Visual and Radar Sensor Fusion for Perimeter Protection and Homeland Security on Edge

Danny Buchman[a,b], Michail Drozdov[a,c], Aušra Mackutė-Varoneckiene[d,e] and Tomas Krilavičius[d,e]

[a] *JVC Sonderus, Vilnius, Lithuania*
[b] *Seraphim Optronics Ltd., Yokne'am Illit, Israel*
[c] *Geozondas Ltd., Vilnius, Lithuania*
[d] *Department of Applied Informatics, Vytautas Magnus University Kaunas, Lithuania*
[e] *Baltic Institute of Advanced Technology, Vilnius, Lithuania*

Abstract

Today, in the border and perimeter protection, it is very common to use RADAR technology and pan-tilt (PT) cameras to have terrain dominance. The common solution uses both sources - while most of the threats are detected by radar, the camera used for inspection of motion, detected by radar. This solution is very dependent on radar performance and not effective for different scenarios when the radar is not capable to monitor the movement of all targets. Inputs from camera and radar are used in close integration to increase detection probability and reduce false alarms. In this work two alternative methods of radar and visual data fusion are proposed, data structures and processing algorithms are defined and results of experimental validation for both proposed methods are shown.

Keywords

Sensor fusion, Radar, Video motion detection, Perimeter protection, Kalman filter

1. Introduction

Sensor fusion is a large research topic. Its goal is to combine multiple data sources to receive joined data, which allows to improve processes or calculations, compared to single-source data usage.

Tracking solution using radar as the only source of data suffers from unreliable detection or even absence of detection when dealing with mostly tangential trajectories of observed objects. An attempt is made to lessen this problem by adding a camera as a second source of data and combining radar tracking with video motion detection (VMD) while keeping a common target state for detections from both sources. It is also expected, that fusion can add the benefit of reduced false detection rate since validation of tracks can be more reliable using two sources of information *redundant* fusion scheme [1]).

This work focuses on research related to the practical application of fusion between radar and video. Two main methods of fusion, namely *data fusion* and *tracks fusion* are defined in this context, with first covering such important parts of the system like data association and state updates and second being more modular and distributed alternative.

Methods based on Kalman family filters [2, 3, 4] are common, when dealing with the data-level fusion because they enable to have process model independent from observation structure [5] while working with uncertain data. In the case of several sensors, such filters allow to incorporate new data into the model as data gets available [3], [6].

Due to the properties of Kalman filter (KF), it is required, that state update of the described dynamic process would be linear. It is common practice to use Cartesian coordinates to describe object state when dealing with mostly linear movement. When trying to fuse camera and radar data, two issues are quite apparent:

1. Both radar and camera are acquiring data in Polar coordinates.
2. While full 3D Cartesian representation can be reconstructed from the radar data, it is not true for camera without several assumptions on the geometry of setup.

The first problem can be solved in several ways. The usual practice is to keep the target state in Cartesian coordinates [2], [7] while measuring in Polar and transforming data before the update (converted measure-

IVUS 2020: Information Society and University Studies, 23 April 2020, KTU Santaka Valley, Kaunas, Lithuania
danny.buchman@seraphimas-hls.com (D. Buchman); michail.drozdov@seraphimas-hls.com (M. Drozdov); ausra.mackute-varoneckiene@bpti.lt (A. Mackutė-Varoneckiene); tomas.krilavicius@bpti.lt (T. Krilavičius)

ments [8], [9], [10]). The covariance matrix, which is used in update and estimation, gets biased if transformed directly. There are exist many solutions for the linearization of space near estimated point to get proper values for the covariance matrix [3], [11] (Extended Kalman Filter, Unscented Kalman filter to name a few). After transformation state can be updated by normal Kalman filter formulas.

Second issue is not addressed by these solutions: if Kalman filter would be used while keeping target state in Cartesian coordinates, camera would change from very precise sensor (*az*and *el* angles) to very unprecise (*x, y, z*), as distance to object is used in Polar to Cartesian transform for any direction and is not directly measured by camera. There are numerous different approaches to get at least some estimation of distance from direct camera measurements:

1. Use radar detections as a base and map camera detections to radar improving angular resolution [12], [13].
2. Use homography[1] estimation methods for camera calibration in the lab.
3. Use corner reflector or another strongly reflective object to map precisely radar and camera detections into 3D [14], [15].
4. Use many assumptions[2] on the positioning of detections relative to the optical axis (ground is straight plane, camera position, and orientation is known, targets are always on the ground, etc.) [16], [17].
5. Use machine learning (ML) techniques, in the cases when targets are specified (e.g. like detecting image size of cars the physical size of which is known) [13, 18].
6. Use the movement of the camera and additional features (lane lines) to acquire distance [19], [20], [21].

For scenarios, arising in perimeter protection or homeland security, method 6) is not applicable or too costly in a sense of performance. Usually, there are no predefined markers and the system is stationary (no translational movement). All other approaches can be explored. Ideal solution, however, would be to use camera only in its strongest domain to augment information received by radar instead of increasing inaccuracies in one dimension while decreasing in others. One such potential solution is to keep the state of Kalman filter in Polar coordinates. This isn't unknown in literature, as bearing only tracking often uses Modified Polar Coordinates (MPC) [8], [22] or Modified Spherical Coordinates (MSC) [23].

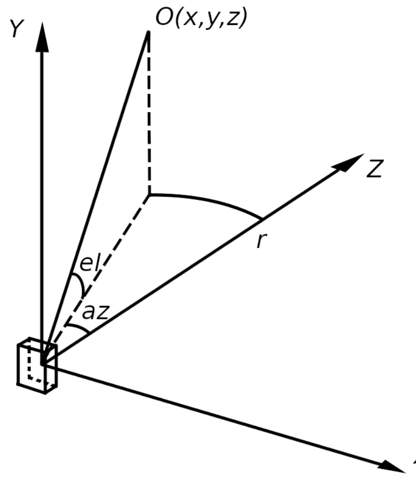

Figure 1: Coordinate systems used in experiments. Unit is pointed in Z direction

Following issues are explored before finalizing data model for fusion:

1. The impact of the origin of the coordinates – to use Cartesian or Polar coordinates in the linear Kalman filter, for the different movement patterns.
2. Improvement (if any) of the accuracy of state estimation in Polar coordinates, if the camera is also used in Kalman filter updates.
3. Detection of distance to objects from camera measurements and the effect of the addition of this result to the Kalman filter measurement model.

As a consequence, some of the following sections (section II, section V and section VI) are split into two main parts - preliminary experiments (simulation) and methods validation. It should be clear, however, that chronologically all preliminary tests were performed and results were analyzed before finalizing fusion models and implementing fusion methods.

2. Data Description

Coordinates, used in this research are defined as shown in Fig.1

[1]A geometrical relation between two images of the same planar surface, described by the transformation matrix

[2]The camera is moving by known pattern, the terrain is flat, camera position and angle to the surface are known, etc.

