3D model-based tracking for UAV indoor localisation

Céline Teulière, Eric Marchand, Laurent Eck

3D model-based tracking for UAV indoor localisation

Céline Teulière, Eric Marchand, Laurent Eck

***Abstract*—This paper proposes a novel model-based tracking approach for 3D localisation. One main difficulty of standard model-based approach lies in the presence of low-level ambiguities between different edges. In this work, given a 3D model of the edges of the environment, we derive a multiple hypotheses tracker which retrieves the potential poses of the camera from the observations in the image. We also show how these candidate poses can be integrated into a particle filtering framework to guide the particle set toward the peaks of the distribution. Motivated by the UAV indoor localisation problem where GPS signal is not available, we validate the algorithm on real image sequences from UAV flights.**

***Index Terms*—model-based tracking, particle filtering, UAV application**

I. Introduction

A. Overview

The past fifteen years have seen a major growth of interest in unmanned aerial vehicles (UAV) [32]. UAV have strong potential applications such as surveillance, search and rescue, or inspection for maintenance. For navigation, existing systems mainly rely on the fusion of GPS and inertial measurements. However, such approaches are usually unsuitable for urban or indoor environments, where GPS is not available or imprecise. For this reason, recent works propose to make use of the visual information provided by one or two cameras [11] [6] [17] for UAV indoor localisation and navigation. Besides being passive, light and cheap sensors, cameras are usually necessary to provide a visual feedback in a surveillance or inspection task, and they provide a rich source of information on the environment. However, using vision for UAV control is also particularly challenging since strong constraints have to be taken into account: during a UAV flight, image quality can be poor (see Figure 2) with substantial noise and motion blur, and large motion disturbances occur, especially with small platforms like those usually considered for indoor missions.

To ensure safe vision-based navigation and control tasks, the robustness of the extraction of visual information is crucial. In particular, the choice of the visual features and their ability to be robustly matched and tracked over time will determine the good achievement of the task. Feature points, which can be extracted and tracked in any textured environment, have been used in various aerial applications, from structure from motion or SLAM [21] [1] to optical flow computation [11]. However, in indoor structured environments with plain floors and walls, feature points are less frequent, leading to robustness issues. They are also sensitive to the typical noise produced by transmission interferences (see Figure 2).

Among the different vision-based localisation approaches, SLAM techniques are particularly interesting in unknown cluttered environments, but they are prone to suffer from those issues related to the use of feature points, and they do not provide an absolute localisation. In this paper, we propose a robust model-based tracking approach to estimate the 3D pose of the UAV through its motion. The vision system thus requires a 3D model of the edges of the environment. Although this condition is not suitable for unstructured outdoor missions, it seems reasonable in the context of indoor inspection tasks, for the maintenance of modelled industrial sites for example. Moreover, edges are very frequent in such structured environments, they offer a good degree of invariance to pose and illumination changes and are easy to detect even in presence of noise or blur. Edges have also been already used in the context of outdoor UAV applications, in particular to track linear structures such as pipes, or landing runway [26] [20] [3] [24]. In such scenes the linear structures are generally distinct in the environment. In a structured environment however, one difficulty of using edges is that they suffer from having very similar appearances. Therefore, some ambiguities can occur typically when different edges get close to each other, which can lead to wrong matches and tracking failures.

To address this issue and ensure the robustness of the pose estimation, our 3D visual tracking approach takes into account multiple hypotheses. Our first contribution is to propose a novel method to retrieve multiple pose hypotheses from image measurements, using a clustering algorithm (Section II). These hypotheses, along with an associated confidence measure, provide a simple approximation of the probability density function. From this, we show that choosing the "best" hypothesis provides us with a tracking system with better performances than single hypothesis registration methods as described in [7], [5]. Second, we derive from this an enhanced approach by combining top-down and bottom-up methods, in a particle filter framework. The hypotheses that result from the image are used to guide the particle set towards regions of interest in the space of 3D poses, allowing to consider fewer particles (Section III). We provide simulation results as well as validation on real sequences from UAV flights. To demonstrate the suitability of such a vision system in a UAV navigation task, we finally provide experimental results on a quad-rotor

C. Teulière was with CEA, LIST, 18 route du Panorama, Fontenay-aux-Roses, F- 92265, France. She is now with the Blaise Pascal University, Pascal Institute, Clermont-Ferrand, France. e-mail: celine.teuliere@univ-bpclermont.fr

E. Marchand is with Université de Rennes 1, IRISA, Inria, Lagadic Project, Rennes, France. email: marchand@irisa.fr

L. Eck is with CEA, LIST, Sensorial and Ambient Interfaces Laboratory, 91191 - GIF-sur-Yvette CEDEX, France. email: laurent.eck@cea.fr

This work was realized in the context of the French ANR national project SCUAV (ANR Psirob 06_174032 SCUAV project ref ANR-06-ROBO-0007-02)

aerial vehicle (Section IV). This paper extends our shortest presentations of our tracker in [30] with additional details and experiments.

B. Related work

Our work is mainly related to the literature 3D pose estimation in computer vision.

The problem of estimating the pose of a moving camera with respect to its modelled environment in real-time has been widely investigated in the past years (see [18] for a survey) and different approaches have been proposed to address it. Most of these approaches can be divided into two categories:

- *Registration methods* use non linear optimisation techniques (Newton minimisation, virtual-visual servoing,...) to find the pose which minimises a given reprojection error between the model and the image edges [19], [7], [5]. The robustness of these methods has been improved by using robust estimation tools [2], [7], [5]. In the UAV context [15] applied model-based tracking to UAV navigation. On main difficulty of such approaches alone is that they can fail in case of large displacements or wrong edge matching, especially in cluttered environment.
- *Bayesian methods*, on the other hand, have been used to perform the same task by estimating the probability density associated to the pose. This can be achieved by Kalman filtering when the probability density function (p.d.f.) can be represented by an uni-modal Gaussian distribution. More recently, the improvement of computational performances has allowed to consider particle filtering approaches [25], [16], [23]. Instead of going from the low level edges to retrieve the camera pose, particle filtering uses a set of hypotheses on the possible camera poses (the particles). The likelihood of each particle is then measured in the image. Since the space of all possible poses is large, the main issue is to keep a fair representation of the different modes of the state probability distribution while using few particles.

To overcome the edge matching issue of registration-based methods, [31] proposed to include multiple low level hypotheses in the robust registration method, showing improved performances. However it still maintain a unique hypothesis on the camera pose. Also motivated by the robustness required for UAV application, [14] proposed to use RANSAC to maintain a multi-modal representation of the posterior density in a particle-filter inspired way. Our tracking approach differs from [14] on two main points:

- First, we propose a novel method for generating multiple edge hypotheses, based on *k-mean* clustering principle (see Section II).
- Second, while in [14] all the pose hypotheses are derived from the image, in our approach we propose to combine hypotheses generated at the pose level, as propagated through a classic condensation scheme [13], with the multiple hypotheses generated thanks to our *k-mean* based registration process (see Section III).

The next two sections describe our vision-based approach in details.

II. Multiple hypotheses registration

Our multiple hypotheses registration algorithm relies on a similar basis as the ones used in [5], [7] and [31]. Assuming the camera parameters and an estimate of the pose are known, the 3D model is first projected into the image according to that pose, which can be the previous one or a prediction obtained from a filter.

