

# A versatile object tracking algorithm combining Particle Filter and Generalised Hough Transform

Antoine Tran<sup>1</sup>, Antoine Manzanera<sup>2</sup>

ENSTA ParisTech U2IS/Robotics & Vision

Université de Paris-Saclay

email: antoine.[tran,manzanera]@ensta-paristech.fr

**Abstract**— This paper introduces a new object tracking method which combines two algorithms working in parallel, and based on low-level observations (colour and gradient orientation): the Generalised Hough Transform, using a pixel-based description, and the Particle Filter, using a global description. The object model is updated by combining information from a back-projection map computed from the Generalised Hough Transform, providing for every pixel the degree to which it may belong to the object, and from the Particle Filter, providing a probability density on the global object position. The proposed tracker makes the most of the two algorithms, in terms of robustness to appearance variation like scaling, rotation, non-rigid deformation or illumination changes.

**Keywords**— Object Tracking, Generalised Hough Transform, Particle Filter, Colour and Edge Features.

## I. INTRODUCTION

Object tracking is a challenging problem in computer vision, with applications in many areas like security, surveillance or human-computer interaction. Difficulties are due to appearance changes of the object and its background, motion variations or occlusion. Many methods have been proposed in the literature. We refer to the survey of Yilmaz [1] for a comprehensive coverage of the problem. In this paper, we propose an original tracking method combining two algorithms: the Particle Filter and the Generalised Hough Transform.

Isard adapted the particle filter for object tracking in his *condensation* algorithm [2]. It belongs to the family of the Bayesian filters, and is based on the estimation of the posterior distribution of the object state using a set of weighted particles. Each particle corresponds to an hypothetical state of the object (typically describing its position and velocity), while the weight represents the importance of the particle. The particle filter is composed of three steps: (1) Observation, to measure how each particle fits the model, (2) Resample, to keep the best particles, and (3) Propagation, to diffuse the set of particles according to a dynamical model. In tracking problems, the observation step is usually based on object visual representations. Different representations have been used to model the target object: Isard [2] models the borders of the object using parametric B-spline curves, Nummiaro [3] uses colour histograms. Pérez [4] splits the object in different parts, each one modelled by a colour histogram. Maggio [5] uses smoothed histograms of gradient orientation for rotation robustness purposes, combined with colour histograms to form a particle filter using two features. Brasnett [6] also uses

multiple features (colour, edge and texture), and proposes a way to adaptively change their relative weights over time, to improve the performance of his tracker. The resample step is also an important process: Arulampalam compares different resampling methods in [7]. Soto [8] proposes a method to adapt the number of particles according to the quality of the observation, and also a way to adapt the dynamical model used in the propagation step over time.

The Hough Transform was originally proposed [10] to detect lines and circles in binary images. Ballard [11] extended the algorithm to arbitrary objects, in the so-called Generalised Hough Transform (GHT). Originally, it consists in building a model using points sampled from the object boundary, and a function mapping the local orientation to the position of every point with respect to the model centre. Initially proposed as a detection algorithm, the GHT was extended to object categorisation [12]. In this last method, the representation was based on image patches built around interest points. The GHT was also used for object tracking. Duffner [15] combined the GHT with a foreground/background model, whereas Godec [13] enriched the model representation, using random forest theories. A similar model was proposed by Gall [14] for action recognition.

The method proposed in this paper combines Hough-based tracking-by-detection with global and predictive particle filtering. The interest is to exploit the complementarity of the two approaches: the GHT is less sensitive than the particle filter to strong apparent accelerations. On the other hand, the GHT lacks any predictive capability and is more sensitive to object deformations. Moreover, while the particle filter is based on a global object model, the GHT uses a pixel-based representation. Finally, even in their simplest form, both algorithms have decent tracking performance, are very versatile, and can be improved in many ways. Our main contribution is the use of the output of the two algorithms to update the object model. It is based on the back-projection map from the GHT, which has proved a powerful tool. Razavi [19] showed that the back-projection can be a base for a multi-view GHT, and can measure the similarity between detections and training examples. It was also used to update the object model, by indicating if a pixel is likely to belong to the object. Our approach is different to those proposed by Duffner or Godec [15], [13], who use segmentation algorithms to strictly determine which pixels belong to the object. Our method uses two multi-valued maps, concurrently produced by two approaches and

combined to softly evaluate the level of belonging to the object for each pixel. Combining a detector with a particle filter for object tracking has already proven effective: Breitenstein [16] proposed a multi-person tracker combining a detector (based on HOG [17] or on GHT [18]), a classifier and a particle filter. However, while the observation of his particle filter combines the detector and classifier outputs, in our case the observations are independent and only based on visual features.

