Real-time Hierarchical Bayesian Data Fusion for Vision-based Target Tracking with Unmanned Aerial Platforms

Andrés F. Echeverri, Henry Medeiros, Ryan Walsh, Yevgeniy Reznichenko and Richard Povinelli

Abstract— **Data fusion algorithms make it possible to aggregate information from multiple data sources in order to increase the robustness and accuracy of robotic vision systems. While Bayesian fusion methods are common in general applications involving multiple sensors, the computer vision field has largely relegated this approach. In particular, most object following algorithms tend to employ a fixed set of features computed by specialized algorithms, and therefore lack flexibility. In this work, we propose a general hierarchical Bayesian data fusion framework that allows any number of vision-based tracking algorithms to cooperate in the task of estimating the target position. The framework is adaptive in the sense that it responds to variations in the reliability of each individual tracker as estimated by its local statistics as well as by the overall consensus among the trackers. The proposed approach was validated in simulated experiments as well as in two robotic platforms and the experimental results confirm that it can significantly improve the performance of individual trackers.**

I. INTRODUCTION

Although recent advances in visual tracking have enabled the emergence of new robotic platforms capable of following objects relatively well, the lack of robustness in a wide range of real-world scenarios is still a significant limitation of existing tracking algorithms. This is in part due to problems that make it difficult to associate images of a target in consecutive video frames under challenging conditions. These problems include: fast motion of the object and/or camera, change of pose or orientation, illumination variation, occlusion, scale change, clutter, and the presence of similar objects in the scene. These common disturbances make tracking with any single approach unreliable in many short term scenarios and nearly impossible in most long term applications. While a specific algorithm might perform well for certain scenarios, it might not work for others. Based on this paradigm, this paper proposes a general tracking approach that fuses the results generated by several algorithms into a unique output. Fusion is done at the bounding box level, where measurements provided by each of the individual tracking algorithms are processed as sensor measurements.

In the literature, sensor fusion is also known as multi-sensor data fusion, data fusion, or combination of multi-sensor information. All of these methods aim for the same goal of creating a synergy of information from several sources [1]. The overall uncertainty of a system that uses only one sensor to observe a physical phenomenon is generally determined by the uncertainty of that particular sensor. Without relying on additional sensors, opportunities to reduce this uncertainty are limited. Furthermore, the failure of the sensor leads to the failure of the entire system. Different types of sensors may produce a spectrum of information, which not only may provide varying accuracy levels but also the ability to operate under different, sometimes complementary, conditions [2].

There are a number of benefits to data fusion. First, with redundant information, the uncertainty can be reduced to increase the overall accuracy of the system. Second, if a sensor is deemed to be faulty, another sensor might compensate for that fault. Furthermore, while one algorithm could be more robust, say, to scale changes, another could be more robust to outlying measurements; a cooperative approach incorporates the best aspects of each method.

In our work, we use the output of vision-based object trackers as our sensors. Due to the substitutable nature of our fusion ensemble, however, many additional data sources such infrared trackers can be incorporated as long as they generate information about the position of the target.

II. RELATED WORK

The first adaptive data fusion methods have been proposed in the 1960s [3], but it was not until the early 1990s that the concept started to be fully explored [4], laying the foundation for adaptive Bayesian approaches using the Kalman filter (KF) and its variations [1], [5], [6], [7] such as the more recent Unscented Kalman Filter (UKF) that uses multiple fading factors-based gain correction [8]. With the recent growth in computational performance, more robust approaches based on the Particle filter (PF) began to emerge [9]. However, both KFs and PFs are known to be susceptible to outliers, and recent studies have tried to solve this problem by introducing extra mechanisms to improve overall robustness [10], [11], [12]. More complex and time consuming algorithms have gone further by considering not only outliers, but also the type of sensor fault in order to resolve this shortcoming [13].

An adaptive fusion approach with a hierarchical architecture was recently proposed that not only adapts but also encodes information from the performance of the sensors [14]. Although that approach is widely used for model regression and classification, training could leave unexplored regions, causing the resulting output to suffer from outlying data. In addition, depending on the selection of experts, the

The authors are with the Department of Electrical and Computer Engineering, Marquette University, Milwaukee, WI, USA {andres.echeverriguevara}, {henry.medeiros}, {ryan.w.walsh}, {yevgeniy.reznichenko}. {richard.povinelli} @marquette.edu

gating network and the inference model, the overall system may no be applicable in real-time applications [15].

While adaptive data fusion has been well studied and established for multi-sensor measurements in general [16], researchers in the computer vision community tend to rely on the computation of multiple pre-define fixed image features to incorporate different image characteristics into tracking algorithms. Methods such as PROST [17], VTD [18], CMT [19], Struck [20], or the well known TLD [21] and its variants [22], [23] fit this framework. However, the aforementioned algorithms provide limited mechanisms to incorporate multiple and complementary feature extraction methods, thereby restricting their practical applicability.

Some of the latest visual tracking fusion approaches suggest fusion at the bounding box level [24], where information such as pixel coordinates are readily available. However, to achieve such fusion, offline training and weight finding must be carried out. This is achieved using ground truth (GT) information as well as performance metrics of the dataset used to train the algorithms. More general fusion approaches have been recently proposed, most of which rely on Sequential Monte Carlo Bayesian methods such as PFs [25], [26], [27]. When compared to KFs, however, PFs are computationally demanding as they tend to require a large number of particles to provide adequate robustness. They are hence too computationally expensive for real-time control applications, and are not popular particularly in applications that involve moderately high dimensional state spaces.

This work aims to create a general Bayesian approach for real-time applications in robotic platforms. The proposed method processes the bounding boxes of the trackers/detectors as sensor measurements. This framework is similar in spirit to a bank of KFs and shares some of the characteristics of the aforecited mixture of experts methods. Furthermore, this scheme addresses some common problems such as data imperfection, outliers and spurious data, measurement delays, static vs. dynamic phenomena, and others discussed in [28]. Our method was tested in simulated signals and two different robotics platforms: An UAV system and a pan-tilt system. Both are capable of following a target.