Formally, the projection of an edge L_i of the 3D model according to the pose ${}^c\mathbf{M}_w$ [1] will be denoted by $E_i = L_i({}^c\mathbf{M}_w)$. Each projected edge E_i is sampled, giving a set of points $\{\mathbf{e}_{i,j}\}$ (see Figure 1). From each sample point $\mathbf{e}_{i,j}$ a search is performed along the edge normal to find strong gradients.

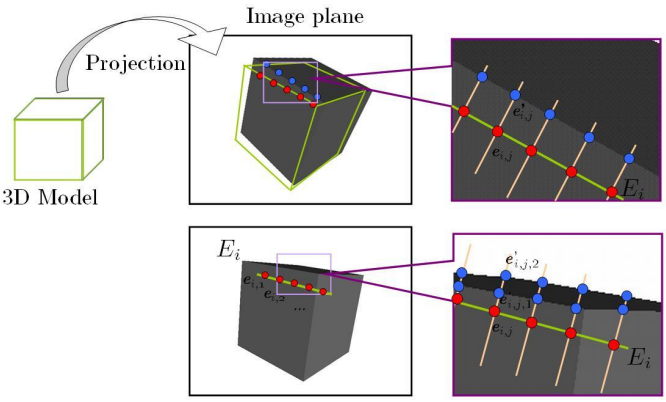


Fig. 1. In classic edge based tracking, the model is projected into the image plane and points are sampled on the projected edges. A search is performed along the normal (top). When multiple strong edges are close in the image, the exploration of the normal can lead to ambiguities (bottom).

As illustrated in Figure 1, two close edges can lead to matching ambiguities. In [5] the point of maximum likelihood with regard to the initial point $\mathbf{e}_{i,j}$ is selected from the exploration step. It is denoted by $\mathbf{e}'_{i,j}$ in the following. A non linear optimisation approach is then used to estimate the camera pose which minimises the errors between the selected points and the projected edges [19], [5], [7], that is:

$$\widehat{{}^c\mathbf{M}_w} = arg\min_{{}^c\mathbf{M}_w} \sum_{i,j} d_\perp(E_i, \mathbf{e}'_{i,j}) \tag{1}$$

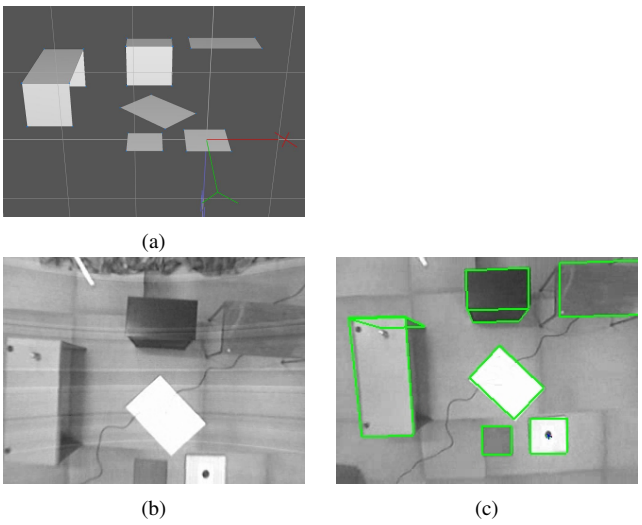
where $d_\perp(E_i, \mathbf{e}'_{i,j}) = d_\perp(L_i({}^c\mathbf{M}_w), \mathbf{e}'_{i,j})$ is the squared distance between the point $\mathbf{e}'_{i,j}$ and the projection E_i of the linear segment L_i of the model. The quantity to minimise is then expressed by:

$$S = \frac{1}{N_e} \sum_i \sum_j \rho\left(d_\perp(E_i, \mathbf{e}'_{i,j})\right) \tag{2}$$

[1] ${}^c\mathbf{M}_w \in SE(3)$ is an homogeneous matrix that gives the position of the camera in the scene frame. It is given by

$${}^c\mathbf{M}_w = \begin{bmatrix} {}^c\mathbf{R}_w & {}^c\mathbf{t}_w \\ 0 & 1 \end{bmatrix}$$

where ${}^c\mathbf{R}_w \in SO(3)$ is a rotation matrix and ${}^c\mathbf{t}_w \in \mathbb{R}^3$ is a translation vector.



(a) (b) (c)

Fig. 2. (a) Example of 3D model, (b) corresponding scene (undistorted) and (c) tracking result.

where N_e is the total number of sampled points, and ρ is a robust estimator.

To be more robust to ambiguous cases, our registration method considers multiple low level hypotheses $\{\mathbf{e}'_{i,j,l}\}$ corresponding to local extrema of the image gradient along the edge normal in $\mathbf{e}_{i,j}$. Instead of performing one single minimisation from these points as in [31] resulting in one single pose, we go from these multiple low level hypotheses to multiple hypotheses on the camera pose itself. This process is described in the next section.

A. Determining the underlying edges

In order to retrieve multiple hypotheses for the camera pose from the detected low level hypotheses, we first determine the underlying lines from the set of points $\{\mathbf{e}'_{i,j,l}\}$. The idea is to use the knowledge we have about the linear components of the model, to assign each detected point to a potential edge and to associate a confidence criterion to these candidate edges (see Figure 3). In [14] this was achieved via a RANSAC approach. Here, we express this as a typical classification problem, and propose to tackle it using a *k-mean* clustering algorithm [10].

In our case the clustering algorithm is slightly modified to group the candidate points into edge hypotheses. For each projected edge E_i, the algorithm segments the candidate points $\{\mathbf{e}'_{i,j,l}\}$ into k_i sets of points or classes $(\mathcal{C}^i_1, ..., \mathcal{C}^i_{k_i})$. The mean of each of the k_i classes corresponds in our case to the line obtained by a least square minimisation on the points of that class. To initialise the algorithm, the number k_i of classes for the edge E_i is set to the maximum number of candidate points detected, that is: $k_i = max_j\{n_{i,j}\}$. The classes $(\mathcal{C}^i_1, ..., \mathcal{C}^i_{k_i})$ are initialised using the order in which the hypotheses have been found on the normal. That is, for each class $\mathcal{C}^i_m$: $\mathcal{C}^i_m = \{\mathbf{e}'_{i,j,m}\}_j$. This initialisation is often close to the correct segmentation, allowing the algorithm to converge faster (see Figure 3 (a)). At each iteration of the algorithm, the "mean line" of each class is computed (Figure 3 (b)). Each point is then assigned to the class with the nearest mean line.

Since the potential edges are not supposed to be normal to the initial edge, we add the constraint that two hypotheses $\mathbf{e}'_{i,j,l_1}$ and $\mathbf{e}'_{i,j,l_2}$ of a same initial sample point $\mathbf{e}_{i,j}$ cannot belong to the same class. The process is summarized in the algorithm 1.

In *k-mean* algorithms the local convergence is ensured by the finite number of classes and the decrease of the cost function at each iteration. Other ending criteria such as a maximum number of iterations and/or a threshold on the cost function decrease rate are often added to set a maximum computational time. In our case, we do not have an analytical proof of convergence and base the choice of this clustering component on experimental performances we observe. The algorithm is deemed to have converged when the assignments no longer change or the iteration number reaches a given threshold which gives an upper limit in terms of computation time (this threshold was set to 30 in our experiments).

