A Hierarchical Tracking Strategy for Vision-Based Applications On-Board UAVs

Carol Martínez, Iván F. Mondragón, Pascual Campoy,
José Luis Sánchez-López, and Miguel A. Olivares-Méndez

Computer Vision Group
www.vision4uav.com
Centro de Automática y Robótica CAR UPM-CSIC
Universidad Politécnica de Madrid, José Gutiérrez Abascal 2, 28006 Madrid, Spain
Email: carolviviana.matinez@upm.es

***Abstract*—In this paper, we apply a hierarchical tracking strategy of planar objects (or that can be assumed to be planar) that is based on direct methods for vision-based applications on-board UAVs. The use of this tracking strategy allows to achieve the tasks at real-time frame rates and to overcome problems posed by the challenging conditions of the tasks: e.g. constant vibrations, fast 3D changes, or limited capacity on-board. The vast majority of approaches make use of feature-based methods to track objects. Nonetheless, in this paper we show that although some of these feature-based solutions are faster, direct methods can be more robust under fast 3D motions (fast changes in position), some changes in appearance, constant vibrations (without requiring any specific hardware or software for video stabilization), and situations in which part of the object to track is outside of the field of view of the camera. The performance of the proposed tracking strategy on-board UAVs is evaluated with images from real-flight tests using manually-generated ground truth information, accurate position estimation using a Vicon system, and also with simulated data from a simulation environment. Results show that the hierarchical tracking strategy performs better than well-known feature-based algorithms and well-known configurations of direct methods, and that its performance is robust enough for vision-in-the-loop tasks, e.g. for vision-based landing tasks.**

I. INTRODUCTION

Robust visual estimation at real-time frame rates is one of the main problems when addressing the visual tracking task on-board UAVs. If the difficulties to obtain it are overcome, the recovered visual information can be used in a variety of vision-based control tasks, allowing to expand the vehicle's capabilities (e.g. vision-based landing, visual inspection), or to cope with vulnerabilities of other on-board sensors (e.g. GPS fallouts, Inertial Navigation System -INS- drift).

In previous works [1], [2], [3], we have used features-based methods [4] to track planar scenes on-board UAVs (Unmanned Aerial Vehicles). We have seen that in the application of tracking on-board UAVs (see Fig. 1), the adopted feature-based strategies are very sensitive to strong motions (e.g. vehicle vibrations and fast 3D changes), being it difficult to find a compromise between achieving real-time and accurate estimations (defining a specific number of good features to track without increasing the processing time). Although multi-resolution (MR) approaches (e.g. [5]) can help coping with strong and large motion problems, constant vehicle vibrations, a low computational capacity available on-board, and delays in the communication (when images are processed on the ground), are problems that make the MR strategies insufficient to properly perform the tracking task. Additionally, it has also been observed that when using feature-based methods under strong motions, the accumulation of errors make the tracking algorithm fail after just a few frames, affecting and making on-line tests difficult.

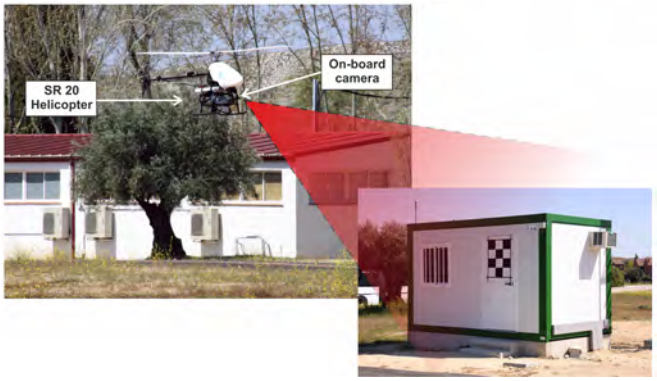

Fig. 1. Tracking on-board UAVs. Robust real-time tracking allows to expand the vehicle's capabilities by using the tracking algorithm in vision-based control tasks, such as landing, visual inspection, etc.; or by using it to cope with vulnerabilities of other on-board sensors, such as GPS drop-outs or INS drift.

In this paper, we present a hierarchical tracking strategy based on direct methods [6] for tracking on-board UAVs. Direct methods have the advantages of solving, without intermediate steps, the motion of the camera and the matching of the pixels using the intensity information of all the pixels of the object to track, without identifying a special set of features. However, in most situations, feature-based methods are preferable to direct methods. This is because direct methods are based on some constraints [6] that are, in some cases, very difficult to preserve, and their speed is dependent on the number of pixels in the image template (the one that contains the object to track), being it sometimes difficult to achieve

real-time frame rates.

Nonetheless, the tracking strategy used in this paper (based on direct methods) is robust under long frame-to-frame motions, and under constant vibrations. This permits to obtain a robust object tracking without compromising the real-time operation required in on-line applications.

In the literature, different strategies have been presented to solve the tracking problem in aerial images. Most of the strategies are based on feature-based methods [7], [8], [3], [9], [10], and just a few have explored the use of direct methods [1], [11].

In this paper, a hierarchical strategy in terms of image resolution and number of parameters estimated in each resolution is used. This strategy permits to improve the tracking task in situations where MR approaches are not enough to cope with long frame-to-frame motions. In the literature, to the authors' knowledge, this strategy has not been presented for solving the on-line tracking problem on-board autonomous vehicles. For this reason, the intention of this paper is also to expand the use of direct methods in real-time applications (e.g. vision-based landing).

Our strategy uses the efficient Inverse Compositional Image Alignment Algorithm ICIA [12] in a Hierarchical Multi-Parametric and Multi-Resolution framework (HMPMR-ICIA), that makes use of two hierarchical structures: the multi-resolution (MR) and the multi-parametric (MP) ones. We have successfully applied this strategy to solve our tracking problem on-board a UAV. We have found that if this strategy is adopted, it is possible to obtain robust estimations at real-time frame rates with complex motion models.

The paper is organized as follows: in Section II, we give a general idea of the visual tracking task based on direct methods. Section III describes the hierarchical strategy for tracking. In this section, we describe the advantages of using at the same time the MP and the MR structures. We also show the different parameters that the HMPMR strategy requires, and we present the HMPMR-ICIA algorithm used for tracking on-board UAVs. The performance of the HMPMR-ICIA algorithm is analyzed under different conditions in Section V. In this section, the hierarchical tracking strategy is compared with well-known feature-based methods: the KLT [5] (pyramidal Lucas Kanade) and the SIFT [13] (Scale-Invariant Feature Transform), and also with ground truth data generated both manually and with a Vicon system (a vision-based motion capture system) [14]. Additionally, in this section, the HMPMR-ICIA algorithm is used to track a helipad in order to conduct a vision-based landing task. Finally, in Section VI, conclusions and the direction of future work are presented.

II. Visual Tracking Based on Direct Image Registration

The 2D visual tracking task consists in determining the position of an object in the image plane in each frame of the sequence, assuming that the 3D displacements of the object can be modeled by a 2D transformation (e.g translation, affine, homography [15]).

This tracking task can be formulated as an incremental image registration task, as shown in Fig. 2. Therefore, using direct methods (i.e. direct image registration), the 2D position in the image plane can be found using the intensity values of the pixels that belong to the object, assuming that an initial position of the object is known (found manually or automatically by detection algorithms), that the motion between frames is small, that the pixels that belong to the object move similarly, and that the appearance of the object does not change over time (the direct methods' constraints [6]).

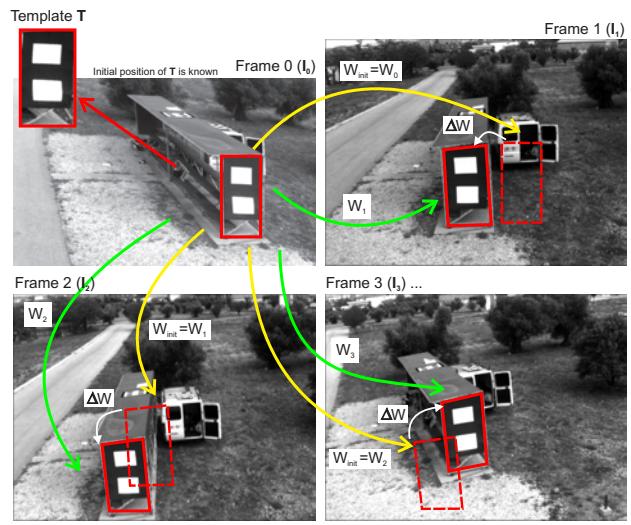

Fig. 2. Tracking as an incremental image registration task.

In the case of image registration the inputs are the two images to be registered: the template image $\mathbf{T}$, and the current image $\mathbf{I}$. These images must overlapped; and the output is a geometrical transformation, which transforms points in one image to points in the other image. Nonetheless, in the case of tracking based on image registration, the inputs are also two images, with the slight difference that in this case the reference image $\mathbf{T}$ is an image or a sub-region of an image that contains the object we want to track, and the current image corresponds to a frame of a sequence. Therefore, the registration is conducted between the template image (fixed image) and the current image of the sequence, using an initial estimation of the motion model: the one that corresponds to the location of the object to track in the previous frame, so that the images are close enough to be registered. The output, in the case of tracking, for every frame analyzed, is also a transformation that relates the reference image and the image of the sequence, but in this case that transformation is used to identify the position of the template image in each frame of the sequence.

As shown in Fig. 2, a reference image (the object to track) is defined in the first frame (template $\mathbf{T}$, Fig. 2, frame 0, upper left image). This reference image corresponds to a sub-image or ROI (Region of Interest), called image template ($\mathbf{T}$), defined in the first frame $\mathbf{I}_0$ (the subscript represents the number of the frame), and is found either manually or automatically by

detection algorithms.

When a new frame is analyzed $\mathbf{I}_1$ (Fig. 2, frame 1, upper right image), the motion between the reference and the current images $\mathbf{W_1}$ (Fig. 2, frame 1, green arrow) is found by an image registration technique, assuming that an initial estimation of the motion $\mathbf{W}_{\text{init}} = \mathbf{W_0}$ is known (Fig. 2, frame 1, yellow arrow). When an initial estimation is not know, this initial estimation can be assumed as the identity matrix, assuming that the frame-to-frame motion is small.

Therefore, the image registration algorithm is in charge of estimating the incremental motion model ($\Delta\mathbf{W}$) in every iteration. Thus, the motion $\mathbf{W_1}$ is estimated, and as a consequence of this, the position of the object to track is found in the current frame.

Then, the estimation found between frame 0 and frame 1 ($\mathbf{W_1}$) is propagated to the next frame, as an initial estimation of the motion ($\mathbf{W}_{\text{init}} = \mathbf{W_1}$ yellow arrow, Fig. 2, frame 2, button left image). The process is repeated with each frame of the sequence: the image registration technique finds $\mathbf{\Delta W}$, the motion between the reference and the current frames $\mathbf{W_2}$ is also found (Fig. 2, green arrow, button left image), and the estimated motion continues being propagated to the following frames.

In the previously mentioned process, the motion model $\mathbf{W}$ represents the trajectory of the object in the image plane while it moves around the scene. It is a 3×3 matrix (1) parameterized by the vector of parameters $\mathbf{p} = (p_1, ...p_n)^{\mathrm{T}}$ in such a way that $\mathbf{W}$ is the identity matrix when the parameters are equal to zero.