While similar approaches have used vision-based trackers to control a small UAV in [29], [30] and [23], previous works did not consider the fusion of several methods at a bounding box level to improve reliability over longer time spans. In addition, unlike previous works that explored the topic of hierarchical data fusion [24], [31], [**?**], [27], our framework incorporates a Bayesian confidence estimation and majority voting scheme to track targets in real time and use this information to control a UAV.

III. System description

To avoid confusion, all the visual trackers used in this work that produce a bounding box such as DSSTtld [23], CMT [19], or Struck [20] will be called detectors from this point forward. These algorithms are processed as sensors that generate measurements. The method proposed in this work, which we call Hierarchical Adaptive Bayesian Data Fusion (HAB-DF), is the main tracker that processes such measurements.

The approach proposed in this paper is a variation of the framework commonly known as mixture of experts [15], which are organized in levels or hierarchies that converge in a gating network. This work substitutes that gating network with a Bayesian approach that adapts its parameters online. Therefore, no training is necessary. In addition, this method is organized in a two-level hierarchy: the experts and the fusion center. Each expert module, K_i, $i = 1,...,n$, works asynchronously from the other modules. Usually, a bank of estimators is applied when the sensors differ in model, as each suffers from different failure types. In this particular case, the experts are KFs, inspired in part by [32] and [14]. Figure 1 shows a representation of this approach.

In the hierarchical model, each expert is equipped with an outlier detection mechanism that calculates a reliability score. The fusion center merges the outputs of each expert through a weighted majority voting scheme.

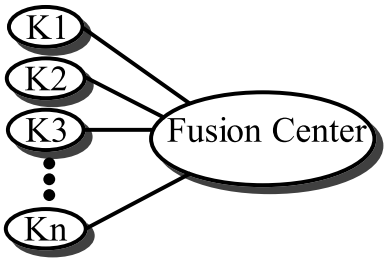


Fig. 1: Hierarchical Adaptive Bayesian Data Fusion approach. The first level of the hierarchy consists of experts that provide a local estimate to the fusion center. The second level is the fusion center.

A. Bayesian State Space Model

Our model is based on a linear Kalman filter [33] in which the state vector is given by $\mathbf{x} = [u\ v\ h\ w\ \dot{u}\ \dot{v}\ \dot{h}\ \dot{w}]$, where u, v are the pixel coordinates of the center of the target, h and w are its height and width, respectively. $\dot{u}\ \dot{v}\ \dot{h}\ \dot{w}$ are the velocities in each dimension. In this work, we adopt a random acceleration model. The object tracking system is then represented as follows:

$$\mathbf{x}(t) = A\mathbf{x}(t-1) + B\mathbf{u}(t) + \mathbf{w}(t) \quad (1)$$

$$\mathbf{y}(t) = C\mathbf{x}(t) + \mathbf{v}(t) \quad (2)$$

where Eq. (1) represents the system dynamics, including the state transition matrix A, the influence of the control action B and the process noise $\mathbf{w}$. Eq. (2) is the measurement model, which includes the observation matrix C and the measurement noise $\mathbf{v}$. The process noise and measurement noise are assumed to be white and Gaussian, with variances R_{ww} and R_{vv} respectively. That is, $\mathbf{w} \sim \mathcal{N}(0, R_{ww})$ and $\mathbf{v} \sim \mathcal{N}(0, R_{vv})$.

B. Hierarchical Adaptive Bayesian Data Fusion

We employ two main strategies to reduce the data fusion uncertainty. The first strategy is concerned with the reliability

of the measurements generated by individual detectors, and it provides a local estimate of the uncertainty based on the Mahalanobis distance [34]. The second strategy is a global approach based on weighted majority voting. As previously mentioned, the overall method is divided into a two-level hierarchy: experts and the fusion center. While each expert uses position and speed for accuracy, the fusion center only fuses direct measurements such as position, but still predicts speeds for better results in subsequent frames. Furthermore, this concept is not limited to KFs. Any Bayesian estimator can be used to accomplish fusion. Nevertheless, KFs are known for being efficient, fast, and ideal for real-time applications.

C. Local Expert Weighting

Without additional strategies to pre-process the measurements, KFs are generally not robust to outliers. Several works have been proposed to solve this problem [11], [35], [36]. The Mahalanobis distance (MD) alleviates this issue by providing a measure of how much a predicted value differs from its expected distribution.

Outliers occur due to modeling uncertainties, incorrect process/measurement noise covariances selection, and other external disturbances. If the estimation error (the difference between the real state and the estimated state) of the KF is beyond a certain threshold, the MD can penalize the expert as being in failure or abnormal mode. Similarly, one can use the predicted measurement to determine outliers. This error is then defined as follows: given a measurement $\mathbf{y} = [y_1\ y_2\ ...\ y_N]^T$, the MD from this measurement to a predicted distribution with mean $\mu = [\mu_1\ \mu_2\ ...\ \mu_N]^T$ and covariance matrix C is given by

$$M(\mathbf{y}) = \sqrt{(\mathbf{y}-\mu)^T C^{-1}(\mathbf{y}-\mu)} \tag{3}$$

Since each expert is equipped with its own MD calculation, an approximate version is used [37]:

$$M(\mathbf{y}) \approx \sum_{i=1}^{N} \left(\frac{(y_i-\mu_i)^2}{C_i} \right)^{1/2} \tag{4}$$

where C_i is the i^{th} value along the diagonal of the innovation covariance C. Eq. (4) decreases the computational burden if a considerable number of experts is needed. Usually, an estimator can be penalized if the MD is beyond a certain threshold. However, doing so yields hard transitions. To allow for smoother transitions, a sigmoid function has been employed [38]:

$$w_m = \frac{1}{1+e^{(-M(\mathbf{y})+\xi)}} \tag{5}$$

where ξ is a value chosen using the χ^2 distribution based on the number of degrees of freedom (DOF) of the system and the desired confidence level. Eq. (5) then allows outliers to be implicitly discarded since w_m represents a local weighting function that reflects the confidence that should be given to that particular measurement.

