AVITRACK

Contract n° AST3-CT-2003-502818

Internal Technical Note

Data Fusion Report

1.0 – Draft 1

IN_AVI_2_014

Contract Number : AST3-CT-2003-502818
Document Title : Data Fusion Report
Document version : 1.0
Document status : Draft 1
Date : 4-April-2005
Availability :
Authors : David Thirde, Mark Borg (UoR)

Abstract Data Fusion work tos date.

Keyword List Data Fusion, Data Association, Multi-Camera System, Multi-View Tracking.

DOCUMENT CHANGE LOG

Document Issue.	Date	Reasons for change
1.0 Draft 1	04/04/05	Initial Release

APPLICABLE AND REFERENCE DOCUMENTS (A/R)

A/R	Reference	Title
	[1] [2] [3] [4] [5] [6] [7] [8]	

Table of contents

INTRODUCTION

This document describes the work performed so far on Data Fusion for the AVITRACK project. This work forms part of work packages D3.2 and D3.6.

The next section (Section 2), gives a brief overview of Data Fusion and introduces the general characteristics of Multi-Camera Tracking Systems that directly influence the data fusion process, together with issues and problems that need to be addressed. Section 3 will then briefly describe how existing systems in the literature go about to solve the data fusion problem. Section 4 will then introduce the work performed for AVITRACK, describe the 3D localisation issues, sensor uncertainty, the different algorithms used for data association, data fusion, track estimations and update rules, and filtering.

SYSTEM OVERVIEW

The data association, fusion and tracking algorithms presented in this document form part of the "Data Fusion" module, which is responsible for performing tracking using the measurements made from all the cameras. This module is used to process XML streams from 8 cameras concurrently. The tracking results of each camera are then sent in XML via CORBA communications to the "Long Term Tracking / Scene Understanding" modules.

The main architecture and algorithms for data fusion have been incorporated into the third delivery of the Data Fusion module, called AvitrackDataFusion v0.3 [?]. The unresolved tasks for data fusion are to make the data association probabilisitic, to add more in-depth reasoning about the splitting and merging of tracks, and also to incorporate feedback into the frame to frame tracker to improve the coherency of the output (by ensuring unique per-object labels at all stages of the tracking process).

The tracking algorithms were mostly tested with the following AVITRACK data sequences: S3-A320, S21-Vehicles and S4-A320, some results of which are presented in this document. A more formal evaluation, using ground-truth information and pre-defined evaluation criteria, will be done in the near future, as part of "Work Package 6.1 – Scene Tracking Evaluation".

MULTI-CAMERA SYSTEMS AND DATA FUSION – OVERVIEW & ISSUES

The main advantages of using a multi-camera tracking system, like in AVITRACK, is:

- *Occlusion Minimisation*. If a target becomes occluded in a camera view (by another target or scene elements), there is a higher probability of observing the same target with a different camera viewpoint, in which it is not occluded.
- A *larger Visible Area*. This consists of the combined area (network field-of-view) observed by all the cameras from their respective viewpoints. This results in targets being potentially observed for a longer period of time over a wider area.
- *Better 3D positions*. A more accurate and reliable 3D position can be computed for targets observed by multiple cameras.

The *Data Fusion* process combines the data seen by each of the individual cameras to maximise the useful information content of the scene being observed and hence achieving the above-mentioned advantages over single-camera systems. Data fusion also helps to minimise the volume of data generated by the many cameras and helps to reduce the bandwidth needed to send information to later processing modules.

Multi-camera architectures normally fall into 2 main categories:

- Cameras with overlapping fields-of-view (FOV). Targets can be potentially seen simultaneously by 2 or more cameras.
- Cameras with non-overlapping fields-of-view. A target exiting one camera's FOV, can later be observed in another.

The main issue for data fusion in the overlapping FOV case is to do the matching of targets observed simultaneously by multiple cameras – this is referred to as the *Data Association* problem. For the non-overlapping case, targets that exit the FOV of one camera and then enter the FOV of a second camera, need to be re-acquired – the so-called *Camera Hand-Off* problem. The AVITRACK system falls into the first category.

The data fusion process also depends on where the tracking module(s) is/are placed in the system. This can be configured as a [2]:

- *Centralised Tracking system*. Tracking is done immediately in 3D for all cameras concurrently.
- *Decentralised Tracking system*. Each camera does its own tracking independently using the image data (in 2D), and then sends its tracking results to the data fusion module.
- *Hierarchical Hybrid system*.

AVITRACK, like most existing systems, uses decentralised tracking. These can be further sub-divided into those that use *feedback* from the data fusion module to the individual camera trackers, and those that don't. It is also possible for feedback to occur between the camera trackers themselves. At the moment, it is not envisaged that feedback will be used for AVITRACK.

The type of *spatial registration* of camera image frames, also plays an important role in data fusion. Several methods are available, such as: using calibrated cameras, homography relations between 2 or more cameras, epipolar lines, uncalibrated cameras and FOV edge visibility, etc.

