3D object pose detection using foreground/background segmentation

Antoine Petit, Eric Marchand, Rafiq Sekkal, Keyvan Kanani

***Abstract*— This paper addresses the challenge of detecting and localizing a poorly textured known object, by initially estimating its complete 3D pose in a video sequence. Our solution relies on the 3D model of the object and synthetic views. The full pose estimation process is then based on foreground/background segmentation and on an efficient probabilistic edge-based matching and alignment procedure with the set of synthetic views, classified through an unsupervised learning phase. Our study focuses on space robotics applications and the method has been tested on both synthetic and real images, showing its efficiency and convenience, with reasonable computational costs.**

I. Introduction

The scope of this paper deals with 3D initial pose estimation of a single known 3D object in a monocular image sequence, with a focus on the case of a space object, such as a satellite or a debris moving in outter space. Solving this issue is a key requirement to initialize a robust and accurate frame-by-frame 3D tracking phase, for instance for space autonomous rendezvous and space debris removal applications.

In the field of monocular 3D object recognition, detection and pose estimation, different classes of approaches can be considered among the large literature. Some methods (single or multi class) are based on global template matching using real training templates of the object. Some of them consider appearance [8], or shape [15, 7, 9] to represent the object. Many others are based on learning local or semi-local features described by descriptors such as SIFT [14], region descriptors such as HOG [5], extracted from real training images of the object. The online recognition and detection phase can then provide, potentially besides the object class, pose or viewpoint estimates, through a pose computation step based on 2D-3D point correspondences with the 3D model [11], using a voting process method [14, 24, 18]. But in our context, these methods, based on real training images, are not suitable since natural images of space objects can hardly be obtained prior to the mission itself. Besides, space objects are often poorly textured or prone to specular effects (for instance due to the insulating film on satellites), making the description of templates or the extraction and description of local features complicated.

Using the 3D model of the object: instead, we propose to rely on another class of approaches which learns the geometry or the shape of the 3D model of the object, which is assumed to be known and present in the image. We deal with industrial objects (spacecrafts, satellites or parts of them), for which accurate geometrical CAD models can be available.

Some template matching methods [26, 17], sparse 2D-3D edge feature matching techniques [13, 23] or multiview learning frameworks based on part or region descriptors [12, 20], suggest to use and learn the 3D model of the target object and its projection. However, approaches such as [13, 23], based on matching geometrical primitives such as lines, can be computationally prohibitive, due to the large search space. Furthermore, they face problems when extracting the considered geometrical primitives from edges in the image with degraded conditions such as noise, blur, or background clutter. Region or part descriptor based methods [12, 20] have recently proposed to overcome the issue of computational costs by efficiently learning the 3D model with shape descriptors. But these solutions, which are more suited for object class detection, still require a certain amount of supervision during the learning step and are restricted to a coarse set of viewpoints.

Towards template matching: Here we design an unsupervised method for single class object pose detection, precise enough to correctly initialize a frame-by-frame tracking process, while keeping computational costs reasonable. We propose to follow the idea of template matching. Some efficient edge or shape based global similarity measures [15, 22, 1] have been considered to cope with occlusion, clutter, noise, and specular effects... Our idea is thus to match an exhaustive set (over the 6D pose parameters), of non photorealistic synthetic views of the object. With the issue of the large search space, we propose to efficiently learn the set of views. In this sense, we rely on the concept of aspect or view graph [4, 25] or of hierarchical view graph [26, 17], leading to sets of reference views of the object. Our pose estimation process can then be performed by matching the input image with these graph structures.

Benefiting from foreground/background segmentation: the considered single object is assumed to be moving with respect to its background. Thus, we suggest to take advantage of a foreground/background segmentation technique, as in [25].

For computational efficiency and robustness concerns, we propose to spread our object localization process over a sequence of successive input images. Only reasoning on the first frame could indeed result in a too coarse pose estimate, or would require a more exhaustive and costlier searching process over the pose parameters. With our system, the retrieval of the pose is then based on progressively

A. Petit was with Inria Rennes, France. He is now with Universita degli Studi di Napoli Federico II, Napoli, Italia, antoine.petit@unina.it

E. Marchand is with Université de Rennes 1, IRISA, Lagadic Team, France, Eric.Marchand@irisa.fr

R. Sekkal was with INSA Rennes, IRISA, Lagadic Team, France

K. Kanani is with Astrium, Toulouse, France

matching and aligning the synthetic views with a short image sequence. At the end of the process the most likely view is determined and selected, along with the stored pose used to render it. Combining this pose with estimated in-plane rotation, translation and scaling parameters to align the view in the image allows us to compute the pose. Indeed, with our applications the dimensions of the object can be assumed to be small ($\sim$ 10m) relatively to its distance from the camera ($\sim$ 100/1000m), a weak perspective projection model can be assumed: an isotropic scaling (equivalent to a translation along the optical axis) precedes an orthographic projection.

In this work we propose an accurate edge-based distance function which is made robust to segmentation errors, by involving both the segmented and the original images. We also suggest to use the image in-plane translation, rotation and scale of the segmented silhouette to coarsely estimate the similarity transformation of the considered tested view to guide the matching process and compute the pose. This framework would thus result in a fast process, and would be less sensitive to local minima, in contrast to [26, 25], which rely on a costlier coarse-to-fine search [26], or an exhaustive probabilistic inference over these parameters [25].

Contributions: As main contributions, this paper suggests a novel probabilistic matching and edge-based alignment framework between the views and the image to retrieve the full 6 DoF pose. A novel foreground/background segmentation method suited for the targeted application combining color statistical modeling and motion-based compensation is also proposed.

Overview of the approach: our method can be outlined by the following steps:

- **Learning step** : based on generated synthetic views of the object, it aims at building a hierarchical model view graph leading to some reference views of the model.
- **Pose estimation step along the image sequence** :
  - Foreground/background segmentation of the object. By computing binary moments of the extracted silhouette, an initial estimate of the image in-plane translation, rotation and scale transformation parameters of the reference views in the image can be determined.
  - With the aim of refining these parameters, particle filtering is performed with respect to them, for each reference view, along the input image sequence.
  - We then determine the most likely model view and an associated estimate of the image similarity transformation of the view in the image, providing the complete pose, along the input image sequence.
  - Finally the matched view and the pose and refined by traversing through the hierarchical view graph.