D. Majority Voting

Weighted voting is one of the simplest approaches for fusing information [3]. There are many ways to determine the weights in such a voting scheme. The method chosen for this application combines the output of multiple detectors, which, in this case, corresponds to the information from multiple bounding boxes. The first step in this method is to calculate the pairwise Euclidean distance between bounding boxes

$$\begin{gathered} d_i(\mathbf{y}^{(1)}, \mathbf{y}^{(2)}) = \left\| \mathbf{y}^{(1)} - \mathbf{y}^{(2)} \right\| \\ i = 1,2,3,\cdots,n \end{gathered} \tag{6}$$

where $\mathbf{y}^{(1)}$ and $\mathbf{y}^{(2)}$ are vectors that represent the coordinates and the size of the bounding boxes for two different detectors D_i and D_j. A measure of agreement, such as the minimum distance value can be used to reach consensus among all the detectors

$$\begin{gathered} min_d = \min(d_i, \cdots, d_n) \\ i = 1,2,3,\cdots,n \end{gathered} \tag{7}$$

Figure 2 shows a scenario in which detector D_3 would be penalized because it is farther from the other two detectors. Note that, although a minimum of three detectors is needed so that a consensus can be reached, this scheme imposes no upper limit to the number of detectors that can be used. The only limitation is computational performance.

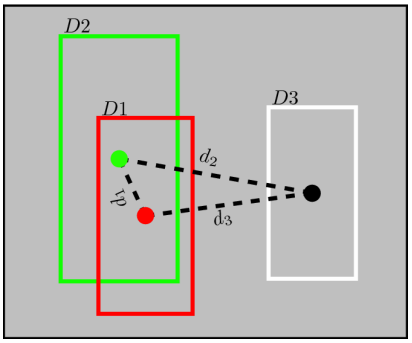


Fig. 2: Majority voting representation. Distances d_i are traced from the center of each detector. While these distances are shown as the center distances among detectors (u and v), they also comprise their heights and widths (h and w). In this scenario, D_1 and D_2 are close to each other, while D_3 is farther away. The consensus will penalize D_3 in this case, since d_1 is the minimum distance.

To calculate a weight that penalizes detectors for being farther from the cluster of detectors, instead of using a hard limiter, a hyperbolic tangent is applied, again allowing a smooth transition among detectors:

$$w_d = \omega_0 + \omega(1+\tanh(min_d - \lambda)) \tag{8}$$

where ω_0 is an initial weight consistent with the observed phenomenon, ω is the desired impact of the penalization function, which determines the overall effect of a particular detector in the fusion if it drifts away, and λ determines the distance at which the penalization starts taking place.

E. Adaptive Fusion Center Strategy

The bank of KFs is composed of one filter for each detector. Each filter/expert in the bank generates a local estimate of the detector assigned to that particular filter. Another KF acts as the fusion center, which adapts itself at each measurement by updating its measurement noise covariance according to

$$R_{vv}(w_d, w_m) = \Gamma w_d + \Delta w_m \tag{9}$$

where w_d and w_m are given by Eqs. 8 and 5, respectively, $\Gamma = diag(\gamma_1, \gamma_2, \cdots, \gamma_n)$, $\Delta = diag(\delta_1, \delta_2, \cdots, \delta_n)$, and $diag(.)$ represents a diagonal matrix whose elements are the function parameters. γ_i and δ_i can be set to 1 if there is no a priori knowledge of the system. Otherwise, γ_i can be set to a value depending on the knowledge of the uncertainty of the detector and δ_i can be set to a value depending on how much drift the detector suffers. This is different from using a standard multiple model Kalman filter because it incorporates the confidence-based majority voting into the Bayesian framework.

Algorithm 1 HAB-DF

Require: Set of n trackers $K_i \in \mathbb{S}$, initial bounding box x_0, set V of images

Ensure: Bounding box s_f representing the fused output

1: Initialize all trackers K_i with x_0.
2: Initialize Kalman filter for each algorithm implementation s_j
3: Initialize Kalman filter for fused data model
4: **while** V has new images **do**
5: Load new image
6: **for** Each tracker $K_i \in S$ **do**
7: Generate bounding box x_j for each tracker K_i
8: Apply Kalman filter (Eq. 1,2) to x_j
9: Compute Mahalanobis Distance weight w_M
10: **end for**
11: Apply majority voting to find w_d
12: Calculate R_{vv} according to Eq. (9)
13: Apply Kalman filter (Eq. 1,2) using R_{vv} as the observation covariance to generate the global estimate x_f
14: **end while**

In summary, the majority voting weight w_d and the MD weight w_m are used by the global tracker to update R_{vv}, which is then used in the global correction stage. The overall structure of our object tracking mechanism is summarized in Algorithm 1, and evaluated in more detail in [39].

IV. Platform description

A pan-tilt system and a small UAV were used to test the proposed method. The algorithm was implemented in C++ and ran in a Lenovo W530 laptop with an Intel® Core™ i7-3630QM CPU @ $2.40GHz \times 8$ processor and a Quadro K1000M graphics card.

A. Pan-Tilt System

The platform was composed of two servo motors that control the $2DOF$ of the system with an on-board Creative Senz3D camera[1]. Two different PID controllers kept the system as close as possible to the center of the image by keeping track of the centroid of the estimate generated by our data fusion approach. The servo motors were driven by the computer using an *Arduino UNO* that converted the position commands into PWM signals for the servo motors. Position commands were sent using serial communication. The implemented PID gains for both the pan and tilt motions were: $Kp = 35$, $Ki = 3.4$ and $Kd = 8$.

