling consists of a tanker aircraft (represented by R1 in the robotic testbed shown in Fig. 1) that deploys a flexible hose with a drogue attached to its end. This drogue is comprised of a canopy to provide stability, and of an automatic coupling for the fuel transfer. A receiver aircraft (represented by R2 in the robotic testbed shown in Fig. 1) is equipped with a probe that must be brought into contact with the drogue. When this is achieved, the automatic coupling is activated and the refuelling process starts.

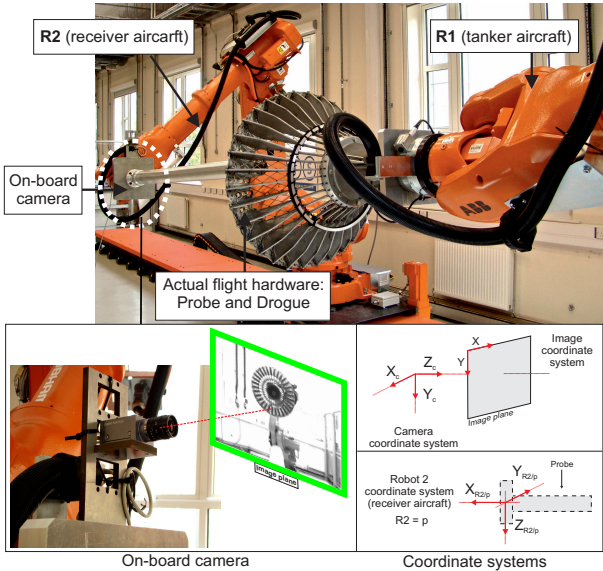


Fig. 1. Testbed for autonomous air-to-air refuelling.

For AAAR capabilities, the receiver aircraft must be able to detect, track, and estimate the relative position of the drogue, taking into account that the drogue's position in the image plane is affected by the effects of the tanker and the receiver aircraft and by environmental conditions. Additionally, because rapid control corrections are required for docking, especially in turbulent conditions, to enable

robust and safe operation of the receiver aircraft, high-frequency updates of the drogue's position are also required (i.e. real-time frame rates).

In order to recreate an AAAR test for the probe and drogue method, the robotic cell shown in Fig. 1 is used to reproduce the relative motion of the receiver and the tanker aircraft during the last stage of the refuelling task, and a camera mounted at the base of the probe (see Fig. 1) is used to sense the drogue.

The visual system will operate in the last stage of the refuelling task (the docking stage), which requires robust and fast position information. Therefore, the initial approaching maneuver between the aircrafts is out of the scope of this study. However it could be conducted based on other sensors, such as GPS.

III. DROGUE TRACKING STRATEGY

The goal of the visual tracking algorithm is to obtain a robust 2D position estimation of the drogue in the image plane, taking into account that the drogue is being imaged by a camera on-board the receiver aircraft, as shown in Fig. 1. Using direct methods, this 2D position can be found in each frame by formulating the tracking task as an incremental image registration task, assuming that the initial position of the drogue is known, and that the motion of the drogue in the image plane can be modeled by a 2D transformation [7].

The image registration process consists of aligning two images, a reference image or image template (\mathbf{T}), and a current image (\mathbf{I}), by finding the transformation or motion model (\mathbf{W}) that best aligns them. The image registration algorithm we use for tracking the drogue is the ICIA algorithm [6]. This is considered to be an efficient algorithm for image registration that permits an efficient estimation of the motion model (\mathbf{W}).

The motion model \mathbf{W} that has been chosen for the AAAR task is a 3×3 matrix (1) parameterized by the vector of parameters $\mathbf{p} = (p_1, p_2, p_3)^T$, as follows:

$$\mathbf{W} = \begin{bmatrix} 1 + p_1 & 0 & p_2 \\ 0 & 1 + p_1 & p_3 \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

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

where p_2 and p_3 represent the translation in the X and Y axes of the image coordinate frame shown in Fig. 1, and p_1 represents the scale factor. This motion model transforms the 2D pixel coordinates \mathbf{x} (where $\mathbf{x} = (x, y, 1)^T$) in image \mathbf{T} into the 2D coordinates $\mathbf{x}' = (x', y', 1)^T$ in image \mathbf{I} .