II. Foreground/background segmentation

We deal with a single object moving in an input image sequence. The aim of this segmentation task is to extract a foreground layer, corresponding to the object silhouette in the image, in the presence of a potentially cluttered and dynamic background. The information provided by the extracted silhouette will further be used in the pose estimation step (see Section IV-E).

Among the vast literature addressing the issue of segmentation, classical approaches rely on background modeling and substraction [21] but these methods requires a prior knowledge of the background and are limited to static or weakly dynamic backgrounds. Other methods [3, 19] have based their solution on the assumption that foreground and background layers have different motion patterns. The general idea is then to automatically extract and classify these patterns and build models of both layers. Segmentation can then be efficiently achieved using graph cuts [2].

Since the apparent motions of both the foreground (the moving object) and the background can potentially be identified, our basic idea is also to use a statistical foreground/background modeling technique. In our application, we can assume that, locally, the Earth is a rigid body at rest in space, and that the apparent motion of the background is due to the camera selfmotion, the processing time of the algorithm being negligible with respect to the Earth self rotation period. For this reason we rely on the idea suggested in [19], consisting in automatically identifying the background and the foreground layers and then to statistically model them.

As in many layer segmentation methods, we use an energy minimization framework, based on statistical models of the foreground and the background.

A. Energy Minimization formulation

For an image $\mathbf{I}_k$, we denote by $\alpha = \{\alpha_i\}_{i=1}^N$ the set of the unknown binary labels of the set of pixels $\{\mathbf{p}_i\}_{i=1}^N$ of $\mathbf{I}_k$ ($\alpha_i = 0$ for the background pixels, $\alpha_i = 1$ for the foreground). Estimating the values $\widehat{\alpha}$ of the labels for an entire image can be formulated as the minimization of an energy-based Markov Random Field objective function $E(\alpha)$, with respect to α:

$$E(\alpha) = E_{data}(\alpha) + \gamma E_{smooth}(\alpha) \quad (1)$$

$$\text{with} \quad E_{data}(\alpha) = \sum_i U_i(\alpha_i) \quad (2)$$

E_{data} is the "data" energy term, with $U_i(\alpha_i)$ a unitary term accounting for the observation probability $p(\mathbf{p}_i \mid \alpha_i)$ of pixel $\mathbf{p}_i$ to belong to the foreground or to the background, based on some image "data" (intensity, color, location...) observed in the image at pixel $\mathbf{p}_i$, using the statistical models previously built for the background and the foreground. More formally, we have $U_i(\alpha_i) = -log(p(\mathbf{p}_i \mid \alpha_i))$. E_{smooth} is the smoothness energy term whose goal is to favor smoothness, or spatial coherence within the pixels [3].

In order to compute the optimal solution of this energy minimization problem and to determine $\widehat{\alpha}$, we employ the *graph cuts* algorithm [2].

In our context, we propose to compute the data energy term using two different terms. One term is obtained through foreground and background statistical modeling (E^m_{data}, see section II-B). The other term is computed by modeling the motion of the background and using homography-based

motion compensation (E^c_{data}, see section II-C). Formally, E_{data} can be rewritten as:

$$E_{data}(\alpha) = \beta E^m_{data}(\alpha) + (1-\beta)E^c_{data}(\alpha). \quad (3)$$

β is a weighting parameter ($0 < \beta < 1$). $p^m(\mathbf{p}_i \mid \alpha_i)$ and let us define $p^c(\mathbf{p}_i \mid \alpha_i)$ as the corresponding observation probabilities for E^m_{data} and E^c_{data}.

B. Feature tracking, clustering and foreground/background modeling

As in [19], we propose to identify and describe both foreground and background layers by processing some feature points that will be tracked over a certain number of frames and will be classified as background or foreground points, consistently with their motions. We choose Harris corner points that are detected on the first frame. By tracking them over the image sequence with the Kanade-Lucas-Tomasi (KLT) tracker, we obtain a set of trajectories on a sliding window. When a frame $\mathbf{I}_{k_0}$ is reached, the goal is then to cluster these trajectories into background or foreground trajectories. Since the background is supposed to be stationary in the world frame, we follow the approach of [19], which uses the rank-constraint for the background. It means that the matrix formed by the projected trajectories of stationary points in the world frame is a rank three matrix, so that background trajectories must lie in a subspace spanned by three basis trajectories. RANSAC is used in order to robustly determine these basis trajectories from the set of all trajectories, and to finally identify trajectories that lie within the resulting subspace. This method enables to efficiently cluster trajectories and the corresponding feature points into foreground trajectory points or background trajectories. We obtain a set of background trajectory points $\{\mathbf{p}^b_i\}_{i=1}^{N_b}$ and a set of foreground trajectory points $\{\mathbf{p}^f_i\}_{i=1}^{N_f}$. We then use these sets to determine the statistical models of both background and foreground layers. For both layers Kernel Density Estimation is employed as probabilistic modeling [19], but in our approach, only a color model is used for the background. The foreground model is instead based on both color and spatial information. A reason for this choice is that the foreground layer is likely to be concentrated in the image, making spatial information more discriminative than for the background.

More formally, the background model is based on the set of vectors $\{\mathbf{z}^b_i\}_{i=1}^{N_b}$, where $\mathbf{z}^b_i = \begin{bmatrix} R_i & G_i & B_i \end{bmatrix}^T$, with R_i, G_i and B_i the RGB color coordinates of pixel $\mathbf{p}^b_i$. The foreground model is based on the set of vectors $\{\mathbf{z}^f_i\}_{i=1}^{N_f}$, where $\mathbf{z}^f_i = \begin{bmatrix} R_i & G_i & B_i & u_i & v_i \end{bmatrix}^T$, with R_i, G_i and B_i the RGB components of $\mathbf{p}^f_i$ and u_i and v_i the pixel coordinates of $\mathbf{p}^f_i$.

Then the probability $p^m(\mathbf{p}_i \mid \alpha_i)$ for a pixel $\mathbf{p}_i$ to belong to the background or the foreground is then computed using Kernel Density Estimation, by selecting the appropriate data $\mathbf{z}_i$ on $\mathbf{p}_i$, and using an Epanechnikov kernel function, which is fast to compute. But kernel density estimation is computationally expensive, and both layers are modeled this way only for the first frame $\mathbf{I}_{k_0}$. For the next ones, based on the data provided by this initial segmented frame, the background and the foreground are represented by smoothed color histograms h which are adaptively learned over successive frames.