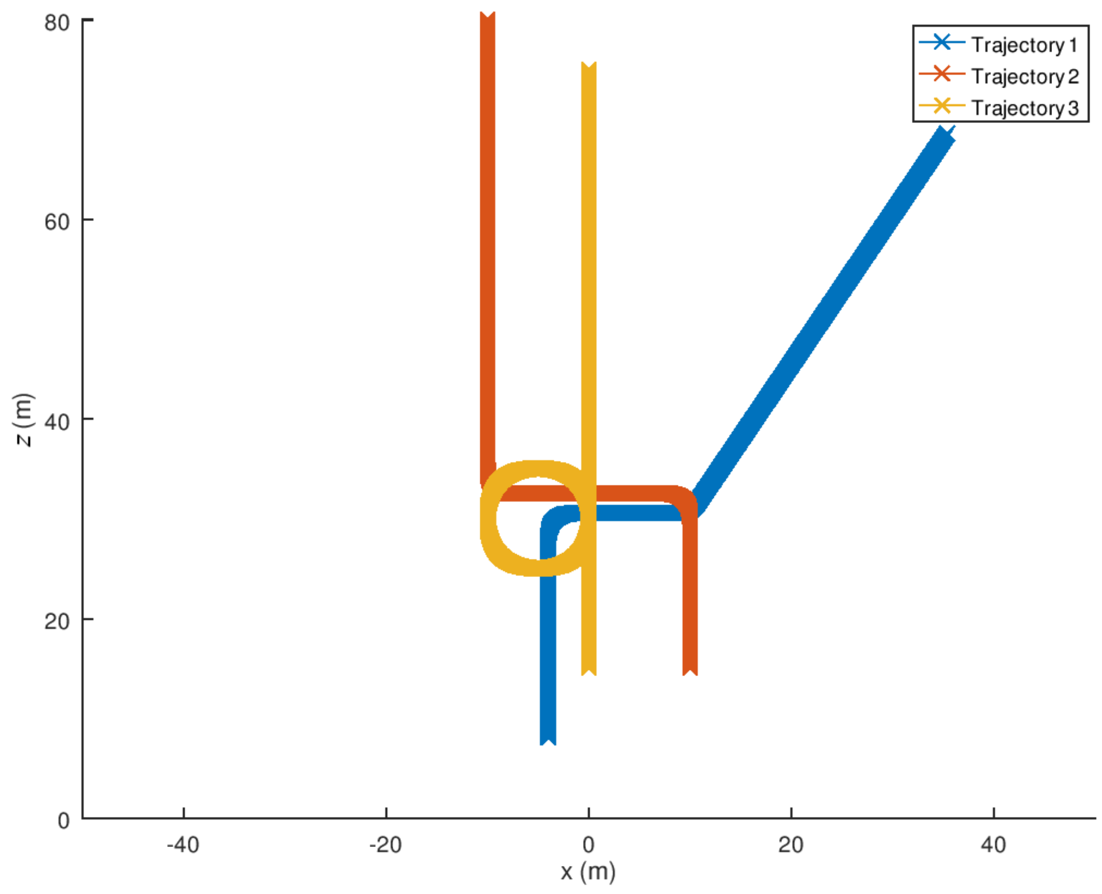

Figure 2: The setup of test trajectories for experiments

2.1. Simulation Data Description

In this part of the research, the definition of custom trajectories for any number of targets as well as radar and video motion detection (VMD) noise models were implemented. The following simulation parameters can be defined for radar:

1. Range noise standard deviation
2. Velocity noise standard deviation
3. Detection angle noise standard deviation
4. False detection rate (per simulation area element)
5. Detection view angle
6. Update rate

The following simulation parameters can be defined for the camera:

1. VMD detection box noise (in pixels)
2. Camera resolution
3. Angular field of view
4. VMD false positive rate and area
5. Update rate

Simulation of detections for VMD and radar and subsequent registration without knowledge of ground truth or noise model is performed.

Kalman filter state is defined by $[x\ v_x\ a_x\ z\ v_z\ a_z]^{\mathrm{T}}$ in case of Cartesian coordinates and by $[r v_r a_r az daz ddaz]^{\mathrm{T}}$ in Polar coordinates, where

1. v_x and v_z are velocities in x and z dimensions respectively,
2. a_x and a_z are accelerations
3. v_r - range change rate
4. az - azimuth angle
5. daz and $ddaz$ - the rate of change for azimuth and the rate of change of daz (angle acceleration)

Measurements of radar and camera are simulated using Gaussian noise model. For camera – different accuracies of VMD were tested ranging from 1 pixel to 5 pixels.

Measurements are filtered and the mean square error (MSE) is calculated for comparison of estimated positions to actual trajectories.

In Fig.2 test trajectories Trajectory 1 – T1, Trajectory 2 – T2, Trajectory 3 – T3, which were used for evaluation (camera and radar are at the point (0,0) in a$x - z$ Cartesian coordinate system) are presented.

2.2. Data for Evaluation of Fusion Methods

In case of track fusion inputs for fusion module are

1. list of VMD outputs as bounding boxes $[x\,y\,w\,h]$ and ID of VMD track
2. list of radar tracker outputs as $[x\,z\,v_x\,v_z]$ and ID of radar track

In case of data fusion inputs are

1. list of bounding boxes $[x\,y\,w\,h]$ received directly from the "blob" stage of VMD pipeline
2. list of radar targets $[r\ v_r\ az]$

The main difference between sources of data for two approaches is, that in case of data fusion there are many false detections, which are not filtered by VMD or radar tracker methods respectively.

3. Data Fusion Methods

Overview of different sensor fusion classifications can be found in [1] for a further reading. Here we employ terminology, used by Castanedo in that paper.

Possible schemes of the sensor fusion in a given application are limited to *redundant* schemes (with other possibilities being *complementary* and *cooperative*) based on the relations of sensors used in the system. Both types of sensors are measuring the same state of objects at the same time in the observed area.

Based on the idea of modularity of the system, it is clear, that radar/camera fusion should not be a decision-making module. While detections are performed by the fusion module, the final decision is affected by additional rule-based filters and the recognition module. The goal of the fusion module is to minimize the false alarm rate (FAR) of the system and provide inputs for decision-making modules. It can be said then, that fusion module approaches can be limited to two (*d*ata in - feature out (DAI-FEO), *f*eature in - feature out (FEI-FEO)), contrary to approaches, which provide data or decisions as outputs (*d*ata in - data out (DAI-DAO), *f*eature in - decision out (FEI-DEO), *d*ecision in - decision out (DEI-DEO)).

Four types of fusion architectures are defined in [1]:

1. centralized architecture,
2. decentralized architecture,
3. distributed architecture,
4. hierarchical architecture.