The motion model has been selected taking into account that due to the symmetric structure of the drogue's appearance, rotations around the Z_c axis (roll motions) and small rotations around the other axes do not have a significant impact on the visual characteristics of the drogue in the image plane. Additionally, \mathbf{W} has been selected taking into account that in the last stage of the task (the docking stage) when the probe moves towards the drogue, most of the

imaged motion of the drogue in the image plane can be represented only by changes in position and in scale.

A. Tracking algorithm

The image registration algorithm selected for this application (the ICIA algorithm) has been adapted to the drogue tracking problem, as follows:

$$\sum M^j(\mathbf{x})[T^j(\mathbf{W}^j(\mathbf{x}; \Delta \mathbf{p})) - I^j(\mathbf{W}^j(\mathbf{x}; \mathbf{p}))]^2 \quad (2)$$

where the sum is conducted only in those pixels of \mathbf{T} that meet the following condition:

$$\forall \mathbf{x} \in \mathbf{T}^j : P^j(\mathbf{W}^j(\mathbf{x}; \mathbf{p})) = 0 \quad (3)$$

Where, M^j and P^j are masks created to exclude some pixels during the minimization process. As shown in Fig. 2, M^j is a constant mask that is used in the minimization process to consider only the pixels in \mathbf{T}^j that belong to the drogue (the drogue is circular, and as a consequence not all the pixels in image \mathbf{T}^j should be used). P^j is another constant mask (the camera is always located in a defined and fixed position with respect to the probe). This mask is used to exclude the pixels of the probe when it is approaching the drogue. This mask plays an important role in the stability of the tracking algorithm, because during the final stage of the refuelling approach the drogue will be occluded by the probe.

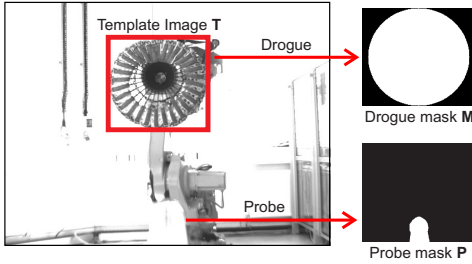


Fig. 2. Masks. M is used to exclude background pixels, and P excludes the pixels of the probe when it is approaching the drogue.

The ICIA algorithm relies on the linearization of the cost function (2). This linearization is valid only when the range of motion is small (when the first-order approximation is valid or close enough to find a minimum). However, in the current application, aspects such as turbulence effects (e.g. wake turbulence) make it difficult to preserve the required small frame-to-frame motion, especially in the final part of the task, when the probe and the drogue are very close to each other. For this reason, the strategy we propose uses the ICIA algorithm as an image registration technique in a hierarchical structure that makes use of a multi-resolution (MR) and a multi-parametric (MP) hierarchy simultaneously: the hierarchical multi-parametric and multi-resolution ICIA algorithm HMPMR-ICIA. Details of the extension of the ICIA algorithm with the HMPMR structure can be found in [8].

The general idea behind the HMPMR strategy, as shown in Fig. 3, is that by estimating only a small number of

parameters in the lowest resolution levels and smoothly increasing the complexity of the motion model through the MR structure, it is possible to obtain a robust estimation of the motion model \mathbf{W} , and to increase the range of motion that the algorithm can tolerate, which is crucial for this application.

The MR structure is created by downsampling the images by a factor of 2, according to the different levels (pL) of the MR structure. The number of levels has been defined as $pL = 3$ (i.e. levels: 0, 1, and 2) taking into account the distance at which the tracking algorithm starts operating (≈ 6 m) and the size of the drogue in the image plane at that distance (≈ 90 pixels).

In the MP structure, different motion models are estimated in each pyramid level. In the lowest level \mathbf{W}^0 (where the superscript represents the level), the motion model described in (1) is estimated; and in levels 1 and 2, only the translation parameters shown in (1) are estimated (i.e. when \mathbf{W}^2 and \mathbf{W}^1 are estimated, $p_1 = 0$, and only p_2 and p_3 are estimated).

In general terms, the algorithm is based on two stages: the initialization and the tracking stages. Details of the algorithm can be seen in Algorithm 1. In the initialization stage, the algorithm requires as input an initial frame, e.g. \mathbf{I}_0 (the subscript represents the number of the frame), that is used to find the template image \mathbf{T} (the drogue), and which in itself is used to define the coordinates of the drogue in the first frame (frame \mathbf{I}_0). \mathbf{T} is detected by a template matching algorithm [9], and corresponds to a sub-image or ROI (Region of Interest) extracted from \mathbf{I}_0 that covers the front part of the drogue (as can be seen in Fig. 2).