***k-mean* algorithm for edge clustering.**
Let $\{\mathbf{e}'_{i,j,l}\}$ be the candidate points corresponding to the projected edge E_i and $k_i = max_j\{n_{i,j}\}$ the maximum number of candidates detected for a sample point. The algorithm is initialised by respecting the order of detection of the candidates: $\mathcal{C}^i_l = \{e'_{i,j,l}\}_j$. Then the process is iteratively run as follows:

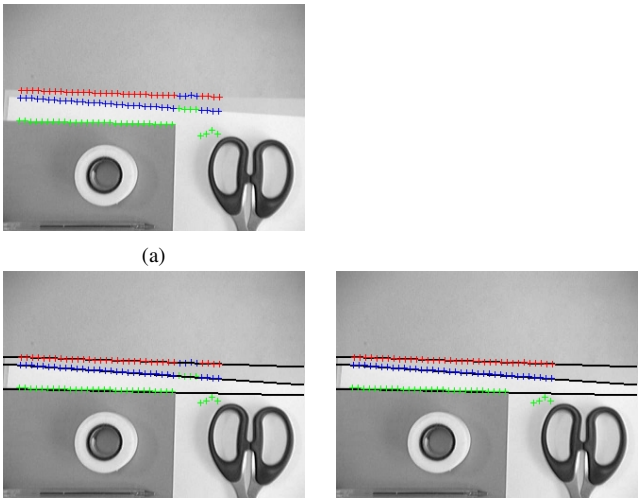
1. For each class $\mathcal{C}^i_l \in \{\mathcal{C}^i_1, \ldots, \mathcal{C}^i_{k_i}\}$ compute the interpolation line of the points in the class. Let r^i_l denote the residual of the interpolation.
2. For each $i,\ j$:
   - compute the distance from the points $\mathbf{e}'_{i,j,l}$ to each computed line,
   - the points $\mathbf{e}'_{i,j,l}$ (for $i,\ j$ fixed) are then associated to different classes, with higher priority for the points closer to the closest line.

The algorithm runs until the assignments no longer change in step 2 or the iteration number exceeds the maximum iteration number.

Algorithm 1: *k-mean* algorithm for edge clustering.

Finally, the *k-mean* algorithm corresponding to the initial edge E_i provides us with a set of classes $\mathcal{C}^i_m = (\{\mathbf{e}'_{i,j,m}\}_j, r^i_m)$ where r^i_m is the residual of the least square minimisation. This residual gives a measurement of the confidence we have in the corresponding candidate edge. Note also that only the lines with a sufficient number of points are taken into account. This allows to avoid cases where 2 or 3 well aligned points would then form a candidate edge considered as very likely without this class being meaningful. We thus eliminate classes with less than five points. Figure 3 shows a simple example of the process. Although the contours considered have been restricted to lines in this study, the approach can be easily adapted to other kinds of contours.

In most cases, the number of classes k_i does not exceed two or three. Figure 5 gives an example of the lines detected from the teabox sequence.



(a) (b) (c)

Fig. 3. An example of the *k-mean* computation, with $k = 3$. Each class is represented by a different color. (a) Initialisation of the classes of points. (b) Mean lines computation. (c) The final segmentation is obtained in one step.

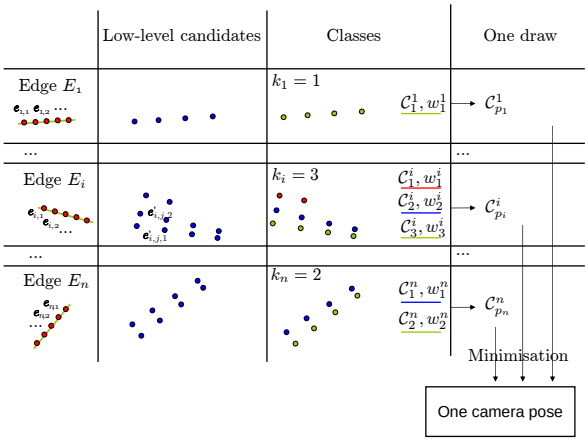


Fig. 4. From low level hypotheses, classes of points are extracted. For each projected edge a random weighted draw is performed among the classes to determine the points that will be used for the minimisation process. The minimisation provides an hypothesis on the camera pose. A different draw would lead to another candidate pose.

B. From edge hypotheses to pose hypotheses

Once candidates have been obtained for each edge in the form of sets of points associated to a residual, random weighted draws are performed. We compute weights w_m^i for each candidate as a function of the residuals. Several choices for computing the weights are possible and we use the following expression that gives satisfying performances:

$$w_m^i = \begin{cases} e^{-\lambda\left(\frac{r_m^i - r_{min}^i}{r_{max}^i - r_{min}^i}\right)^2} & \text{if } r_{max}^i \neq r_{min}^i \\ 1 & \text{otherwise.} \end{cases} \quad (3)$$

where λ is a parameter that can be tuned according to the selectivity that is desired. A value of $\lambda = 1$ has been used in our experiments.

One weighted draw will denote here the draw of one candidate per edge, that is, for each edge E_i a class $\mathcal{C}_{p_i}^i$ is drawn from the k_i classes. From each draw, a numerical non-linear minimisation is performed according to (4), using the set of points corresponding to the picked classes, resulting in a camera pose. We use the virtual visual servoing minimization process [5] but any other non-linear minimisation method could be used for this step.

$$S = \frac{1}{N_e} \sum_i \sum_{\mathbf{e}'_{i,j,l} \in \mathcal{C}_{p_i}^i} \rho\left(\Delta_{E_i}(\mathbf{e}'_{i,j,l})\right) \quad (4)$$

Since the optimisation is deterministic, it is only computed when the sets of candidates are different. The weighted draw allows to favour, among all the possible combinations, the ones with the candidates of lowest residual, which are more likely to correspond to a real edge. Several hypotheses on the camera pose are thus obtained from the low level detected hypotheses. The process is summarized in Figure 4. In practice, since the number of candidate lines per edge is small, so will be the number of optimisations to be performed and thus the number of pose candidates obtained. In our experiments we set the number of optimisations (and thus pose hypotheses) to 3.

C. Experiments

To illustrate the benefits of this approach in terms of robustness, Figure 5 shows an example where the multiple hypotheses approach allows to avoid failure. At this particular frame, extracted from a video sequence, a single hypotheses tracker fails. While running the multiple hypotheses algorithm, it appears (see Figure 5) that only one candidate is found for almost each projected edge, except the top back one. For this edge, two candidates have been found, which lead to two different camera poses. Whereas the single hypothesis tracker fails due to a wrong match, the multiple hypotheses tracker finds the correct pose (Figure 5-(2-b)).

To validate the proposed approach, we also used a simulated sequence, for which the ground truth is known. The comparative results between the classic registration method and our multiple hypotheses method are shown in Figure 6.

In the single hypothesis case, only the maximum likelihood point is selected in the search along the normal. The tracking fails when confronted to ambiguities, that is especially when two edges get close to each other, or when a new face appears (Figure 6-(a) and (c)).

In the multiple hypotheses case, the output considered at this stage is the camera pose which gives the lowest residual in the minimisation process. The object is successfully tracked even in cases of ambiguities. However, at some ambiguous frames where two candidates give almost the same residual, the tracker selects the wrong one as the best, which results in some jitter on the camera trajectory (Figure 6 (d)). In the same way, the minimisations which lead to Figure 5 (1-b) and (2-b) correspond to minima giving almost the same residuals. By selecting only the "best" one, some information given by other candidates can be lost. Moreover the tracker still needs frame to frame motion to be small to converge and could benefit from a prediction.