Another important issue for the data fusion process is whether the video output from the individual cameras is *synchronised* or not. This is especially important for decentralised systems, where each camera operates as an independent process. This can give rise to temporal drift between the image frames acquired by each camera, which will affect the fusion accuracy. To solve this, the data fusion module must either use methods that can handle non-synchronised tracking results or perform *temporal registration* of the tracking data.

In the case of AVITRACK, the cameras should be synchronised by the video server, but from an initial examination it appears that there might be a slight synchronisation problem. This is under investigation, but if present, will require awareness of the problem at data fusion level.

LITERATURE REVIEW

This section gives a brief review of some of the data fusion methods that are mentioned in the computer vision literature (listed in no particular order).

1. The system developed by Remagnino, Shihab and Jones [1], is a decentralised (see section 2) multi-camera system, with the software architecture of the system composed of agents (The use of software agents is the main focus of this paper). Each camera's tracker runs independently and it uses the camera's calibration information (which is learnt automatically from the scene), to map the image coordinates of tracked targets onto the 3D ground plane.

Data fusion occurs in a distributed fashion by the software agents created for each tracked object, which communicate with each other. Two tracked objects from different cameras match with each other if they have similar trajectories on the 3D ground plane. If the distance between the past *N* positions making up the 2 trajectories is less than some pre-defined threshold *T*, then the two objects are fused together. This paper uses the last 5 seconds of positions for trajectory comparison. This is a simple data association technique, but is effective, although it suffers from problems and is limited to handling the fusion of 2 objects at a time.

2. Much of the work on data fusion in computer vision is based on extensive work done previously in the radar-tracking domain. Several data fusion methods used in radar tracking have been applied successfully to computer vision applications. Bar-Shalom and Li [2], introduce several statistical techniques for data association. These are:

   - Nearest Neighbour Filter.

     This is the simplest data association method, where a tracked object from one camera is fused with the nearest neighbour from another camera (as determined from the 3D positions of the objects). A variation of this technique uses the strongest neighbour instead of the nearest one. This method suffers from problems when objects are too close to each other, cluttered scenes and when the tracking results are affected by measurement and detection errors (noise).

   - Joint Probabilistic Data Association Filter (JPDAF).

     The JPDAF method is more robust to 3D position and detection errors. It calculates the association probabilities for each detected object at the current time (measurements) with the known targets (tracked in previous frames), taking into account the potential presence of noisy measurements. These probabilities are then used to compute weighted measurements, and used to update the known targets. Because of this, the JPDAF is referred to as the “all neighbours” filter [2]. JPDAF performs quite well in the presence of scene clutter and when targets approach and pass each other.

   - Multiple Hypothesis Tracking (MHT).

     The MHT data association method considers all possible combinations of matches between the previous targets and the current measurements as detected by the cameras. These are arranged in a tree-like structure of hypothesis, and includes all possible new target initiations (i.e. targets that enter/appear in the scene for the first time). The MHT method performs quite well in tracking a large number of simultaneous targets, even if the correct association is not immediately made (example when targets are not visually separated).

3. The work by Ruan [3], describes 2 other data association techniques:

   - Probabilistic Multiple Hypothesis Tracking (PMHT).

     This is a generalisation of the MHT method [2], that eliminates the assumption that a target can generate only one measurement (observation). This is useful for cases where an object gets fragmented when seen by one of the cameras.

   - S-D Assignment.

     The S-Dimensional Assignment method considers the data fusion problem as a multi-dimensional matching problem, and a subset of all feasible measurement to known-target associations is found by minimising a cost function.

4. All the data association methods reviewed so far are based on fusing 'point targets', where tracked objects are represented as a single 3D point (usually set to the object's 3D centroid or object's 3D ground position). This is especially true of methods originating from work done in the radar domain. But vision-based tracking applications track extended objects and so other cues, apart from the 3D position, can be used during fusion. These cues can also be integrated together to get a better result. The object features selected as cues must be insensitive to the camera viewpoint to be useful for data fusion.

   The work by Collins et al. (VSAM project), [4] and [5], uses a combination of colour information, 3D position and coarse object classification to aid data fusion. Their application consists of a decentralised multi-camera system made up of different sensor types, including mobile and airborne cameras. In this project, data fusion is mainly used for solving the camera hand-off problem and for sensor slaving (a wide FOV camera controlling pan-tilt-zoom cameras).

   The 3D positions of tracked objects are computed by geolocation – intersecting viewing rays from a camera with a non-planar ground terrain model. Together with the 3D position, a variance measure is also calculated. This is determined from variation in the image coordinates of the tracked object over a period of time. The object's colour information consists of 3 coarse R, G, and B histograms. The histograms are normalised to handle different colour responses in the cameras. The classification information consists of the classes: person, group of persons, and vehicles.

   During data association, 3 match score functions are calculated for all potential measurements and hypotheses pairs (previously tracked objects). The 3D position score uses the distance between the predicted position of the hypothesis (using a simple constant velocity value) and the measurement's ground position together with the covariance value. The classification match score uses heuristics and pre-defined constants, while the colour match score uses a simple colour histogram difference. The 3 scores are then combined into a single value and the best pair is selected (strongest neighbour approach), if it is above a certain score threshold. If no match is found, then the measurement is assumed to be a new target.