Once the coordinates of \mathbf{T} (the drogue) are known, \mathbf{I}_0 is downsampled according to the different levels of the MR structure, thus creating a template image \mathbf{T}^j for each level (where j represents the pyramid level), as shown in the template images of Fig. 3. Additionally, in this initialization stage, for each level of the MR pyramid different elements of the ICIA algorithm are calculated (e.g. the Hessian matrix, and the masks shown in Fig. 2).

During the tracking stage, when a new frame is analyzed (e.g. frame \mathbf{I}_1), the ICIA algorithm is used to find the position of the drogue in the current frame in each level of the hierarchical structure. In each level, the minimization is done only with respect to parameters of the motion model defined for that level. When the stopping conditions have been reached, the parameters are propagated to the next level of the pyramid (see [8] for details).

At the lowest level of the pyramid, the motion model \mathbf{W}^0 is used to determine the position of \mathbf{T} (i.e. the drogue) in the current image \mathbf{I}_1 . This motion model is propagated as initial guess to the highest level of the pyramid ($j = 2$) of the next frame. The propagation of the parameters from the lowest level of the pyramid in the previous frame to the highest level of the pyramid in the new frame helps to validate the linearization of (2) done by the image registration algorithm, so that when a new frame is analyzed, by using the estimation of \mathbf{W} in the previous frame, images \mathbf{T} and \mathbf{I} are close enough to find a minimum.

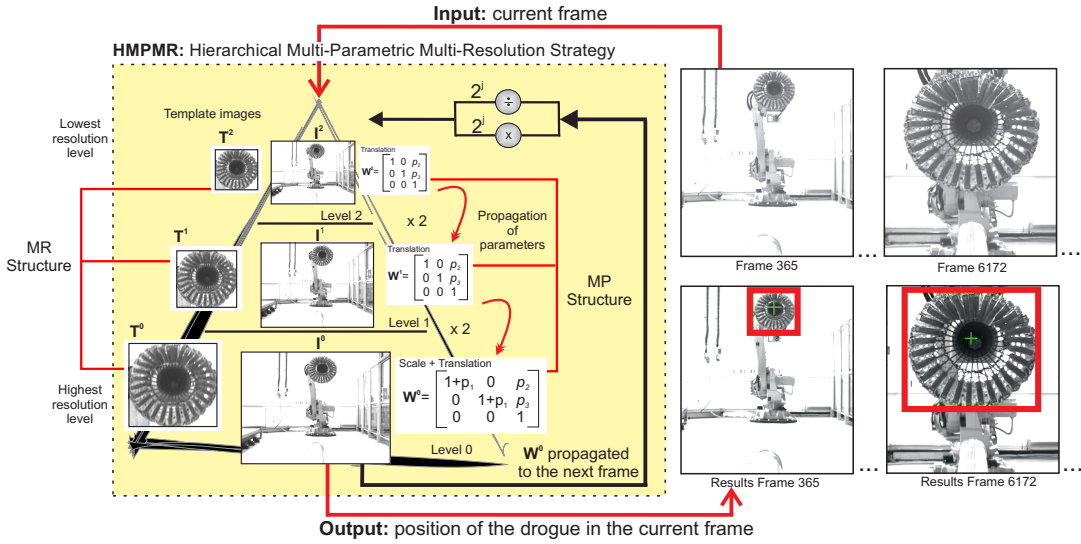


Fig. 3. HMPMR Tracking Strategy. The MR structure is created by downsampling the images. Different motion models are estimated in each level (MP structure). The motion model found in the lowest level of the pyramid (level 0) is used to determine the position of the drogue in the current image.

In the AAAR task, the previously described tracking algorithm will start operating in the last stage of the task (the docking stage). Therefore, taking into account that the algorithm will not operate for long periods of time, an update of the template is not required, we assume that during the tracking stage, the template image is fixed (the one found by the template matching algorithm when the tracking algorithm started operating).