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

$$\mathbf{W} = \begin{bmatrix} 1+p_1 & p_2 & p_3 \\ p_4 & 1+p_5 & p_6 \\ p_7 & p_8 & 1 \end{bmatrix} \tag{1}$$

As shown in (1), $\mathbf{W}$ is the motion model that transforms the 2D pixel coordinates $\mathbf{x}$ (where $\mathbf{x} = (x, y, 1)^{\mathrm{T}}$) in image $\mathbf{T}$ into the 2D coordinates $\mathbf{x}' = (kx', ky', k)^{\mathrm{T}}$ in image $\mathbf{I}$.

$\mathbf{W}$ can model different 2D transformations with different numbers of parameters [15], e.g. translation (2 parameters), rotation + translation (3 parameters), similarity (4 parameters), affine (6 parameters), and homography (8 parameters). If $\mathbf{W}$ represents the homography, then $k = xp_7 + yp_8 + 1$. Otherwise, $k = 1$.

In our application, the assumption of 2D motion models is enough, considering that the tracking algorithm will be used for tracking planar surfaces (building inspection, helipad for landing) or non planar surfaces that can be assumed planar when flying at high altitudes, as shown in [16].

III. The Hierarchical Tracking Strategy

The tracking strategy used on-board UAVs, based on direct methods, is a hierarchical multi-parametric and multi-resolution strategy (HMPMR). It makes use of two hierarchical structures: the multi-resolution (MR) and the multi-parametric (MP) ones, as shown in Fig. 3. The MR structure is created by downsampling the images [17], [18]. The MP estimation takes place inside this pyramidal structure in resolution.

For each level of the pyramid, as shown in Fig. 3, a specific motion model is recovered (different motion models are estimated in each level). The idea is that the number of estimated parameters increases (i.e. the complexity of the motion model increases) with the resolution of the image, as shown in Fig. 3.

There are different advantages of integrating the MP and the MR strategies. As pointed out in [19], the MR strategy has been focused on computational efficiency and accuracy, suggesting the idea that at low resolutions, the vector of motion is smaller and long displacements can be better approximated by improving the estimation using higher resolution information.

In a strategy using only a MR approach, the same motion model is estimated in each level of the pyramid. The higher the frame-to-frame motion is, the bigger the number of levels the MR structure requires to be able to cope with the large displacement. Nonetheless, if many levels are required, it may be possible that due to the subsampling of the image, the information at low resolutions could be insufficient (depending on the quality and size of the images) to find a robust estimation of a motion model with a high number of parameters, presenting an unstable behavior when estimating motion models with a high number of parameters.

If, on the contrary, less pyramid levels are considered in order to avoid the loss of information due to the low resolution, then this reduction of levels will cause a reduction in the range of motion the algorithm can tolerate. For these reasons, for our application, sometimes MR approaches fail to solve the tracking problem. Nevertheless, by integrating the MR and the MP structures, the HMPMR approach will allow to continue taking advantage of the low resolution information to find a large range of motion even when motion models with a high number of parameters are estimated.

The HMPMR strategy requires the definition of different parameters, such as the number of levels (pL) of the MR structure and the motion models in the MP structure.

The different levels in the MR pyramid (pL) are defined as a function of the size of the template image $\mathbf{T}$, so that in the lowest resolution level, i.e. the $j_{\max}$ level (where j represents the level), an image with not less than a defined number of pixels ($minPixels$) will be used. Therefore, pL is defined as follows, taking into account that the images are downsampled by a factor of 2:

$$2^{pL} = \frac{lowS}{minPixels} \tag{2}$$

Where $lowS$ represents the lowest size between the width and height values of image $\mathbf{T}$ (the ROI that contains the object of interest), and $minPixels$ is defined as the minimum size the template must have in the lowest resolution image (e.g. 5 pixels). Thus, with (2), the different levels of the MR structure can be defined automatically, depending on the size of the image template.

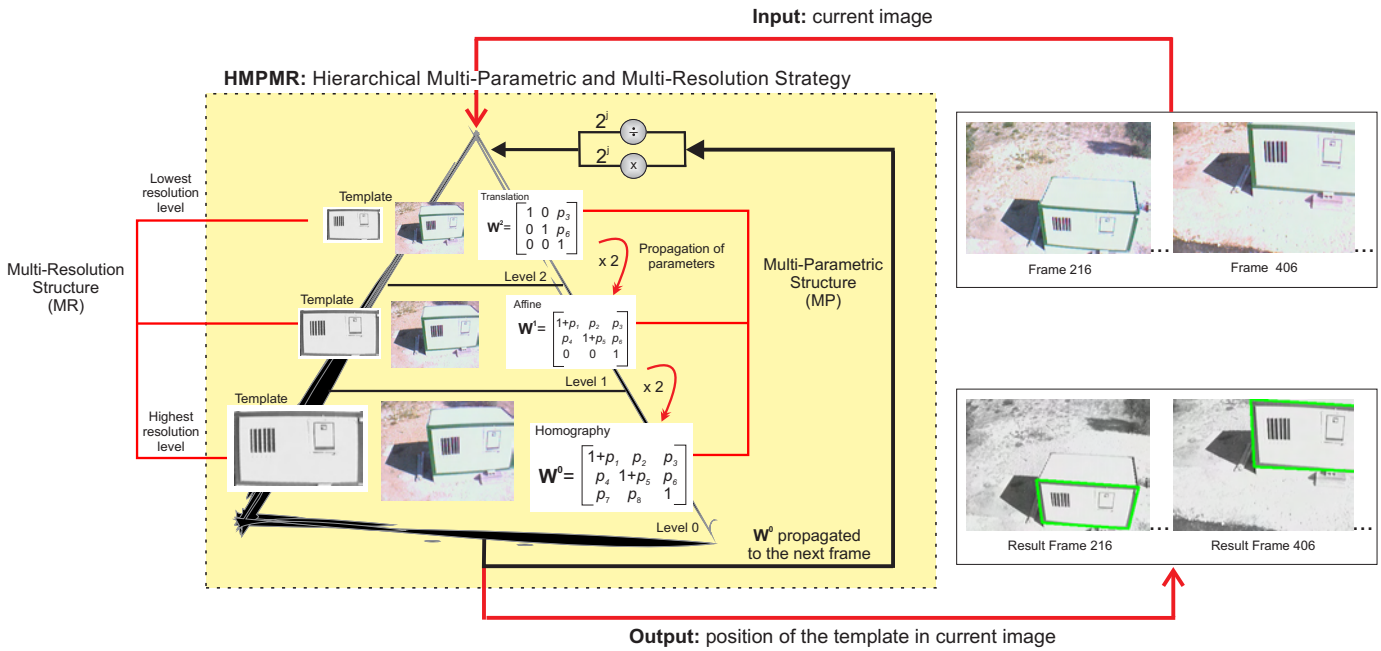

Fig. 3. Hierarchical Tracking Strategy. A multi-parametric (MP) structure inside a multi-resolution (MR) scheme is used to improve the tracking problem on-board UAVs, especially when the range of motion between frames is large. The MR structure is created by downsampling the images. Inside this pyramidal structure in resolution, the MP estimation takes place. Different motion models with different numbers of parameters are estimated in each level. The motion model found in the lowest level of the pyramid (level 0) permits to find the position of the template image in the current image. Additionally, this motion model is propagated as initial guess to the next frame.

On the other hand, the MP structure is defined according to the motion model selected at the lowest level of the pyramid $\mathbf{W}^0$ —the highest resolution level— (the superscript represents the level). In this level, $\mathbf{W}^0$ must be chosen as the best transformation that represents the motion of the object or the motion of the camera in the image plane.

Additionally, in order to ensure the detection of large frame-to-frame motion, the translation motion model is chosen for the highest level of the pyramid $\mathbf{W}^{j_{\max}}$ (the level that has the lowest resolution image), while for the other levels the motion models are selected, so that a smooth transition of the number of parameters from the highest to the lowest level of the pyramid is achieved.

If a camera is moving in the 3D space, then a possible combination of motion models can be 8-4-3-2 in a pyramidal structure with four levels. The first number corresponds to the motion model that will be estimated in the lowest pyramid level —highest resolution image— $\mathbf{W}^0$, in this case the homography. The last number corresponds to the motion model that will be estimated in the highest pyramid level —lowest resolution image— $\mathbf{W}^{j_{\max}}$, in this case the translation; and the other numbers correspond to the motion models estimated in the intermediate levels, in this case the similarity (4 parameters) and the translation+rotation motion model (3 parameters).

A. HMPMR-ICIA Algorithm

The image registration process consists in aligning two images, a reference image or image template ($\mathbf{T}$) and the current image ($\mathbf{I}$), by finding the transformation ($\mathbf{W}$) that best aligns them. This transformation or motion model ($\mathbf{W}$) is normally found iteratively, by minimizing the sum of squared differences (SSD) between the reference image and the current image [20].

Different minimization algorithms have been used in different fields: pose estimation [21], tracking [22], motion segmentation [23], and mosaics [24]. However, the gradient descent optimization (based on a first order Taylor series approximation of the SSD) is one of the most used approaches because of its efficiency [25], [26], [27]. In [25], gradient descent approaches were classified according to the update rule of the parameters of the motion model as: forwards additive [26], forwards compositional [24], inverse additive [27], and inverse compositional [12].

The image registration algorithm we use for tracking is the Inverse Compositional Image Alignment algorithm (ICIA) proposed in [12]. It is considered an efficient algorithm for image registration (or image alignment) that permits an efficient estimation of the parameters that define the motion of the object $\mathbf{W}$.