B. UAV Platform

The UAV used in this work was the Parrot AR.Drone 2.0, controlled over a Wi-Fi link. The $4DOF$ platform is controlled using the same heuristic proposed in [29]. However only a PD controller was used, with the following gains:

- Pitch(θ): $Kp_\theta = 0.020$ and $Kd_\theta = 0.020$.
- Roll(ϕ): $Kp_\phi = 0.699$ and $Kd_\phi = 0.400$.
- Yaw(ψ): $Kp_\psi = 0.120$ and $Kd_\psi = 0.020$.
- Throttle: $Kp_T = 0.430$ and $Kd_T = 0.021$.

Furthermore, in addition to attempting to keep the target at the center of the image using its centroid position (u,v), the UAV also used the target's relative scale variations, based on h and w, to keep a constant distance from the target.

V. Experimental Results

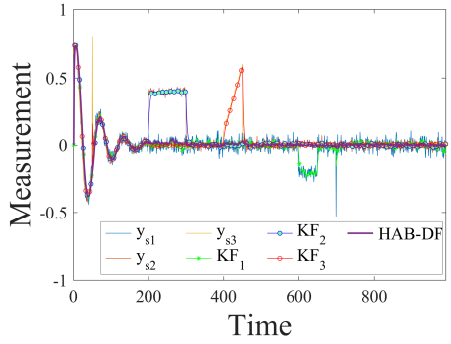
This section describes the experiments that were conducted to evaluate the proposed HAB-DF approach. We first show results from a simulation-based experiment and then discuss two real applications using the pan-tilt system and the UAV platform described in Sections IV-A and IV-B.

A. Simulations

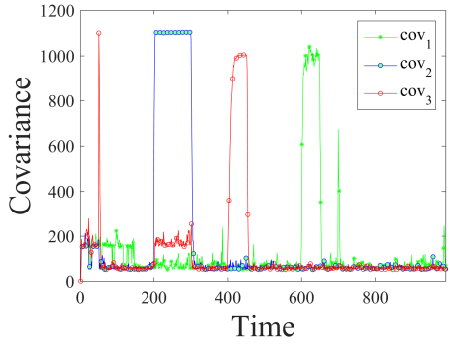
A simulation using the HAB-DF is shown in Figure 3. To emulate a scenario in which different sensors have distinct characteristics, each signal in the simulation suffers from different types of noise and faults. Each expert in the first level of the hierarchy fed the fusion center with its own estimate. Having redundancy in sensor data produced estimations that no single method could accomplish alone. Moreover, the way that the approach adapts itself along the run allows it to eliminate noise and faults. This can be seen in Figure 3b, where higher covariance values indicate that each expert in the first hierarchy is deemed faulty depending on its performance.

Compared to other works such as [13], the HAB-DF takes outliers into consideration by using the Mahalanobis distance, thereby reducing their impact. Unlike [13], HAB-DF does not learn the fault types, as learning specific types can leave unexplored regions outside the scope of the training scenarios. Additionally, the majority voting penalizes any faulty sensor.

[1]Only RGB images were used in this work. Depth data was discarded.



(a) Simulated signals



(b) Adaptive Covariance

Fig. 3: Simulation of a second order system. y_{si}, $i = 1,2,3$ are the sensor measurements. The proposed HAB-DF method is able to accurately track the signal by fusing the output of each KF in the first hierarchy. Each sensor suffers from different types of faults: Gaussian noise, spikes, drifts and shocks (a constant offset for an given time).

B. Pan-tilt System

This section describes the experiments carried out using the pan-tilt platform presented in Section IV-A. The evaluation consisted of testing each of the estimators individually with their respective detector and then the fusion of all of the detectors. This experiment consisted of tracking the face of a human subject using the pan-tilt system. All the experiments were run using similar light conditions and with the same face at similar starting distances. Each run continued until the target was out of the image frame or noticeable tracking loss occurred. As a results, for each run of the experiment, the system follows the target for a different number of frames. Furthermore, to compare each individual detector's overall performance, each test was labeled by hand. Since manual labeling is a time consuming task, we performed five repetitions of the experiment for each individual detector as well as for the proposed HAB-DF, which generated a total of 20 data sets. Figure 4 shows snapshots from these selected sequences.

Performance and reliability were measured with an overlap score (also known as the Jaccard index), given by

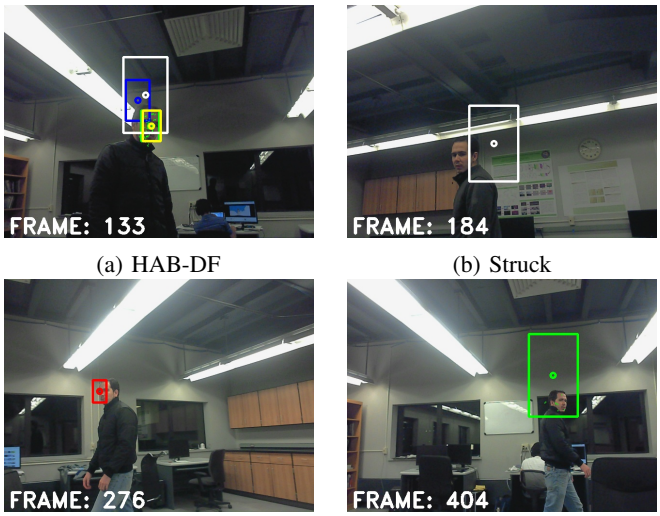
$$J_{IDX}(A_{bb},A_T) = \frac{A_{bb} \cap A_T}{A_{bb} \cup A_T} \tag{10}$$

where A_{bb} and A_T are the areas in pixels of the bounding boxes of each approach and of the GT, respectively. J_{IDX} measures the area of overlap between the bounding boxes generated by each approach and the labeled GT. The closer to 1, the better the performance. In addition to the J_{IDX}, the four-dimensional Euclidean distance d used in the majority voting also reflects the dissimilarity among each approach and the GT.