IV. RESULTS AND DISCUSSION

The visual tracking algorithm is tested in two scenarios. In the first one, basic motions of the probe and the drogue performed at different speeds are used to analyze the performance of the HMPMR-ICIA algorithm tracking the drogue. In the second scenario, motions of the probe towards the drogue simulate the refuelling procedure in two different situations: without turbulences and with light turbulences, are used.

• Experimental setup

The HMPMR-ICIA algorithm is tested using the AAAR testbed shown in Fig. 1. The robot motion is governed by data sets produced from a numerical simulation of a probe and drogue refuelling procedure, based on an F16 flight dynamics model. A grey scale Sony XCD-V60 FireWire camera is mounted on robot R2, at the base of the probe. This camera is capable of capturing images at 60 Hz, and is connected to a 2.7 GHz Intel core I7 MacBook Pro laptop, where the visual system runs. The camera settings were adjusted in order to emphasize the drogue structure and to avoid perturbations of background objects (e.g. the window, the wall, the robot structure) which are not present in a refuelling scenario.

The HMPMR-ICIA algorithm was developed in C++, and the OpenCV libraries [9] were used for managing image data. The KLT feature-based algorithm [10] (pyramidal Lucas

Kanade) used in the second test is based on the version of the algorithm implemented in the OpenCV libraries [9]. The masks shown in Fig. 2 were also applied. The maximum number of features was defined as 100, a window size of 5 was used, and four pyramid levels were used in the MR structure of the algorithm.

A. Test 1: Drogue Tracking Evaluation

The image sequence used in this test contains basic motions of the drogue (moving left, right, up and down) where the background information changes (due to the robot structure, the window, and the wall), and also contains basic motions of the probe in relation to the drogue (forwards and backwards), that in some situations, cause occlusions of the drogue by the probe, or make the drogue go out of the FOV of the camera.

The evaluation of the results is based on a comparison of the motion estimated by the tracking algorithm with the position data recorded from the robots, that is used as ground-truth (GT) data. The 2D position of the drogue found by the HMPMR-ICIA algorithm is used to estimate the 3D position of the drogue using the method presented in [11], assuming that the camera is calibrated, and that the dimension of the drogue is known.

Fig(s). 4 and 5 show the results of the position of the drogue with respect to the Y_p (the drogue moves to the left and to the right) and the Z_p axes (the drogue moves up and down), at three different speeds: 400 mm/s, 1000 mm/s, and 2000 mm/s.

Fig. 6 shows the results of the estimated motion in the X_p axis (scale changes). In this sequence, the probe is kept stationary in frames 0-1750, 1900-3632, 6920-8710, and 8860-10000; and from frame 3720 to frame 6900 the Robot 2 is commanded to move towards the stationary drogue and backwards, at two different speeds: 400 mm/s and 1000 mm/s.

input : $\mathbf{I}_0, pL, \mathbf{I}, \mathbf{x}$, configuration of \mathbf{W}^j
output: \mathbf{W}^0 , position of \mathbf{T} in \mathbf{I}
Pre-compute
 1. Downsample \mathbf{I}_0 according to pL and create \mathbf{T}^j
 2. Initialize $j_{\max} = 2$
for $j \leftarrow 2$ **to** 0 **do**
 3. Evaluate the gradient $\nabla \mathbf{T}^j = \left(\frac{\partial \mathbf{T}^j}{\partial x}, \frac{\partial \mathbf{T}^j}{\partial y}, \mathbf{1} \right)$
 4. Evaluate the Jacobian $\mathbf{J}^j = \frac{\partial \mathbf{W}^j}{\partial \mathbf{p}}$ at $(\mathbf{x}, 0)$
 5. Compute the steepest descent images SDI
 $\text{SDI}(\mathbf{x}) = \nabla \mathbf{T}(\mathbf{x})^j \frac{\partial \mathbf{W}^j}{\partial \mathbf{p}}$
 6. Compute the Hessian matrix and its inverse
 $\mathbf{H}^j = \sum_{\mathbf{x}} [\text{SDI}(\mathbf{x})^T \text{SDI}(\mathbf{x})]$, and \mathbf{H}^{j-1}
end
Iterate
foreach *new frame* \mathbf{I} **do**
 7. Downsample \mathbf{I} according to pL to create \mathbf{I}^j
 8. Initialize $\mathbf{W}^{j_{\max}}$ (see [8]). If it is the first frame, $\mathbf{W}^{j_{\max}}$ is the identity matrix
 for $j \leftarrow 2$ **to** 0 **do**
 repeat
 foreach \mathbf{x} in $\mathbf{T} : P^j(\mathbf{W}^j(\mathbf{x}; \mathbf{p})) = 0$ and $M^j(\mathbf{x}) = 1$ **do**
 9. Warp $\mathbf{W}^j(\mathbf{x}; \mathbf{p})$ to find $I^j(\mathbf{W}^j(\mathbf{x}; \mathbf{p}))$
 10. Compute $E^j = [I^j(\mathbf{W}^j(\mathbf{x}; \mathbf{p})) - T^j(\mathbf{x})]$
 11. Compute $\mathbf{b}^j = \mathbf{b}^j + \text{SDI}(\mathbf{x})^T E^j$
 end
 12. Compute $\Delta \mathbf{p} = \mathbf{H}^{j-1} \mathbf{b}^j$
 13. Update the warp
 $\mathbf{W}^j(\mathbf{x}; \mathbf{p}) \leftarrow \mathbf{W}^j(\mathbf{x}; \mathbf{p}) \circ \mathbf{W}^j(\mathbf{x}; \Delta \mathbf{p})^{-1}$
 until $\|\Delta \mathbf{p}\| \leq \varepsilon$;
 14. Propagate the parameters in $\mathbf{W}^j(\mathbf{x}; \mathbf{p})$ to the next level [8]
 end
 15. Use \mathbf{W}^0 to find the position of \mathbf{T} in \mathbf{I}
 16. Draw results
end