5. The work by Turolla, Marchesotti and Regazzoni [6], uses a similar approach to the previous one. This paper also mentions that for extended targets, data association can occur at different levels: pixel level (colour), blob/object level (shape, corners) and event level (dynamics).

   In their implementation, the object features used as cues for the data fusion process are: target speed, 3D position and colour histogram. The speed is estimated using a median filter, while a Kalman filter is used for estimating the 3D position. During data fusion, the Euclidean distance metric is used for the 3D position, the Bhattacharyya coefficient is used to compare the colour histograms, while a vector metric is used for speed. The values are then thresholded and pre-defined association rules ('and', 'or', 'majority' rules) are used to select the best match.

6. In the previous two works, several target features were used as cues for data fusion and their related scores were integrated using score addition or association rules. One problem with combining scores in a linear way is that a poor feature score can degrade the scores of the other features (example, colour can become unreliable when a target moves from an illuminated to a shadow region), and data fusion may fail during these instances.

   To solve this problem, Hsu et al. [7], introduces a new data association method called Rank And Fuse (RAF). During the rank stage, all possible matches of measurements to hypotheses are enumerated and for each feature, a score is calculated using a similarity metric. Then during the fusion stage, all the rankings are combined together using rank correlation and rank combination.

   In their implementation, the object features selected for data fusion are the 3D position and the average colour. Their RAF system performs better than using a score combination method.

7. Nummiaro et al. [8], use a multi-camera system for a smart room application. In this system, data fusion is used for selecting the best view of people as they are tracked by multiple cameras in the room. Each camera has its own tracking module running on its own, but the trackers can also communicate with each other to exchange tracking results and to re-acquire targets if these are lost due to occlusion, clutter or when they leave the camera's FOV. The first part of data fusion is done at this stage.

   When cameras exchange tracking results, they use their epipolar geometry to determine where lost objects should be re-acquired or new objects should be seen. If a target is only seen in one camera, then it is searched in other camera views by taking samples along the single epipolar line. If seen in more than one camera, then it is searched in the remaining cameras by taking samples near the intersection of the multiple epipolar lines. To determine the correct association between targets seen in different cameras, the target's colour histogram is used together with the Bhattacharyya coefficient as the similarity metric. The same feature and metric is also during the selection of the best view for the tracked object. This system suffers from incorrect matches if there are several equally good candidates in the vicinity of epipolar lines (crowded scenes).

8. The system by Mittal and Davis [9], consists of a centralised and synchronised multi-camera system. Like the previous work, epipolar geometry is used for the data fusion process, which is done on pairs of cameras at a time. In this system, data fusion is performed at an early stage and before any target tracking occurs and uses low-level information.

   Motion regions of constant colour in one camera view are matched along the epipolar line with similar-coloured motion regions in the second camera view. The midpoints of the matched regions are then mapped to 3D points using calibration information and projected on to the ground plane. A Gaussian kernel is then used to add a probability value to a ground-plane probability map. This map is used to indicate the probability of the presence of an object at that point on the ground plane. If during fusion, a region in one camera matches with many regions of the same colour in the second camera view, then all are considered as potential matches and added to the probability map. The probability map is then thresholded and objects detected and tracked on the 3D ground plane.

   This system performs quite well in crowded scenes and the accumulation of probabilities in the probability map helps to reject outliers.

9. Snidaro et al. [10], use a decentralised multi-camera system for outdoor surveillance. For data fusion, they use a technique they call "measurement gating and assignment" (the term *gate* is borrowed from the radar domain and refers to that part of the multidimensional data space around the measurement in which a search is made). Given the known targets (known from previous frames), the data fusion process will compare each measurement with the predicted 3D position of the known targets. Only those measurements which fall within the gating distance of the predicted position are considered for fusion. The gating distance is derived from the normalised (Mahalanobis) distance and the predicted state of the target.

   Because of the variability in weather conditions in outdoor environments, this paper also introduces an Appearance Ratio metric that dynamically measures sensor reliability. The camera closest to the target may not always be the best sensor under adverse weather conditions (example, fog) and the appearance ratio is higher for cameras in which the target has a stronger visibility. Then, during data fusion, the final matching score is calculated from a combination of the gating distance metric and the appearance ratio. The measurement with the best score is fused with the target.

10. Most of the reviews so far, have either used synchronised camera systems or else assumed synchronisation. But, as mentioned in section 2, a temporal drift can occur between cameras running independently. For fast moving targets, such as vehicles, this temporal drift can create large differences in 3D positions of the target as seen by different cameras and may cause problems for data fusion. Ohya, Utsumi, and Yamato [11 chapt.2], describe how multiple measurements from non-synchronous cameras can be integrated together by using a Kalman Filter algorithm. Their system is used to track multiple persons.

    Each camera has its own processing module, which the authors call the Observation Node. It detects feature points for the persons it observes – these are the points with highest value after the distance transform is applied to the binary motion image. Then it matches these feature points, against the tracking models of already known targets, using the Mahalanobis distance. This data association is done in a decentralised fashion by the observation nodes.