The goal of the ICIA consists in finding the vector of parameters $\mathbf{p}$ of the motion model (1) by minimizing:

$$\sum_{\mathbf{x}} [T(\mathbf{W}(\mathbf{x};\Delta\mathbf{p})) - I(\mathbf{W}(\mathbf{x};\mathbf{p}))]^2 \tag{3}$$

The increment of the parameters ($\Delta\mathbf{p}$) is found after a first-order Taylor series expansion of (3). Then, the motion model

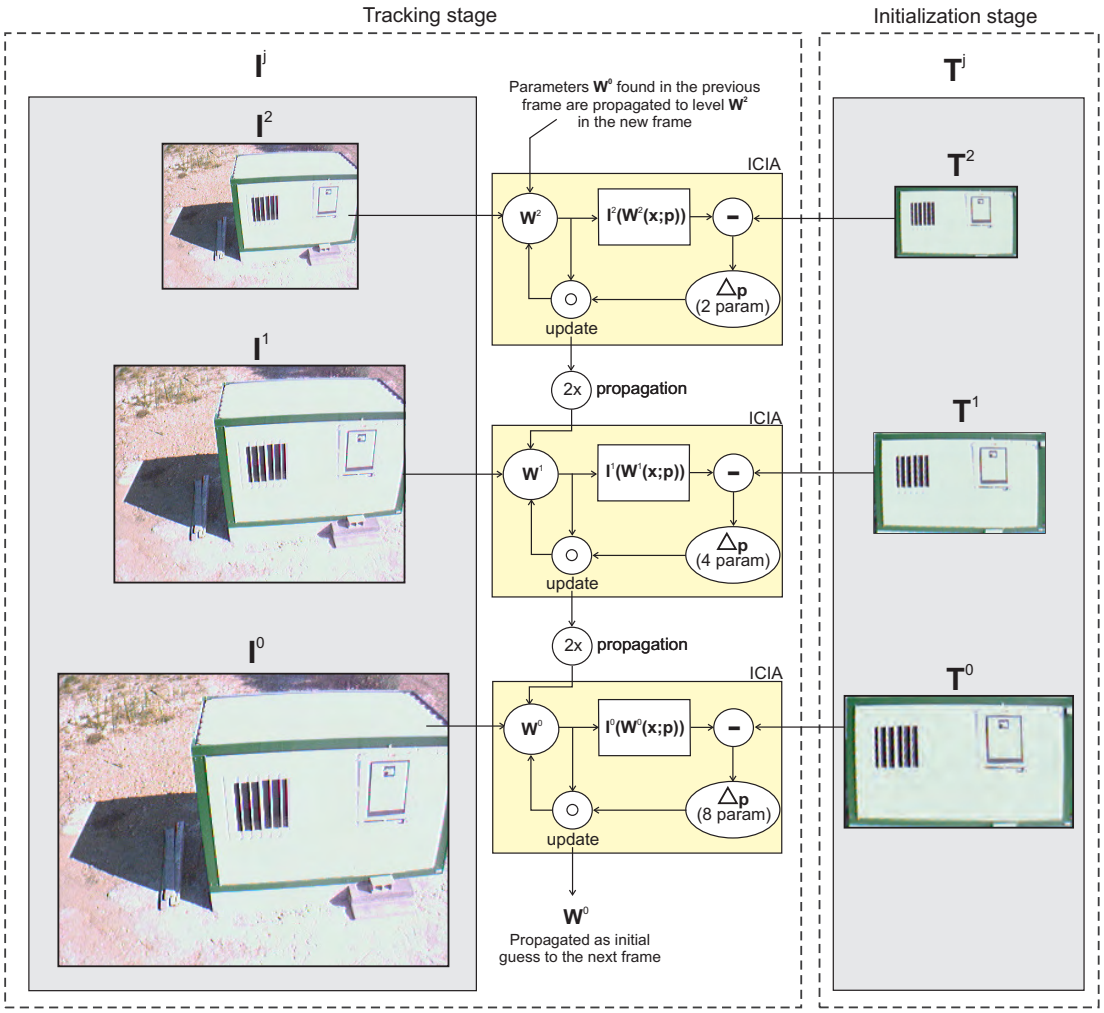

Fig. 4. HMPMR-ICIA. Images $\mathbf{I}$ and $\mathbf{T}$ are downsampled to create the MR structure. In each level, the $\mathbf{x}$ coordinates in $\mathbf{T}$ are transformed by $\mathbf{W}^j$, and the errors between the intensity values in $\mathbf{I}^j(\mathbf{W}(\mathbf{x};\mathbf{p}))$ and in $\mathbf{T}^j$ are calculated. Then, $\Delta\mathbf{p}$ is calculated, and the parameters are updated. The minimization is conducted with respect to the parameters defined in each level. When the stopping conditions have been reached, the parameters are propagated to the next level. At level $j = 0$, the most complex motion model is estimated. This final estimation $\mathbf{W}^0$ corresponds to the transformation that allows to find the template image $\mathbf{T}$ in image $\mathbf{I}$.

is updated, as follows:

$$\mathbf{W}(\mathbf{x};\mathbf{p}) \leftarrow \mathbf{W}(\mathbf{x};\mathbf{p}) \circ \mathbf{W}(\mathbf{x};\Delta\mathbf{p})^{-1} \tag{4}$$

The increment in the parameters $\Delta\mathbf{p}$ of the motion model (1) is estimated iteratively until stopping criteria are reached, denoting the best local alignment solution. In our implementation we have defined three criteria: the minimum is reached if the increment of the parameters is below a threshold $||\Delta\mathbf{p}|| \leq 10^{-5}$, if the MAE (mean absolute error) between $\mathbf{T}$ and $\mathbf{I}$ does not decrease after a defined number of iterations (10 iterations), or if the maximum number of iterations have been reached (100 iterations).

The efficiency of the ICIA algorithm comes from the change of roles of images $\mathbf{I}$ and $\mathbf{T}$ in (3), and the way the motion model is updated (4). This change of roles makes the Hessian matrix be constant, calculated at the beginning of the tracking task, and so a fast alignment is achieved.

Nonetheless, this iterative algorithm relies on the assumption that a previous estimation of the parameters of the motion model is known, and that after a linearization of the cost function (3), the algorithm iteratively solves the increment of the parameters. Nevertheless, this linearization is valid only when the range of motion is small.

In our application (tracking on-board UAVs) as well as in other ones, this constraint can not be always ensured (limited capacity on-board, and so low processing units on-board; fast 3D motions; etc), and although MR approaches were proposed to help dealing with this problem, the use of a HMPMR strategy instead of only a MR will help increasing the range of motion that the algorithm can tolerate.

Therefore, by using the ICIA algorithm, an efficient tracking algorithm can be achieved using direct methods; and by integrating it with the HMPMR structure, robust motion estimations are achieved, allowing to track objects during long periods of time at real-time frame rates.

Fig. 4 shows the diagram of the HMPMR structure for tracking using the ICIA algorithm, and Algorithm 1 describes the different steps in more detail.

As input, the algorithm requires the information of $\mathbf{I}_0$ (the first frame), and the coordinates ($\mathbf{x}$) in $\mathbf{I}_0$ of the object to track. These coordinates can be found manually or automatically using detection algorithms, e.g using template matching approaches [28]. Additionally, the algorithm requires the definition of the levels of the MR structure (pL) defined using (2), and the definition of the different motion models in the MP structure $\mathbf{W}^j$. The definition of the motion models depends on different criteria: the complexity of the task, the application (building inspection, landing), and the configuration of the camera in the UAV (forwards-looking or downwards looking). As explained previously, according to these criteria $\mathbf{W}^0$, must be defined as the best transformation that represents the motion of the object or the motion of the camera.

Once this information is known, $\mathbf{I}_0$ is downsampled according to the different levels (pL) of the MR structure, thus creating the template image $\mathbf{T}^j$ for each level, as shown in Fig. 4 (initialization stage, right column). Additionally, in this initialization stage, for each level of the pyramid, the Hessian matrix and its inverse are calculated, as shown in more detail in Algorithm 1, steps 1–6. These steps are carried out only once, at the beginning of the tracking task.

When a new frame $\mathbf{I}$ is analyzed, it is first downsampled to create the MR structure, as shown in Fig. 4 (tracking stage, left column). The motion model at the highest level (lowest resolution) ($\mathbf{W}^{j_{\max}}$) is initialized. Because this is the first frame, $\mathbf{W}^{j_{\max}}$ is the identity matrix.

For each level of the pyramid, as illustrated in Fig. 4, the HMPMR-ICIA algorithm is applied as follows:

1) The coordinates $\mathbf{x}$ in $\mathbf{T}$ are warped using $\mathbf{W}^j$; and $\forall \mathbf{x}$, the error between $T(x)$ and $I^j(\mathbf{W}^j(\mathbf{x};\mathbf{p}))$, is calculated (steps 9-11, Algorithm 1).
2) The increment of the parameters is found using step 12, Algorithm 1.
3) The motion model is updated using (4), step 3, Algorithm 1.
4) In each level of the pyramid, 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 as follows, taking into account that the images have been scaled by a factor of two:

$$\begin{aligned} p_i^{j-1} &= p_i^j \quad for \quad i=\{1,2,4,5\} \\ p_i^{j-1} &= 2p_i^j \quad for \quad i=\{3,6\} \\ p_i^{j-1} &= \frac{p_i^j}{2} \quad for \quad i=\{7,8\} \end{aligned} \tag{5}$$

being,

$$j=\{j_{\max}, j_{\max}-1,\ldots,0\}=\{pL-1, pL-2,\ldots,0\}$$

Where the subscript i represents the parameters defined in (1), and j represents the level of the pyramid. j is initialized as $j=j_{\max}$, where $j_{\max}=pL-1$, where pL is the number of levels the pyramid has, as defined in (2).

At the lowest level of the pyramid (i.e the one that has the image with the highest resolution), the motion model $\mathbf{W}^0$ will contain the parameters that minimize the differences between the template and the current images. This motion model is the best approximation to the motion of the object in the image plane. With this information, the position of $\mathbf{T}$ (i.e. the object to track) in the current image $\mathbf{I}$ can be determined (steps 15-16, Algorithm 1).

The motion model found in this frame is propagated as initial guess to the highest level of the pyramid, $j_{\max}$, of the next frame, as follows:

$$\begin{aligned} p_i^{j_{\max}} &= p_i^0 \quad for \quad i=\{1,2,4,5\} \\ p_i^{j_{\max}} &= \frac{p_i^0}{s} \quad for \quad i=\{3,6\} \\ p_i^{j_{\max}} &= sp_i^0 \quad for \quad i=\{7,8\} \end{aligned} \tag{6}$$

Where $s=2^{j_{\max}}$ (step 8, Algorithm 1).

This 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 permits to validate the linearization of (3) 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 each other to find a minimum.

The pseudocode of the HMPMR-ICIA algorithm is presented in Algorithm 1.

IV. 3D Position Estimation

In this section a pose estimation algorithm, widely used in the literature [29] [30] [2], is used to estimate the 3D position of the object to track. The method is based on the "world homography", which transforms points on the world plane to points in the image plane [29].

Assuming that the visual tracking algorithm is capable of obtaining the ROI where the object to track is located, the position estimation algorithm uses that ROI in order to calculate the "world homography", and then from this homography to extract the position between the world and the camera coordinate systems.

Therefore, the 2D position in the image plane of the object to track (see Fig. 5) obtained by the visual tracking algorithm are transformed into 3D positions assuming that the dimensions of the object to track are known, that the 3D points of the object to track lie on a plane, and that the camera calibration parameters [31] are also known.