Algorithm 1: HMPMR-ICIA for AAAR tasks.

As can be seen in the figures, the thumbnail images show that the drogue was tracked throughout the sequence (the red/dark box covers the drogue) in spite of e.g. the significant changes in scale shown in Fig. 6 (from 0 m to 6 m). Additionally, it can be seen that the motion recovered by the vision algorithm (red/dark-dashed line) has behavior and values that are similar to the motion conducted by the robot (green/light-solid line). The RMSEs obtained for p_y was < 11 cm, for p_z was < 34 cm, and for p_x was < 7 cm. These errors are low considering that the estimation is based on a monocular system.

B. Test 2: Refuelling tests

In this test, simulated data is sent to one of the robots (R2, the one that simulates the receiver aircraft, shown in Fig. 1).

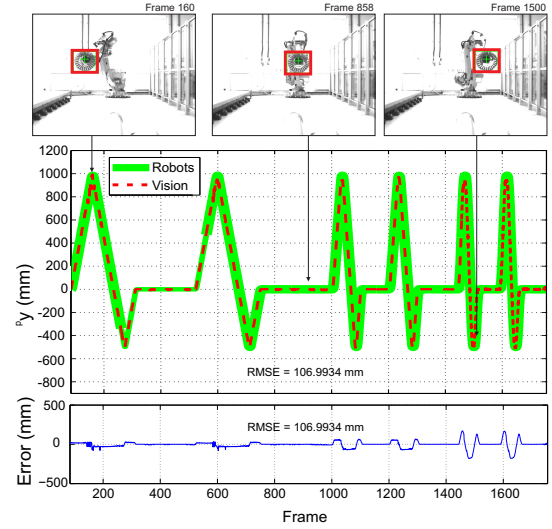


Fig. 4. Motion estimation comparison: Y_p axis. The upper graphic shows the comparison of the motion estimated by the robots (green/light-solid line) with the motion estimated by the HMPMR-ICIA algorithm (red/dark-dashed line). The lower graphic shows the errors in the estimation. Thumbnail images show the results of the tracking task.

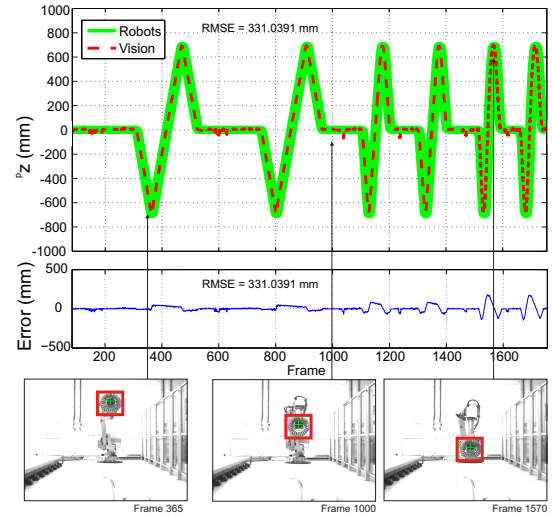


Fig. 5. Motion estimation comparison: Z_p axis.