Centralized architecture (data from all sources is processed in single module) is expected to be theoretically optimal in case of proper synchronization of data sources and sufficient bandwidth for data transfer. It can suffer, however, from lack of distribution of bandwidth and processing in case these resources are limited for given task. Alternatives, solving this issue, are *decentralized architecture* (fusion nodes incorporate raw data in different order and composition) and *distributed architecture* (fusion nodes receive single sensor data and provide features to be fused). In our view, decentralized architecture would introduce unnecessary complexity and implementation difficulty, so distributed architecture is considered as only another option to centralized architecture.

4. Proposed Fusion Methods

Based on the discussion in the previous section and the results of preliminary evaluations (described in section VI), two main fusion approaches are established:

1. data fusion
2. tracks fusion

Data fusion is a centralized mixed input (DAI-FEO and FEI-FEO) method, accepting raw outputs of radar and intermediate results of VMD (bounding boxes of detected blobs).

Tracks fusion is distributed FEI-FEO method, accepting features (tracks) from VMD and radar tracker modules.

Both should solve the problem defined, but the pros and cons of approaches are different. Fusion of tracks is performed after data from both sources is processed and there is track detected on both sets of data. Then both tracks are matched (track to track association) to better reflect the behavior of the object being tracked. In the case of one of the sources not returning track while other is returning track, different policies can be used to favor reduced false detection rate over extended tracking duration or vice versa. Comparing with data fusion, tracks fusion is easier to debug and tune, since separate modules can be tested and tuned faster due to overall reduced complexity. Data fusion works on a lower level than tracks fusion. It has the benefit of incorporating updates from both sources of data into common state update converging to a true state of the object being tracked faster. The role of policies in the case of one of the sources missing detections is reduced since both sources of raw data can be treated similarly (both update Kalman filter state). There is no need to tune policies for different cases (no recent radar detection, no recent camera detection, higher than an average mismatch between sources, etc.), it is handled by common update. The downside of data fusion compared to track fusion is longer debugging and tuning. If conditions of experiments/use cases or equipment changes, that could add delivery time overhead.

4.1. Data Fusion

General strategy for data-level fusion can be summarized as follows:

1. Use radar detections as a base for track validation.
2. Any VMD detection can create a persistent track, but it will be validated without radar data only after a relatively long track age is reached.
3. Any track with both recent VMD and Radar detections has a higher probability to be validated as a real track, as a consequence, it is validated at the lower age of the track.

Such choices are proposed due to relative ease of radar data validation - if the movement of a potential target in the area under test contradicts Doppler velocity, reported by radar, such target can be quickly invalidated. There is no similar process for VMD.

Full track state representation consists of the following parts:

1. Track state vector at time t (based on definitions in Section II). 6x1 vector:

$$X_t = \begin{bmatrix} x \\ v_x \\ a_x \\ z \\ v_z \\ a_z \end{bmatrix}, \tag{1}$$

   or its counterpart in Polar coordinates. In case of constant velocity model (4x1 vector):

$$X_t = \begin{bmatrix} x \\ v_x \\ z \\ v_z \end{bmatrix}. \tag{2}$$

2. The track covariance matrix P_t. 6x6 matrix or 4x4 matrix (if accelerations removed), describing the amount of variance in data and interstate dependencies (covariance).
3. The track age. Time elapsed from the moment of track creation by VMD or radar. It is updated if either VMD or radar detection can be associated with the current state vector.
4. The track innovation error. Value, which is compared to predefined threshold values for track validation and removal. Track innovation error (squared Mahalanobis distance) is calculated from state measurement residual (innovation) and residual (innovation) covariance. It is one of the main criteria for positive detection.
5. Track update timestamp for VMD
6. Track update timestamp for radar
7. Object size. It can be calculated if both VMD and radar detected an object.
8. Object visual distance state. 2x1 vector (distance, rate of distance). It is estimated from the camera and can be used if the positioning of the device is known. It is highly unstable due to partial obstructions and because of that is separated from the track state vector. It can be used by higher-level decision-making modules.
9. Object visual distance covariance matrix. 2x2 matrix.
10. The total duration of detection for VMD. It is one of the main criteria for positive detection.
11. The total duration of detection for radar. It is one of the main criteria for positive detection.

The fusion scheme consists of data association, state update, and management of tracks. Data association (point to track) in the first prototype is implemented as simple Nearest Neighbour (NN) estimation in state space, favoring simplicity and speed. Joint Probability Data Association (JPDA) based association method [24] is used as an alternative in later versions. In NN based data association each measurement is compared against the estimated state $X_{t|t-1}$ at time t by calculating Mahalanobis distance based metric (discussed further). The prefiltering of potential associations can be performed using some simple heuristics like 2D distance.

Linear KF is used for state estimation and update. Estimation step:

$$X_{t|t-1} = AX_{t-1|t-1}, \tag{3}$$

where A is process matrix defined for a state with accelerations as:

$$A = \begin{bmatrix} 1 & dt & dt^2 & 0 & 0 & 0 \\ 0 & 1 & dt & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & dt & dt^2 \\ 0 & 0 & 0 & 0 & 1 & dt \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4}$$

for state without acceleration as:

$$A = \begin{bmatrix} 1 & dt & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & dt \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{5}$$

here $X_{t-1|t-1}$ means posterior state from the previous update, dt is time passed from the last update. Further, all the calculations will be described for state without accelerations to shorten notations, state vector dimensionality m will be used to describe sizes of matrices. The predicted covariance matrix is calculated as

$$P_{t|t-1} = AP_{t-1|t-1}A^{\mathrm{T}} + Q, \tag{6}$$

where Q is a square mxm matrix defining process noise.

The next step of processing is mixed - state update and data association step. Innovation is calculated for each pair of track and a prefiltered measurement to later allow optimal NN data association:

$$Y_t = Z_t - HX_{t|t-1}, \tag{7}$$

here H is kxm measurement matrix, Z_t is the measurement vector of kx1 size. The size of the measurement vector and the measurement matrix k depend on the type of sensor and coordinate systems used for measurements and the state. k can be understood as number of parameters measured by a specific sensor and H as mapping between state and measured parameters. If transformation from sensor coordinates to state coordinates is linear, it can be performed by H directly. In our experiment of using Polar representation of target state space

$$X_t = \begin{bmatrix} r \\ v_r \\ az \\ daz \end{bmatrix}. \tag{8}$$

The data vector of radar and measurement matrix:

$$Z_t = \begin{bmatrix} r \\ v_r \\ az \end{bmatrix}, \tag{9}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{10}$$

which simplifies calculations. In the case of camera (VMD) update without the knowledge of the geometrical setup:

$$Z_t = \begin{bmatrix} x_p \end{bmatrix}. \tag{11}$$

here x_p is just x position of the center of the bounding box in screen coordinates. The measurement matrix then

$$H = \begin{bmatrix} 0 & 0 & \frac{w}{\mathrm{HFOV}} & 0 & \frac{w}{2} \end{bmatrix}, \tag{12}$$

and the state vector needs to be augmented with additional element 1 as the last row to allow matrix multiplication in (7). In the above formula, w is the width of the video in pixels. The single element matrix Y_t can also be calculated by just using az element of the state vector as:

$$Y_t = \begin{bmatrix} x_p - \frac{az}{\mathrm{HFOV}} w + \frac{w}{2} \end{bmatrix}. \tag{13}$$