Figure 5 displays several metrics that illustrate the performance of the approaches. Figure 5a shows the average performance of the different detectors and the proposed approach according to J_{IDX}. As shown, Struck showed the worst performance among all the detectors, having problems with scale changes caused by the target moving closer and farther from the camera. CMT, DSSTtld and the proposed approach performed similarly until the 400^{th} frame. DSSTtld showed the best performance for a few frames in terms of accuracy (between the 400^{th} and 600^{th} frame) but was not able to handle pose changes nor out-of-plane rotations of the target, which resulted in a sudden drop in confidence level and consequently losing track of the target. While CMT was able to handle distortions caused by rotation, its J_{IDX} degraded with scale changes. As a result, it kept track of the target longer than the other detectors, albeit with substantially reduced accuracy. If the intrinsic properties of the detectors are combined, the Bayesian approach is not only more robust but also more accurate than only using a single detector. Also, if one of the detectors is not performing well, such as Struck in the aforementioned scenario, it is possible to see that the fusion is not affected, since that particular detector will be considered unreliable by our framework. Figure 5c shows a comparison of the accuracies of the different approaches. This plot considers a threshold between J_{IDX} and d of what is considered a successful frame. On average, the Bayesian fusion yielded better results and outperformed each individual estimator. This is consistent with the results in [39] that show that this method outperforms the constituent trackers.

An additional experiment was conducted using a recycling bin as target because of its distinct appearance. Figure 7 exhibits different images from the experiment. Figure 6 shows the different metrics collected during the experiment. Figure 6a shows the J_{IDX} for each approach. Up to the 100^{th} frame, all approaches have similar performance, with HAB-DF leading in accuracy most of the time. In this scenario, Struck showed better performance, since the object was kept almost at a constant distance. It was not until frame 700 that Struck lost track. Figure 6b shows the Euclidean distance d. In this case, DSSTtld showed the worst performance due to pose variations and out-of-plane rotations of the object, while CMT had a reasonable performance throughout the run. Furthermore, the HAB-DF leads in performance among all approaches, relying mainly on the best detectors at each frame. Figure 6c shows once again that HAB-DF outperforms all of the other approaches.

Figure 6d displays how the adaptation of the HAB-DF took place. When the distortion of DSSTtld became too high, the MD increased accordingly. Between frames 100-300 and 500-800 the detector did not overcome distortions caused



(a) HAB-DF (b) Struck

(c) DSSTtld (d) CMT

Fig. 4: Pan-tilt system experiment (best seen in colors). The frames shown here are random frames selected from the dataset. Each of them presents a different tracking approach. The target was moving sideways with some vertical disturbances, and gradually increasing the distance from the camera. In (a) HAB-DF is shown in yellow and DDSTtld is shown in blue because it is lost.

by out-of-plane rotations of the object, lowering DSSTtld's confidence, and consequently losing track. CMT showed several spikes caused by substantial delays in processing key points. This behavior does not affect the overall approach, as asynchronous measurements are implicitly accounted for by the MD and majority voting.

Figures 6e and 6f illustrate the object position (u_{pos}, v_{pos}) in the frame with respect to the desired set-point ($Sp_u = 320$ and $Sp_v = 240$ which are the pixel center coordinates of the image). This graph shows that the experiment was consistent with the motion of the target. Despite some detectors being lost along the experiment, the transition among them was smooth.

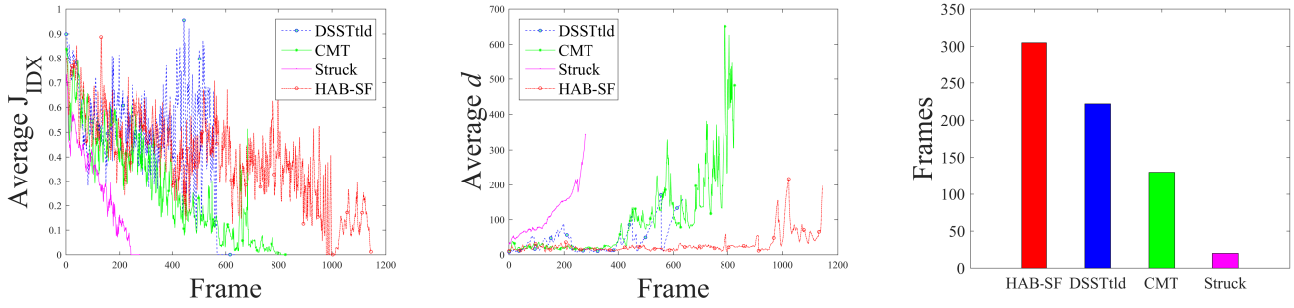
C. UAV Platform

Figure 8 shows snapshots of experiments using a small UAV. These experiments were carried out indoors and consisted of following different targets in a hallway and in a gym. The results of one of these experiments can be seen in Figure 9. Figure 9a displays the relative distance to the target as estimated by the ratio between the area of the target and the image area. The initial ratio is used as the set point, and the error is used to control the UAV pitch. Figures 9b and 9c show the vertical and horizontal target positions within the frame and the corresponding set points. The offset observed in Figure 9c is due to the coupled effect of the pitch and throttle controllers as the target moves (i.e., as the UAV moves forward, its camera faces down). Although this effect is unavoidable with a fixed camera, it could be resolved with a camera that can be controlled independently from the UAV. Figure 9d shows the amount of penalization suffered by each tracker throughout the trial. It is interesting to note that in this scenario Struck shows improved performance in comparison to the pant-tilt system experiments. This is a result of the fact that the target scale remains approximately constant as the UAV follows it.