C. Homography based motion compensation

Relying on the important assumption that the background can be considered as planar, the idea is to evaluate pixel observation probabilities through motion compensation. It is based on the estimation of the homography transformation induced by the motion of the background in successive frames. With this motion compensation framework, the idea is to compensate for errors induced by a poor modeling of foreground and the background layers, due to some misclassifications of the trajectory points.

By using background trajectory points identified at a frame $\mathbf{I}_k$, and the same points at frame $\mathbf{I}_{k-k_H}$, with $k_H \leq k_0$ we can compute the homography ${}^k\mathbf{H}_{k-k_H}$ between the two background planes, with a RANSAC robust procedure. ${}^k\mathbf{H}_{k-k_H}$ is then applied to the whole frame $\mathbf{I}_{k-k_H}$, to compensate for the background motion between $\mathbf{I}_{k-k_H}$ and $\mathbf{I}_k$, and thus to discriminate the foreground layer, which has a different motion, by computing the error $\mathbf{e}(\mathbf{p}_i)$:

$$\mathbf{e}(\mathbf{p}_i) = \mathbf{I}_k(\mathbf{p}_i) - \mathbf{I}_{k-k_H}({}^k\mathbf{H}_{k-k_H}(\mathbf{p}_i)) \quad (4)$$

Likelihoods can then be evaluated by a Gaussian kernel, with a bandwidth σ:

$$p^c(\mathbf{p}_i \mid \alpha_i = 0) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{\|\mathbf{e}(\mathbf{p}_i\|)^2}{2\sigma^2}} \quad (5)$$

$$p^c(\mathbf{p}_i \mid \alpha_i = 1) = 1 - p^c(\mathbf{p}_i \mid \alpha_i = 0) \quad (6)$$

III. Hierarchical model view graph

A. Generation of synthetic views

The purpose of our template matching method is to align a sequence of successive initial input images with synthetic views generated from the 3D model. These views are generated on a view sphere centered on the 3D model. This is managed by a 3D rendering engine by setting virtual cameras at uniformly spaced viewpoints (see Figure 1). [4, 25, 17] process these views as silhouette shapes. Instead we propose to extract both silhouette and internal edges of the rendered views through a Laplacian filter computed on the depth maps, in order to avoid ambiguities between silhouettes (Figure 1). For each generated view V, we store the pose ${}^{cV}\mathbf{M}_o$ used to render the 3D model. Besides, we also store the centroid $\overline{\mathbf{c}}^V = \begin{bmatrix} \overline{u}^V & \overline{v}^V \end{bmatrix}$, orientation α^V and area A^V of the silhouette of the projected 3D model (Figure 1 right). These parameters are evaluated using image moments.

B. Building a hierarchical model view graph

Since the process of matching the whole set of views with the input images can be computationally challenging, we iteratively cluster the views into a hierarchical view graph, as in [26, 17]. We consider an unsupervised clustering technique based on Affinity Propagation [6], similar to [17]. At the first level of the hierarchy, we build clusters within disjoint neighborhoods on the sphere. This is done by comparing the views with each other in each neighborhood using a

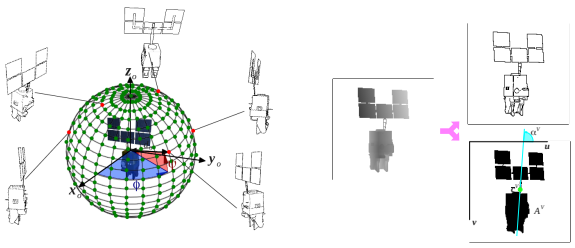


Fig. 1: Generation of synthetic views on a view sphere centered on the 3D model. Contours are extracted by processing the depth buffer of the rendered 3D model, and silhouette parameters are compute using image moments.

distance function (see Section III-C). A slight overlap is considered between the different neighborhoods to consider inter neighborhood variabilities. The result is a set of clusters, represented by reference views, which define the first level of our hierarchical structure. We proceed in the same way with these views. Since this set has an acceptable size, we do not consider spatial neighborhoods from this level. We can then iteratively build successive hierarchical levels until a reasonable number N_r of reference model views is reached. A set $\{V^j\}_{j=1}^{N_r}$ of reference views is finally obtained.

C. An edge-based distance function between synthetic views

The proposed distance function D, intended to compare two synthetic views, is based on all the extracted edges of the two views, instead of silhouette contours as with the Shape context used in [25], which is based on the silhouette contours. From the sets of edge points $\{\mathbf{c}_k^i\}_{k=1}^{N_i}$ and $\{\mathbf{c}_k^j\}_{k=1}^{N_j}$ (both silhouette and internal edges) on views V^i and V^j, we compute an oriented Chamfer distance by looking for the closest edge from one view to the other, so that $D_{i,j} = \frac{1}{2}(d_{i,j} + d_{j,i})$. $d_{i,j}$ is related to the distance for each edge point $\mathbf{c}_k^i$ of V^i to the closest one in V^j. It also takes into account the difference between the orientation $\theta(\mathbf{c}_k^i)$ of the edge point $\mathbf{c}_k^i$ and the orientation of the corresponding closest edge point in V^j. It results in an accurate and robust discriminative distance function by taking into account distance between edges of the views and the difference between their corresponding orientations.

IV. A PROBABILISTIC FRAMEWORK FOR ALIGNING AND MATCHING REFERENCE VIEWS WITH INPUT IMAGES

Our problem consists in matching and aligning the reference model views to each input image and finding the most likely one. Once a first image is segmented, at time step k_0, the next input images are used to determine a pose estimate. In order to ensure smooth transitions between the matched and aligned model views, we propose a probabilistic framework to determine the best view. Let us first describe how the pose can be determined from the alignment of a given synthetic view V with an input image $\mathbf{I}$.