Next, innovation covariance is calculated

$$S_t = H_t P_{t|t-1} H_t^{\mathrm{T}} + R, \tag{14}$$

here R is kxk diagonal measurement error matrix. It is defined based on the parameters of sensors used. Error ranges from the datasheet of the sensor is a good starting point. Kalman gain

$$K_t = P_{t|t-1} - H^{\mathrm{T}} S_t^{-1}, \tag{15}$$

here inverse of S_t is taken. Innovation error, which is not used directly in the Kalman filter update, but is the main criterion for data association:

$$\epsilon = Y_t^{\mathrm{T}} S_t^{-1} Y_t, \tag{16}$$

Lastly, if it is found, that there is the best match between the track-measurement pair, the Kalman filter update step is finished by calculating the posterior state and covariance:

$$X_{t|t} = X_{t|t-1} + K_t Y_t, \tag{17}$$

$$P_{t|t} = (I - K_t H) P_{t|t-1}, \tag{18}$$

here I is mxm identity matrix. The best match between track and measurement is found by comparing ϵ for each prefiltered pair. It is possible to fail to find a matching pair if all ϵ are larger than some predefined threshold τ. In such a case, a new track state for the track without measurement:

$$X_{t|t} = X_{t|t-1}, \tag{19}$$

$$P_{t|t} = P_{t|t-1}. \tag{20}$$

Track merging is performed if (similarly to (16)):

$$D = (X_{1t} - X_{2t})^{\mathrm{T}} P^{-1}_{\mathrm{match}} (X_{1t} - X_{2t})) < D_{\mathrm{thresh}}, \tag{21}$$

where X_{1t} and X_{2t} are states of two tracks to be compared, P_{match} is a diagonal matrix, can be treated as typical variances of track state elements (with large values of P_{match} more tracks will be matched), D_{thresh} is a threshold for matching.

With the main parts of the data association and track update discussed, global view of data fusion can be defined (see Algorithm 1).

During the step of managing unmatched tracks, innovation error of track (normally calculated as (16)), is increased. In current implementation following empirically found formula is used:

$$\epsilon = \epsilon * \max(1 + \sqrt{dt}/3, 1 + 6dt/T_{\mathrm{tr}}), \tag{22}$$

here T_{tr} is the age of the track. The idea is to increase ϵ for new tracks faster, that older tracks, as the track age usually shows, how reliable the current track is.

The age of track is increased if there is VMD or Radar detection. If detection was from previous measurement (contrary to measurements, which came from both sources on the same processing step) it is increased by the time difference between current and previous measurements. If track is picked up after an absence of matching measurements, delta time is added based on the type of update (frame update time for VMD or frame update time for Radar). Tracks without current matching measurements do not update age value.

For VMD only state (no recent radar detection) previous angular movement is used along with size to create view space gating for measurement to tracks matching. It is part of measurement prefiltering, mentioned earlier. If angular velocity isn't initialized (the track age is low), the maximum possible rate of movement based on typical size and velocity ratio is used to create view space gate.

For a track, having recent Radar detection or overall high radar detection duration, the exact estimated position is calculated and new detections of VMD and Radar are projected on common space to use spatial gating.

Tracks are created/initialized with every moving object detection. For radar detection movement condition is non-zero Doppler velocity by default or can

```
/* merging of existing tracks                  */
for each track t in memory do
    for each track t2 in memory except t do
        apply (21);
        if (21) condition met then
            merge t and t2;
            remove t2 from memory;
        end
    end
end
/* updating of tracks by
   measurement                                 */
if have a new measurement of any source then
    for each track t in memory do
        estimate prior state (3) and covariance
          (6) prefilter list of possible
          measurements;
        for each measurement with eps small
          enough do
            if measurement was already used
              then
                check for more precise updates
                  in earlier tracks;
                if no better found then
                    update Kalman state (17),
                      (18);
                    push updated state to stack
                      of potential updates;
                    mark measurement as used;
                end
            else
                update Kalman state (17), (18);
                push updated state to stack of
                  potential updates;
                mark measurement as used;
            end
        end
    end
    for each track t in memory do
        apply the best update from the updated
          states stack;
    end
end
/* manage unmatched tracks                     */
for each unmatched track do
    increase track innovation error;
    if innovation error reaches threshold then
        remove a track from memory
    end
end
/* create potential tracks                     */
for each unmatched measurement do
    if satisfies movement conditions then
        create a track with an initial state and
          initial covariance;
    end
end
```

Algorithm 1: Data fusion algorithm

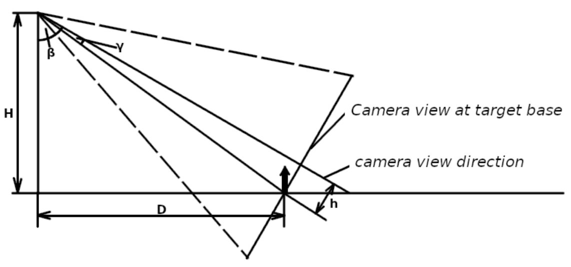

Figure 3: Distance estimation from camera

be set as one of the many algorithm parameters. All VMD detections treated as moving by definition. After the creation track is in a non-validated state. Tracks are considered to be validated after predefined age depending on the following properties:

1. Existence or absence of recent VMD/Radar detection. Radar detections (without VMD detection) having a higher impact on validation threshold than another way around.
2. Track innovation error.
3. Total previous duration of detections for radar and VMD
4. Trajectory type - mostly tangential movement with radar only detections should be verified for a longer duration.
5. Velocity thresholds.

Tracks to be removed from potential tracks list if:

1. Track innovation error grows too large. It is calculated based on new measurements and also incremented after the absence of radar detection (22).
2. The track is created by VMD detection and visual-only innovation error grows to large. It is calculated from angular measurements only and updated similarly to (22).

4.2. Distance Estimation from a Camera

Distance from camera measurements is calculated based on assumptions, that:

1. target is not blocked by some other objects
2. height and elevation of the camera are known
3. detection is of ground-based targets

The main features used for calculations are presented in Fig.3. Algorithm:

1. Calculate angle to the base of target based on the bottom edge of VMD detection and knowledge of VFOV of the camera as:
$$\gamma = \frac{h\mathrm{VFOV}}{n_v}, \tag{23}$$
where h is projection of position, where target touches the ground on camera view, n_v - vertical resolution (number of pixels along the vertical axis of image).
2. Subtract this angle from the angle formed by straight-up direction and camera "looking" direction:
$$\beta' = \beta - \gamma. \tag{24}$$
3. Calculate distance, by knowing one side of the triangle (height of camera) and the angle between this side and hypotenuse:
$$D = H \tan \beta'. \tag{25}$$

There are other ways to estimate the distance to the target from the camera only. For example, if the target is identified, the relative size of the target on the image could signal distance. That requires, however, a feedback loop between the fusion module and the recognition module.