Using the pinhole camera model [15], 3D coordinates can be related to the 2D image coordinates, as follows:

$$\mathbf{x}_{\mathrm{f}} = \lambda \mathbf{K}[\mathbf{R} \mid \mathbf{t}]\mathbf{x}_{\mathrm{w}} \tag{7}$$

input : $\mathbf{I_0}$, pL, $\mathbf{I}$, $\mathbf{x}$, configuration of $\mathbf{W}^j$
output: $\mathbf{W}^0$, transformation that finds the position of $\mathbf{T}$ in $\mathbf{I}$
Pre-compute
1. Downsample $\mathbf{I_0}$ according to pL and create $\mathbf{T}^j$
2. Initialize $j_{\max} = pL - 1$
for $j \leftarrow j_{\max}$ **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 $(\mathbf{x}, \mathbf{0})$
  5. Compute the steepest descent images SDI $\mathbf{SDI}(\mathbf{x}) = \nabla \mathbf{T}(\mathbf{x})^j \frac{\partial \mathbf{W}^j}{\partial \mathbf{p}}$
  6. Compute the Hessian matrix and its inverse $\mathbf{H}^j = \sum_{\mathbf{x}} \left[\mathbf{SDI}(\mathbf{x})^{\mathrm{T}} \mathbf{SDI}(\mathbf{x})\right]$, 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 $\mathbf{W}^{j_{\max}}$ according to (6). If it is the first frame, $\mathbf{W}^{j_{\max}}$ is the identity matrix
  for $j \leftarrow j_{\max}$ **to** 0 **do**
    repeat
      foreach $\mathbf{x}$ *in* $\mathbf{T}$ **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 $\mathbf{b}^j = \mathbf{b}^j + \mathbf{SDI}(\mathbf{x})^{\mathrm{T}} E^j$
      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 using (5)
  end
  15. Use $\mathbf{W}^0$ to find the position of $\mathbf{T}$ in $\mathbf{I}$
  16. Draw results
end

Algorithm 1: HMPMR-ICIA tracking algorithm

Where $\mathbf{x}_{\mathrm{f}} = (x_{\mathrm{f}}, y_{\mathrm{f}}, 1)$ are the 2D image coordinates of a point; $\mathbf{x}_{\mathrm{w}} = (x_{\mathrm{w}}, y_{\mathrm{w}}, z_{\mathrm{w}}, 1)$ are the 3D world coordinates of the same point; λ is a scale factor; $\mathbf{R}$ and $\mathbf{t}$ are, respectively, 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 [31] using the camera calibration toolbox for Matlab [32].

As shown in Fig. 5, the known dimensions of the object to track can be used to define four 3D points ($\mathbf{x}_w^i$) that lie on the ground plane where the world coordinate system is located. Additionally, assuming that the tracking algorithm robustly estimates the location of the ROI of the object to track in each frame; then with the four 3D points and the four 2D points

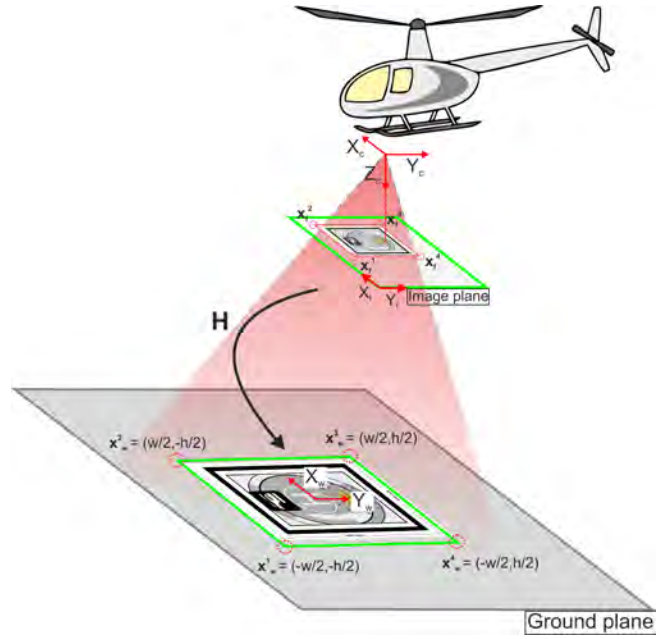

Fig. 5. Position estimation strategy. The 2D positions of the object to track in the image plane are transformed into 3D positions assuming that dimensions of the object are known, that the known 3D points lie on a plane, and that the camera calibration parameters are known.

of the ROI that defines the object to track in the image plane, expression (7) is simplified for the planar case ($z_{\mathrm{w}}^i = 0$), as follows:

$$\begin{pmatrix} x_{\mathrm{f}}^i \\ y_{\mathrm{f}}^i \\ 1 \end{pmatrix} = \lambda \mathbf{K}[\mathbf{r_1}\ \mathbf{r_2} \mid \mathbf{t}] \begin{pmatrix} x_{\mathrm{w}}^i \\ y_{\mathrm{w}}^i \\ 1 \end{pmatrix} \qquad \mathbf{x}_{\mathrm{f}}^i = \mathbf{H}\mathbf{x}_{\mathrm{w}}^i \tag{8}$$

where $\mathbf{x}_{\mathrm{w}}^i$ contains the coordinates of one of the four points that lie on the ground plane, the index i represents each corner of the ROI that inscribes the object 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 Fig. 5.

Therefore, with the point-to-point (2D-3D) correspondence of the four corners of the object, and reorganizing (8), a system of equations of the form $\mathbf{Ah} = \mathbf{b}$ can be created in order to estimate $\mathbf{H}$, where $\mathbf{h}$ corresponds to the components of $\mathbf{H}$ stacked into a vector.

Once $\mathbf{H}$ is estimated, the translation vector $\mathbf{t}$ is estimated using the method described in [33], assuming that the camera parameters are known. Thus, $\mathbf{t}$ is found based on (8), and taking into account that $\|\mathbf{r_1}\| = \|\mathbf{r_2}\| = 1$, as follows:

$$\begin{aligned} \mathbf{H} &= [\mathbf{h_1}\ \mathbf{h_2}\ \mathbf{h_3}] = \lambda \mathbf{K}[\mathbf{r_1}\ \mathbf{r_2}\ \mathbf{t}] \\ \lambda &= \|\mathbf{K}^{-1}\mathbf{h_1}\| = \|\mathbf{K}^{-1}\mathbf{h_2}\| \\ \mathbf{t} &= \frac{1}{\lambda}\mathbf{K}^{-1}\mathbf{h_3} \end{aligned} \tag{9}$$

Using (9), the 3D position of the object to track with respect to the camera coordinate system is found.

V. Results

Tests are conducted using different types of images under different conditions, in which the most complex of the 2D transformation (the homography) is estimated.

In the first test, a comparison of different configurations of the ICIA algorithm is conducted: the ICIA without hierarchies, the MR-ICIA, and the HMPMR-ICIA (all of them based on direct methods). In this test, we analyze the advantages of using, simultaneously, the MP and MR hierarchies during the tracking task on-board UAVs.

A second test is conducted in order to compare the performance of the HMPMR-ICIA with feature-based tracking algorithms when tracking objects on-board UAVs. In this test, the HMPMR-ICIA tracking strategy based on direct methods is compared with well-known feature-based methods: the SIFT [13] and the pyramidal Lucas Kanade [5] (KLT) algorithms.

Finally, in a third test, the information that is recovered by the HMPMR strategy is used to estimate vision-based position information of the state of the UAV, which can be used later for autonomous landing and take-off tasks. We present results comparing the vision-based position estimations with the estimation recovered by a Vicon system [14] in a laboratory test where the movements and visual conditions of a landing and take-off tasks are simulated. Additionally, we present results of using the 2D and the 3D positions of the object to track to conduct a vision-based landing task.

Different criteria, explained in each experiment, are used to evaluate the performance of the different tested algorithms. The evaluation is based on a visual examination of the tracking results, based on a comparison with ground truth data, and also based on a comparison of the speed reached by the algorithms.

Videos of the tests can be seen in [34].

- **Experimental setup**

The data used in tests 1 and 2 correspond to different flights conducted with the Rotomotion SR20 electric helicopter (the Colibri III system [35]), shown in Fig. 1. The images used in test 3 correspond: to a laboratory test conducted using the Vicon system (a vision-based motion capture system) and a FireWire camera; and to simulation tests conducted using a virtual environment that uses the ROS (Robot Operating System) framework [36], the 3D simulator Gazebo [37], and the Starmac aircraft model [38].

The HMPMR-ICIA and the MR-ICIA algorithms were developed in $\mathrm{C++}$ and the OpenCV libraries [28] were used for managing image data.

On the other hand, the KLT feature-based algorithm used in the second test is based on the version of the algorithm implemented in the OpenCV libraries. The maximum number of features was defined as 100, a window size of 5 was used, and four pyramid levels were used in the multi-resolution structure of the algorithm. The SIFT algorithm used in test 2 is the implementation developed by Rob Hess [39], [40]. The values of the different parameters the algorithm requires correspond to the standard values that come with the implementation of the algorithm that was used.

A. Test 1: comparison with direct methods

In this test, we evaluate the performance of the HMPMR strategy tracking part of a structure affected by the 3D motion of the UAV. We compare the proposed HMPMR-ICIA algorithm with other configurations of the ICIA algorithm: with the ICIA without hierarchies, and also with a MR-ICIA. In this test, we analyze the advantages of using, at the same time, the MR and MP hierarchies during the tracking task when large frame-to-frame motions are presented.

The object to track in the image sequence used in this test corresponds to a flat section of a 3D structure. The UAV is flying around the structure during the task. The size of the images is 640×480 pixels, and the size of the template is 84×170 pixels; so according to (2) $pL = 4$, considering that the minimum size the template should have is $minPixels = 5$. The camera on-board the UAV is in a forwards-looking configuration, and the homography (8 parameters) is chosen as the transformation that best describes the changes of the scene due to the UAV movements.

Therefore, the ICIA recovers 8 parameters (the homography), i.e. no hierarchical structure is used. The MR-ICIA recovers the same number of parameters in the different levels of the pyramid. Thus, the combination of motion models used is in the form 8-8-8-8, and the HMPMR-ICIA recovers different motion models in its structure 8-4-3-2: the homography in the lowest level of the pyramid (8 parameters), the translation in the highest level (2 parameters), and the similarity (4 parameters) and rotation+translation (3 parameters) in the intermediate levels.

The selected image sequence contains jumps of the visual information, so that long frame-to-frame motions affect the object to track (sometimes 5 and 10 pixels from frame-to-frame). This characteristic makes this sequence challenging from the visual tracking point of view.

In this first test, the evaluation of the results obtained with the different algorithms is based on a visual examination of the tracking results (if the green/light box is covering the tracked area during the sequence).

Fig. 6 presents the result of the tracking task using the ICIA algorithm without any hierarchy, recovering 8 parameters (the homography). The green/light box indicates the result of the tracking task.

As can be seen in Fig. 6, the ICIA was not able to continue tracking the template after frame 360. The large frame-to-frame motion in some parts of the sequence violates one of the main constraints of direct methods (small motion), and so the ICIA is not able to track the template in this sequence.

Fig. 7 presents a collection of images that shows the performance of the MR-ICIA during the tracking task (8 parameters are found in the four levels of the hierarchical structure). A green/light box indicates the results in each frame.

Analyzing Fig. 7, we can see that the MR-ICIA 8-8-8-8 configuration fails after frame 20. The MR-ICIA is not able to track the template in the image sequence. As mentioned in Section III, a multi-resolution hierarchy is not always enough

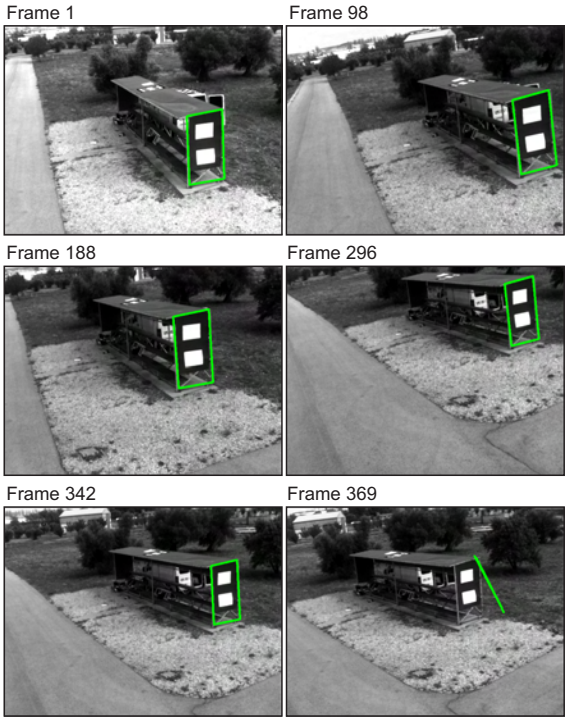

Fig. 6. Visual examination of the tracking results: ICIA. The green/light box indicates the result of the tracking task. Without using any hierarchy, the ICIA is not able to track the template when there are large motions in the sequence (> 20 pixels).