A. Rough pose computation assuming a weak perspective model

We assume a weak perspective projection model, justified in our applications by the fact that the dimensions of the target space object are small relatively to the distance from the camera. Based on this assumption, the pose ${}^c\mathbf{M}_o$ between the camera and the object can be retrieved using the stored pose ${}^{c_V}\mathbf{M}_o$ used to generate the considered synthetic view V and the similarity transformation which aligns the view V with the image $\mathbf{I}$. This similarity transformation can be represented by four parameters: the in-plane rotation, expressed by a rotation angle β, the 2D translation vector $\mathbf{t} = \begin{bmatrix} t_x & t_y \end{bmatrix}^T$, and the scaling s. Let $\mathbf{R}_{-\beta}$ denotes the 3D rotation matrix of angle $-\beta$ around the optical axis $\mathbf{z}_c$. The rotation matrix ${}^c\mathbf{R}_o$ of ${}^c\mathbf{M}_o$ can then be computed through ${}^c\mathbf{R}_o = \mathbf{R}_{-\beta}\ {}^{c_V}\mathbf{R}_o$, with ${}^{c_V}\mathbf{R}_o$ the rotation matrix of ${}^{c_V}\mathbf{M}_o$. Since synthetic views are generated on a view sphere centered on the 3D model, at a fixed distance d_0, and since scaling is assumed isotropic, the translation vector ${}^c\mathbf{t}_o$ of ${}^c\mathbf{M}_o$ is given by ${}^c\mathbf{t}_o = s\ d_0 \begin{bmatrix} t_x & t_y & 1 \end{bmatrix}^T$.

Next, we present our solution to align each reference view V_r^j with the input images (section IV-B), giving a fair similarity transformation, and to determine the best matching (or most likely) reference view (section IV-D).

B. Aligning a reference view by refining similarity transformation parameters

Using the segmentation technique presented in section II, the silhouette of the object can be extracted on the first segmented frame $\mathbf{I}_{k_0}$. The centroid $\overline{\mathbf{c}} = \begin{bmatrix} \overline{u} & \overline{v} \end{bmatrix}$, orientation α and area A of this silhouette can be then evaluated, using the image moments.

Given a reference view V^j and using its stored silhouette parameters $\begin{bmatrix} \overline{u}^j & \overline{v}^j & \alpha^j & A^j \end{bmatrix}^T$, we can retrieve the similarity transformation to align V^j with $\mathbf{I}_{k_0}$. This similarity transformation can be expressed by vector $\mathbf{x}^j$:

$$\mathbf{x}^j = \begin{bmatrix} t_u & t_v & \beta & s \end{bmatrix} \quad (7)$$

$$= \begin{bmatrix} \overline{u} - \overline{u}^j & \overline{v} - \overline{v}^j & \alpha - \alpha^j & \sqrt{\frac{A}{A^j}} \end{bmatrix} \quad (8)$$

Based on the process presented in section IV-A, the pose ${}^{c_{V^j}}\mathbf{M}_o$ (used to generate V^j) and $\mathbf{x}^j$ can provide us with a pose ${}^c\mathbf{M}_o$, for the considered view V^j. However, due to some segmentation errors, $\begin{bmatrix} \overline{u} & \overline{v} & \alpha & A \end{bmatrix}^T$ may be too coarsely computed. We thus propose, for each reference view V^j, to refine the parameters $\mathbf{x}^j$.

C. Refining as particle filtering

With the aim of estimating and refining $\mathbf{x}^j$ by minimizing a distance function between a reference view and the observed input image, we propose an efficient and robust solution by using particle filtering, which is particularly suited to deal with the non-linearity of this distance function, in contrast to local deterministic minimization techniques such as Gauss-Newton or Levenberg-Marquardt. It is also more optimal and computationally efficient than coarse-to-fine searches.

Given a reference model view V^j and a first image $\mathbf{I}_{k_0}$, we estimate and refine the corresponding parameters $\mathbf{x}^j$ using particle filtering, and we propose to use the CONDENSATION [10] formulation of the filter, whose steps are recalled hereafter, and which is illustrated on Figure 2.

In this sense, the similarity transform $\mathbf{x}_k^j$, for a frame $\mathbf{I}_k$, is represented by a finite set $\{\mathbf{x}_k^{(i,j)}\}_{i=1}^{N_j}$ of N_j samples, or particles, associated with weights $\{w_k^{(i,j)}\}_{i=1}^{N_j}$, with $\sum_{i=1}^{N_j} w_k^{(i,j)} = 1$. Then, after initialization, the process consists in predicting the states of the particles according to a motion model (here a simple Gaussian noise is used), in updating the weights using a likelihood function and in performing a random weighted draw among the particles to avoid degeneracy. The estimate $\widehat{\mathbf{x}}_k^j$ is then the estimator of probability expectation.

Likelihood evaluation: a likelihood function needs to be evaluated for each particle to compute its weight. The function chosen here is derived from the distance function presented in section III-C. For a model view V^j, a particle $\mathbf{x}_k^{(i,j)}$, an image $\mathbf{I}_k$ and its corresponding segmented image $\mathbf{I}_k^{seg}$, it consists in the distance $D(\mathbf{x}_k^{(i,j)})$ between the contour points $\{\mathbf{p}_l^{i,j}\}_{l=1}^{M_\mathbf{x}}$ extracted from V^j translated, scaled and rotated around its centroid with respect to $\mathbf{x}_k^{(i,j)}$, and the corresponding closest contour points of both sets $\{\mathbf{p}_{l,k}\}_{l=1}^{N}$ and $\{\mathbf{p}_{l,k}^{seg}\}_{l=1}^{P}$ extracted from $\mathbf{I}_k$ and $\mathbf{I}_k^{seg}$ using a Canny edge detector. It gives $D(\mathbf{x}_k^{(i,j)}) = \rho d(\mathbf{x}_k^{(i,j)}, \mathbf{I}_k) + (1-\rho) d(\mathbf{x}_k^{(i,j)}, \mathbf{I}_k^{seg})$. ρ is a scalar tuning the balance between the original image and the segmented one. $d(\mathbf{x}_k^{(i,j)}, \mathbf{I}_k)$ and $d(\mathbf{x}_k^{(i,j)}, \mathbf{I}_k^{seg})$ are respectively computed in similar ways to $d_{i,j}$ (section III-C). By computing the distance on both the edge maps of the original image and the segmented image, the idea is that edges resulting from a potentially cluttered background can be discarded through the segmented image, whereas potential segmentation errors can be compensated by keeping the original image edges.