The next section presents an enhanced algorithm of our pose estimation tracker that deals with these issues and ensures

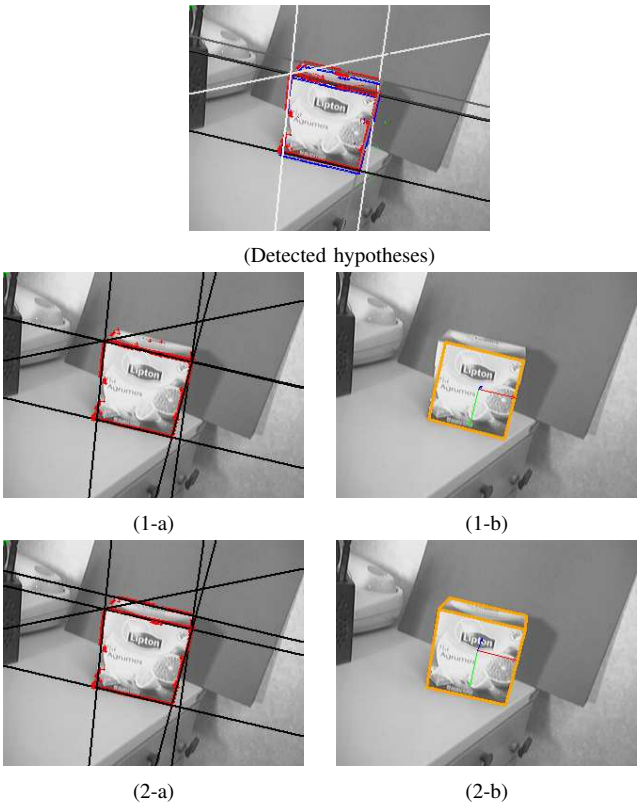


Fig. 5. Example of multiple hypotheses ambiguity. On the top frame, all the candidate lines and their corresponding points have been represented. The gray level of the line corresponds to its likelihood. When tracking the top-back edge, two hypotheses have been found, one of them corresponding to the top-front edge. (1-a) and (2-a) show two different draws from the initial set of candidates, resulting in two different camera poses (1-b) and (2-b). In the first draw, the back edge has been mixed up with the front one, leading to a tracking failure. Using multiple hypotheses allows to be more robust to such situations.

temporal coherence of our pose estimation.

III. Oriented particle filter

To tackle the above issues, we consider particle filtering. Particle filtering offers a very interesting framework in that it provides both temporal coherence through the filtering process and multi-hypotheses handling through the particle representation. More specifically the main interests of the particle filtering framework at this point are:

- the temporal filtering which can incorporate easily some prior knowledge on the UAV motion and prevents the system to fail in case of large occlusion.
- the possibility to handle multi-modal representations, which means concretely that, if 2 poses have similar residual, the particle filter will keep particles corresponding to both hypotheses, while the optimisation-based approach presented in the Section II will detect several, choose one, and forget about the others.

The remainder of this section presents how we combined the optimisation process presented above with the particle filtering framework.

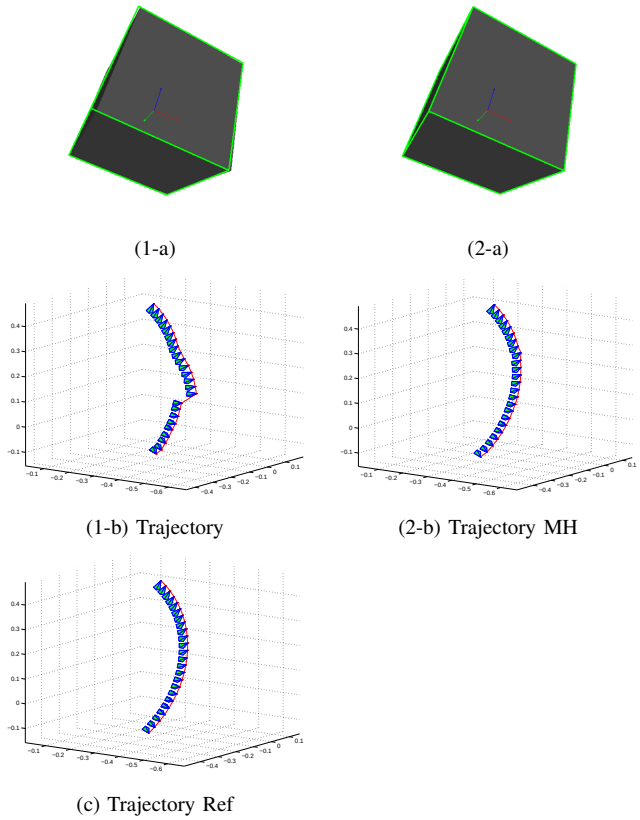


Fig. 6. Comparative results on a simulated sequence with auto-occlusion. The basic algorithm with single hypothesis (1-a), (1-b) fails when a new face appears. While considering multiple hypotheses (2-a), (2-b), the object is successfully tracked. The ground truth for the camera pose is shown in (c).

A. Overview

As mentioned in the introduction, particle filtering approaches [13] have been recently introduced in model-based tracking as an alternative to numerical optimisation methods, showing promising performances [25], [16], [23].

Our approach relies on the same basis as these works, that is the classic CONDENSATION algorithm [12], which represent the probability density function $p(\mathbf{x}_k \mid \mathbf{z}_{1:k})$ of the state $\mathbf{x}_k$ which in our case will be the pose at frame k, by a finite set $\{(s_k^{(i)}, \pi_k^{(i)})\}_{i=1..N}$ of N samples, or particles, $s_k^{(i)}$ associated with the weights $\pi_k^{(i)}$. Each particle $s_k^{(i)}$ represents a potential camera pose and $\mathbf{z}_{1:k}$ are the observations until frame k, deduced from image measurements. For each new frame, the particles first evolve according to a given dynamic model. Then, the likelihood of every particle is measured in the image and a weight is derived. The output considered is usually the weighted mean of the resulting set of particles. The particle set is updated by performing a random weighted draw among the particles.

It is interesting to note that whereas the tracker presented in the previous section was a bottom-up approach, in which multiple hypotheses on the camera pose were derived from low level hypotheses, particle filtering does the opposite. Multiple hypotheses are made on the camera pose at start, and the likelihood of these hypotheses is measured on the low level, in the image.

The main difficulty with these top-down approaches, as proposed in [16] and [23], results from the great size of the considered state space. For the tracking to be accurate enough, a large number of particles is needed. [16] and [23] use particle annealing method as a hierarchical approach to reduce the particle number. However, the likelihood functions proposed still need to be very fast to compute, since they have to be called for each particle. These functions may not distinguish enough between low level ambiguities.

In this paper, we propose to use particles resulting from our multiple hypotheses tracker to guide the particle set towards the local maxima of the distribution. The coherence is ensured using the so-called importance sampling method [13].

1) State space: The state space considered is the special Euclidean group $SE(3)$ of all possible pose matrices which transform points from homogeneous world coordinates to the camera coordinate frame.

$$ {}^{c}\mathbf{M}_{w} = \begin{bmatrix} {}^{c}\mathbf{R}_{w} & {}^{c}\mathbf{t}_{w} \\ 0 & 1 \end{bmatrix} $$

where ${}^{c}\mathbf{R}_{w} \in SO(3)$ is a rotation matrix and ${}^{c}\mathbf{t}_{w} \in \mathbb{R}^3$ is a translation vector. $SE(3)$ is the group of rigid body transformations.