This paper is organised that way: Section II focuses on the particle filter, then Section III deals with the GHT. Section IV shows how the two algorithms are combined, and Section V presents results and discusses the performance of the tracker.

## II. PARTICLE FILTER

### A. Principle

The goal of the particle filter is to estimate the object state using a set of particles. In our case, we will work with a fixed number of particles:  $N_p = 500$ . Each particle represents an hypothesis of the state, and is associated with a weight, whose role is to evaluate the adequation between the hypothesis and the model. In our experiments the position of the object is given by its bounding box. We then used the following four-dimensional state space:

- $(x, y)$ : coordinates of the bounding box centre
- $(w, h)$ : width and height of the bounding box

We will not consider the orientation of the object. The state of a particle  $i$  at frame  $t$  will be denoted by  $\mathbf{X}_t^i = \{x_t^i, y_t^i, w_t^i, h_t^i\}$ . The three steps of the particle filter are defined as follows, for each frame:

- **Observation:** for each particle with state  $\mathbf{X}_t^i$ , we associate a colour-based weight  $\omega_t^i$ , obtained by comparing the colour histograms  $\hat{H}_t$  of the object model and  $H_t^i$  of the image at frame  $t$  within the bounding box given by  $\mathbf{X}_t^i$ . The RGB colour space is used, using  $n_c \times n_c \times n_c$  bins (with  $n_c = 8$ ), and the histogram distance is based on the Bhattacharyya coefficient:

$$d_c(\hat{H}_t, H_t^i) = 1 - \sum_{u \in \{1, \dots, n_c\}^3} \sqrt{\hat{H}_t(u) \cdot H_t^i(u)} \quad (1)$$

and then, as proposed by Nummiaro [3], the colour weight is set as:

$$\omega_t^i = \exp[-\lambda \cdot (d_c(\hat{H}_t, H_t^i))^2] \quad (2)$$

with  $\lambda = 50.0$ , chosen experimentally. The use of colour histogram has the advantage to be robust to scaling and rotations issues.

From the observations, the global state of the object is estimated as:

$$\hat{\mathbf{X}}_t = \sum_{i=1}^{N_p} \mathbf{X}_t^i \cdot \omega_t^i \quad (3)$$

with  $N_p$  the number of particles.

- **Resampling:** Arumpamlam proposed [7] different methods to resample the particles. For the sake of simplicity,

the particle set was generated using a multivariate Gaussian Process. For each particle  $\mathbf{X}_t^i = \{x_t^i, y_t^i, w_t^i, h_t^i\}$ , the multivariate Gaussian Process is done using four independent Gaussian Process :

- For the spatial states  $(x_t^i, y_t^i)$ , we use two independent Gaussian of parameters  $(\hat{x}_{t-1}, c \cdot \hat{w}_{t-1})$  and  $(\hat{y}_{t-1}, c \cdot \hat{h}_{t-1})$  ( $\hat{x}_{t-1}$  and  $\hat{y}_{t-1}$  corresponding to the spatial estimation from the previous frame, while  $\hat{w}_{t-1}$  and  $\hat{h}_{t-1}$  corresponding to the width and height estimated in the previous frame). We choose  $c = \frac{1}{2 \cdot (2 \cdot \sqrt{2} \cdot \ln 2)}$ , because  $(2 \cdot \sqrt{2} \cdot \ln 2)$  is the coefficient linking the *full width at half maximum* (FWHM) of a Gaussian and its standard deviation  $\sigma$ :  $FWHM = (2 \cdot \sqrt{2} \cdot \ln 2) \cdot \sigma$ . To goal is to ensure that most of the particle are inside the last bounding box estimated.
- For the scale states, given the estimation from the last frame  $(\hat{w}_{t-1}, \hat{h}_{t-1})$ , we are generating a multiplicative coefficient  $\beta$  using a centralised Gaussian with a standard deviation of 0.05. Then  $w_t^i = \beta \cdot \hat{w}_{t-1}$  and  $h_t^i = \beta \cdot \hat{h}_{t-1}$ .
- **Propagation:** by denoting  $\mathbf{A}_t$  a transition vector modelling a dynamical model, and  $\mathbf{W}_t$  a vector modelling a Gaussian noise, the propagation model to update the particle state from  $\mathbf{X}_{t+1}^i$  to  $\mathbf{X}_t^i$  is given by:

$$\mathbf{X}_{t+1}^i = \mathbf{A}_t^i \cdot \mathbf{X}_t^i + \mathbf{W}_t^i \quad (4)$$