    After matching, the measurements from the different cameras and with different timestamps are sent to a central tracking module, which feeds the measurements for a particular target to its Kalman filter, and the Kalman filter in turn updates its tracking state. The Kalman filter will then generate new predicted positions to be used in the next matching step and sends these to the observation nodes.

    Amongst the advantages of using non-synchronous cameras, quoted by the authors, one can find: no need for mechanisms for synchronising the cameras; the multi-camera system is more scalable; and each camera can run at its own processing rate unhindered by the other cameras.

11. The system by Black, Ellis and Rosin [12] and Ellis et al. [13], is also using non-synchronised cameras. But the authors use a temporal alignment technique to actually register the frames from the cameras. This needs to be performed only once at the beginning and uses a least median of squares method to geometrically align object tracks and generate a time offset for pairs of cameras.

    For tracking objects, a 2-level Kalman filter-based tracking is used. Objects are first tracked by a Kalman filter in 2D image coordinates with the object state consisting of the image position and velocity. The minimum Mahalanobis distance between the 2D Kalman prediction and the observed position are used to match the observations with the known targets for that particular camera. The 3D position is then estimated from the 2D Kalman state by using the ground plane constraint and the camera calibration information. The second level Kalman filter tracks objects in 3D and performs the data fusion step, by combining all the 3D positions for a certain target reported by each camera tracker.

    This system also describes a method for estimating the 3D measurement uncertainty by combining the uncertainty of the camera calibration (higher error, the farther away from the camera) and the tracked 2D object states. This uncertainty is expressed as a 2D image covariance and is projected on to the ground plane to get the 3D uncertainty.

12. The multi-camera system by Jiao et al. [14] uses a similar approach to the previous system, with a 2-level Kalman filters for tracking and data fusion and it also makes use of non-synchronous cameras. Temporal registration is performed by observing a 3D trajectory, an invariant signature of the 3D trajectory is determined from its 2D projections, followed by correlating these invariant trajectories from the different cameras.

    The lower-level 2D Kalman filter takes the image trajectory as the observation and uses the image positions, velocity and acceleration, expressed in the local camera reference frame, as the object state. Then the top-level 3D Kalman filter receives the image positions, velocity and acceleration from the individual 2D Kalman filters as its input and fuses the data to get the 3D state of the tracked object.

A feedback mechanism is included so that the top-level module (the one doing the data fusion) sends fused data back to the lower-level camera trackers. This feedback is used to handle cases where individual camera trackers lose track of objects when they are occluded or out of the camera's FOV. And on top of the 2D Kalman filters, a layer of multiple hypotheses verification is added to handle these lost/occluded cases that normally cause the 2D Kalman filter to fail. The last known fused position (received from the feedback from the 3D Kalman) is then used and backprojected on to the image to try and re-acquire the object.

13. Dockstader and Murat Tekalp [15] also use a 2-level Kalman Filter system, but with the addition of a Bayesian Belief Network which does the actual data fusion. The low-level camera trackers use sparse feature points and their corresponding motion estimate and track these points using 2D Kalman filters. The output of these trackers is then sent to the data fusion module, which consists of the Bayesian belief network, and on top of it, a 3D Kalman filter.

    The Bayesian belief network uses a probabilistic weighting scheme for data association and fusion. It fuses independent observations from multiple cameras by iteratively resolving independency relationships and confidence levels within the graph structure of the network. The output of the Bayesian belief network is the most likely 3D state estimates given the available data, and is sent to the top-level 3D Kalman filter. This filter helps to smooth out 3D trajectories and is also used to generate predicted target positions that can be fed back into the low-level camera trackers.

14. The multi-camera system by Xu, Lowey and Orwell [16] is used track football players. It uses a 2-level Kalman filter configuration. The lower-level 2D Kalman filter uses the image position and the bounding box for the object's state. The covariance is used to generate an estimate of the measurement error and this is projected on to the ground plane to get the related 3D measurement error. The 3D ground positions and their related error from the individual cameras, are then sent to the top-level multi-view module to be fused together

    The multi-view tracking module uses a 3D Kalman filter to represent the state of established targets i.e. targets tracked in previous frames. Data fusion is performed by associating the observations with the established targets using an association matrix. This association matrix is populated with scores obtained from the Mahalanobis distance between the measured positions and the predicted positions and also using the measurement uncertainties. The nearest neighbour algorithm is then used to do the matching.

    Any remaining measurements that are not matched to existing targets, are considered to be new objects if they appear in more than one camera and are within a certain distance of each other on the ground plane.

15. The multi-camera system by Kang, Cohen and Medioni [17], performs simultaneous tracking in 2D (image coordinates) and 3D (ground plane positions) using a centralised tracking system. To be able to do this, the images from all the cameras are pre-registered together using a ground plane homography.

    For tracking, a set of 3 probabilistic models is defined – a 2D image motion model, a 3D ground plane motion model and an appearance model using colour information. The motion models are specified using a Kalman filter, while the appearance model is structured in a way to make it invariant to 2D rigid transformations. The joint probability model is then defined to be the product of the above 3 probability models.