Also, due to potential ambiguities in the computation of the image moments and the direction of the principal axis, we actually use the distance $D_{min}(\mathbf{x}_k^{(i,j)}) = min(D(\mathbf{x}_k^{(i,j)}), D(\mathbf{x}_{k,\pi}^{(i,j)}))$, where $\mathbf{x}_{k,\pi}^{(i,j)}$ is the particle $\mathbf{x}_k^{(i,j)}$ rotated by π in the image plane. The likelihood $\pi_k^{(i,j)}$ of $\mathbf{x}_k^{(i,j)}$ for a frame $\mathbf{I}_k$, with τ a tuning parameter, is thus given by:

$$w_k^{(i,j)} \propto \pi_k^{(i,j)} = e^{-\tau^{-1} D_{min}(\mathbf{x}_k^{(i,j)})^2} \tag{9}$$

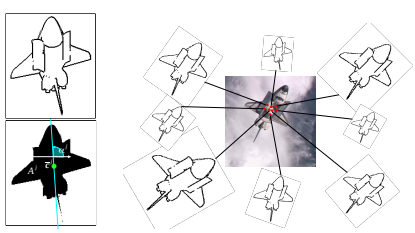


Fig. 2: Illustration of particle filtering for a reference view. The view is translated, rotated and scaled according to the particles, and the likelihoods are evaluated w.r.t the input image and its segmented frame.

D. Matching the reference views within a probabilistic framework

Once particle filtering is performed for all the reference views V^j for a frame $\mathbf{I}_k$, the goal is then to find the most likely view, while ensuring smooth transitions with respect to previous selected views. For this purpose, probabilistic graphical models can be considered. We have chosen to employ Hidden Markov Models (HMM) which define a joint distribution over the sequence of the matched model views and the sequence of observations which are the initial input images.

The sequence of the matched views as a Hidden Markov Model: a HMM supposes that the sequence of matched reference views $\{V_l\}_{l=k_0}^{k}$ follows a hidden Markov process. Besides, each observation $\mathbf{I}_l$ is assumed to only depend on V_l. Based on these assumptions, the joint probability of the sequence $\{V_l\}_{l=k_0}^{k}$ and the sequence $\{\mathbf{I}_l\}_{l=k_0}^{k}$ can be written as:

$$p(V_{k_0:k}, \mathbf{I}_{k_0:k}) = \prod_{l=k_0}^{k} p(\mathbf{I}_l|V_l) p(V_l, V_{l-1})$$

$$\text{with } p(\mathbf{I}_l|V_l = V^j) = \frac{1}{A}\sum_{i=1}^{N_j} \pi_l^{(i,j)}, \quad A = \sum_{j=1}^{N_r}\sum_{i=1}^{N_j} \pi_l^{(i,j)} \tag{10}$$

The probability $p(\mathbf{I}_l|V_l)$ refers to the observation probability of a given matched model view V_l, with V^j the view corresponding to V_l in $\{V^j\}_{j=1}^{N_r}$. $\pi_l^{(i,j)}$ is the weight of particle $\mathbf{x}_l^{(i,j)}$ and A is a normalization factor so that we deal with a probability distribution.

$p(V_l, V_{l-1})$ is the transition probability between matched views V_l and V_{l-1}. It can be determined offline by:

$$p(V_l, V_{l-1}) \propto e^{-\frac{acos(\mathbf{u}_l{}^T \mathbf{u}_{l-1})^2}{2\sigma_v{}^2}} \tag{11}$$

where $\mathbf{u}_l$ corresponds to the viewpoint vector of the matched view V_l in the set $\{V^j\}_{j=1}^{N_r}$. This viewpoint vector is related to the azimuth and elevation angles used to generate the synthetic view corresponding to V_l. σ_v a fixed parameter related to the variance of the viewpoints.

Inference of the HMM: in order to maximize equation (10) with respect to the sequence of views, in the set $\{V^j\}_{j=1}^{N_r}$ at each time step k and thus to determine V_k (the last element of the estimated sequence), we use the classical Viterbi algorithm on the whole sequence of observations until k. The resulting reference view V^{j^*} corresponding to V_k is thus chosen as the most likely one.

As an estimate of the similarity transformation parameters, we propose to consider the whole set of reference views to compute a global estimate $\widehat{\mathbf{x}}_k$, given their respective probabilities $\{p(\mathbf{I}_k|V^j)\}_{j=1}^{N_r}$ and their estimate $\{\widehat{\mathbf{x}}_k^j\}_{j=1}^{N_r}$. It gives $\widehat{\mathbf{x}}_k = \sum_{j=1}^{N_r} p(\mathbf{I}_k|V^j)\widehat{\mathbf{x}}_k^j$.

The particles $\{\mathbf{x}_k'^{(i,j)}\}_{i=1}^{N_j}$ of each V^j are reweighted with respect to $\widehat{\mathbf{x}}_k$, prior to being processed in the particle filters of the different reference views, for the next frame $\mathbf{I}_{k+1}$.

E. Refinement as graph search and pose computation

Once a certain number of frames k_F is reached, the most likely reference model view V_{k_F}, serves as a starting point of a best match search among its child views on the hierarchical view graph and among the whole set of its associated particles. More formally, if V^{j^*} denotes the

Objet	Azi. step	Elev. step	L0	L1	L2
Spot	8°	8°	2303	436	53
Atlantis	8.6°	8.6°	1765	304	44
Soyuz	5.1°	5.1°	1225	268	38

TABLE I: Parametrization of the view sphere for each object and results of the learning step, with the number of reference determined at each level L of the hierarchical view graph.

reference view corresponding to V_{k_F}, the process results in a view $V^{l_{j^*}}$ determined at the bottom level on the view graph, and in a best particle $\widehat{\mathbf{x}}_{k_F}$. With $\widehat{\mathbf{x}}_{k_F}$ and $V^{l_{j^*}}$ and using the steps described in section IV-A, we can compute a pose ${}^c\mathbf{M}_o{}^{k_F}$. This pose is finally directly used to initialize a frame-by-frame tracking algorithm [16].