The first matrix is used to describe the dynamical changes of the model (velocity, scale change), while the second is used to diffuse the particle set in the state space, but given our resampling model, the operation of diffusion has already been done before. In the experiments shown in this paper, the transition matrix  $\mathbf{A}_t^i = \mathbf{I}_4$  is the identity, and  $\mathbf{W}_t^i$  is a null-vector.

## III. GENERALISED HOUGH TRANSFORM

### A. Construction of the model

The GHT model is based on a so-called *R-table*  $\mathbf{T}$ , built using a reference point  $R$  (the centre of mass of the object for example) and the gradient map. The R-table is indexed by orientation values, and each entry contains a list of couples  $(\vec{u}, \omega)$ , where  $\vec{u}$  is a displacement, and  $\omega$  its weight. Let  $\mathbf{M}$  be the magnitude map, and  $\mathbf{O}$  the quantified orientation of the gradient (with  $n_o = 8$  bins).

For all pixels  $P$  such that  $\mathbf{M}(P)$  is above a certain threshold (set to 10 in our experiments), let  $\vec{u}_P = \overline{PR}$  be the displacement of  $P$  relative to the centre, and  $\Theta_P$  the quantified orientation of the gradient at  $P$ , we add in  $\mathbf{T}(\Theta_P)$  the couple  $(\vec{u}_P, \mathbf{M}(P))$ . Like Duffner in [15], for each table indexed by its orientation, we only store the  $D$  displacements with the highest weights (in our case  $D = 50$ ). It has the advantage to limit the memory use and also significantly improve the speed (otherwise, each table can contain hundreds of displacements).

## B. Detection

Given an image, the goal is to detect an object using the R-Table. The decision is made using an accumulation map (the Hough transform), denoted  $\mathbf{HT}$ . The magnitude map  $\mathbf{M}$  and the quantified orientation map  $\mathbf{O}$  are computed. For each pixel  $P$  with  $\mathbf{M}(P)$  above the threshold, and orientation  $\Theta_P$ , we consider all the  $n_{\Theta_P}$  entries  $\{(\vec{u}_j, \omega_j)\}_{j \leq n_{\Theta_P}}$  of the R-Table  $\mathbf{T}$  at the index  $\Theta_P$ . For each couple  $j$ , the accumulator  $\mathbf{HT}(P + \vec{u}_j)$  is incremented by the quantity  $\omega_j$ . The detector output for the best object position (if needed) may be given by  $P_{max} = \arg \max_P \mathbf{HT}(P)$ .

## C. Back-projection map

Under this form, the GHT has some weaknesses: in spite of a certain robustness to small deformations, it is very sensitive to scaling or rotating. Moreover, in a tracking context, it lacks a model updating process. Godec and Duffner proposed in [13] and [15] to use a back-projection map as a support for updating. In their case, after having detected the pixel  $P_{max}$  with the highest value in the accumulation map, they build the back-projection map by determining which pixels voted for  $P_{max}$ , for they are the more likely to belong to the object. Our approach uses a "softer" version of the back-projection map: each voting pixel is attributed the value of its best pixel in the accumulation map, more precisely:

$$\mathbf{BP}(P) = \max_{j \leq n_{\Theta_P}} \mathbf{HT}(P + u_j) \quad (5)$$

The map is then soft and denser compared to the one proposed by Duffner or Godec, however, a lot of pixels from the background may have a non-zero value. A threshold is not satisfying, since background regions similar to the object may have good votes in bad detection situation. Godec proposed to use the GrabCut segmentation algorithm from [20] to extract the foreground, while Duffner used a probability model to segment the foreground from the background. In our case, we do not use hard segmentation but combine the rough back-projection map with the output of the particle filter.