The data corresponds to motions of the probe towards the drogue (the drogue is kept stationary), simulating the aerial refuelling task (with both light turbulence and no turbulence effects being considered).

In the first test, the performance of the HMPMR-ICIA algorithm (a direct method) is compared with the one of the pyramidal Lucas Kanade (KLT) algorithm (a feature-based method). Although the algorithms we are comparing have the same basis: the sum of square differences between the intensity values of the template and the current images, this comparison is conducted with the only purpose of showing the advantages in the motion estimation process, of using all the pixels of the drogue instead of a sparse set of features, as the KLT algorithm does, especially in situations where the center of the drogue is occluded by the probe or where

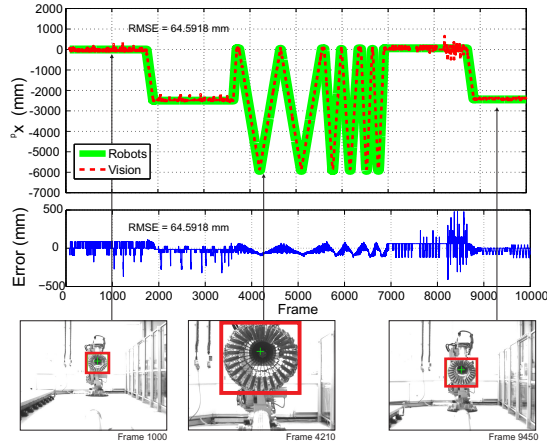


Fig. 6. Motion estimation comparison: X_p axis.

part of the drogue is outside the FOV of the camera. These situations are found typically during the final stage of the refuelling approach, in this stage the drogue is almost always occluded by the probe

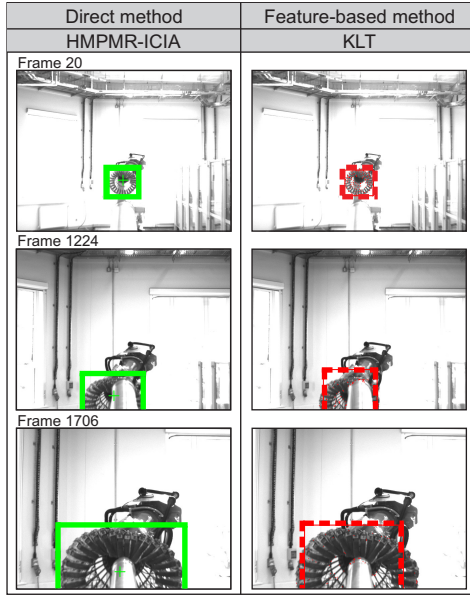


Fig. 7. Comparison of the tracking results during a refuelling test.

Fig. 7 presents representative images that illustrate the performance of the tracking algorithms. As can be seen, the HMPMR-ICIA algorithm (first column green/light box) tracks the drogue during the entire refuelling task, despite of occlusions of the drogue by the probe and the fact that part of the drogue is out of FOV of the camera (e.g. frames 1224 and 1706). In Fig. 7, second column, it can be seen that the KLT algorithm fails to find a good motion model for tracking the drogue, especially in the final part of the refuelling procedure (e.g. the red/dark box in frames 1224 and 1706 does not cover the extent of the drogue).

On the other hand, Fig. 8 shows a visual analysis of the motion models recovered by each algorithm. In each frame, the estimated motion model is used to transform the current

frame to the coordinate frame of the first image (i.e. back-warpping the current image), where the template image was initialized. If the motion model is correctly estimated, the template image and the back-warped template should be very similar.