4.3. Tracks Fusion

A general strategy for tracks fusion can be summarized as follows:

1. Both sources can create fusion tracks with another component not present until the match will be found at a later stage.
2. Input tracks are not verified against Kalman state estimate (no data association), because this step is already done in the tracker module.
3. VMD and radar tracker tracks can be merged if merging requirements are met. From such moment fusion track has both components.
4. The track can be split, if it is detected, that visual and radar data diverge too much. Track splitting/merging is performed at each data update step.

Additionally to track structure discussed in the previous section following fields are defined for tracking state:

1. Fused track ID (different from components).
2. *visualSeparated*- boolean value, which shows, that track recently had a visual component but lost it due to diverging of visual and radar.

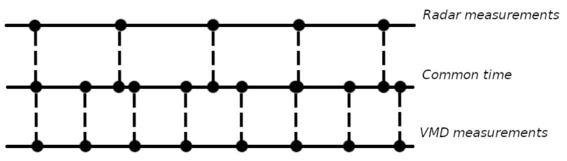

Figure 4: Time alignment of radar and VMD tracks for matching

3. VMD track ID. With new data updates, the degree of matching between fused tracks components is first checked for stored best previous matches
4. Radar tracker track ID. Same as above.

The track fusion algorithm is overviewed in Algorithm 2. The operation of mismatch calculation, used in algorithm description on many occasions can be explained looking at Fig. 4. First, measurement timestamps are created for all entries of both types of tracks. Then estimations for matching are calculated by interpolation (or extrapolation, if on edges). Average azimuth mismatch is used as a matching parameter. The early exit of the matching function is possible if mismatch grows to a predefined value.

The age of track is updated as per data fusion with each new radar tracker output considered as new radar measurement with time step equal to radar update duration. Track time out is increased, if no measurements were added to track. This step is the same for component tracks and the fused track. Time out for deletion calculation, mentioned in Algorithm 2, is calculated based on the current number of tracks. It is defined as 3 s if the number of tracks is less than N_{max} - the maximum number of tracks. If, on the other hand, the number of tracks is higher, allowed time out reduces:

$$T_{\text{timeout}} = 3\frac{2N_{\text{max}} - N_{\text{cur}}}{N_{\text{max}}}. \tag{26}$$

This assures, that all tracks are cleared if N_{cur} reaches $2N_{\text{max}}$. If a number of tracks for some reason grows more than $2N_{\text{max}}$, all tracks are cleared.

Tracks are created/initialized with every moving object detection from radar and every VMD detection. After creation, VMD track is in non-validated state, but track, created from radar tracker data directly, is in a validated state. Fused track having both components can be split, if VMD measurements diverge from tracker output too much. *visualSeparated* is set to *true* in this case. It is done to prevent the reacquisition of the same VMD track with a high mismatch factor. VMD only part of such split inherits range data and all

```
/* updating tracks structures                    */
if have a new radar tracker frame then
    for each track in frame do
        if The ID of a track can be found in already existing then
            append a new measurement;
        else
            create a new list of track entries with new ID;
        end
    end
end
if have new frame of VMD tracks then
    same as for radar tracks;
end
/* matching of tracks                            */
for each fused track do
    calculate mismatch of radar and video;
    for all non-fused tracks of both types do
        calculate mismatch with appropriate (radar vs. VMD) track;
        if a better match found then
            assign a new component to fused;
            release the previous component as non-fused;
        end
    end
end
/* generation of tracks                          */
for each combination of radar and VMD track do
    calculate mismatch; if mismatch small enough then
        create a new fused track with matched components
    end
end
/* destruction of tracks                         */
calculate time out of track for deletion;
delete all tracks with higher time out than allowed;
/* tracks state update                           */
for each fused track do
    if any of the track components received updates then
        a full Kalman filter update
    else
        update the state as an estimation only (17),(18)
    end
end
```

Algorithm 2: The track fusion algorithm

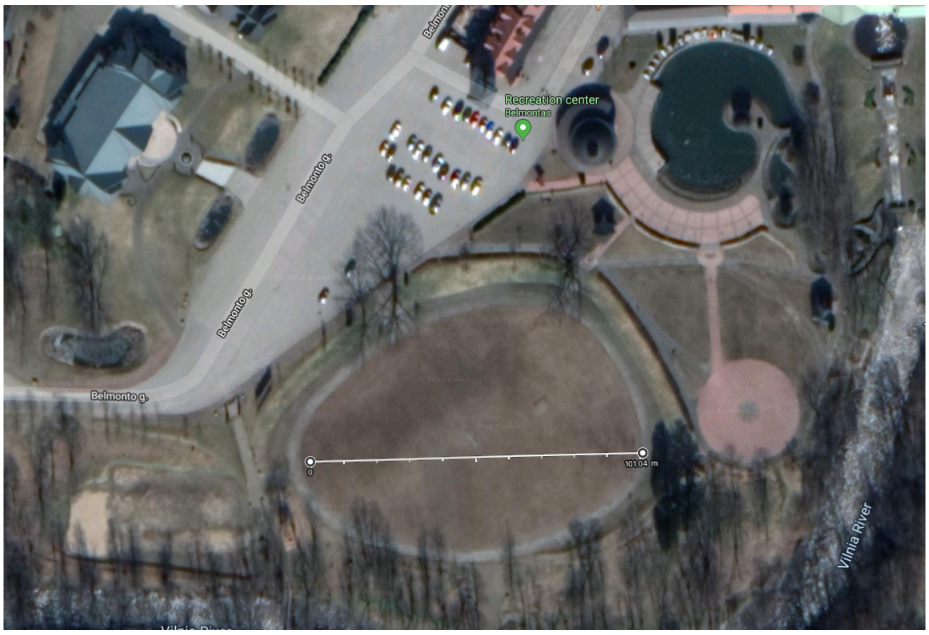

Figure 5: Area of testing for fusion evaluation. Position of equipment marked as 0 distance. Distance from equipment to one of the points of trajectory is shown

data, which is relevant to durations of detections and age of the track. It can be matched again later after *visualSeparated* expires. The split tracks get invalidated for some short duration (less than second) by setting it's innovation error parameter to some high value and gradually reducing it after new measurements.

5. Experimental Setup

The area, selected for experiments is shown in Fig. 5. The selected area allows the testing performance of tracking on big enough distance (more than 100 m) with parts of trajectories being almost exactly tangential while moving around the edge of the stadium. A small amount of moving background (cars, people, trees) allows more control over experiments.