Redundant information allows the platform to track the target for longer periods of time. In the sequence shown in Figure 9, HAB-DF was able to keep track of the target for 7132 frames, until all the detectors lost track of the target simultaneously. In comparison, DSSTtld first lost track at frame 220, Struck at frame 275, and CMT at frame 1738. While these trackers were often able to recover from failure because the target was eventually brought back to the center of the image, had the control actions been taken according to any one of those trackers individually, the platform would likely not have been able to continue following the target. The proposed scheme allows the system to ignore lost detectors and rely on those that provide confident estimates. Failures are evident in Figure 9d, which shows to what extent each detector is penalized.

VI. Conclusion

In this work, a Hierarchical Adaptive Bayesian Data Fusion method was presented. While the algorithm is not limited to specific applications, the main scenario under consideration was vision-based robotic control. The method out-



(a) Jaccard Index (b) Euclidean distance d (c) Measure of success

Fig. 5: Average performance. Figure 5a shows the average of the J_{IDX}. A decrease in J_{IDX} indicates tracking performance degradation. A value of zero indicates a complete failure in which there is no overlap between the GT and the detector. Figure 5b shows average of d for each approach. A value close to zero means that the GT and the tracker are similar. Figure 5c shows the success bar graph, a frame is considered successfully tracked when $J_{IDX} \geq 0.5$ *and* $d \leq 50$.

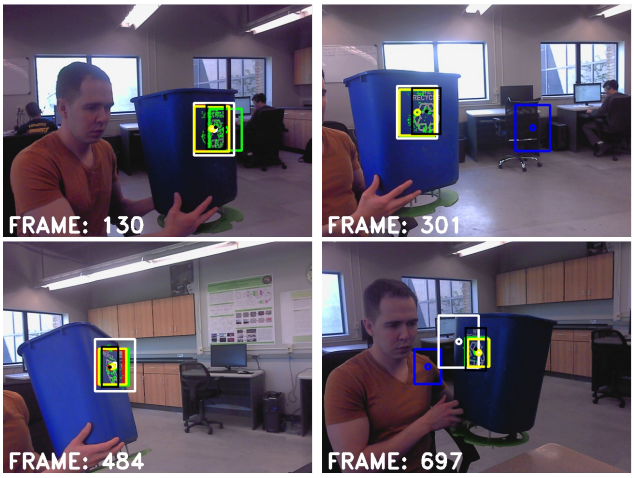
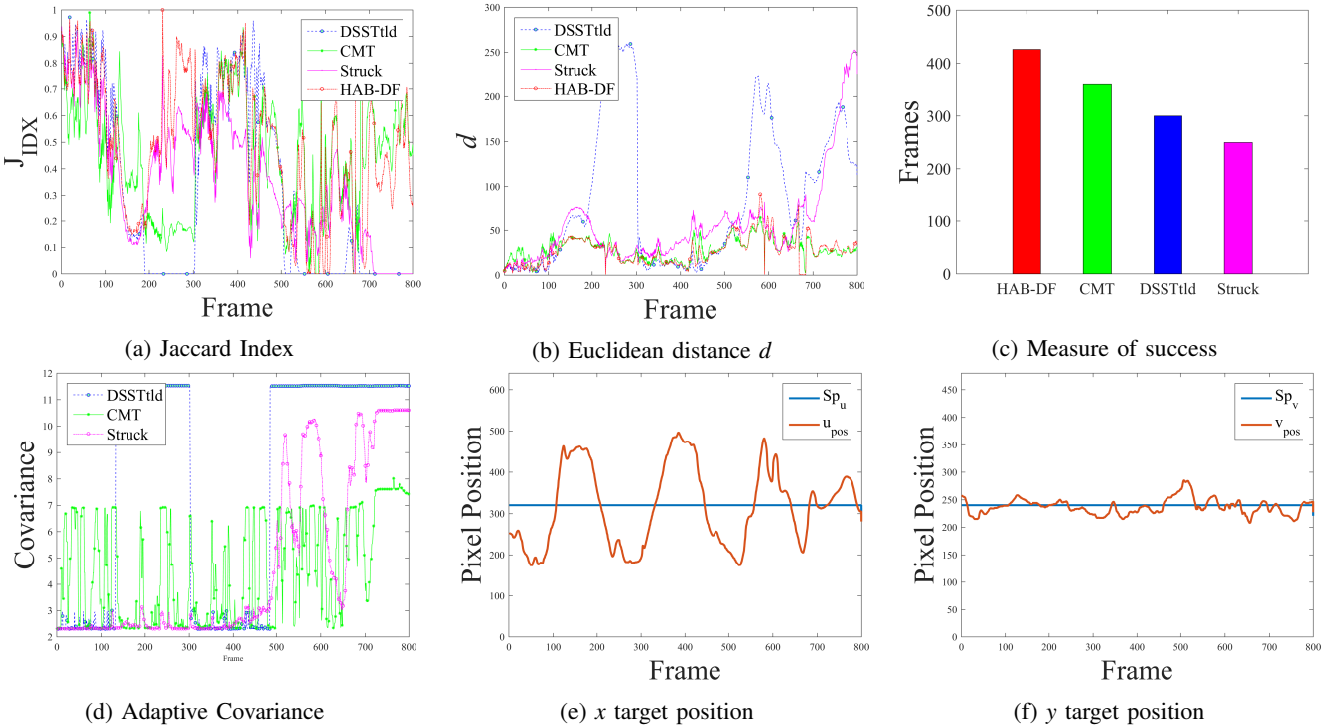


Fig. 7: Tracking a recycling bin. Frame 130 shows when DSSTtld loses track due to out-of-plane rotation of the target, while the other approaches continue tracking normally. Frame 301 shows the majority voting taking place. While Struck and CMT are tracking the recycling bin, DSSTtld could not recover. Frame 484 shows all the approaches working together giving a good estimate before DSSTtld loses track again. At frame 697, Struck drifts and DSSTtld loses track due to an out-of-plane rotation in a previous frame. Despite these problems, HAB-DF is able to keep track of the target for the entire sequence.