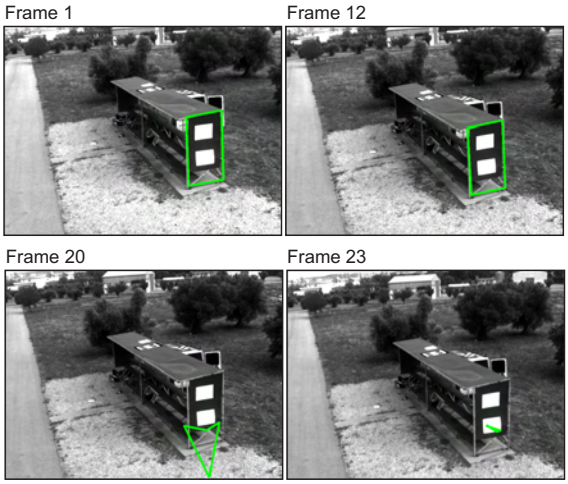

Fig. 7. Visual examination of the tracking results: MR-ICIA. The green/light box indicates the result of the tracking task. As can be seen, the MR-ICIA strategy can not track the template in all the sequence.

in our application to solve the tracking problem when large frame-to-frame motions are presented.

Additionally, as it was also mentioned in Section III, one of the problems with MR approaches is that at low resolutions the quality and quantity of the available information is not good enough to find a good estimation of motion models with a high number of parameters. For this reason, it can be seen that the MR-ICIA fails earlier than the ICIA algorithm without hierarchies.

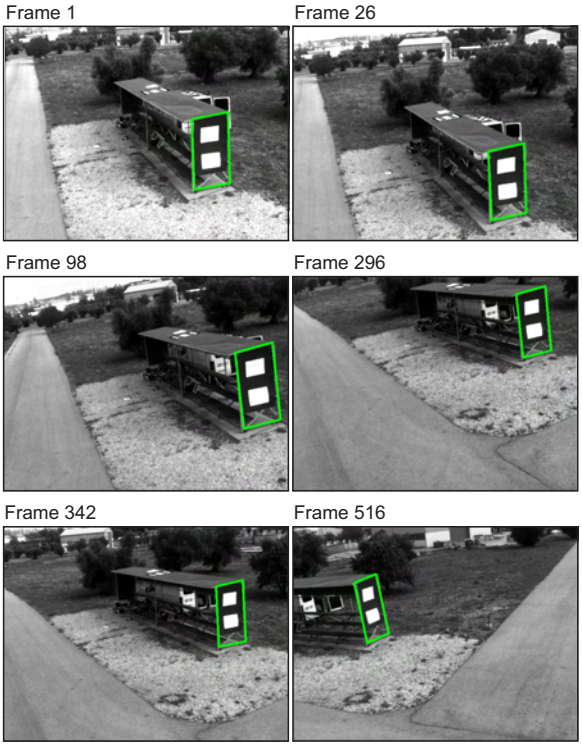

Fig. 8. Visual examination of the tracking results: HMPMR-ICIA. The green/light box indicates the result of the tracking task. The HMPMR-ICIA tracks the template throughout the sequence.

Finally, the proposed HMPMR strategy using the ICIA algorithm is tested. Fig. 8 presents a collection of images illustrating the performance of the tracking task using the HMPMR-ICIA algorithm. As can be seen, the HMPMR strategy is able to track the template in all the frames in spite of the jumps the sequence has and of the 3D changes of the sequence.

As a result of the different algorithms tested, we can conclude that the MR approach is not enough to overcome frame-to-frame motions that are > 5 pixels, whereas a well-configured HMPMR strategy can deal with large frame-to-frame motions > 5 pixels. Additionally, we could see that the HMPMR is more robust than the MR approach recovering motion models with high numbers of parameters.

B. Test 2: comparison with feature-based methods

In the previous test, it was shown that by configuring the direct method with MR and MP hierarchies, the results of the tracking task present a more robust behavior than when using only MR hierarchy or none of the hierarchies. As a consequence, using the ICIA algorithm with a HMPMR strategy is more robust than using only a MR approach in our application.

This second test compares the performance of the HMPMR-ICIA algorithm with two feature-based algorithms: the SIFT and the KLT (pyramidal Lucas Kanade). The comparison is also performed in the most difficult situation: when tracking planar objects that are affected by perspective effects due to

the 3D movements of the UAV.

In this test, a UAV is flying around a "house" with a forwards-looking camera configuration. The front of the "house" is used as template image $\mathbf{T}$. The size of the images is 320×240 pixels, and the size of the template is 213×123 pixels. Therefore, $pL = 4$ (4 pyramid levels).

The selected sequence was chosen due to some particular features found in it that help testing the performance of the algorithms. First, the images contain constant changes in positions because of the UAV's vibrations. Additionally, the sequence includes: changes in the appearance of the object to track (due to 3D movements), low texture information, and loss of information when the object goes out of the field of view (FOV) of the camera.

Taking into account the different changes in perspective throughout the sequence, the homography (8 parameters) is chosen as the transformation that best describes the changes of the scene due to the UAV movements. Therefore, the combination of motion models used in the HMPMR-ICIA algorithm is 8-4-3-2: the homography in the lowest level of the pyramid (8 parameters), the translation in the highest level (2 parameters), and the similarity (4 parameters) and rotation+translation (3 parameters) in the intermediate levels.

The evaluation of the results is based on a visual examination of the tracking results, based on the analysis of the transformation recovered by the tracking algorithms using ground truth data, and also based on the frame rate reached by the algorithms.

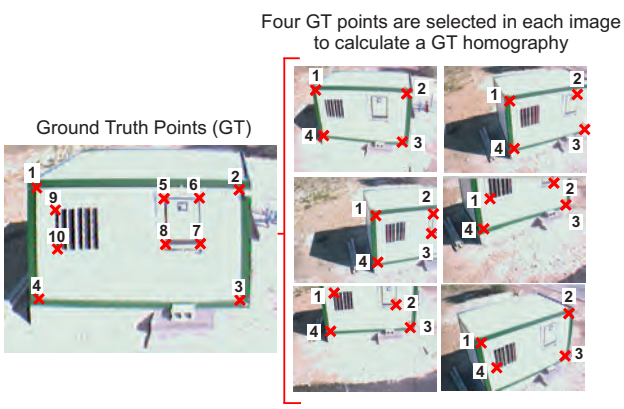

Fig. 9. Ground Truth Data. Four points (right images) of the possible 10 points (left image) are selected to calculate a ground truth homography that relates points in the first frame to points in each frame of the sequence

Ground truth data is used to analyzed the homography recovered by the algorithms. Fig. 9 shows the ground truth data. As can be seen in the image located on the left, 10 different ground truth points (GT) can be used. Nonetheless, taking into account that due to the movements of the UAV the front of the "house" goes out of the FOV of the camera, only 4 GT points (Fig. 9, right images) well distributed over the template are manually selected in each frame in order to calculate a ground truth homography, as follows:

$$\mathbf{x}_n^i = \mathbf{H}\mathbf{x}_1^i$$

This GT homography ($\mathbf{H}$) relates points in the first frame (template image) to points in each frame of the sequence.

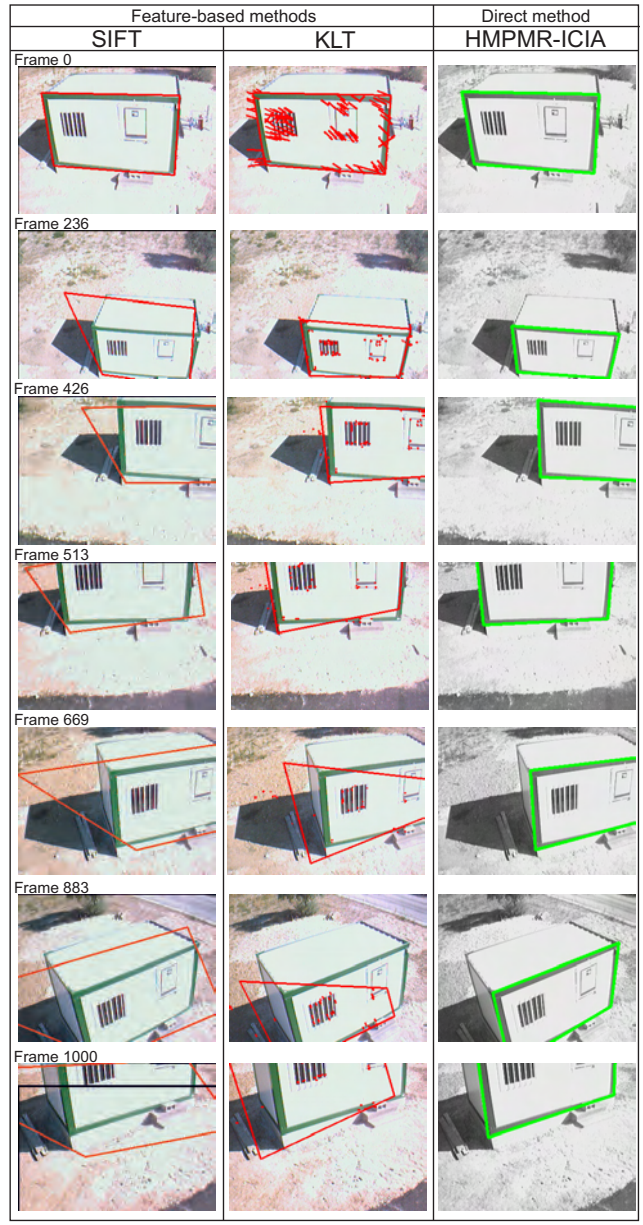

Fig. 10. Visual examination of the tracking results: SIFT, KLT, and HMPMR. The red polygons indicate the template estimated by the feature-based methods (SIFT, first column, and KLT, second column).The green polygon indicates the position of the template estimated by the direct method HMPMR-ICIA. As can be seen, the latter is the only one that tracks the template in all the sequence.

Fig. 10 shows a collection of images illustrating the performance of the tracking task and comparing the results obtained by the three algorithms: two feature-based methods (SIFT and KLT), and one based on direct methods the HMPMR-ICIA.