Three experiments are performed:

1. A person moving clockwise around the edge of the stadium without stopping
2. A person moving counter-clockwise around the edge of the stadium, stopping, then proceeding
3. A person moving clockwise around the edge of the stadium, then changing the moving pattern from mostly tangential to radial

An example frame of video with detection displayed is presented in Fig. 6. Data were acquired using the NXP iMX6 SoM based embedded system with a quad-core 1.2 GHz processor. Video recording and radar data acquisition was performed simultaneously. Time synchronization was assured by knowing the starting time of video and frame rate and storing radar raw data or radar tracks with exact timestamps. Although the discussed algorithms were not running at the time of these measurements, close to real-time performance is achieved later using the same hardware.

6. Results of Experimental Investigation

Experimental investigation was subdivided into two stages. First, preliminary experiments were performed to select the state model and update strategy for fused tracks. During the second stage, the tracks fusion and the data fusion approaches were validated.

6.1. Comparison of Cartesian Coordinates and Polar Coordinates for KF State Representation

The impact of the selected system of coordinates to performance is presented in (a) - (c) pictures in Fig. 7. KF state representation by Cartesian or Polar coordinates produces very close results and a visual separation of results is hardly noticeable. In Table 1, the results of models, in which Polar or Cartesian coor-

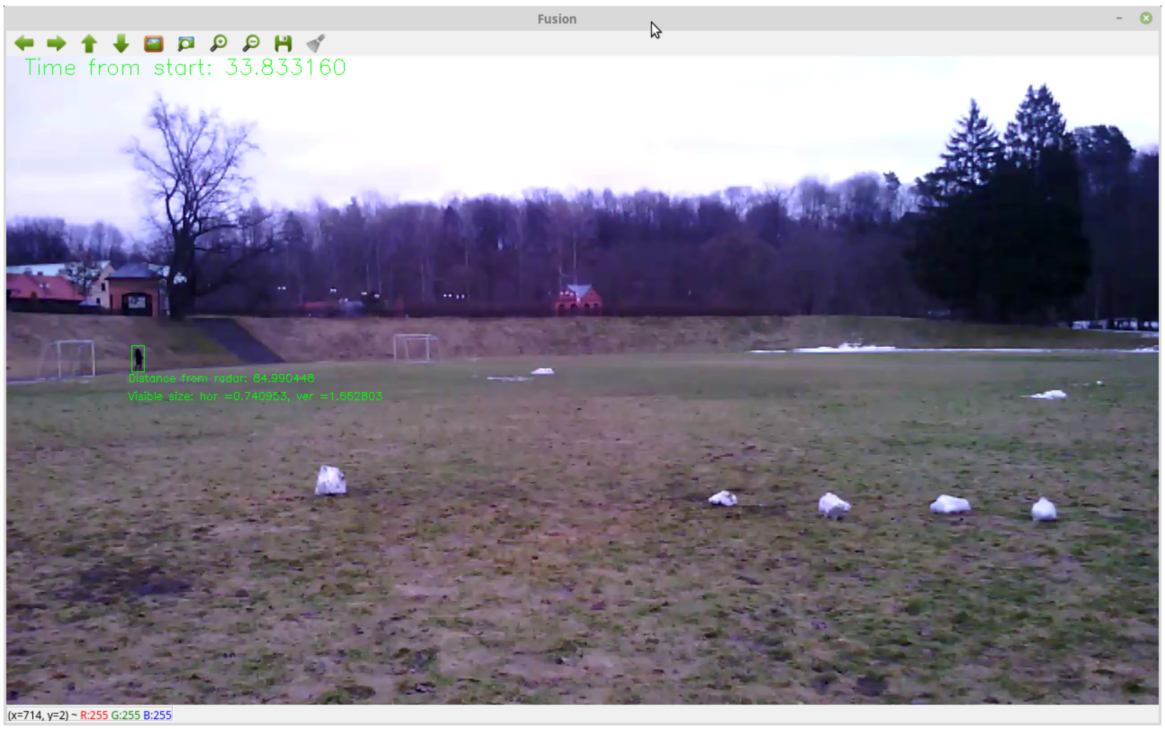
Fusion
Time from start: 33.833160
Distance from radar: 84.990448
Visible size: hor =0.740953, ver =1.662803
(x=714, y=2) ~ R:255 G:255 B:255

Figure 6: Example frame of first sequence

Table 1
Comparison of performance of models in which Polar or Cartesian coordinates ar used for KF state representation using MSE

Trajectory	Measured error	KF error Cartesian	KF error Polar
T1	2.4647	0.85995	0.87358
T2	2.2787	0.62444	0.58846
T3	2.7926	0.60758	0.63428

Table 2
Comparison of performance of models with/without adding camera to state using MSE

Trajectory	Measured error	KF error Polar	Fusion error
T1	2.4647	0.87358	0.72844
T2	2.2787	0.58846	0.51873
T3	2.7926	0.63428	0.58261

dinates are used for KF state representation performances, are presented. Since results are very similar, it can be concluded, that there is no significant difference in different KF state representations. To obtain error for each model, 10 simulations were performed for each and mean MSE calculated.

6.2. Accuracy of Filtering with Camera Data added

The resulting performances of models with/without adding camera to state update, represented through MSE, are shown in Table 2. It can be observed, that KF state updated using camera output represents ground truth more accurately than updated by radar data only. As before, mean MSE is calculated by running simulation 10 times for each trajectory.

6.3. Accuracy of Filtering with Distance Calculation from Camera Data

Results of fusion error and fusion with distance models performance evaluation using MSE statistics are presented in Table 3. The best minimum values as well as mean values of MSE shows, that fusion with distance evaluation model performance outperforms model without distance evaluation performance.

An example of a typical simulation run with different models evaluated is shown in Fig. 8

6.4. Fusion Methods Evaluation

The main metrics for evaluation of two fusion approaches were Object count accuracy (OCA) and FAR. OCA is