Fig. 8: Images from UAV Trials. The frames show all the detectors working properly in the hallway and gym scenarios while the HAB-DF fuses their measurements. During the trial, DSSTtld loses track several times, as illustrated in frame 2520, while CMT and Struck continue to track and HAB-DF properly combines their outputs. Struck shows significant scale disparity, while the combined output correctly estimates the size of the target. Frame 3132 shows a different target in which all three detectors are working albeit with some positional inaccuracy. The combined estimate is more accurate.

performed single detectors, with better accuracy and keeping track for longer periods of time. Moreover, no training data was used while most approaches in this field rely on machine learning techniques, most of which require large amounts of training data for good performance. Even when substantial amounts of training data are available, these methods may be unable to handle situations that were not properly explored during training. The HAB-DF relies instead on the local statistical performance of the individual data sources. In addition, the decentralized architecture allows the experts to operate asynchronously, while penalizing measurements that are delivered to the fusion center with significant delays.



(a) Jaccard Index

(b) Euclidean distance d

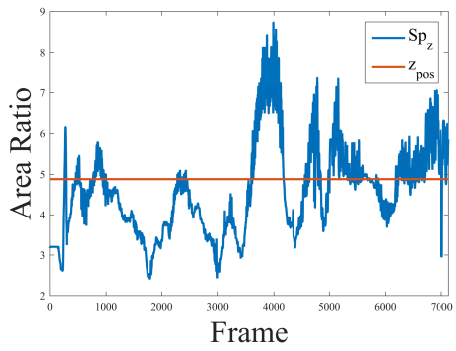
(c) Measure of success

(d) Adaptive Covariance

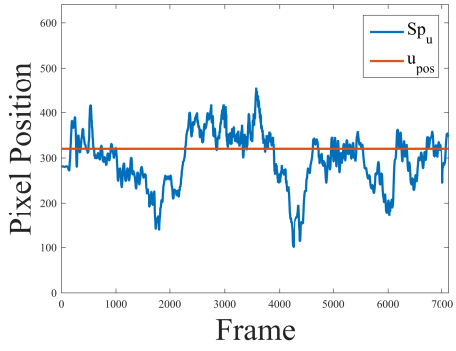
(e) x target position

(f) y target position

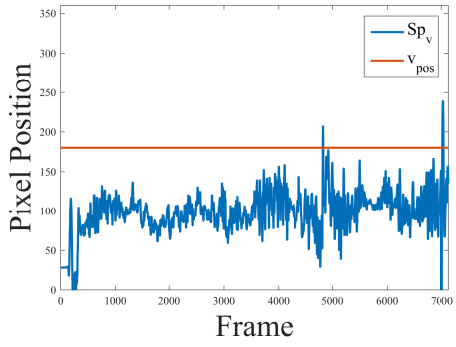
Fig. 6: Evaluation of the performance of the experiment in which a recycling bin was tracked. Figure 6a and Figure 6b show that DSSTtld has a degraded performance (around frames 100-300 and 500-800). This is consistent with Figure 6d, where DSSTtld suffers of a sudden drop of confidence value resulting in an increment of the covariance that is ruled by the MD and the majority voting scheme. The HAB-DF has the best performance among all the approaches as seen in Figure 6c. Moreover, the transition between detectors is soft, allowing for the smooth motion control that can be seen in Figure 6e and Figure 6f.



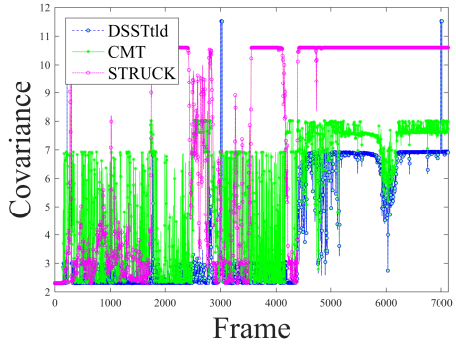
(a) Relative distance to target



(b) Horizontal target coordinate



(c) Vertical target coordinate



(d) Adaptive covariance

Fig. 9: Tracking a person in a gym with a UAV. Figures 9a, 9b, and 9c show the behavior of the UAV along the trial. Figure 9d shows the adaptive behavior of the HAB-DF during the experiment.

Finally, the weighted majority voting scheme allows sensors that provide measurements which are discrepant or have low confidence to be automatically discarded from the estimation.

Moreover, the two platforms tested show that this algorithm is suitable for real-time applications with good performance. Both platforms were able to follow practical objects with different characteristics without any prior training. Additionally, it shows that when detectors with different performances are combined, they can outperform individual methods.

REFERENCES

[1] Y. Gu, J. N. Gross, M. B. Rhudy, and K. Lassak, "A fault-tolerant multiple sensor fusion approach applied to uav attitude estimation," *International Journal of Aerospace Engineering*, vol. 2016, no. 2016, 2016.

[2] A. M. Rahimi, R. Ruschel, and B. Manjunath, "Uav sensor fusion with latent-dynamic conditional random fields in coronal plane estimation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[3] J. Llinas, D. L. Hall, and M. E. Liggins, *Handbook of Multisensor data fusion: theory and practice*. CRC Press Broken Sound Parkway NW, 2009.

[4] L. Hong, "Adaptive data fusion," in *Systems, Man, and Cybernetics, 1991.'Decision Aiding for Complex Systems, Conference Proceedings., 1991 IEEE International Conference on*. IEEE, 1991, pp. 767–772.

[5] P. J. Escamilla-Ambrosio and N. Mort, "Hybrid Kalman filter-fuzzy logic adaptive multisensor data fusion architectures," in *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, vol. 5. IEEE, 2003, pp. 5215–5220.

[6] A. D. Tafti and N. Sadati, "Novel adaptive Kalman filtering and fuzzy track fusion approach for real time applications," in *Industrial Electronics and Applications, 2008. ICIEA 2008. 3rd IEEE Conference on*. IEEE, 2008, pp. 120–125.