## IV. FUSION

### A. Combine the outputs

The goal is to exploit the output data from both the particle filter and the back-projection map to determine which pixels are more likely to belong to the object, and use them to update the two models. As written in Section III, the back-projection map provides local confidence measure related to the GHT model (the R-table). However, our "soft" back-projection map may also give good score to pixels far from the object. We then combine this value with the information provided by the particle filter. Let the observation map of the particle filter be the image  $\mathbf{OP}$  defined from the position states of the particles, such that for all  $i \leq N_p$ ,  $\mathbf{OP}(x_t^i, y_t^i) = \omega_t^i$ , and  $\mathbf{OP} = 0$  elsewhere. Multiply the back-projection map  $\mathbf{BP}$  with  $\mathbf{OP}$  would be the natural combination. However, the latter is only

a sparse estimation of the density map, and so, we need to smooth it. While fitting it with a joint probability distribution can give good results, it relies on a global estimation scheme that may be time-consuming. Instead we spread it using a morphological dilation, with an elliptic structuring element  $\mathbf{SE}$  of size  $(\frac{\hat{w}}{N_p^{\frac{1}{3}}}, \frac{\hat{h}}{N_p^{\frac{1}{3}}})$ , with  $\hat{w}$  and  $\hat{h}$  the estimated width and height of the object. Finally the fusion map  $\mathbf{B}$  is defined as follows:

$$\mathbf{B} = \mathbf{BP} \cdot \delta_{\mathbf{SE}}(\mathbf{OP}) \quad (6)$$

$\mathbf{B}$  is then normalised to  $[0, 1]$  and all values inferior to  $\epsilon = 0.05$  are set to 0 (corresponding to discard pixels whose importance in the fusion map are less than 5% of the maximum value).

### B. Update models

Given the normalised fusion map  $\mathbf{B}$ , the two models are updated as follows:

- For the particle filter, we evaluate the colour histogram (denoted  $H_t^{\mathbf{B}}$ ) from the sub-image defined by the bounding box of all pixels  $P$  such that  $\mathbf{B}(P) > 0$ . Then, we update the model according to this formula:

$$\hat{H}_{t+1} = \alpha \cdot \hat{H}_t + (1 - \alpha) \cdot H_t^{\mathbf{B}} \quad (7)$$

with  $\alpha = 0.9$ .

- to update the R-table, the magnitude map  $\mathbf{M}$  and quantified orientation map  $\mathbf{O}$  of the gradient are computed just like in the construction of the R-table. Then, let  $R$  be the centre of mass of  $\mathbf{B}$ . For all non-zero pixels  $P$  from  $\mathbf{B}$ , we consider its quantified orientation  $\Theta_P$ , and its fusion map value  $\mathbf{B}(P)$ :
  - if  $\vec{u}_P = \vec{PR}$  is not in the table  $\mathbf{T}(\Theta_P)$ , then we add the entry with a weight  $M(P)$ .
  - if it is already in, we update its weight  $\omega_{\Theta_P, \vec{u}_P}$  using both the magnitude  $\mathbf{M}(P)$  and the fusion map value  $\mathbf{B}(P)$ :

$$\omega_{\Theta_P, \vec{u}_P} \leftarrow \frac{\alpha \cdot \omega_{\Theta_P, \vec{u}_P} + (1 - \alpha) \cdot \mathbf{B}(P) \cdot \mathbf{M}(P)}{2} \quad (8)$$

To avoid overloading the memory and slowing down the process, the R-Table size is still limited to the  $D = 50$  displacements with highest weights for each  $\Theta_P$  index.

### C. Tracking decision

The final step is to provide the estimated state of the tracked object. In one hand, the particle filter gives an estimation of the object state. In the other hand, following the idea that a pixel in back-projection map is as high as it is more likely to belong to the object, we will use this back-projection map to estimate the state. Like the updating process, the use of the fusion map  $\mathbf{B}$  is the natural combination. So, we decide to estimate the state of the tracked object by the bounding box of all non-zero pixels of  $\mathbf{B}$ .

The Figure 5 synthesises the complete tracking process as presented in this section and the two previous ones.