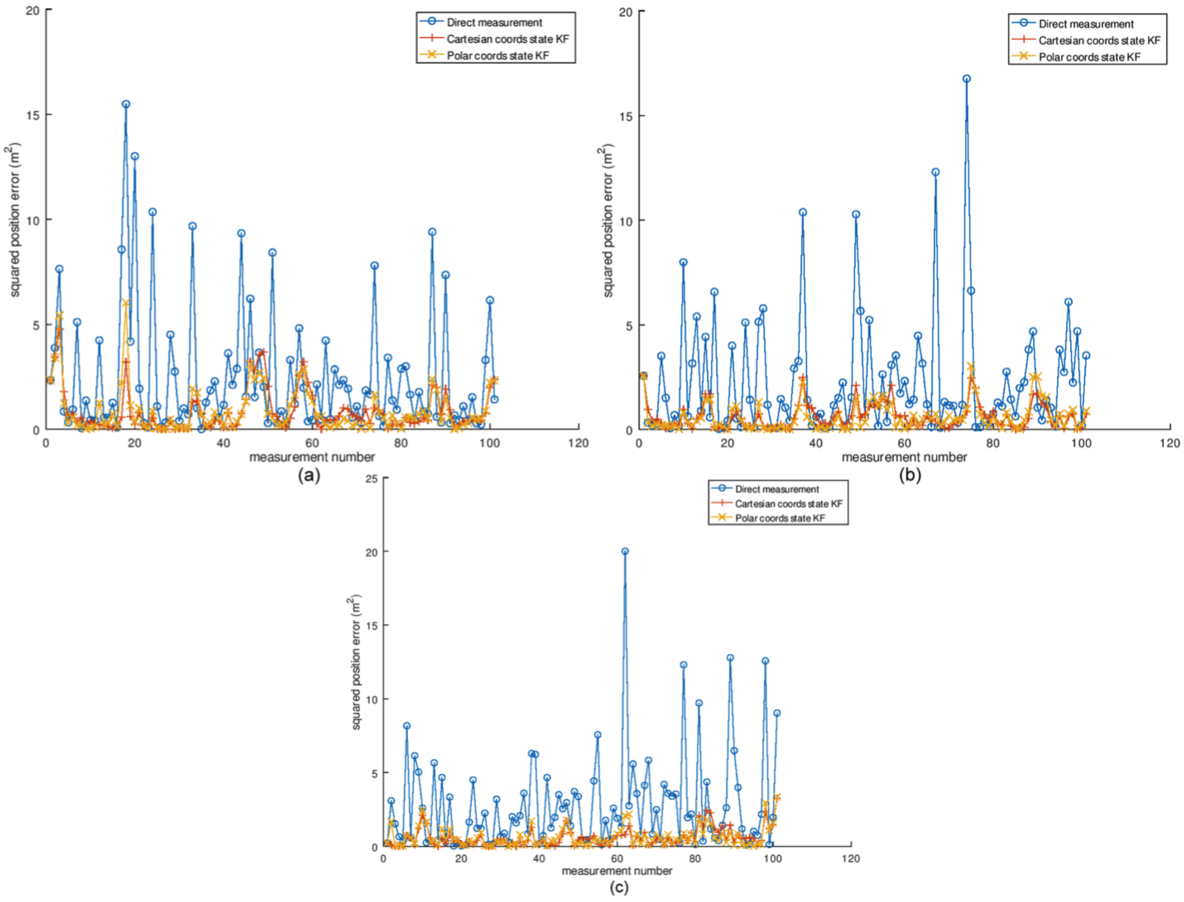

Figure 7: Comparison of performance of models in which Polar or Cartesian coordinates for KF state representation are used: (a) – test trajectory T1, (b) – test trajectory T2, (c) – test trajectory T3.

Table 3
Fusion error and fusion with distance estimation error comparison using MSE

Trajectory		MAX	MIN	MEAN	STD
T1	Measured error	3.0179	2.2735	2.6342	0.2677
	KF error	0.8749	0.4311	0.6654	0.1456
	Fusion error	0.8065	0.3300	0.5491	0.1471
	Fusion with DIST error	0.8438	0.2631	0.5346	0.1720
T2	Measured error	2.8525	1.9410	2.4447	0.3272
	KF error	1.1779	0.4676	0.7667	0.2336
	Fusion error	0.9825	0.3526	0.5915	0.2170
	Fusion with DIST error	0.9841	0.3333	0.5543	0.2203
T3	Measured error	3.0052	1.5744	2.3672	0.4143
	KF error	0.9347	0.3607	0.6654	0.1919
	Fusion error	0.7708	0.2087	0.5219	0.1843
	Fusion with DIST error	0.6282	0.1687	0.4243	0.1536

defined as

$$\mathrm{OCA}_t(P_t^G, P_t^D) = \frac{\min(M_t^G, M_t^D)}{\frac{M_t^G + M_t^D}{2}}, \tag{27}$$

where P_t^G and P_t^D are sets of ground truth points and detected points in measurement frame t respectively, M_t^G and M_t^D are quantities of ground truth and detected instances respectively. Overall OCA is defined as the average OCA of all frames of measurements.

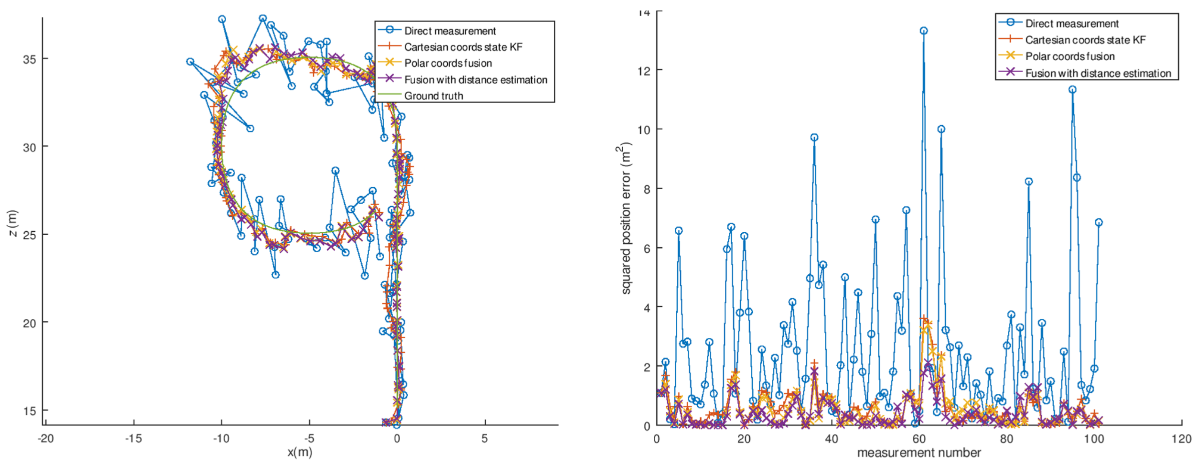

Figure 8: Typical results of simulation run: (a) – measured positions (T3 trajectory), (b) position estimation squared error

Table 4
Data fusion and tracks fusion evaluation results

	Radar tracker		Data fusion		Tracks fusion	
Sequences	mean OCA	FAR	mean OCA	FAR	mean OCA	FAR
1st	0.5034	0	0.9935	0	0.9839	0
2nd	0.2771	0	0.8909	0	0.8832	0
3rd	0.907	0.0135	0.9800	0	0.9360	0

FAR is defined as number of frames with false tracks, divided by total observed frames:

$$\mathrm{FAR} = \frac{N_\tau}{N}. \tag{28}$$

Any false track appearing in the frame constitutes to given frame becoming a false positive.