    The tracking problem is then expressed as computing the optimal position in the 2D image and the 3D world coordinates of the moving object by maximising all the probability models. This is done through the use of the Joint Probability Data Association Filter (JPDAF).

Table 1: Main characteristics of reviewed papers.

No.	Main characteristics	Ref
1.	Data association: 3D Trajectory similarity.	[1]
2.	Data association: nearest neighbour, JPDAF, MHT methods.	[2]
3.	Data association: PMHT, S-D Assignment methods.	[3]
4.	Data association: 3D position + colour + object classification, cost functions, heuristic rules.	[4], [5]
5.	Data association: 3D position + colour + speed, distance metrics, association rules.	[6]
6.	Data association: 3D position + average colour, RAF method.	[7]
7.	Decentralised trackers exchange results and target re-acquisition, Data association: epipolar geometry, colour histogram.	[8]
8.	Low-level data fusion occurs before tracking, Data association: epipolar geometry, motion regions with constant colour, ground-plane probability map, Centralised tracking performed in 3D on the ground plane.	[9]
9.	Data association: 3D position, gating distance, appearance ratio.	[10]
10.	Non-synchronised multi-camera system, Data association: 3D position, Kalman filter prediction, Mahalanobis distance, Centralised tracking module using Kalman filter, with feedback to observation nodes.	[11]
11.	Temporal registration for non-synchronous cameras, 2-level Kalman Filters, Data association: 3D position, Kalman filter prediction, Mahalanobis distance.	[12], [13]
12.	Temporal registration for non-synchronous cameras, 2-level Kalman filters with feedback, Data association: 3D position, Kalman filter prediction.	[14]
13.	2-level Kalman filters with feedback, Data association: Bayesian Belief Network.	[15]
14.	2-level Kalman filters, Data association: 3D position, Mahalanobis distance, Nearest Neighbour.	[16]
15.	Simultaneous 2D and 3D tracking, Data association: JPDAF mehod.	[17]

ALGORITHMS FOR DATA FUSION

4.1 3-D OBJECT LOCALISATION

Prior to the data fusion stage, the detected and tracked objects in each camera require accurate location in the 3-D world co-ordinates. The location of an object is determined to be the centre of gravity on the ground plane (i.e. in the x and y world co-ordinates). In this section we describe various methods used to attempt to estimate the location of the objects in the 3D world co-ordinates.

4.1.1 Bottom centre bounding box – This was the first approach applied to the AVITRACK project. The bottom centre of the bounding box was found to be only suitable for localising objects with negligible depth away from the camera.

4.1.2 Projected bottom centre bounding box – to alleviate the problem of localising objects with significant depth the bottom centre of the bounding box was projected vertically in the image to the pixel location at which motion has been detected. This generally improved the localisation for vehicles, but presented errors for objects where the boundary of the detected motion is non-rigid i.e. for people. A hybrid strategy was applied using the output of the classifier to distinguish people from other objects, after which the method presented in 4.1.1 or 4.1.2 was applied.

4.1.3 Angle of camera to object – to improve the localisation in a generic manner with few assumptions made about the object types the hybrid strategy presented in 4.1.2 was improved. For people objects the bottom centre of the bounding box is used as the point at which the object is localised on the ground plane, barring motion detection errors (e.g. shadow and reflection) this is a reasonable assumption. For the remaining object types the image point at which the centre of gravity is localised is chosen to be relative to the angle of the camera to the object. For a camera lying on the ground plane the centroid will be relatively close pixelwise to the bottom centre of the bounding box, whereas for an object viewed from directly overhead (i.e. $1/2\pi$ radians angle between the camera and object) the centre of gravity will be close (pixelwise) to the centre of the bounding box.

Using this observation a simple function was formulated to estimate the vertical position of the centroid in the image based on the (2-D) camera angle to the object. Taking a to be the angle measured between the camera and the object, the proportion p of the bounding box height (where $0 \le p \le 1/2$) was estimated as $p = 1/2(1 - \exp(-\lambda a))$ where $\lambda \equiv \ln(2)/(0.15 \times 1/2\pi)$. The horizontal position of the object centroid is taken to be the horizontal centre point of the bounding box, since this is generally an reasonable estimate.

4.2 SENSORY UNCERTAINTY FIELD

The location measured for an object inherently has some *uncertainty* about that measurement due to the sensor properties and the location estimation technique. For a generic camera the 3D measurement uncertainty (also known as the measurement noise covariance), $\mathbf{R}$, is estimated by propagating an image plane covariance to the world co-ordinate system at a given value of height for the world location.