[7] E. Bostanci, B. Bostanci, N. Kanwal, and A. F. Clark, "Sensor Fusion of Camera, GPS and IMU using Fuzzy Adaptive Multiple Motion Models," no. March 2016, 2015.

[8] H. E. SÃűken and C. Hajiyev, "Adaptive unscented Kalman filter with multiple fading factors for pico satellite attitude estimation," in *Recent Advances in Space Technologies, 2009. RAST'09. 4th International Conference on*. IEEE, 2009, pp. 541–546.

[9] G. Loy, L. Fletcher, N. Apostoloff, and A. Zelinsky, "An adaptive fusion architecture for target tracking," in *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*. IEEE, 2002, pp. 261–266.

[10] M. K. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," *Journal of the American statistical association*, vol. 94, no. 446, pp. 590–599, 1999.

[11] J.-A. Ting, E. Theodorou, and S. Schaal, "A Kalman filter for robust outlier detection," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*. IEEE, 2007, pp. 1514–1519.

[12] G. Agamennoni, J. I. Nieto, and E. M. Nebot, "An outlier-robust Kalman filter," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1551–1558.

[13] S. Reece, S. Roberts, C. Claxton, and D. Nicholson, "Multi-sensor fault recovery in the presence of known and unknown fault types," in *Information Fusion, 2009. FUSION'09. 12th International Conference on*. IEEE, 2009, pp. 1695–1703.

[14] A. Ravet, S. Lacroix, G. Hattenberger, and B. Vandeportaele, "Learning to combine multi-sensor information for context dependent state estimation," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 5221–5226.

[15] S. E. Yuksel, J. N. Wilson, and P. D. Gader, "Twenty years of mixture of experts," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 23, no. 8, pp. 1177–1193, 2012.

[16] S. Yao, S. Hu, Y. Zhao, A. Zhang, and T. Abdelzaher, "Deepsense: A unified deep learning framework for time-series mobile sensing data processing," *arXiv preprint arXiv:1611.01942*, 2016.

[17] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof, "Prost: Parallel robust online simple tracking," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 723–730.

[18] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 1269–1276.

[19] G. Nebehay and R. Pflugfelder, "Clustering of static-adaptive correspondences for deformable object tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2784–2791.

[20] S. Hare, A. Saffari, and P. H. Torr, "Struck: Structured output tracking with kernels," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 263–270.

[21] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 7, pp. 1409–1422, 2012.

[22] G. Nebehay, "Robust object tracking based on tracking-learning-detection," Master's thesis, TU Wien, 2012.

[23] K. Haag, S. Dotenco, and F. Gallwitz, "Correlation filter based visual trackers for person pursuit using a low-cost quadrotor," in *Innovations for Community Services (I4CS), 2015 15th International Conference on*. IEEE, 2015, pp. 1–8.

[24] C. Bailer, A. Pagani, and D. Stricker, "A superior tracking approach: Building a strong tracker through fusion," in *European Conference on Computer Vision*. Springer, 2014, pp. 170–185.

[25] I. Leichter, M. Lindenbaum, and E. Rivlin, "A general framework for combining visual trackersâĂŞthe" black boxes" approach," *International Journal of Computer Vision*, vol. 67, no. 3, pp. 343–363, 2006.

[26] L. Zhang, Y. Gao, A. Hauptmann, R. Ji, G. Ding, and B. Super, "Symbiotic black-box tracker," in *International Conference on Multimedia Modeling*. Springer, 2012, pp. 126–137.

[27] T. A. Biresaw, A. Cavallaro, and C. S. Regazzoni, "Tracker-level fusion for robust Bayesian visual tracking," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 25, no. 5, pp. 776–789, 2015.

[28] B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi, "Multisensor data fusion: A review of the state-of-the-art," *Information Fusion*, vol. 14, no. 1, pp. 28–44, 2013.

[29] J. Pestana, J. L. Sanchez-Lopez, S. Saripalli, and P. Campoy, "Computer vision based general object following for gps-denied multirotor unmanned vehicles," in *American Control Conference (ACC), 2014*. IEEE, 2014, pp. 1886–1891.

[30] C. Fu, "Vision-based tracking, odometry and control for uav autonomy," Ph.D. dissertation, Universidad Politecnica de Madrid, 2015.

[31] C. Y. Chong and S. Mori, "Track association using augmented state estimates," in *2015 18th International Conference on Information Fusion (Fusion)*, July 2015, pp. 854–861.

[32] W. S. Chaer, R. H. Bishop, and J. Ghosh, "A mixture-of-experts framework for adaptive Kalman filtering," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 27, no. 3, pp. 452–464, 1997.

[33] J. V. Candy, *Bayesian signal processing: Classical, modern and particle filtering methods*, Second ed. John Wiley and Sons, 2016.

[34] P. C. Mahalanobis, "On the generalized distance in statistics," *Proceedings of the National Institute of Sciences (Calcutta)*, vol. 2, pp. 49–55, 1936.

[35] Z. M. Durovic and B. D. Kovacevic, "Robust estimation with unknown noise statistics," *IEEE Transactions on Automatic Control*, vol. 44, no. 6, pp. 1292–1296, 1999.

[36] S. Chan, Z. Zhang, and K. Tse, "A new robust Kalman filter algorithm under outliers and system uncertainties," in *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on*. IEEE, 2005, pp. 4317–4320.

[37] R. R. Pinho, J. Tavares, and M. Correia, "Efficient approximation of the Mahalanobis distance for tracking with the Kalman filter," in *CompIMAGE*, 2006, pp. 349–354.

[38] A. E. "Guevara, A. Hoak, J. T. Bernal, and H. Medeiros, "Vision-based self-contained target following robot using bayesian data fusion," in *12th International Symposium in Visual Computing, ISVC 2016*, 2016.

[39] R. Yevgeniy and H. Medeiros, "Improving target tracking robustness with bayesian data fusion," in *British Machine Vision Conference*, September 2017.