In this figure, it can be seen that the feature-based algorithms (Fig. 10, first and second columns) failed tracking the template almost at the same time. The multi-resolution structure of the KLT tracker is not enough to help the

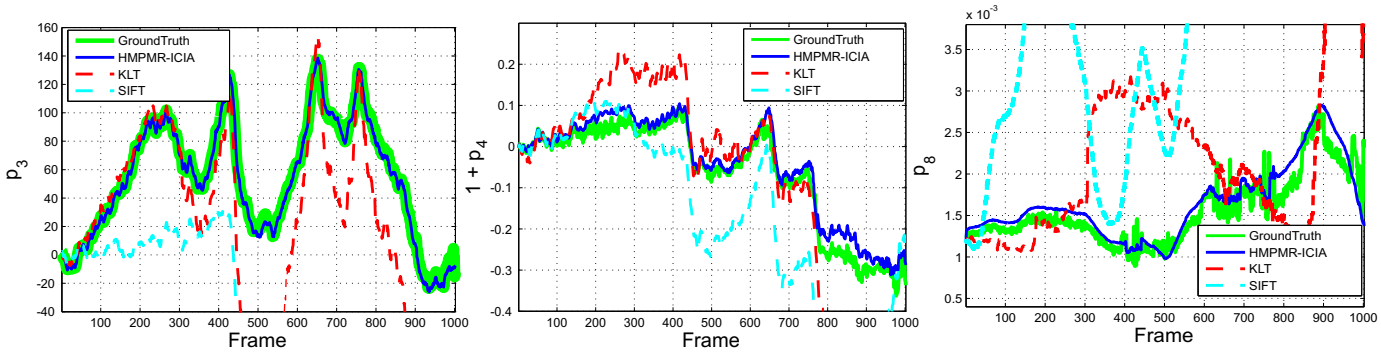

Fig. 11. Homography comparison with ground truth data. The parameters of the homography (p_3, p_4, and p_8, shown in (1)) estimated by each algorithm (HMPMR-ICIA, blue/dark solid line; KLT, red/dark dashed line; and SIFT, cyan/light dashed line) when tracking the front of the "house" are compared with the parameters of the ground truth homography (green/light solid line). Each graphic shows the comparison of each parameter shown in (1). The HMPMR-ICIA (blue/dark dark line) is the only algorithm in which the parameters show behavior and values that are similar to the ones of the ground truth homography (green/light solid line)

algorithm track this sequence. Nonetheless, in Fig. 10, third column, it can be seen that the HMPMR-ICIA with the 8-4-3-2 configuration tracked the template in all the frames of the sequence (third column).

Additionally, when comparing the parameters estimated by each algorithm with the ones of the GT homography, the same results are found. Fig. 11 shows the comparison among some parameters of the homographies found by the HMPMR-ICIA (blue/dark solid line), the KLT (red/dark dashed line), the SIFT (cyan/light dashed line), and the GT homography (green/light solid line).

In Fig. 11, it can be seen that the SIFT algorithm (cyan/light dashed line) fails earlier than the KLT algorithm (red/dark dashed line). However, in the figure, it can be seen that the KLT also fails in the first frames of the sequence. None of the parameters of the homographies recovered by the tested feature-based algorithms show a behavior similar to the parameters of the ground truth homography (green/light solid line).

On the other hand, comparing the parameters of the homography estimated by the HMPMR-ICIA (blue/dark solid line) with the ones of the ground truth homography (green/light solid line) in Fig. 11, it can be seen that the values of the parameters estimated by the HMPMR-ICIA (blue/dark solid line) have behavior and values that are similar to the ones of the ground truth data (green/light solid line).

The comparison of the parameters of the homography recovered by each algorithm shows that after a few frames the feature based algorithms failed detecting a correct transformation in spite of the different features that were found (an average of 80 features in the KLT, and 30 in the SIFT). Nonetheless, the low texture information of the object to track (the template), and the previously mentioned characteristics of this sequence, made the trackers fail.

Finally, Fig. 12 shows the average speed of the three algorithms, expressed in FPS (frames per second). As expected, the KLT feature-based algorithm (red box) tracks the template faster (average speed 27 FPS matching ≈ 85 features per frame) than all the other methods. This algorithm is

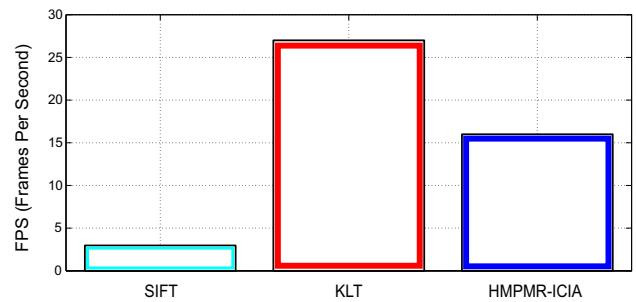

Fig. 12. Speed comparison. The average fame rate of the different tested algorithms is plotted. The SIFT (cyan box), 3 FPS; the KLT (red box), 27 FPS; and the HMPMR-ICIA (blue box),16 FPS.

widely used in different applications because of its efficiency, although we have shown that its performance is not robust enough in our application to track the template appropriately. The SIFT algorithm (Fig. 12, cyan box) has an average speed of 3 FPS, obtaining the slowest speed of the different tested algorithms (this is due to the high computational overheads in the different steps of the algorithm: e.g the calculation of the descriptor for each point, matching of points, etc).

On the other hand, we can see that in this test the direct method HMPMR-ICIA algorithm (Fig. 12, blue box) reaches an average speed of 16 FPS. This speed is fast enough to use the visual information for a vision-in-the-loop application.

It is also important to consider that the direct method analyzes each pixel of the template in each level of the pyramid (around 26000 pixels must be analyzed only in the highest resolution level). Despite the amount of information the algorithm analyzes, we can see that by using the MP and MR strategies at the same time a robust real-time tracking algorithm is obtained.

C. Test 3: visual estimation for take-off and landing maneuvers

The previous tests have shown that the proposed tracking strategy HMPMR-ICIA has been able to track objects from a UAV, recovering complex motion models with a performance

that is better than the one obtained with feature-based methods, and is able to track the object in the sequences that were used.

Another test is conducted: with it, we want to analyze the performance of the HMPMR-ICIA algorithm tracking a template for landing and take-off. We analyze its behavior under the visual conditions present in these applications: e.g. large frame-to-frame motions and rapid changes in scale.

Two experiments are conducted. In the first one, the vision-based position estimation explained in Section IV is tested experimentally in a laboratory facility using a Vicon system. In the second experiment, the information recovered by the HMPMR-ICIA algorithm is used for a vision-in-the-loop task, a landing task, based simultaneously on image-based and position-based visual servoing strategies.

1) **Position estimation**

As can be seen in Fig. 13, a scaled helipad is used as template (the object to track) for the experiment. A FireWire camera moves forwards and backwards simulating the take-off and landing processes (from the image point of view). This camera captures the image data used in the test. It captures images of size 1024×740 pixels at a frame rate of 7.5 FPS in order to generate image data with a large frame-to-frame motion.

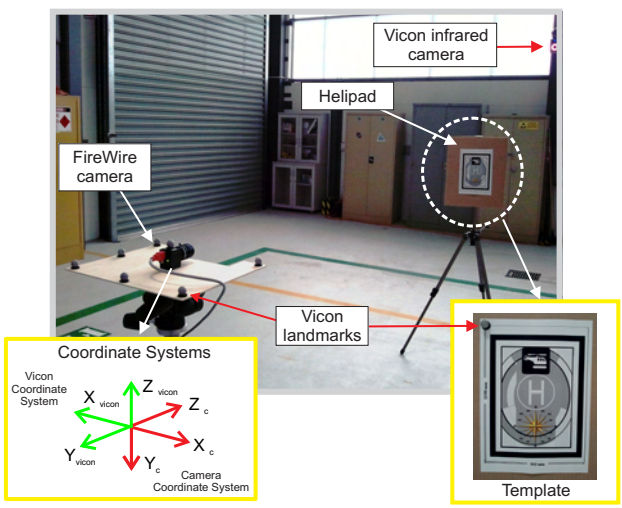

Fig. 13. Experimental setup. A FireWire camera that is moved manually simulates the UAV during take-off and landing tasks. This camera captures images of a scaled helipad. Ground truth data is generated using a Vicon system that tracks infrared landmarks located on the FireWire camera and the helipad.

The Vicon system [14], composed of five infrared cameras, is in charge of detecting the position and orientation of the template image (the helipad) and the FireWire camera, by detecting and tracking infrared landmarks (see Fig. 13). The system provides accurate 3D position information (with sub-milimeter and sub-degree precision) of the helipad and the FireWire camera with respect to the Vicon coordinate system shown in Fig. 13, at real-time frame rates (100 Hz). This information is used as ground truth data in order to analyze the visual estimation obtained with the position estimation algorithm described in Section IV.

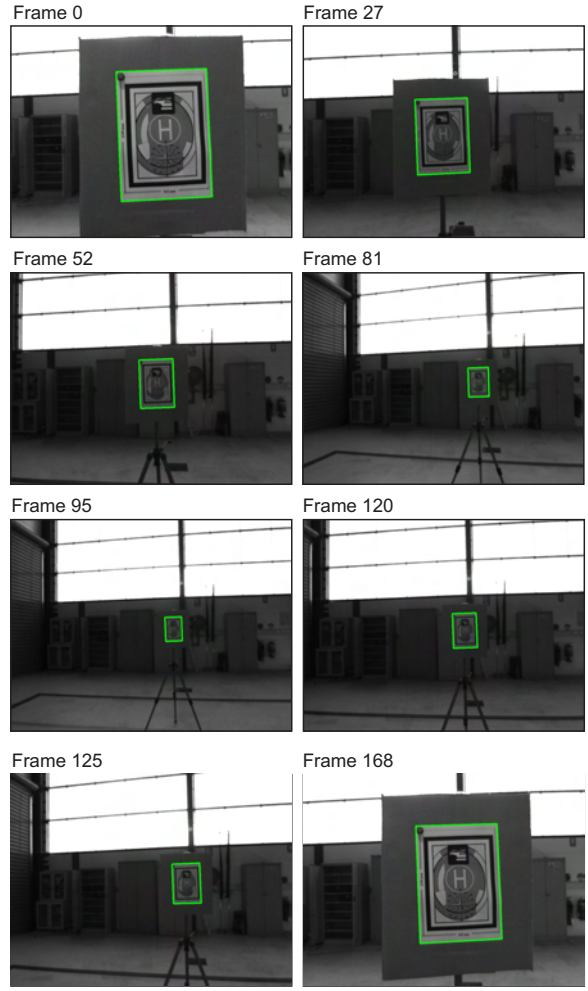

Fig. 14. Visual examination of the tracking results: HMPMR-ICIA. The green/light box indicates the estimated 2D position and extent of the helipad.

Fig. 14 presents a collection of images illustrating the performance of the tracking task. The green/light box indicates the results of the HMPMR-ICIA algorithm. The helipad was tracked during the entire task, in spite of the different changes in scale (e.g. see Fig. 14, frames: 0, 95, 168), the quality of the images (dark images), vibrations (the camera was moved manually), and the large frame-to-frame motion of the sequences (images were acquired at 7.5 FPS)

In order to compare the data, the 3D position of the FireWire camera is estimated using the method presented in Section IV, assuming that the camera is calibrated and that the dimension of the helipad is known. The vision-based positions are obtained with respect to the camera coordinate system, and then transformed to the Vicon coordinate system shown in Fig. 13.