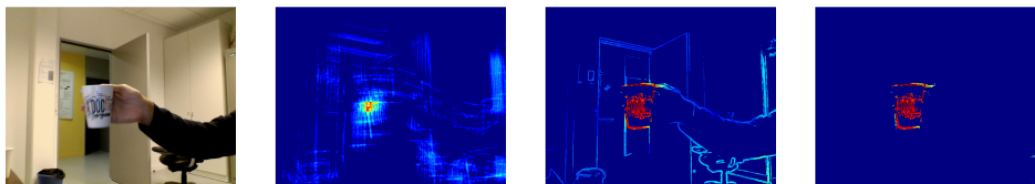


Fig. 1. Generalised Hough Transform used alone. After several frames, the peak of the accumulation map (second image) becomes less visible. The third image represents the back-projection map, and the third map shows the thresholded version (using Otsu’s method explained in [21]). Some pixels from the background, in the bottom-right corner, are still visible

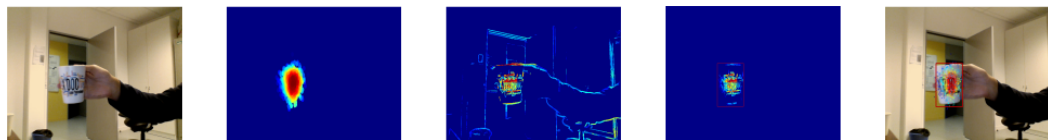


Fig. 3. Success case for the combined tracker for position and scale changes. Alone, the particle filter was not able to detect the scale change. But with the back-projection map (the third image), pixels from the border of the mug, having high value, are still present in the fusion map (the fourth image), making success estimation of the scale and the position.

## V. EXPERIMENTS

We have performed a series of preliminary experiments, whose purpose is to show that the combination of the two algorithms provide significant improvement with respect to each one used individually. In the working sequence, the objective is to track the mug. Basically, the movement is composed of a translation and a scaling, followed by a descaling. Both algorithms are used in their simplest possible form.

On the one hand, the GHT is rapidly turning bad (See Figure 1): the accumulation map shows multiple local maxima with high values, and the lack of information from the previous frames (the GHT is just a detector) does not provide a way to select the best maximum. Moreover, without other information (such as the classification model used by Duffner [15]), or an effective segmentation algorithm (like GrabCut used by Godec [13]), extracting the object boundaries is a difficult task.

On the other hand, while the Particle Filter, in the simple form proposed in our paper, gives decent results in the spatial estimation, the absence of propagation model makes it impossible to detect scale change (see Figure 2). However, it is possible to improve the tracking using different solutions: using a propagative model taking in consideration the motion can be a solution.

By combining the two algorithm, the tracker turns more effective to scaling issues (see Figure 3). However, it possesses the same weakness as the simplified Particle Filter: without a strong model of resampling and propagation, it is difficult to follow an abrupt variation of scale, and also possess the inherent problem of the GHT: without scale or orientation parameters, the detectors fails in rotation or scaling task (see Figure 4).



Fig. 2. Particle Filter used alone. Each coloured pixel corresponds to the projection of the particle in the spatial space. Its colour represents its weight (Equation 2): red corresponding to high weight, blue to low weight. The rectangle is the state estimated with the Particle Filter, processed alone, using the equation 3. Its colour corresponds to the weight of the observation, for its state.

## VI. CONCLUSION

We have presented an original and versatile object tracking algorithm combining Hough based tracking-by-detection and Particle Filtering. These two algorithms present fundamental complementarities: the GHT is insensitive to the acceleration while the Particle Filter has a strong predictive basis. Furthermore they operate on two different levels: pixel level for the GHT, and global level for the Particle Filter. The main rationale for this approach is in the hypothesis that the combination of two complementary methods under their most basic form provide more benefits that sophisticating each approach individually. Our contribution is focused on the method to fuse the outputs of the algorithms: combining the back-projection map from the GHT and the estimation states given by the Particle Filter provides an effective method to evaluate the level of belonging to the object at the pixel level. Our experiments showed that working with very light versions of the two algorithms, with a small R-Table for the GHT, and a fixed and small number of particles, the combined