This estimation of measurement uncertainty allows formal methods to be used to determine the association of observations originating from the same measurement, as well as providing mechanisms for fusing the observations into a single, estimated, measurement. For the measurement covariance $\mathbf{\Lambda}$ at location (x,y) in the image plane of camera c, the measurement uncertainty $\mathbf{R}(x_w,y_w,z_w)$ at a given height $z_w=0$ (i.e. the ground plane) in the world co-ordinate system is given by [18]:

$$\mathbf{R}\left(x_w, y_w, z_w\right) = \mathbf{J}\left(x_c, y_c\right) \mathbf{\Lambda} \mathbf{J}\left(x_c, y_c\right)^T$$

where $\mathbf{J}$ is the Jacobian matrix found by taking the derivatives of the two mapping functions between the image and world co-ordinate systems. The measurement uncertainty field is demonstrated in the figure below for camera 6, notice that the uncertainty becomes elongated perpendicular to the sensor in the far-field.

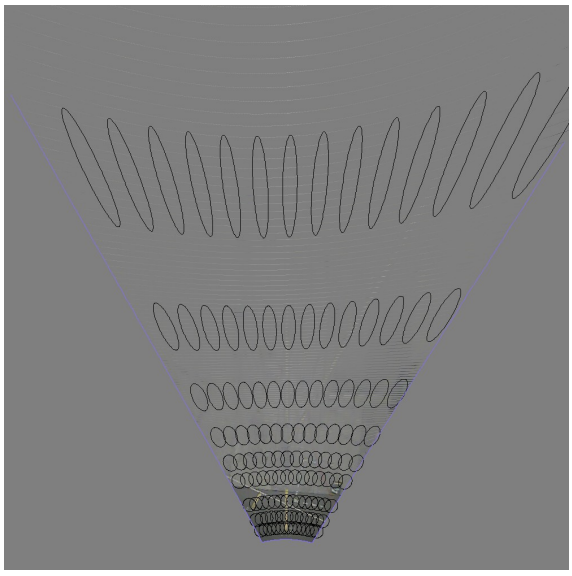

Figure 1 The Sensory Uncertainty Field for camera 6 computed using the method presented in Section 4.2.

The measurement covariance $\mathbf{\Lambda}$ can be biased with *a priori* knowledge about the expected uncertainty of the location estimation for a given object. For each detected object in the image plane the measurement uncertainty $\mathbf{\Lambda}$ is dependent on the pixel uncertainty, the dimensions of the object (i.e. more uncertain for larger objects) and the quality of the 2-D measurement (e.g. reflection and shadow can introduce error).

The overall uncertainty of the object location in the image plane is therefore estimated as :

$$\mathbf{\Lambda} = \mathbf{\Lambda}\left(x_c, y_c\right) + \mathbf{\Lambda}_{O_i} + \mathbf{\Lambda}_{M_j}$$

$\mathbf{\Lambda}(x_c,y_c)$ is the estimated uncertainty in the measurement of a pixel and is defined by $\mathbf{\Lambda}(x_c,y_c) = \mathbf{I}\,\sigma^2$ where σ^2 is some nominal variance. $\mathbf{\Lambda}_{Oi}$ represents the uncertainty due to the dimensions of the observed object and can be computed as a proportion of the measured height and width of the bounding box for that object i.e.

$$\mathbf{\Lambda}_{O_i} = \begin{bmatrix} w_{O_i} & 0 \\ 0 & h_{O_i} \end{bmatrix} \begin{bmatrix} r_w & 0 \\ 0 & r_h \end{bmatrix}$$

where (w_{Oi},h_{Oi}) are the measured dimensions of the 2-D bounding box for the *i'th* object and the values of r are chosen empirically to scale the uncertainty for each dimension (relating to the width and height). $\mathbf{\Lambda}_{Mj}$ is an additional bias factor to account for expected error bounds due to misdetection of objects.

4.3 DATA ASSOCIATION

Two methods were investigated to compute the data association of the tracks.

4.3.1. NEAREST NEIGHBOUR ALGORITHM

The initial method investigated was the nearest neighbour filter. This type of filter is popular due to its simplicity. There are two variants of the nearest neighbour approach, in the case of a single target these both consist of the following steps:

(1) Validation of the measurements to the predicted track location.

(2) Selection of one of the validated measurements for each track.

(3) Update of the track state assuming this measurement is the correct one (e.g. with a Kalman filter).

In Step (1) a validation gate is applied to limit the potential matches, the validation gate is determined by a threshold τ on the normalised innovation squared distance between the predicted track states and the observed measurements:

$$d^2 = \left(\mathbf{H}\widehat{\mathbf{X}}_k^- - \mathbf{Z}_k\right)^T \mathbf{S}_k^{-1} \left(\mathbf{H}\widehat{\mathbf{X}}_k^- - \mathbf{Z}_k\right)$$

where $\mathbf{S}$ is the innovation covariance, $\mathbf{X}$ is the *a priori* state estimate and $\mathbf{Z}$ is the observed measurement at time k. .Values for τ can be readily obtained from tables of the X^2 distribution, with the degrees of freedom equal to the dimension of the measurement.

Step (2) can be achieved in one of two ways. In the *nearest neighbour standard filter* the nearest measurement is chosen using the normalised innovation squared metric. In the *strongest neighbour standard filter* the validated measurement with the strongest signal is associated to the track.