Fig. 15 (upper left and right plots, and bottom left plot) shows the comparison of the position estimation obtained by the Vicon system (green/light line) with the position estimated using the homography recovered by the HMPMR-ICIA algorithm (red/dark line). As can be seen, the position estimated by the HMPMR-ICIA algorithm (red/dark line) shows a behavior

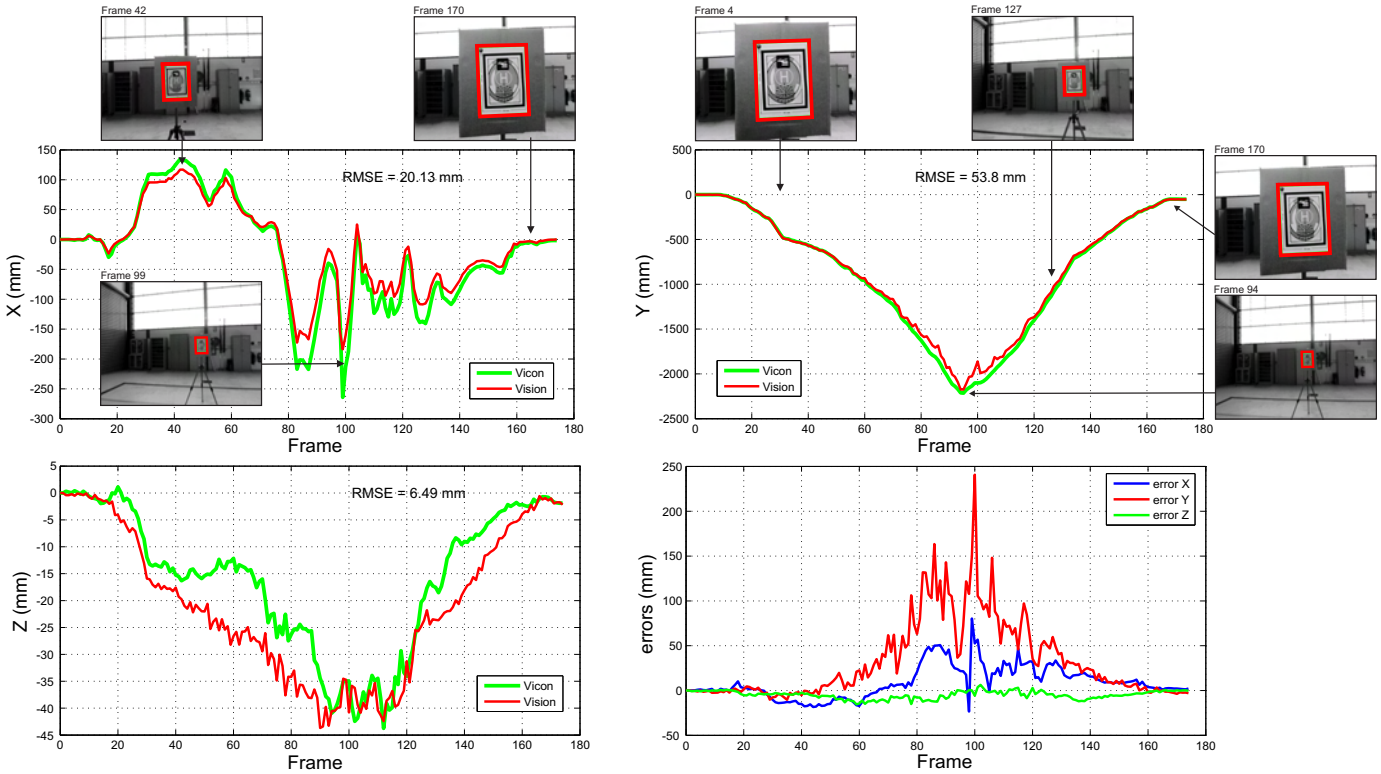

Fig. 15. Comparison with Ground Truth Data. The position of the FireWire camera estimated by the Vicon system (green/light line) is used as ground truth data and is compared with the position estimated using the homography recovered by the HMPMR-ICIA algorithm (red/dark line). The bottom-right plot shows the errors obtained in each axis. Both data are expressed with respect to the Vicon coordinate system.

that is similar to the position estimated by the Vicon system (green/light line). The RMSE (Root Mean Squared Errors) obtained in the three axes are < 6 cm.

The bottom-right plot of Fig. 15 shows the errors in each axis. We can see that during the whole sequence, the errors were always below 10 cm, and only in one point an error of 25 cm was obtained in the Y axis (the one that corresponds to the UAV height estimation). Nonetheless, these errors are low considering that errors in GPS-based position estimations are around 1 m under good conditions.

Thumbnail images in Fig. 15 show the correlation of the visual data with the estimated data. These images have been manually enhanced (compared with the real ones shown in Fig. 14) to allow a clear distinction of the template image and the result of the tracking algorithm.

From this test, we can see that the information obtained by the HMPMR-ICIA algorithm can be used for obtaining robust important information (position information) for vision-in-the-loop tasks.

2) **Vision-based landing**

In this second experiment, the 2D position in the image plane of the object to track obtained with the HMPMR-ICIA algorithm and the altitude of the UAV obtained with the position estimation algorithm presented in Section IV are used to send vision-based control commands to the UAV in order to perform a vision-based landing task.

The test is conducted using a virtual environment that uses the ROS (Robot Operating System) framework. In the test, an helipad is located on the ground of the simulation environment, a camera located on-board a quadrotor is used to capture images of the helipad, and the HMPMR-ICIA algorithm is used to find the 2D position of the helipad in the image plane (i.e. is used to track the helipad). The control task of this test consists, first, in placing the quadrotor over the helipad at a fixed altitude (10 m). In this first stage, we use image-based control commands in order to locate the helipad in the center of the image plane, i.e. in the coordinate $(320, 240)$, taking into account that the image is 640×480 pixels size.

When the helipad is close to the center, we use the vision-based altitude of the quadrotor, estimated using the method described in Section IV to send altitude commands to the flight controller in order to make the quadrotor descend to a defined position (1 m from the ground). In this last stage, both the control in the image plane and the altitude control operate simultaneously.

In Fig. 16, it is possible to see the trajectory of the quadrotor during the task. The blue line corresponds to the position data obtained by the Starmac-ros package [38]. In the figure, the positions to which the quadrotor was commanded to move to can be seen. It can be seen that first the quadrotor moves towards the helipad in order to locate the helipad in the center of the image plane (Setpoints: Sp $\mathrm{X_f} = 320$ pixels and Sp $\mathrm{Y_f} = 240$ pixels), and then starts to decend until reaching the altitude setpoint (Sp $\mathrm{Z} = 1$ m). When the quadrotor is

decending, it continues maintaining the helipad centered with respect to the image plane.

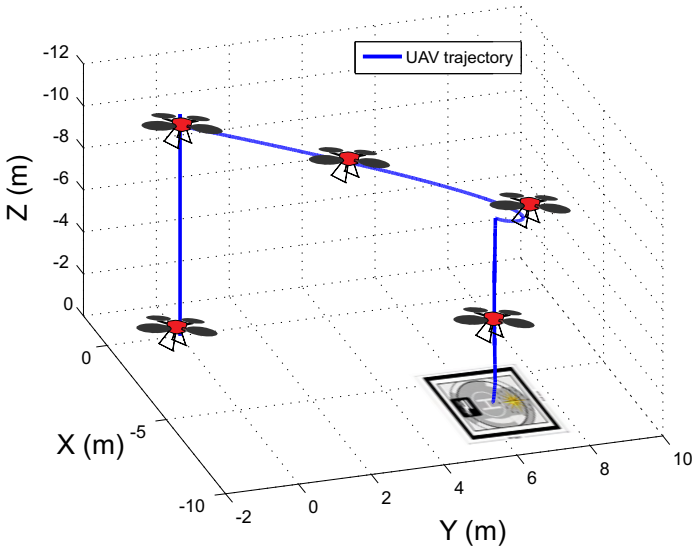

Fig. 16. 3D trajectory of the UAV during the positioning and landing tasks.

Fig. 17 shows the first stage of the landing approach. As mentioned previously, this stage is conducted based on the information recovered by the HMPMR-ICIA algorithm. In the graphic, it can be seen that the helipad is centered in the X_f and Y_f axes (i.e. in the center of the image plane) and remained centered during the task, i.e. the red/dark solid line reached the Sp $X_f = 320$ pixels, and the magenta/light solid line reached the Sp $Y_f = 240$ pixels (green/light dashed lines). The blue/dark dashed line is a control flag that indicates when the visual control in the X_f and Y_f axes (i.e. in the image plane) is operating. It can be seen that the image-based control was also operative during the descent process.

In the thumbnail images of Fig. 17, it can be seen how the helipad is centered in the first part of the task (frames 300-4871) and then remained centered when the quadrotor was descending (frames 4872-7338).

On the other hand, Fig. 18 shows the results of the position-based visual control task. When the helipad is centered (after frame 4871), a control flag is activated (blue/dark dashed line) and position-based control commands, based on the vision-based altitude estimation, are sent to the flight controller in order to make the quadrotor descent over the helipad (frames 4872-7338). In Fig. 18, it can be seen that the vision-based altitude estimations (red/dark solid line) have values and behavior that are similar to the altitude estimated by the flight controller of the quadrotor (cyan/light solid line). The thumbnail images show that the HMPMR-ICIA algorithm tracked the template throughout the task, and that when the quadrotor is descending the helipad remains centered in the image plane.

This test reveals that the visual information recovered by the HMPMR-ICIA algorithm can be used for an image-based and/or a position-based landing task.

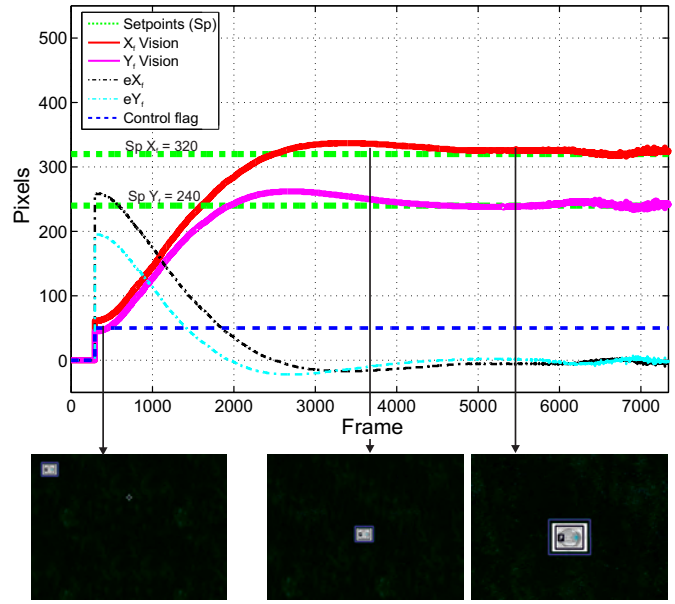

Fig. 17. Image-based control for landing. When the control flag is active (blue/dark dashed line), the 2D position of the helipad in the image plane (red/dark solid line and the magenta/light solid line) is used to send control commands to the quadrotor in order to center the helipad in the image plane (the green/light dashed lines represent the setpoints).