2) Particles propagation on $SE(3)$: As in [16], the propagation model that has been considered in the experiments is a simple Gaussian noise centered on the previous pose. Since $SE(3)$ is a Lie group, there exists an exponential map between $SE(3)$ and its Lie algebra $se(3)$. Gaussian noise is first added on the canonical exponential coordinates and the resulting pose matrix is computed using the exponential map. Formally,

$$ \mathbf{x}_{pred} = \mathbf{M}_{\boldsymbol{\sigma}}.\mathbf{x} \tag{5} $$

where $\mathbf{M}_{\boldsymbol{\sigma}} = exp(\mathbf{v})$, $\mathbf{v} \sim \mathcal{N}_{\mathbf{0},\boldsymbol{\sigma}^2\mathbf{I_6}}$, $\mathbf{v} \in \mathbf{se(3)}$ and $\boldsymbol{\sigma}$ is the vector of the covariances associated to the components of $\mathbf{v}$.

3) Mean and averaging on $SE(3)$: Since the addition is not a binary operation on $SO(3)$ (and thus $SE(3)$), the arithmetic mean $\overline{\mathbf{R}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{R}_i$ of a set of rotation matrices $\mathbf{R}_i$ is usually not a rotation. It is however possible to define a meaningful average as the point in $SO(3)$ which minimises the sum of squared distances to the considered points. This distance can be extrinsic when we use the vector space embedding $SO(3)$ or intrinsic when a Riemannian distance is considered [9]. In this work an extrinsic distance was used, following [22]. The average rotation is thus computed as the arithmetic mean $\overline{\mathbf{R}}$, followed by the unique projection onto SO(3) given by the unique polar factor in the polar decomposition of $\overline{\mathbf{R}}$. Let $\overline{\mathbf{R}} = \mathbf{U\Sigma V}$ be the singular value decomposition of $\overline{\mathbf{R}}$, then the mean rotation $\mathbf{R}_m$ is given by:

$$ \mathbf{R}_m = \begin{cases} \mathbf{VU}^{\top} & \text{if } det(\overline{\mathbf{R}}) > 0 \\ \mathbf{VHU}^{\top} & \text{otherwise,} \end{cases} \tag{6} $$

where $\mathbf{H} = diag(1, 1, -1)$.

Finally, the weighted mean of the particle set is computed using this average rotation and the arithmetic mean of the translations.

4) Likelihood evaluation: Each particle represents a potential camera pose which has to be evaluated according to image measurements. In [16] the contours are projected according to the particle s to evaluate, and the ratio between the number n of pixels of the projected contours which do correspond to an edge in the image, and the total number v of pixels on the visible contours is computed. The likelihood of the particle s is then derived from this ratio by:

$$ p(\mathbf{z} \mid \mathbf{x} = s) = e^{\lambda \frac{n}{v}} \tag{7} $$

where λ is a parameter to be tuned. To decide whether a pixel do correspond to an edge in the image, a distance map [8] is first computed, providing for each pixel of the image the distance to the closest edge and its direction (see Figure 7). Then a threshold on the distance has to be set to determine the inlier/outlier count. The distance map has to be computed only once per frame, which make the likelihood value very fast to compute. [16] showed the computation can be performed in real time on a graphics processing unit (GPU).

In this paper, the distance map is directly used to compute a mean distance:

$$ d(s) = \frac{1}{N}\sum_{i} d_i \tag{8} $$

where d_i is the distance given by the distance map for the pixel i, that is the distance between the pixel i and the closest contour in the image. The pixels i are the pixels of the projected edges. The use of the direction to the nearest edge could improve the discriminative power of the distance function. However, we found that this measure was accurate enough in our experiments. Figure 8 shows the shape of the distance d with respect to in-plane translations for the frame of Figure 7. The likelihood is derived from this distance by:

$$ p(\mathbf{z} \mid \mathbf{x} = s) \propto \begin{cases} e^{-\lambda\left(\frac{d(s)-d_{min}}{d_{max}-d_{min}}\right)^2} & \text{if } d_{max} \neq d_{min} \\ 1 & \text{otherwise.} \end{cases} \tag{9} $$

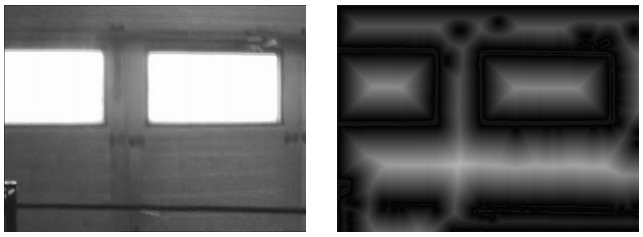


Fig. 7. Window frame (left) and its distance map (right). The darkest values correspond to the smallest distances.

B. Importance sampling

In our work we propose to use the registration method presented in Section II to guide the particle set towards the regions of interests. The approach is inspired from hybrid particle filters like [28] [4] [27] where some particles are moved to local maxima of the likelihood by a local optimisation. In our case, the optimisation corresponds to the multiple hypotheses tracker described in Section II. To reduce computational complexity, the optimisation is only applied to a subset of particles whom likelihood is above a given percentage of the maximum likelihood. The resulting set of

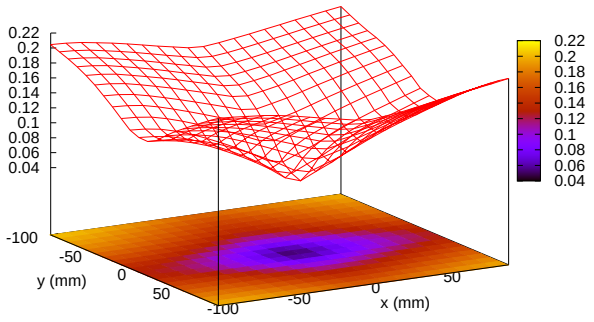


Fig. 8. Distance function with respect to x and y translations for the window frame. The zero position corresponds to the true camera pose.

particles still provides a good representation of the main modes of the density, but it is not directly sampled from the prior distribution $f_k(\mathbf{x}_k) = p(\mathbf{x}_k \mid \mathbf{z}_{1:k-1})$ as required by Bayesian filtering theory. However, the new particles $\{(s_k^{*(i)})\}_{i=1..N^*}$ can be regarded as sampled from an importance function $g_k(\mathbf{x}_k)$. A corrective term f/g can then be applied to the weights of the particles according to the importance sampling theory [13] to maintain a fair distribution. As in [4], $f_k(\mathbf{x}_k)$ and $g_k(\mathbf{x}_k)$ are approximated by Gaussian mixtures to evaluate the corrective term:

$$f_k(\mathbf{x}_k) = \frac{1}{N}\sum_{i}^{N} \mathcal{N}(s_k{}^{(i)}, \Sigma)(\mathbf{x}_k) \tag{10}$$

$$g_k(\mathbf{x}_k) = \frac{N}{N+N^*}\Big(\frac{1}{N}\sum_{i=1}^{N} \mathcal{N}(s_k{}^{(i)}, \Sigma)(\mathbf{x}_k) + \frac{1}{N^*}\sum_{i=1}^{N^*} \mathcal{N}(s_k^{*(i)}, \Sigma)(\mathbf{x}_k)\Big) \tag{11}$$

where $\mathcal{N}(s_k{}^{(i)}, \Sigma)$ denotes the 6-dimensional normal distribution of covariance Σ centered on the exponential coordinates of the pose s. The whole algorithm is summarized in the algorithm 2.

Note that to avoid a loss of diversity in the particles, known as the *sample impoverishment issue*, the initial particles are not removed but combined with the optimised ones.