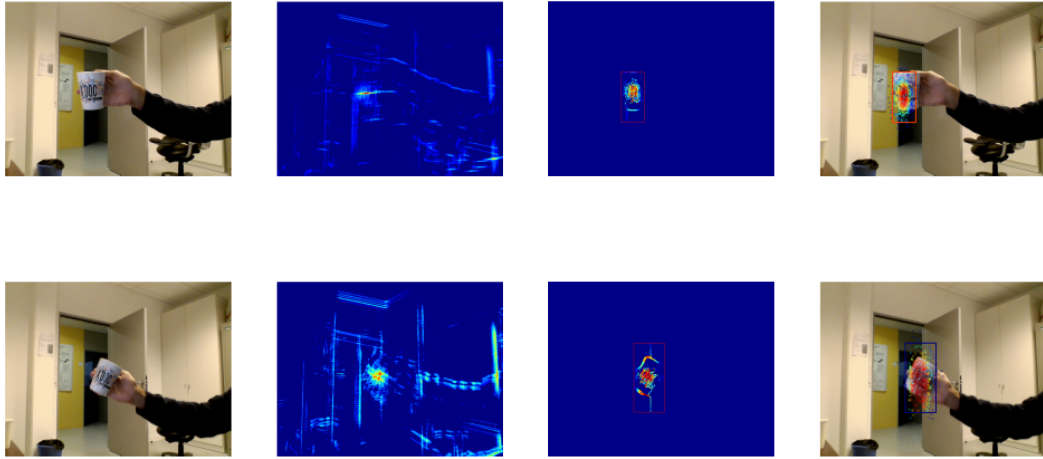


Fig. 4. Fail case for the combined tracker. In the first line, after several frame of fast descaling, the tracker is no longer able to estimate the scale. In the second line, after several frames of rotation, the GHT causes the fail: the second image shows different maximums at different places, and near of the center of the mug, several local maximums exists, perturbing the updating process, by adding data from the background.

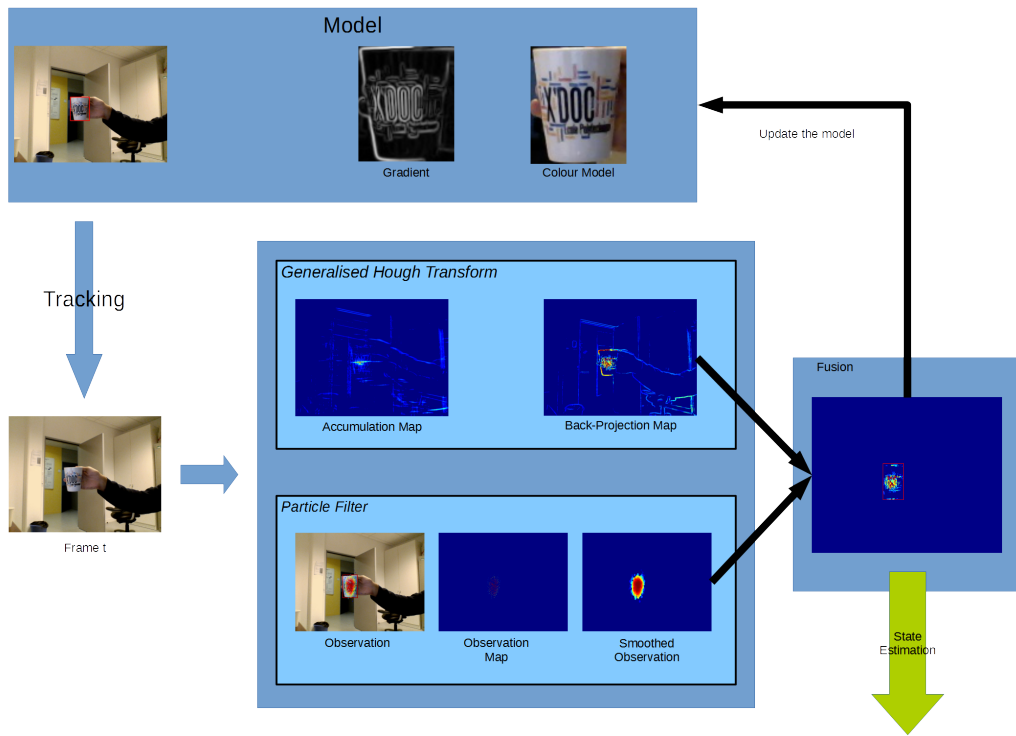


Fig. 5. Summary of the tracker. At the top, the object is represented by its bounding box (the red rectangle) described using two features: gradient and colour. The goal is to estimate the state vector at each frame of the sequence. The GHT and the Particle Filter are operating independently: the first one generates the back-projection map, the second one the (spreaded) observation map. The two output are combined (Fusion step), while the tracker output is given by the Particle Filter.

tracker provides significant improvement compared to the two algorithms used independently.

In the close future, we will focus on the extension of the state space to orientation parameter. We will also improve the fusion process, by proposing alternate combination between the two confidence maps. At middle-term, we will consider the

different enhancement possibilities, at the visual representation level (e.g. use of different features) and on the algorithm itself (e.g. by implementing stronger resampling and propagation processes), with the objective to maintain the simplicity and lightness of the combined tracker. Quantitative evaluation of enhanced versions will be performed using academic datasets.