The main problem associated with the nearest neighbour filter is that it is a *discrete* association, leading to an over-confidence in our belief that we have the correct measurement associated with a given track. By not handling the probability of association, the nearest neighbour filter is likely to lose tracks even in moderate clutter.

The extension of the nearest neighbour filter to multiple targets viewed from multiple sensors is thus:

(1) For each track, obtain the validated set of measurements (one set for each camera).

(2) For each track, associate the nearest neighbour (for each camera) with this track (stored in an association matrix, β)

(3) For each track, fuse the associated measurements into a single (fused) measurement.

(4) Update of each track state with the fused measurement (e.g. with a Kalman filter).

(5) Inter-sensor association of remaining measurements. These are subsequently fused into potential candidates for new tracks.

It is noted that (2) is a *sequential* operation as opposed to a *batch* analysis of the track to measurement associations; this is a more efficient (although sub-optimal) strategy for associating the tracks and measurements.

4.3.2. JPDAF ALGORITHM

As mentioned in the previous section, the Nearest Neighbour Filter is discrete in nature – i.e. it selects one observed measurement (the nearest or the strongest) as being the 'correct' measurement and uses only this observed measurement to update the state of the object being tracked. This is susceptible to errors when a mis-association occurs, i.e. when a measurement originating from noise is chosen over the true measurement. In the presence of a large rate of noise measurements (due detection errors, object fragmentation, etc.), the performance of the nearest neighbour algorithm begins to degrade.

The Joint Probabilistic Data Association Filter (JPDAF), which is an extension of the Probabilistic Data Association Filter (PDAF), does not make a discrete selection, but it uses all of the surrounding measurements with different weights (probabilities) [2]. This helps to make the association of measurements with tracks more robust to the presence of noise. For this reason, PDAF/JPDAF is also referred to as an *all-neighbours* approach.

The PDAF algorithm assumes only one tracked object is being observed in the presence of noise, while JPDAF extends PDAF for the case of multiple tracked objects. For the PDAF case with *N* measurements and 1 tracked object, *N+1* hypotheses are generated – hypothesis H_0 represents the case where all the measurements originate from noise (tracked object is unobserved at time *t*); while hypotheses H_i for *i =1* to *N,* represent the case where measurement *i* originates from the tracked object. The noise measurements are assumed to be independent identically distributed measurements with uniform spatial distribution β . To limit the number of potential hypotheses, a validation region is used. Only measurements that fall within this validation region are considered and the equation for the validation gate is the same as that used in step (1) for the nearest neighbour (see d^2 in previous section). The probabilities of the hypotheses is then given by:

$$p'_0 = \beta^N (1 - P_{TD})$$

$$p'_i = \frac{\beta^{N-1} P_{TD} e^{\frac{-d_i^2}{2}}}{2\pi\sqrt{|S|}} \quad for\ i = 1..N$$

$$p_0 = \frac{p'_0}{p_{total}},\ and\ p_i = \frac{p'_i}{p_{total}},\ where:\ p_{total} = \sum_{j=0}^{N} p'_j$$

where P_{TD} is the probability that the tracked object is detected at time *t*, d_i^2 is the normalised innovation squared distance as computed in the previous section, and S is the covariance matrix. The last step in the above equations is a probability normalisation step. These probability values p_i are then used to build the association matrix.

In the multiple tracked object case of the JPDAF algorithm, given *N* measurements and *T* tracked objects, first the measurements are validated by considering only those that fall within the validation gate of a tracked object *j* – some measurements will be shared between tracked objects due to overlapping validation gates. Next all feasible hypotheses H_i are generated, by considering the total number of undetected tracked objects N_{ND} (equivalent of H_0 in PDAF), the number of validated measurements N_j for each of the tracked objects *j*, and the total number N_F of measurements assumed to be generated by noise.

The association probability calculation is done in a similar way to that of PDAF, except that now multiple tracked objects are involved, requiring multiple probability terms in the association probability equation:

$$g_{ij} = \frac{e^{-d_{ij}^2/2}}{2\pi\sqrt{|S|}}$$

$$p'(H_l) = \beta^{N_j - (T - N_{ND})} (1 - P_{TD})^{N_{ND}} (P_{TD})^{(T - N_{ND})} g_{ij} \cdots g_{mn}$$

$$p(H_l) = \frac{p'(H_l)}{\sum p'(H)}$$

where g_{ij} represents the probability that measurement *i* originates from tracked object *j*. The number of g_{ij} terms in $p'(H_l)$ is $(T - N_{ND})$, i.e. the number of tracked objects that are assumed visible at a particular time.

This process of enumerating all the feasible hypotheses can be quite computationally intensive in the presence of a large number of closely-spaced tracked objects and large number of measurements. For the initial implementation, an exhaustive search is being used. This can be improved later on, by using optimisation techniques and previous information from the tracked objects themselves to reduce the number of hypotheses generated.