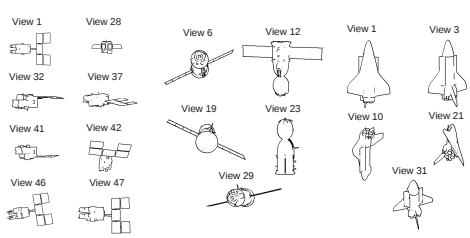
V. Results

We have evaluated and validated our technique with a focus on space objects. The algorithm has been run, using a standard laptop (2.8GHz Intel Core i7 CPU), on synthetic images featuring a Spot satellite (512×512 images are processed). Concerning real sequences, a first one shows the Soyuz TMA-12 spacecraft approaching the International Space Station (ISS). A second one features the Atlantis Space Shuttle performing its pitch maneuver towards the ISS.

A. Learning step

Table I shows the parametrization of the view sphere for the different objects, with sampling steps for both azimuth and elevation angles.

It also presents the results of the building of the hierarchical model view graph and the number of reference views obtained at each level L of the graph. For each object, we have stopped the learning process at level 2 in order to get reasonable numbers of reference views. Some examples of these reference views can be seen on Figure 3.



(a) Spot (b) Soyuz (c) Atlantis

Fig. 3: Some reference views determined at the second level of the view graph.

B. Segmentation

As explained in Section II, the pose detection process starts by segmenting the initial images of the sequence. Figures 4(a,d,g) shows the Harris corner points being tracked using the KLT tracker (red trajectories) for the different sequences. Blue and green dots represent the dots respectively classified as belonging to the foreground object or the background, spread on regular grids. We observe that the clustering process explained in section III-A is performed correctly, with very few misclassified pixels for Spot (Figure 4(a)) and Soyuz (Figure 4(d)) whereas we find more errors for Atlantis (Figure 4(g)). Figures 4(b,e,h) depicts the mean image between the current image and its homography-based compensated one, enhancing the motion of the object with respect to the background, with the zone featuring the object being blurred (with $k_H = 5$).

Finally, Figures 4(c,f,i) shows both object (colored) and the background (black) layers after the segmentation phase, which starts at $k_0 = 8$, with satisfactory results, despite the cluttered background. This segmentation step is executed in around $0.6s$ using kernel density estimation and in $0.38s$ using histograms.

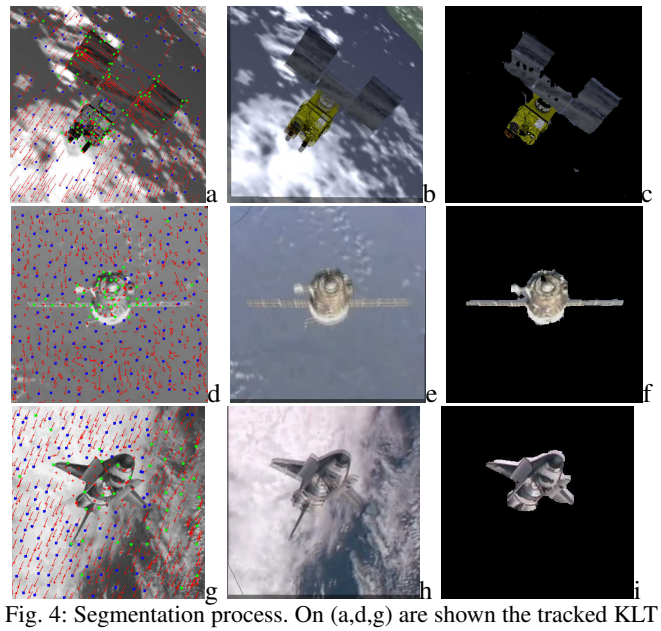


Fig. 4: Segmentation process. On (a,d,g) are shown the tracked KLT trajectory points, in green those classified as foreground points and in blue as background points. (b,e,h) represent the mean image after homography based motion compensation and (c,f,i) the resulting segmented image.

C. Results for the initial pose estimation

Several sequences, with some of them presented on Figure 4, have been processed within our initial pose estimation framework. For the considered objects, reference views collected at the second level (L2) of their respective hierarchical view graph (Figure 3) are selected to perform the matching and alignment phase, over 10 initial input images. The particle filters on the similarity transformation parameters process 100 particles in these tests.

Results for the different sequences are depicted on Figures 5- 7. The probabilistic alignment phase is represented by the superimposition of the most likely reference view. The view refinement and pose computation step, performed at frame 9, is also shown. Finally, the initialization of a frame-by-frame model-based tracking algorithm [16] is featured.

In order to show the advantage the Hidden Markov Model used to match the reference views with the input images (section IV-D), the marginal joint probabilities, provided by

the Viterbi algorithm, of the different reference views along the input sequence are plotted.

For sequences on Figure 6(a)-6(d), 7(b)-7(d), 6(f)-6(i) and 5(l)-5(n), we observe that consistent reference views are matched and realigned to the image through particle filtering. The benefit of the HMM is visible through its ability to smooth, by estimating the optimal sequence, the determination of the most likely view at each time step. For the sequence on Figures 6(a)-6(d), three reference views (views 1, 46 and 47) still have similar appearance, despite the hierarchical clustering technique described in section III-A. Thanks to the inference of the HMM, view 1 is progressively rejected, in terms of marginal joint probability, and the marginal probability of view 46 is increasing. A false positive can be observed on the initial match for the sequences on Figures 5(a)-5(d). However, the HMM rapidly discard this inconsistent view and a fair one is finally dominant. Ambiguities can also be observed for the Soyuz sequence (Figures 6(f)- 6(i)) between view 6 and view 19.

On Figures 7(b)-7(d) (see also the provided video), due to the coarse segmentation, the orientation of the object in the image is initially not proper, as seen on Figure 7(b), but through the particle filtering framework, the most likely view, which is consistent with the image, is progressively aligned, its marginal probability increasing. Without the alignment procedure, tracking then fails.

The alignment and matching step is executed in around $0.35s$ per frame. Including segmentation, the process is executed in less than $1fps$.