Besides, the addition of optimised particles generated from a bottom-up optimisation process as well as the resampling step also prevent the approach from degeneracy issues.

C. Experiments

1) Comparative results: To underline the improvement in robustness brought by particle filtering framework, the tracker was tested on different sequences taken from a UAV. Comparative results are presented in the Figures 9 and 10.

The window sequence presents important frame to frame motion and occlusions, as much as great illumination changes. Results are shown in Figure 9. The registration process alone fails when the occlusion is too important (Figure 9 1-c).

Algorithm summary.
Given the set $\left\{(s_{k-1}{}^{(i)}, \frac{1}{N})\right\}_{i=1..N}$ of N particles of equal weights $\frac{1}{N}$ at frame $k-1$, the algorithm goes as follow:

- **Propagation** of the particles according to (5), giving the new set: $\left\{(s_k^{'(i)}, \frac{1}{N})\right\}_{i=1..N}$.
- **Distance measurement** for each particle, computed from equation (8), to determine which particle to optimise
- **Optimisation of the best particles:** the multiple hypotheses tracker is applied from the camera poses corresponding to the best particles (with a distance below a given threshold). One optimisation can lead to several hypotheses. A set of optimised particles $\left\{(s_k^{'*(i)}, \frac{1}{N^*})\right\}_{i=1..N^*}$ is obtained.
- **Distance measurement** for the optimised particles.
- **Combination** of the particles $s_k^{'(i)}$ and $s_k^{'*(i)}$ to get a set $\left\{(s_k^{(i)}, \frac{1}{N+N^*})\right\}_{i=1..N+N^*}$.
- **Weight computation** for each particle using a corrective term: $\pi_k^{(i)} \propto \frac{f_k(s_k^{(i)})}{g_k(s_k^{(i)})} p(\mathbf{z}_k \mid \mathbf{x}_k = s_k^{(i)})$, with $\sum_{i=1}^{N+N^*} \pi_k^{(i)} = 1$. See III-A4. It gives the set $\left\{(s_k^{(i)}, \pi_k^{(i)})\right\}_{i=1,..N+N^*}$
- **Estimation** of the tracking result as the weighted mean of the particle set (see III-A3).
- **Resampling** by performing a weighted draw of N particles among the $N+N^*$ particles

Algorithm 2: Algorithm summary.

Embedded in a particle filtering framework, the window is tracked all along the sequence.

This experiment shows the interest of introducing temporal filtering, with respect to the optimisation-based approach of Section II.

Thanks to the optimisation of some of the particles, a small number of particles is needed. For the window sequence of Figure 9, only 100 particles were used.

Figure 10 shows comparative results for a complex structured sequence with ambiguities (see rows 3 and 5), noise (2nd row for instance) and sometimes few information when few edges are visible in the image (4th row). The 1st column presents the results from the single optimisation approach. On the 2nd column a particle filter with 25 particles has been applied, which is not enough to track the camera pose. With 200 particles the filter does not diverge anymore but the accuracy is still poor (3rd column). Our hybrid approach (4th column) allows good estimation with very few particles (25 particles only were used in this example).

2) Computational complexity and execution time: In the different tracking approaches considered in this paper, for a given image and model, the computational complexity depends on different parameters:

- the number K of optimisations,
- the average number M of iterations per optimisation,
- the number N of particles when particle filtering is used.

If M denotes the average number of iterations necessary for the classic registration method, then the complexity of this

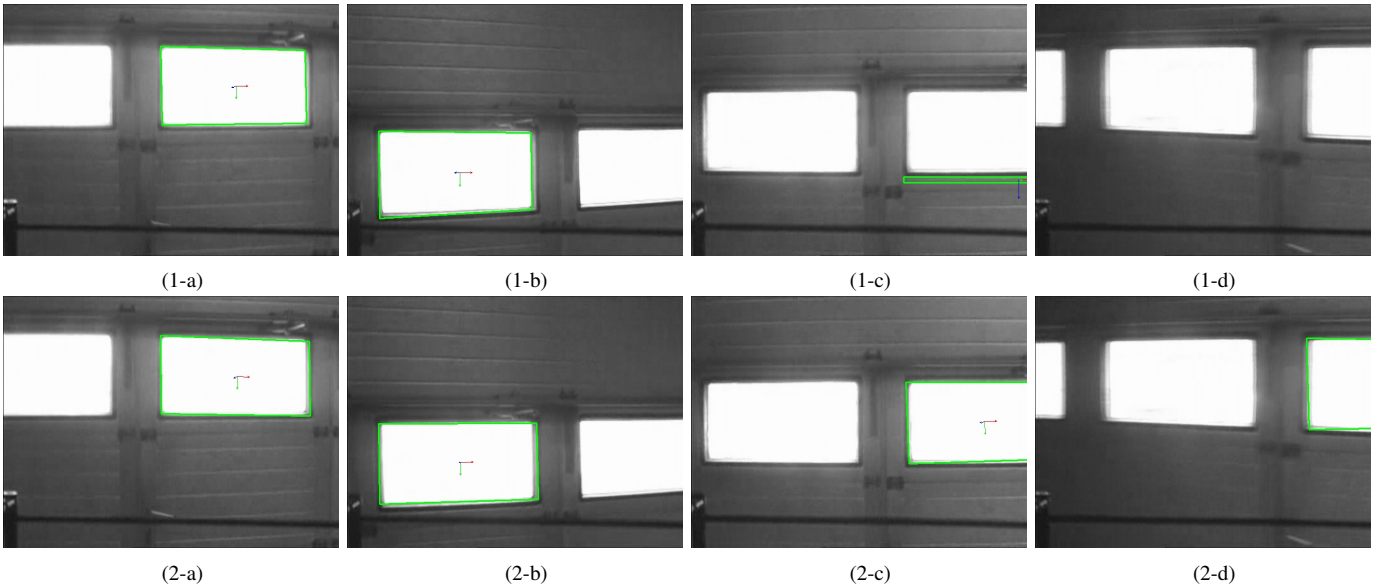


Fig. 9. Window sequence. The registration method alone (first row) fails when large occlusions occur (1-c). The multiple hypotheses tracker embedded in particle filtering framework (second row) tracks the object successfully.

method is $C = O(M)$.

The multiple hypotheses optimisation process of Section II requires the computation of several optimisations when ambiguities are met. In practice, the number K of optimisations needed for a real robustness improvement does not exceed 4 or 5, and this approach is still suitable for real time: $C = O(M)$.

In the case of particle filtering integration III, the complexity mainly depends on the number of particles and the optimisations applied: $C = O(N \times M \times K)$. However, the number N of particles required is significantly lower than for classic particle filtering. Moreover, the mean number of iterations required by the optimisation process is reduced thanks to the prediction of the particle filter, and the fact that the optimisation is only applied to the best particles.

In the example of Figure 10 the execution time of the hybrid approach was 10 frames per second, without specific optimisation, on a standard PC computer. Since particle filtering is well adapted to parallelisation, the approach is suitable for real-time applications.

IV. Suitability of the approach for UAV positioning tasks.

In Sections II and III we presented a multiple hypotheses framework to estimate the pose of a camera and we showed how it allows to improve the tracking robustness with regard to the constraints associated with UAV flight. This section presents an experimental validation of the suitability of the presented approach in the UAV indoor navigation context. Since our current implementation of the algorithm of Section III is still slow for UAV control, we consider here a simplified version to validate the feasibility of the global approach. We thus estimate the pose from our multiple hypotheses tracking (Section II) with 3 optimisations, and fuse it with inertial data in a Kalman filter instead of particle filtering. With such simplification, the pose estimation is still better than the classical approach, as shown in Section II, since it still retrieves several poses from the images. However, compared to the particle filter, Kalman filtering is mono-modal and does not keep a memory of all the candidate poses when one is selected. The main objective is to test our vision-based approach in a real UAV experiment, faced with the challenges of flight conditions, to validate the feasibility of the overall system.