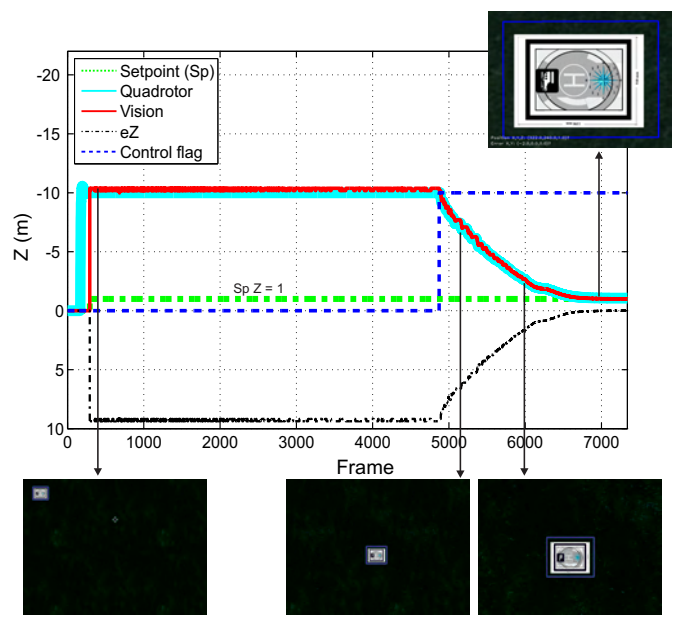

Fig. 18. Position-based control for landing. When the quadrotor is over the helipad, a control flag is activated (blue/dark dashed line), and the vision-based height estimations (red/dark solid line) are used to command the quadrotor to an altitude of 1 m (green/light dashed line) with respect to to the ground. The cyan/light solid line corresponds to the altitude estimated by the state estimator of the quadrotor.

VI. CONCLUSIONS AND FUTURE WORK

Our goal was to present a tracking strategy for tracking planar structures (or structures that can be assumed to be planar) on-board UAVs, that can deal with large frame-to-frame motions, that can recover complex motion models (e.g.

the homography), that can obtain real-time frame rates, and that recovers information that can be used for different vision-based applications on-board UAVs (e.g. building inspection, landing, take-off).

In this paper, we have presented a hierarchical tracking algorithm HMPMR-ICIA for tracking on-board UAVs using direct methods, thus extending the use of direct methods for real-time applications. Previous works in this area have often been based on feature methods. Nonetheless, we have shown that our tracking strategy performs better than well-known feature-based algorithms (SIFT and KLT) and well-known configurations of direct methods (MR-ICIA), in the presence of strong changes in position, fast changes in appearance, in situations where part of the template is out the FOV of the camera, and under constant vibrations. Concerning the latter aspect, this is accomplished without requiring any specific hardware or software for video stabilization.

Different evaluation mechanisms were used to analyze the performance of the HMPMR-ICIA algorithm: images from real-flights, manually generated ground truth data, accurate position estimation using the Vicon system, and a simulation environment for a vision-based landing task were used.

The results show a good performance of the algorithm tracking planar structures affected by perspective effects, and also show a good correlation of position data estimated using the information obtained by the visual tracking algorithm, that validates the proposed strategy and makes it useful to provide valid vision-based data for UAV applications, as was demonstrated in the landing test.

Due to the amount of information that direct methods have to evaluate, these kinds of methods are not commonly used for real-time applications. Nonetheless, we have shown that by using the proposed strategy and without optimizing the code in any way, direct methods can be employed for real-time tracking, and are able to achieve frequencies of 16 fps when estimating 8 parameters. It is important to notice that the speed is highly dependent on the number of parameters estimated (faster responses -30 and 50 fps- are achieved when estimating motion models with a lower number of parameters). Additionally, the speed of the algorithm is dependent on the size of the template and the parameters estimated in each level of the pyramid.

Taking this into account, future work will focus on creating criteria to control the performance of the alignment task inside the MR structure, in order to create a dynamic strategy that decides which levels of the MR and MP structure are evaluated. This dynamic strategy can help speeding up the algorithm.

Finally, considering the inherent appearance changes in our application when conducting outdoors operations (illumination changes), and taking into account the UAV 3D movements, future work will focus on establishing criteria in aspects as the the update of the template and outliers rejection in order to deal with possible drift problems that may emerge due to the propagation of the parameters throughout the image sequence, especially when the template's appearance notoriously changes in the sequence.

VII. Acknowledgments

The work reported in this paper is the result of several research stages conducted at the Computer Vision Group of the Universidad Politécnica de Madrid. The authors would like to thank the Universidad Politécnica de Madrid, the Consejería de Educación de la Comunidad de Madrid, and the Fondo Social Europeo (FSE) for the Ph.D. scholarships of some of the authors. The authors would also like to thank the Australian Research Centre for Aerospace Automation (ARCAA) for allowing us to collect data from the Motion Caption System (Vicon). This work has been supported by the Spanish Ministry of Science under grant MICYT DPI2010-20751-C02-01.

References

[1] Pascual Campoy, Juan F. Correa, Ivan Mondragón, Carol Martínez, Miguel Olivares, Luis Mejías, and Jorge Artieda. Computer vision onboard uavs for civilian tasks. *Journal Intelligent and Robotic Systems*, 54(1-3):105–135, 2009.

[2] Ivan F. Mondragón, Pascual Campoy, Carol Martinez, and Miguel. Olivares-Mendez. 3D pose estimation based on planar object tracking for UAVs control. In *Proceedings of IEEE International Conference on Robotics and Automation 2010 ICRA2010*, Anchorage, AK, USA, April 2010.

[3] I.F. Mondragon, P. Campoy, J.F. Correa, and L. Mejias. Visual model feature tracking for uav control. In *Intelligent Signal Processing, 2007. WISP 2007. IEEE International Symposium on*, pages 1 –6, 3-5 2007.

[4] Philip H. S. Torr and Andrew Zisserman. Feature based methods for structure and motion estimation. In *ICCV '99: Proceedings of the International Workshop on Vision Algorithms*, pages 278–294, London, UK, 2000. Springer-Verlag.

[5] Jean Y. Bouguet. Pyramidal implementation of the Lucas Kanade feature tracker: description of the algorithm. Technical report, OpenCV Document, Intel Microprocessor Research Labs, 2002.

[6] M. Irani and P. Anandan. About direct methods. In Bill Triggs, Andrew Zisserman, and Richard Szeliski, editors, *Vision Algorithms: Theory and Practice*, volume 1883 of *Lecture Notes in Computer Science*, pages 267–277. Springer Berlin / Heidelberg, 2000.

[7] Hong Zhang and Fei Yuan. Vehicle tracking based on image alignment in aerial videos. In *EMMCVPR'07: Proceedings of the 6th international conference on Energy minimization methods in computer vision and pattern recognition*, pages 295–302, Berlin, Heidelberg, 2007. Springer-Verlag.

[8] Saad Ali and Mubarak Shah. Cocoa - tracking in aerial imagery. In *Proc. Int. Conf. on Computer Vision*, 2005.

[9] Luis Mejias, Srikanth Saripalli, Pascual Campoy, and Gaurav Sukhatme. Visual servoing approach for tracking features in urban areas using an autonomous helicopter. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 2503–2508, Orlando, Florida, May 2006.

[10] Jay Hyuk Choi, Dongjin Lee, and Hyochoong Bang. Tracking an unknown moving target from uav: Extracting and localizing an moving target with vision sensor based on optical flow. In *Automation, Robotics and Applications (ICARA), 2011 5th International Conference on*, pages 384 –389, dec. 2011.

[11] A. Dame and E. Marchand. Accurate real-time tracking using mutual information. In *IEEE Int. Symp. on Mixed and Augmented Reality, ISMAR'10*, pages 47–56, Seoul, Korea, October 2010.

[12] Simon Baker and Iain Matthews. Equivalence and efficiency of image alignment algorithms. In *Proceedings of the 2001 IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 1090 – 1097, December 2001.

[13] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[14] Vicon Motion Systems. VICON MX digital optical motion capture system. http://www.vicon.com, 2011.

[15] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2003.

[16] C. Martinez, L. Mejias, and P. Campoy. A multi-resolution image alignment technique based on direct methods for pose estimation of aerial vehicles. In *Proceedings of the International Conference on Digital Image Computing Techniques and Applications (DICTA), 2011*, pages 542 –548, dec. 2011.

[17] C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden. Pyramid methods in image processing. RCA Engineer, vol. 29, no. 6, pp. 33-41, Nov./Dec., 1984.

[18] P. Burt and E. Adelson. The laplacian pyramid as a compact image code. *Communications, IEEE Transactions on*, 31(4):532 – 540, April 1983.

[19] James R. Bergen, P. Anandan, Th J. Hanna, and Rajesh Hingorani. Hierarchical model-based motion estimation. In *Proceedings of the European Conference on Computer Vision, pp. 237252.*, pages 237–252, 1992.

[20] Richard Szeliski. Image alignment and stitching: a tutorial. *Found. Trends. Comput. Graph. Vis.*, 2(1):1–104, 2006.

[21] Jose Miguel Buenaposada and Luis Baumela. Real-time tracking and estimation of plane pose. In *In ICPR*, pages 697–700, 2002.

[22] S. Benhimane and E. Malis. Real-time image-based tracking of planes using efficient second-order minimization. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, 2004.

[23] Michal Irani and Shmuel Peleg. Motion analysis for image enhancement: Resolution, occlusion, and transparency. *Journal of Visual Communication and Image Representation*, 4:324–335, 1993.

[24] H.-Y Shum and R. Szeliski. Panoramic vision. chapter Construction of panoramic image mosaics with global and local alignment, pages 227–268. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001.

[25] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(1):221 – 255, March 2004.

[26] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[27] G.D. Hager and P.N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(10):1025–1039, October 1998.

[28] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly, 2008.

[29] Gilles Simon and Marie-Odile Berger. Pose estimation for planar structures. *IEEE Comput. Graph. Appl.*, 22(6):46–53, November 2002.

[30] J.M. Buenaposada and L. Baumela. Real-time tracking and estimation of plane pose. In *Proceedings 16th International Conference on Pattern Recognition, 2002*, volume 2, pages 697 – 700 vol.2, 2002.

[31] Zhengyou Zhang. Flexible camera calibration by viewing a plane from unknown orientations. *Computer Vision, IEEE International Conference on*, 1:666, 1999.

[32] J. Y. Bouguet. Camera calibration toolbox for Matlab, 2008.

[33] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(11):1330–1334, November 2000.

[34] Videos. http://www.vision4uav.com/?q=HMPMRtracker.

[35] Computer Vision Group. Universidad Politécnica de Madrid. CVG 2010. http://www.vision4uav.com/.

[36] Morgan Quigley, Ken Conley, Brian P. Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. ROS: an open-source Robot Operating System. In *ICRA Workshop on Open Source Software*, 2009.

[37] ROS-Gazebo. http://www.ros.org/wiki/simulator_gazebo, 2012.

[38] Starmac-ros package. http://www.ros.org/wiki/starmac-ros-pkg.

[39] R. Hess. SIFT feature detector implementation in C. http://www.web.engr.oregonstate.edu/h̃ess/index.html, Feb. 2007.

[40] Rob Hess. An open-source SIFT library. In *Proceedings of the international Conference on Multimedia*, pages 1493–1496, 2010.