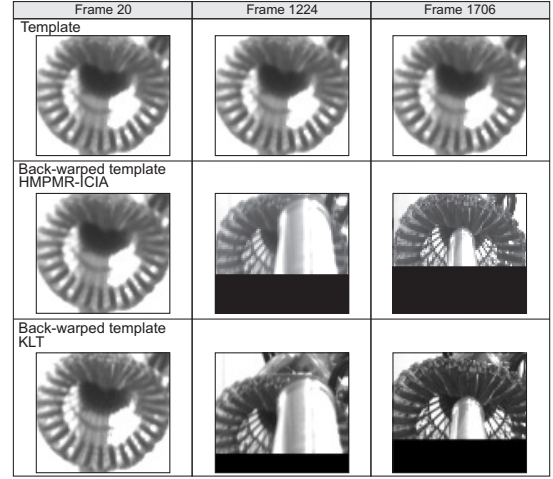


Fig. 8. Visual analysis of the estimated motion. The first row shows the template image (i.e. the object to track). The second and third rows show the back-warped templates.

In Fig. 8, the first row shows the template image. The second row shows the back-warped template found using the motion model estimated by the HMPMR-ICIA algorithm, and the third row shows the back-warped template found using the motion model estimated by the KLT algorithm. In this figure, it can be seen that the transformation recovered by the HMPMR-ICIA (second row) appropriately transforms the current frame to the coordinate frame of the first image. Whereas in the third row, it can be seen that in some frames, e.g. frames 1224 and 1706, the KLT can not find a good motion model in the final part of the refuelling test (the most critical one). As a consequence, as it was also shown in Fig. 7, the KLT fails to track the drogue. With regard to execution times, both algorithms run at real-time frame rates: the KLT runs > 40 fps and the HMPMR-ICIA > 30 fps

In a second test, the HMPMR-ICIA algorithm is tested in a refuelling exercise with light turbulence conditions. Fig 9 shows the position and orientation data recorded from the robots which is used as ground-truth data.

Light turbulence conditions represent a more challenging scenario for the tracking algorithm than those exhibited in the no turbulence case, with changes in orientation, including roll, pitch and yaw (Fig. 9); occlusion of the drogue by the probe; large changes in scale; and sudden motions.

In the thumbnail images shown in 9, it can be seen that the HMPMR-ICIA algorithm tracked the drogue during the entire refuelling task in spite of the motions caused by the turbulence effects (e.g. orientation changes).

Fig. 10 shows the errors obtained when the position estimated by the tracking algorithm is compared with the position of robot R2 (the receiver aircraft). As can be seen, the obtained vision-based position data have similar values

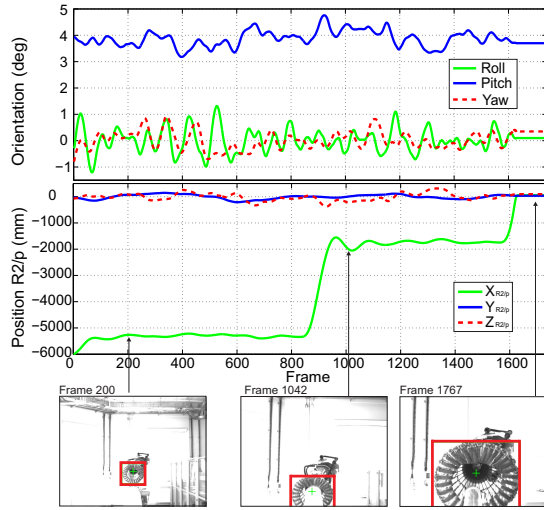


Fig. 9. Motions of a refuelling task under light turbulence conditions. The AAAR testbed reproduces simulated motion of the receiver aircraft. Thumbnail images show the results of the tracking algorithm.

as the ones of the GT data. The RMSEs obtained in the three axes are < 55 mm. These errors are low, considering that the estimation is based on a monocular system. Videos of the tests can be seen in [12].

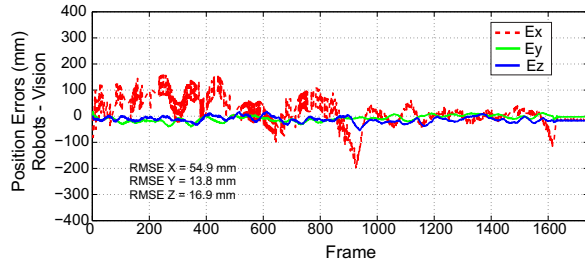


Fig. 10. Position errors. Errors obtained from the comparison of the position estimated by the tracking algorithm with the position of robot R2 (the receiver aircraft).