D. Discussion

An issue regarding our pose detection system to be discussed concerns its reliance on the segmentation process and on image moments. In this study we have dealt with cases for which moments are quite distinct. Besides, potential ambiguities in the determination of the direction of the principal axis of the object are treated by computing the distance function w.r.t both a considered particle and itself rotated by π. However, in the case of more ambiguous moments (a spherical object for instance) or on large segmentation errors, the determination of the image plane rotation could be problematic, despite our robust distance function and alignment procedure, solving the issue to a certain extent. Finally, as another issue and as suggested in the results shown above, ambiguities between different reference views may result in false positives. To improve segmentation, a solution for our applications would be to use the localization information provided by the chaser spacecraft with respect to the earth to provide prior information on the earth apparent motion. Otherwise, some priors on the apparent motion or color of the earth surface could be used. Besides, to relax the reliance on segmentation or to cope with indistinct moments, we could think of integrating in the process some local or region descriptors based learning methods.

VI. Conclusion

In this paper we propose a method to address the challenging issue of full-viewpoint detection and initial pose estimation in the case of complex poorly textured known 3D objects such as spacecrafts. The idea is to match and align synthetic views of the 3D model of the known object with successive initial frames. In order to efficiently cover the parameter space, the views are classified into a hierarchical view graph. We also take advantage of a segmentation technique, which guides the probabilistic edge-based matching process to provide a sufficiently precise pose to initialize a classical frame-by-frame tracking.

References

[1] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE PAMI*, 24(4), April 2002.

[2] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. In *IEEE PAMI*, Nov. 2001.

[3] A. Criminisi, G. Cross, A. Blake, and V. Kolmogorov. Bilayer segmentation of live video. In *IEEE CVPR*, pages 53–60, 2006.

[4] C. Cyr and B. Kimia. A similarity-based aspect-graph approach to 3d object recognition. *IJCV*, 57:5–22, 2004.

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE CVPR*, pages 886–893, 2005.

[6] B. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315:972–976, 2007.

[7] D. Gavrila and V. Philomin. Real-time object detection for smart vehicles. In *IEEE ICCV*, Vancouver, 1999.

[8] S. Hinterstoisser, V. Lepetit, S. Ilic, P. Fua, and N. Navab. Dominant orientation templates for real-time detection of texture-less objects. In *IEEE CVPR*, San Francisco, 2010.

[9] S. Holzer, S. Hinterstoisser, S. Ilic, and N. Navab. Distance transform templates for object detection and pose estimation. In *IEEE CVPR*, Miami, 2009.

[10] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *IJCV*, 29(1):5–28, Jan. 1998.

[11] V. Lepetit and P. Fua. Keypoint recognition using randomized trees. *IEEE Trans. on PAMI*, 28(9):1465–1479, 2006.

[12] J. Liebelt and C. Schmid. Multi-view object class detection with a 3d geometric model. In *IEEE CVPR*, San Francisco, June 2010.

[13] D. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3):355–394, Mar. 1987.

[14] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[15] C. Olson and D. Huttenlocher. Automatic target recognition by matching oriented edge pixels. *IEEE T. on IP*, 6(1):103–113, 1997.

[16] A. Petit, E. Marchand, and K. Kanani. A robust model-based tracker combining geometrical and color edge information. In *IEEE/RSJ IROS'2013*, Tokyo, Japan, November 2013.

[17] C. Reinbacher, M. Ruether, and H. Bischof. Pose estimation of known objects by efficient silhouette matching. In *IAPR ICPR*, 2010.

[18] J. Rodrigues, J.-S. Kim, M. Furukawa, J. Xavier, P. Aguiar, and T. Kanade. 6D pose estimation of textureless shiny objects using random ferns for bin-picking. In *IEEE IROS*, Vilamoura, 2012.

[19] Y. Sheikh, O. Javed, and T. Kanade. Background subtraction for freely moving cameras. In *IEEE ICCV*, Kyoto, Japan, 2009.

[20] M. Stark, M. Goesele, and B. Schiele. Back to the future: Learning shape models from 3d cad data. In *BMVC*, Aberystwyth, Aug. 2010.

[21] C. Stauffer and E. Grimson. Adaptive background mixture models for real-time tracking. *IEEE CVPR*, Miami, 1999.

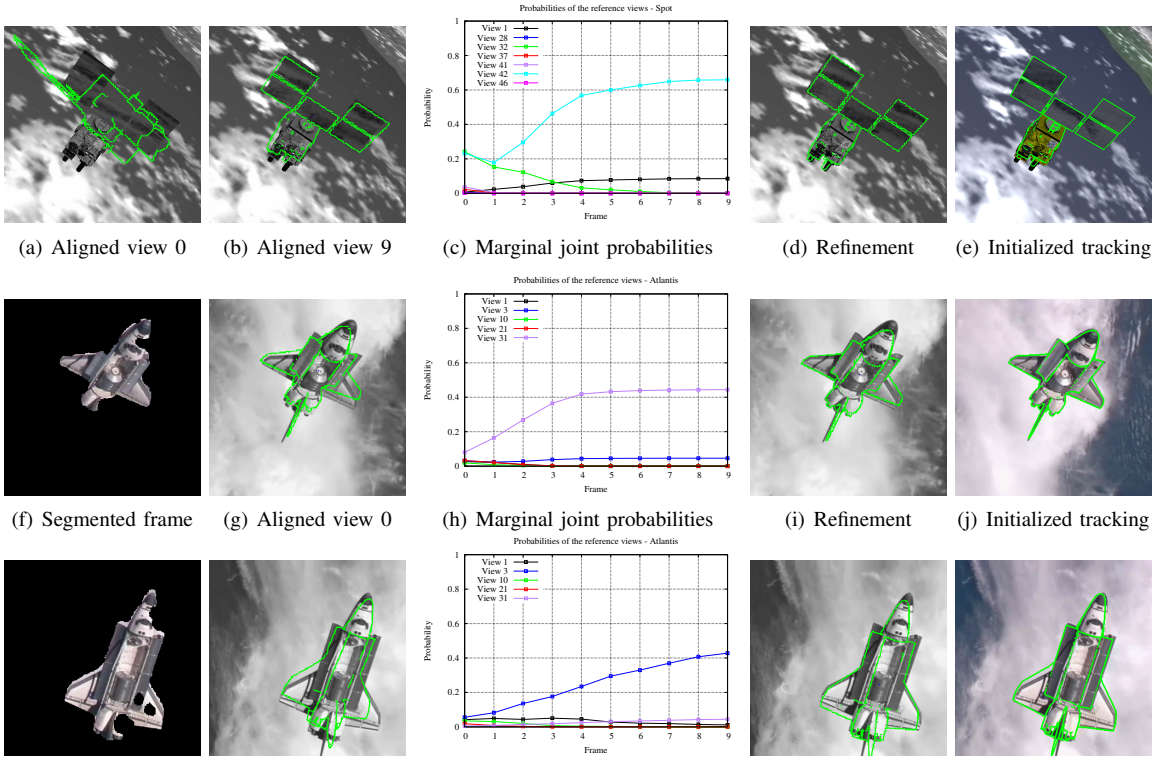
[22] C. Steger. Occlusion, clutter, and illumination invariant object recognition. *Int. Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, 34(3/A):345–350, 2002.