A. Experimental setup

The experiments have been conducted on the quad-rotor (X4-flyer) developed by the CEA LIST (Figure 11).

Fig. 11. Quad-rotor UAV.

The UAV sends the images from its embedded camera to the ground station (PC) via a wireless analogical link of 2.4GHz. Images are processed on the ground station with a framerate of 20Hz. A scene was built, combining planar and 3D objects (see Figure 2). The tracking initialisation has not been considered in this paper. During the experiment, the tracking is automatically initialised by detecting a black dot in its first position (Figure 2). Then the vehicle can locate itself thanks to the model-based tracking, without using the dot anymore.

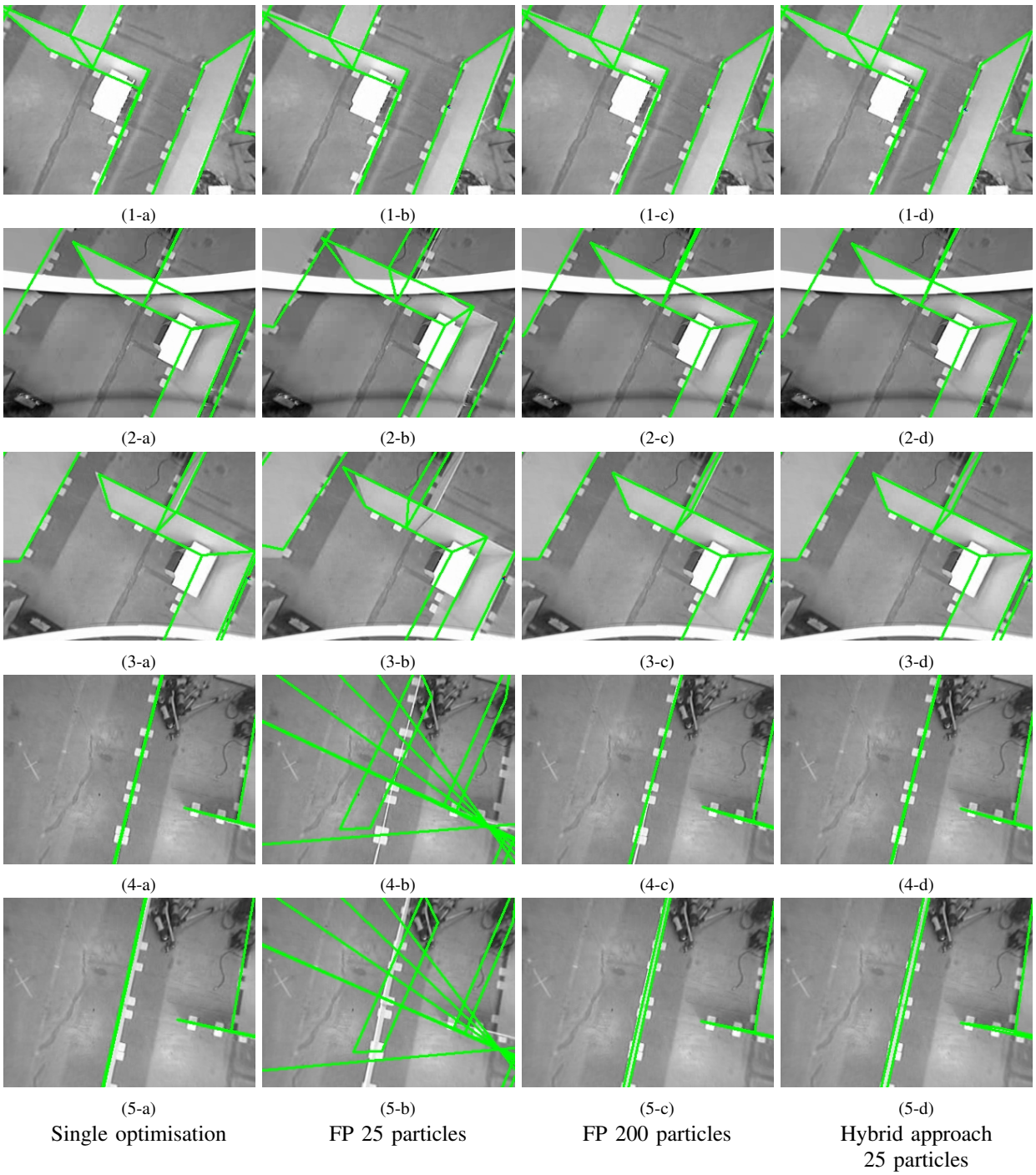


Fig. 10. Comparative results for the apartment sequence.

B. Accuracy of the estimation

To evaluate the accuracy of the estimation of the position of the UAV, we compared it to a ground truth obtained using a metrologic laser tracker Leica[2] of micrometric precision.

Figures 12 and 13 show the comparative results obtained for the pose estimation for a flight in the modelled scene.

The standard deviation for the pose estimation is about 16cm in translation which is much better than the localisation achieved with standard GPS. The main limiting factor here is

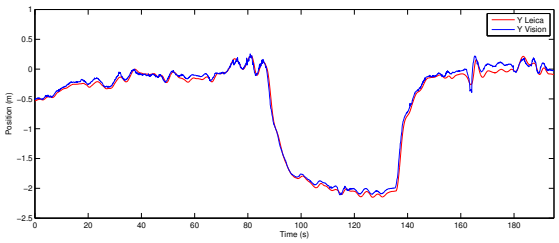


Fig. 12. Position estimated with the model-based tracking (blue) and ground truth (red).

[2]http://metrology.leica-geosystems.com/

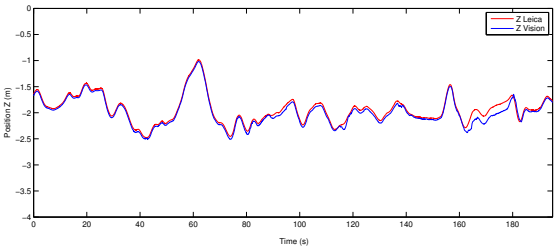


Fig. 13. Position estimated with the model-based tracking (blue) and ground truth (red).

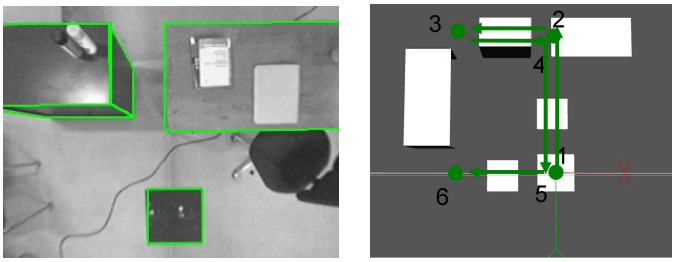
the accuracy of the 3D model, which was obtained by coarsely measuring the different elements.

This experiment also shows the robustness of the approach to typical interference noise that was present in the flight (see Figure 2 (b)).

C. Navigation

In [29] we show that such a localisation method can be used for the UAV control. We do not focus here on the control strategy which is not the aim of this paper. We show however that the proposed approach was successfully validated on a way-point navigation trajectory.