V. CONCLUSIONS

Results of using visual information for AAAR tasks for UAVs have been presented. We propose to use a hierarchical implementation of the ICIA algorithm (HMPMR-ICIA). This strategy makes use of the information of all the pixels that belongs to the drogue in order to estimate the motion, with the advantages that the installation of specific hardware on-board is not required, and that the robustness in the estimation of the motion model is increased (large image motion is tolerated). The afore mentioned advantage is particularly important during the refuelling task, taking into account that turbulence effects will produce strong and large motions in the image plane.

The algorithm has been tested in a laboratory facility using real visual information from real refuelling hardware (a drogue). Results showed that the HMPMR-ICIA algorithm can solve the drogue tracking problem for AAAR tasks, overcoming adverse conditions, such as those present when

the drogue is being occluded by the probe, when part of the drogue goes out of the field of view of the camera, fast motions, and orientation changes. Results also demonstrate that for the drogue tracking task, the implemented direct method obtains better results than the feature-based KLT algorithm. With the direct method, robust position estimations of the drogue at real-time frame rates (> 30 fps) were obtained (without optimizing the code in any way) with an average accuracy of 6 cm in the position estimation for the light turbulence test.

These results demonstrate that this algorithm is a suitable candidate for the AAAR task for UAVs, being fast enough to form a basis for automated aerial refuelling sensing capabilities. Future work will focus on increasing robustness of the algorithm, such as considering possible illumination changes during the operation. In future work this algorithm will be augmented by other sensors and incorporated into control algorithms.

VI. ACKNOWLEDGEMENTS

This work is jointly funded by a Ph.D. Scholarship from the Universidad Politécnica de Madrid (UPM), the Spanish Ministry of Science MICYT DPI2010-20751-C02-01, and by Cobham Mission Equipment as part of the ASTRAEA Programme. Website: <http://www.astraea.aero/>

REFERENCES

- [1] J. H. Spencer. Optical Tracking for Relative Positioning in Automated Aerial Refueling. MS Thesis, Air Force Institute of Technology, March 2007.
- [2] Lorenzo Pollini, Giampiero Campa, Fabrizio Giulietti, and Mario Innocenti. Virtual Simulation Set-Up for UAVs Aerial Refuelling. In *AIAA Guidance Navigation and Control Conference and Exhibit*, pages 1–8, August, 2003.
- [3] John Valasek, Kiran Gunnam, Declan Hughes, John Junkins, Jennifer Kimmitt, and Monish Tandale. Vision-Based Sensor and Navigation System for Autonomous Air Refueling. *Journal of Guidance Control and Dynamics*, 28(5):979–989, 2005.
- [4] James Doebbler, Theresa Spaeth, John Valasek, Mark J Monda, and Hanspeter Schaub. Boom and Receptacle Autonomous Air Refueling Using Visual Snake Optical Sensor. *Journal of Guidance Control and Dynamics*, 30(6):1753–1769, 2007.
- [5] Soujanya Vendra, Giampiero Campa, Marcello Napolitano, Marco Mammarella, Mario Fravolini, and Mario Perhinschi. Addressing Corner Detection Issues for Machine Vision Based UAV Aerial Refueling. *Machine Vision and Applications*, 18:261–273, 2007.
- [6] Simon Baker and Iain Matthews. Equivalence and Efficiency of Image Alignment Algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 1090 – 1097, December 2001.
- [7] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2003.
- [8] Carol Martinez, Pascual Campoy, Ivan F. Mondragón, Jose Luis Sanchez-Lopez, and Miguel Olivares-Mendez. A Hierarchical Strategy for Real-Time Tracking On-Board UAVs. In *International Conference on Unmanned Aircraft Systems*, Philadelphia, PA, USA, June, 2012.
- [9] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly, 2008.
- [10] Jean Y. Bouguet. Pyramidal Implementation of the Lucas Kanade Feature Tracker: description of the algorithm. Technical report, OpenCV Document, Intel Microprocessor Research Labs, 2002.
- [11] Ivan F. Mondragón, Pascual Campoy, Carol Martinez, and Miguel Olivares-Mendez. 3D Pose Estimation Based on Planar Object Tracking for UAVs Control. In *ICRA2010*, Anchorage, AK, USA, April 2010.
- [12] Computer Vision Group 2013. Videos Autonomous Air-to-Air Refuelling. <http://www.vision4uav.com/?q=HMPMRtracker>.