[23] R. Strzodka, I. Ihrke, and M. Magnor. A graphics hardware implementation of the generalized hough transform for fast object recognition, scale, and 3d pose detection. *ICIAP*, pages 188–193, 2003.

[24] A. Thomas, V. Ferrar, B. Leibe, T. Tuytelaars, B. Schiel, and L. Van Gool. Towards multi-view object class detection. In *IEEE CVPR*, pages 1589–1596, New York, 2006.

[25] A. Toshev, A. Makadia, and K. Daniilidis. Shape-based object recognition in videos using 3d synthetic object models. *IEEE CVPR*, pages 288–295, 2009.

[26] M. Ulrich, C. Wiedemann, and C. Steger. Cad-based recognition of 3d objects in monocular images. *IEEE ICRA*, pages 2090–2097, 2009.

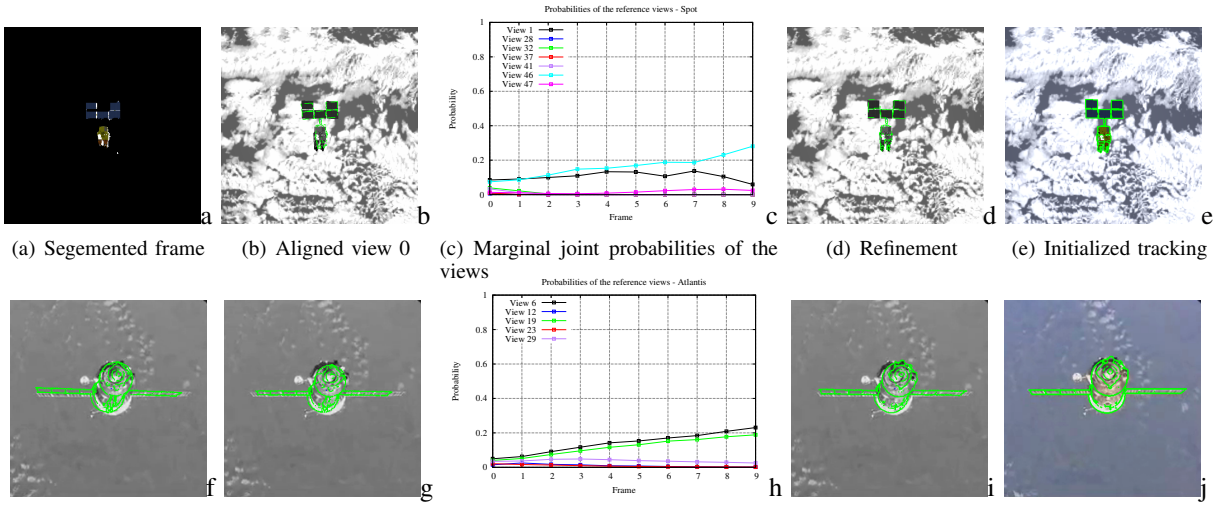


(a) Aligned view 0 (b) Aligned view 9 (c) Marginal joint probabilities (d) Refinement (e) Initialized tracking

(f) Segmented frame (g) Aligned view 0 (h) Marginal joint probabilities (i) Refinement (j) Initialized tracking

(k) Segemented frame (l) Aligned view 0 (m) Marginal joint probabilities (n) Refinement (o) Initialized tracking

Fig. 5: Matching and alignment procedure of the reference views. From the segmented frame (see 4(a) for the first row), the alignment of the most likely reference view is shown. Probabilities of the views are plotted on (c,h,m). (d,i,n) depict the pose determined after the refinement step on frame 10 by traversing through the view graph, from the most likely reference view. Finally the tracking can be initialized (e,j,o).



(a) Segemented frame (b) Aligned view 0 (c) Marginal joint probabilities of the views (d) Refinement (e) Initialized tracking

(f) Aligned view 0 (g) Aligned view 9 (h) Marginal joint probabilities (i) Refinement (j) Initialized tracking

Fig. 6: Ambiguous cases. Two similar reference views are predominant. However, thanks to the distance function and the probabilistic framework, the most consistent reference view tends to be more likely (c,h). From this view, the refinement step (d,i) enables to initialize the tracking (e,j).

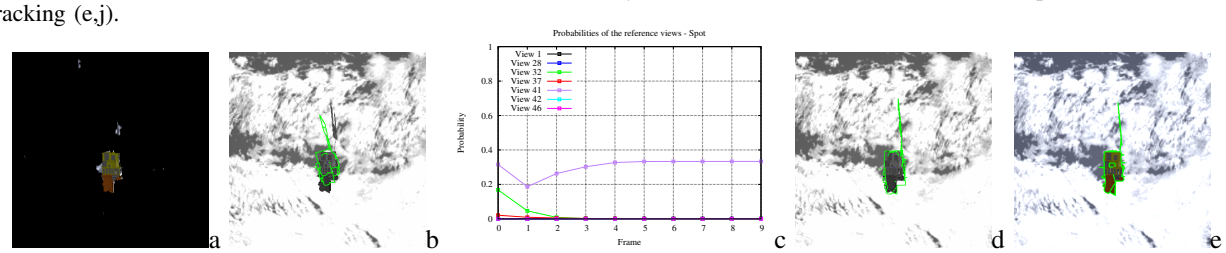


Fig. 7: Coarse segmentation. In the case of a coarse segmentation (a), particle filtering allows to correctly estimate the silhouette parameters on the next frames (b,d) and to match the proper reference view. Tracking can then be achieved (e).