The task considered was to autonomously reach several set points successively: first the vehicle is stabilised 2 meters above the dot target ($\mathbf{p} = (0, 0, -2)$). Then the set points are successively set to $(0, -2, -2)$, $(-2, -2, -2)$, $(0, -2, -2)$, $(0, 0, -2)$, $(-2, 0, -2)$ (see Figure 14b).



(a) (b)

Fig. 14. Example of camera view with model reprojection (a) and required trajectory (b).

Figure 15, shows the estimated and ground truth trajectories. The system was able to localise itself despite noise and large interframe motion.

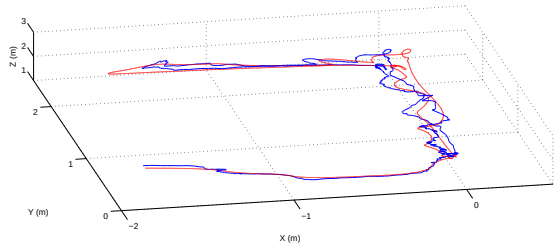


Fig. 15. UAV trajectory as estimated by our vision algorithm (blue) and ground truth (red).

V. Conclusions

In this paper, we proposed a novel model-based tracking system. Our multiple hypotheses registration process provides us with multiple pose hypotheses by performing several minimizations, corresponding to different sets of points. We show how these hypotheses can be integrated into a particle filter. The multiple hypothesis tracker is applied to the best particles to move them to the local maxima of the likelihood function. The particle set is therefore guided toward the candidate poses emerging from the multiple hypothesis tracker. Although the state space is large, a small number of particles are needed. In that case, the tracking approach benefits from both the temporal coherence and multi-modal representation of the Bayesian framework as well as the accuracy of a registration process. The resulting approach has been validated on different sequence from UAV flights. A simplified approach has been tested on positioning tasks on a quad-rotor UAV. The accuracy of the estimates has been evaluated and the experiment shows the feasibility of the proposed approach in indoor structured environments.

References

[1] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer. 2D simultaneous localization and mapping for micro aerial vehicles. In *Proceedings of the European Micro Aerial Vehicles (EMAV 2006) conference*, 2006.

[2] M. Armstrong and A. Zisserman. Robust object tracking. In *Proc. Asian Conference on Computer Vision*, volume 1, pages 58–61, 1995.

[3] O. Bourquardez and F. Chaumette. Visual servoing of an airplane for auto-landing. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 1314–1319, 2007.

[4] M. Bray, E. Koller-Meier, and L. Van Gool. Smart particle filtering for 3D hand tracking. *IEEE International Conference on Automatic Face and Gesture Recognition*, 1:675–680, 2004.

[5] A.I. Comport, E. Marchand, M. Pressigout, and F. Chaumette. Real-time markerless tracking for augmented reality: The virtual visual servoing framework. *IEEE Trans. on visualization and computer graphics*, 12(4):615–28, July/August 2006.

[6] J. Courbon, Y. Mezouar, N. Guenard, and P. Martinet. Vision-based navigation of unmanned aerial vehicles. *Control Engineering Practice*, 18(7):789 – 799, 2010. Special Issue on Aerial Robotics.

[7] R. Drummond, T. Cipolla. Real-time visual tracking of complex structures. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(7):932–946, 2002.

[8] R. Fabbri, L. Da F. Costa, J. C. Torelli, and O. M. Bruno. 2D euclidean distance transform algorithms: A comparative survey. *ACM Comput. Surv.*, 40(1):1–44, 2008.

[9] P.T. Fletcher, C. Lu, and S. Joshi. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*.

[10] J.A. Hartigan. *Clustering algorithms*. John Wiley & Sons, 1975.

[11] B. Hérissé, T. Hamel, R. Mahony, and F.-X. Russotto. A terrain-following control approach for a VTOL unmanned aerial vehicle using average optical flow. *Autonomous Robots*, pages 1–19, 2010.

[12] M. Isard and A. Blake. CONDENSATION: conditional density propagation for visual tracking. *Int. Journal of Computer Vision*, 29(1):5–28, 1998.

[13] M. Isard and A. Blake. ICONDENSATION: Unifying low-level and high-level tracking in stochastic framework. In *European Conf. on Computer Vision*, volume 1, pages 893–908, 1998.

[14] C. Kemp and T. Drummond. Multi-modal tracking using texture changes. *Image and Vision Computing*, 26(3):442–450, 2008.

[15] Christopher Kemp. *Visual control of a miniature quad-rotor helicopter*. PhD thesis, University of Cambridge, 2006.

[16] G. Klein and D. Murray. Full-3d edge tracking with a particle filter. In *British Machine Vision Conf.*, volume 3, pages 1119–1128, 2006.

[17] S. Klose, J. Wang, M. Achtelik, G. Panin, F. Holzapfel, and A. Knoll. Markerless, vision-assisted flight control of a quadrocopter. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Taipei, Taiwan, October 2010.

[18] V. Lepetit and P. Fua. Monocular model-based 3D tracking of rigid objects: A survey. In *Foundations and Trends in Computer Graphics and Vision*, pages 1–89, 2005.

[19] D.G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13:441–450, 1991.

[20] R. Mahony and T. Hamel. Image-based visual servo control of aerial robotic systems using linear image features. *IEEE Transactions on Robotics*, 21(2):227–239, 2005.

[21] M. Meingast, C. Geyer, and S. Sastry. Vision based terrain recovery for landing unmanned aerial vehicles. In *IEEE Conf. on Decision and Control*, volume 2, pages 1670 – 1675, dec. 2004.

[22] M. Moakher. Means and averaging in the group of rotations. *SIAM journal on matrix analysis and applications*, 24:1–16, 2002.

[23] S. Nuske, J. Roberts, and G. Wyeth. Visual localisation in outdoor industrial building environments. In *IEEE Int. Conf. on Robotics and Automation*, pages 544–550, 2008.

[24] A. Petit, E. Marchand, and K. Kanani. Tracking complex targets for space rendezvous and debris removal applications. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2012.

[25] M. Pupilli and A. Calway. Real-time camera tracking using known 3D models and a particle filter. In *Int. Conf. on Pattern Recognition*, pages 199–203, August 2006.

[26] S. Rathinam, Zu Kim, A. Soghikian, and R. Sengupta. Vision based following of locally linear structures using an unmanned aerial vehicle. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, pages 6085–6090, 2005.

[27] C. Shan, T. Tan, and Y. Wei. Real-time hand tracking using a mean shift embedded particle filter. *Pattern Recognition*, 40(7):1958–1970, 2007.

[28] J. Sullivan and J. Rittscher. Guiding random particles by deterministic search. In *IEEE Int. Conf. on Computer Vision*, volume 1, pages 323 –330, 2001.

[29] C. Teulière, L. Eck, E. Marchand, and N. Guenard. 3D model-based tracking for UAV position control. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Taipei, Taiwan, 2010.

[30] C. Teulière, E. Marchand, and L. Eck. Using multiple hypothesis in model-based tracking. In *IEEE Int. Conf. on Robotics and Automation*, Anchorage, Alaska, May 2010.

[31] L. Vacchetti, V. Lepetit, and P. Fua. Combining edge and texture information for real-time accurate 3d camera tracking. In *IEEE Int. Symposium on Mixed and Augmented Reality*, pages 48–57, 2004.

[32] K. Valavanis. *Advances in Unmanned Aerial Vehicles. State of the Art and the Road to Autonomy*. Springer-Verlag, 2007.