At long-term, the goal will be to extend the tracker to the multi-target context [16].

#### ACKNOWLEDGMENT

We gratefully acknowledge funding support by the French General Directorate for Armament (DGA).

#### REFERENCES

- [1] Yilmaz, Alper and Javed, Omar and Shah, Mubarak. Object tracking: A survey. *Acm computing surveys (CSUR)*, 38(4), 2006.
- [2] Isard, Michael and Blake, Andrew. Condensation: conditional density propagation for visual tracking. *International journal of computer vision*, 29(1): 5–28, 1998
- [3] Nummiaro, Katja and Koller-Meier, Esther and Van Gool, Luc and others. A color-based particle filter. *First International Workshop on Generative-Model-Based Vision*, 1: 53–60, 2002
- [4] Pérez, Patrick and Hue, Carine and Vermaak, Jaco and Gangnet, Michel. Color-based probabilistic tracking. *Computer Vision?ECCV 2002*, 661–675, 2002
- [5] Maggio, Emilio and Smeraldi, Fabrizio and Cavallaro, Andrea. Combining Colour and Orientation for Adaptive Particle Filter-based Tracking. *Proceedings of the British Machine Vision Conference*, 1: 53–60, 2005
- [6] Brasnett, Paul and Mihaylova, Lyudmila and Bull, David and Canagarajah, Nishan. Sequential Monte Carlo tracking by fusing multiple cues in video sequences. *Image and Vision Computing*, 25(8): 1217–1227, 2007
- [7] Arulampalam, M Sanjeev and Maskell, Simon and Gordon, Neil and Clapp, Tim. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *Signal Processing, IEEE Transactions on*, 50(2): 174–188, 2002
- [8] Arulampalam, M Sanjeev and Maskell, Simon and Gordon, Neil and Clapp, Tim. Self adaptive particle filter. *IJCAI*, 1398–1406, 2005
- [9] Yang, Changjiang and Duraiswami, Ramani and Davis, Larry. Fast multiple object tracking via a hierarchical particle filter. *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, 212–219, 2005
- [10] Duda, Richard O and Hart, Peter E. Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1): 11–15, 1972
- [11] Ballard, Dana H. Generalizing the Hough transform to detect arbitrary shapes. *Pattern recognition*, 13(2): 111–122, 1981
- [12] Leibe, Bastian and Leonardis, Ales and Schiele, Bernt. Combined object categorization and segmentation with an implicit shape model. *Workshop on statistical learning in computer vision, ECCV*, 2(5), 2005
- [13] Godec, Martin and Roth, Peter M and Bischof, Horst. Hough-based tracking of non-rigid objects. *Computer Vision and Image Understanding*, 117(10): 1245–1256, 2013
- [14] Gall, Juergen and Lempitsky, Victor. Class-specific hough forests for object detection. *Decision Forests for Computer Vision and Medical Image Analysis*, 143–157, 2013
- [15] Duffner, Stefan and Garcia, Christophe. PixelTrack: a fast adaptive algorithm for tracking non-rigid objects. *Computer Vision (ICCV), 2013 IEEE International Conference on*, 2480–2487, 2013
- [16] Breitenstein, Michael D and Reichlin, Fabian and Leibe, Bastian and Koller-Meier, Esther and Van Gool, Luc. Robust tracking-by-detection using a detector confidence particle filter. *Computer Vision, 2009 IEEE 12th International Conference on*, 1515–1522, 2009
- [17] Dalal, Navneet and Triggs, Bill. Histograms of oriented gradients for human detection. *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*: 886–893, 2005
- [18] Leibe, Bastian and Leonardis, Aleš and Schiele, Bernt. Robust object detection with interleaved categorization and segmentation. *International journal of computer vision*, 77(1-3): 259–289, 2008
- [19] Razavi, Nima and Gall, Juergen and Van Gool, Luc. Backprojection revisited: Scalable multi-view object detection and similarity metrics for detections. *Computer Vision–ECCV 2010*, : 620–633, 2010
- [20] Rother, Carsten and Kolmogorov, Vladimir and Blake, Andrew. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (TOG)* 23(3): 309–314, 2004
- [21] Otsu, Nobuyuki. A threshold selection method from gray-level histograms. *Automatica* 11(285-296): 23–27, 1975