The focus of the experimental investigation was on elimination of false detections and reduction of missed detections rate. Progress towards both goals can be successfully monitored using selected metrics [25, 26]. Rather conservative policies for tracks validation were chosen for both versions of fusion to highlight the possibility to avoid false alarms while still keeping high enough detection rate (indirectly shown by OCA) for all practical purposes. Evaluation results are presented in Table 4. Best results are obtained by Data fusion and close to the best results are obtained by Tracks fusion.

7. Conclusions

1. It was observed experimentally, that radar only tracking suffers from many missed detections, if the target trajectory is close to tangential.
2. While radar only tracker performs without false alarms during the first two sequences, it is demonstrated with the third sequence, that target direction changes can cause false tracks to appear.
3. Two issues, mentioned above, can be solved with any of two fusion of radar and camera approaches, as it is seen from evaluation results. OCA increased drastically in both cases compared to radar only tracking.
4. Data fusion offers slightly better performance, reflected by higher OCA values. In practice, it means faster track validation and more robust tracking with missed detections from either VMD or radar.
5. The addition of distance measurements from the camera didn't prove to be stable method for tracks matching or state updates in practice. Although simulation was suggesting accuracy improvement, real measurements were highly unstable while using this approach.

8. Acknowledgments

Research is partially funded by Lithuanian Research Council Project Nr. 09.3.3-ESFA-V-711-01-0001, and partially funded by Lithuanian Business Support Agency Project Nr. 01.2.1-LVPA-T-848-01-0002.

References

[1] F. Castanedo, A review of data fusion techniques, The Scientific World Journal 2013 (2013).

[2] C. Napoli, E. Tramontana, M. Wozniak, Enhancing environmental surveillance against organised crime with radial basis neural networks, in: 2015 IEEE Symposium Series on Computational Intelligence, IEEE, 2015, pp. 1476–1483.

[3] Y. B. Shalom, Multitarget-multisensor tracking: advanced applications, Artech House, Boston, MA (1990).

[4] Y. Bar-Shalom, X.-R. Li, Multitarget-multisensor tracking: principles and techniques, volume 19, YBs Storrs, CT, 1995.

[5] D. Willner, C. Chang, K. Dunn, Kalman filter algorithms for a multi-sensor system, in: 1976 IEEE Conference on Decision and Control including the 15th Symposium on Adaptive Processes, IEEE, 1976, pp. 570–574.

[6] N. Kaempchen, K. Dietmayer, Data synchronization strategies for multi-sensor fusion, in: Proceedings of the IEEE Conference on Intelligent Transportation Systems, volume 85, 2003, pp. 1–9.

[7] E. A. Wan, R. Van Der Merwe, S. Haykin, The unscented kalman filter, Kalman filtering and neural networks 5 (2001) 221–280.

[8] D. Laneuville, C. Jauffret, Recursive bearings-only tma via unscented kalman filter: Cartesian vs. modified polar coordinates, in: 2008 IEEE Aerospace Conference, IEEE, 2008, pp. 1–11.

[9] J. Lian-Meng, P. Quan, F. Xiao-Xue, Y. Feng, A robust converted measurement kalman filter for target tracking, in: Proceedings of the 31st Chinese Control Conference, IEEE, 2012, pp. 3754–3758.

[10] S. V. Bordonaro, Converted measurement trackers for systems with nonlinear measurement functions, Ph.D. thesis, The school of the thesis, Doctoral Dissertation, 2015.

[11] S. Blackman, R. Popoli, Design and analysis of modern tracking systems(book), Norwood, MA: Artech House, 1999. (1999).

[12] D. Y. Kim, M. Jeon, Data fusion of radar and image measurements for multi-object tracking via kalman filtering, Information Sciences 278 (2014) 641–652.

[13] X. Wu, J. Ren, Y. Wu, J. Shao, Study on target tracking based on vision and radar sensor fusion, in: WCX World Congress Experience. SAE International, 2018.

[14] G. Alessandretti, A. Broggi, P. Cerri, Vehicle and guard rail detection using radar and vision data fusion, IEEE transactions on intelligent transportation systems 8 (2007) 95–105.

[15] S. Sugimoto, H. Tateda, H. Takahashi, M. Okutomi, Obstacle detection using millimeter-wave radar and its visualization on image sequence, in: Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004., volume 3, IEEE, 2004, pp. 342–345.

[16] G. P. Stein, O. Mano, A. Shashua, Vision-based acc with a single camera: bounds on range and range rate accuracy, in: IEEE IV2003 Intelligent Vehicles Symposium. Proceedings (Cat. No. 03TH8683), IEEE, 2003, pp. 120–125.

[17] F. Liu, J. Sparbert, C. Stiller, Immpda vehicle tracking system using asynchronous sensor fusion of radar and vision, in: 2008 IEEE Intelligent Vehicles Symposium, IEEE, 2008, pp. 168–173.

[18] F. Beritelli, G. Capizzi, G. L. Sciuto, C. Napoli, F. Scaglione, Rainfall estimation based on the intensity of the received signal in a lte/4g mobile terminal by using a probabilistic neural network, IEEE Access 6 (2018) 30865–30873.

[19] A. Sole, et al., Solid or not solid: Vision for radar target validation, in: IEEE Intelligent Vehicles Symposium, 2004, IEEE, 2004, pp. 819–824.

[20] C. Kreucher, S. Lakshmanan, K. Kluge, A driver warning system based on the lois lane detection algorithm, in: Proceedings of IEEE international conference on intelligent vehicles, volume 1, Stuttgart, Germany, 1998, pp. 17–22.

[21] R. Deriche, O. Faugeras, Tracking line segments, Image and vision computing 8 (1990) 261–270.

[22] S. D. Gupta, J. Y. Yu, M. Mallick, M. Coates, M. Morelande, Comparison of angle-only filtering algorithms in 3d using ekf, ukf, pf, pff, and ensemble kf, in: 2015 18th International Conference on Information Fusion (Fusion), IEEE, 2015, pp. 1649–1656.

[23] D. V. Stallard, Angle-only tracking filter in modified spherical coordinates, Journal of Guidance, Control, and Dynamics 14 (1991) 694–696.

[24] T. Fortmann, Y. Bar-Shalom, M. Scheffe, Sonar tracking of multiple targets using joint probabilistic data association, IEEE journal of Oceanic Engineering 8 (1983) 173–184.

[25] F. Beritelli, G. Capizzi, G. Lo Sciuto, C. Napoli, M. Woźniak, A novel training method to preserve generalization of rbpnn classifiers applied to ecg signals diagnosis, Neural Networks 108 (2018) 331–338.

[26] G. Capizzi, G. Lo Sciuto, C. Napoli, D. Polap, M. Wozniak, Small lung nodules detection based on fuzzy-logic and probabilistic neural network with bioinspired reinforcement learning, IEEE Transactions on Fuzzy Systems 28 (2020) 1178–1189.