4.4 DATA FUSION

Once the data association is determined, the measurements can be combined into single fused estimates. This is achieved using one of two strategies[18] - *covariance accumulation* and *covariance intersection*. For discrete data association covariance accumulation estimates the fused uncertainty $\mathbf{R}_{\text{fused}}$ for N matched observations as:

$$\mathbf{R}_{fused} = \left(\mathbf{R}_1^{-1} + \ldots + \mathbf{R}_N^{-1}\right)^{-1}$$

The covariance intersection method is conceptually similar to the accumulation except that the observation uncertainty covariances are weighted in the summation:

$$\mathbf{R}_{fused} = \left(w_1\mathbf{R}_1^{-1} + \ldots + w_N\mathbf{R}_N^{-1}\right)^{-1} \text{ where } w_i = \frac{w_i'}{\sum_{i=1}^{N} w_i'} \text{ and } w_i' = \frac{1}{\mathrm{Tr}\left(\mathbf{R}_i^c\right)}$$

where R_i^c is the measurement uncertainty of the *i'th* associated observation (made by camera c); Covariance intersection therefore weights in favour of the sensors that have more certain measurements.

The resulting fused observations are demonstrated in Figure 2 for the `Services Vehicle' object; the covariance accumulation method results in a more localised estimate of the fused measurement than the covariance intersection approach.

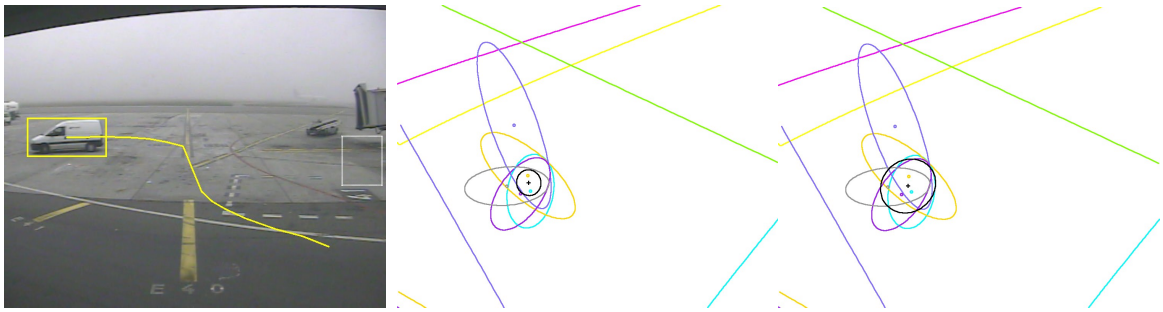

Figure 2 (Left) Frame 9126, camera 6 of sequence S21-Vehicles. **(Middle)** the fused measurement from all eight cameras (in black) for the services vehicle using the covariance accumulation method **(Right)** the fused measurement using covariance intersection.

For probabilistic data association the conditional probability of the association is used in the weighting. For covariance accumulation the association matrix β (containing the per camera association probabilities) is incorporated thus[16]:

$$\mathbf{R}_{fused} = \left(\sum_c \sum_j \beta_{ij}^c \mathbf{R}_j^{-1}\right)^{-1}$$

where $\boldsymbol{\beta}_{ij}^c$ represents the association probability between the *i'th* track and the *j'th* measurement made from the *c'th* camera. The covariance intersection method can be updated in a similar manner. The fusion of discretely associated tracks is therefore equivalent to the probabilistic approach with a binary association matrix i.e. the entries of β are 0 or 1.

4.5 TRACK ESTIMATION

Data association and fusion provides mechanisms for assigning the relationship between tracks and per sensor measurements. Intertwined with this work is the dual problem of how to update and estimate the tracks using the information about the incoming data. The Kalman filter is an optimal recursive algorithm used to estimate an unknown state based on a set of measurements; this state can be further applied to predict *future* measurements and hence improve the tracking of targets.

In the AVITRACK test sequences the filtering process was simplified by assuming that all targets are on the ground plane, and hence all the motion is 2-D. While this may not be necessarily true for all objects (e.g. people walking up the stairs on the jetbridge) the assumption holds for all the objects we are currently interested in tracking. The type of Kalman filter applied is a constant velocity model, with a state vector $\boldsymbol{X}=[x, y, \dot{x}, \dot{y}]$ and a measurement vector $\boldsymbol{Z}=[x, y]$.

Using the Kalman filter the set of tracks can be *predicted* and subsequently matched to the new set of measurements. Those tracks with measurements are *corrected* with the new observation. Tracks missing observations become more uncertain due to the recursive addition of expected (unmodelled) process noise in the prediction step.

4.6 TRACK UPDATE RULES

To ensure that only valid tracks are output rules are required to resolve tracking ambiguities. One of the main challenges is to filter out short-lived spurious measurements (i.e. *false alarms*), these type of measurements can occur due to camera noise / shake, detection failure due to shadows/reflections etc, or occlusions and object interactions. The following rules are therefore suggested to alleviate these problems.

Creation - A track is initialised for any fused measurement that is not associated with an existing track. The confidence of the track is initially set to zero. As supporting observations are made the confidence is increased as $1-\exp(-\lambda t)$ where t is the age of the track and λ is set such that after a predetermined time period the track is deemed confident i.e. mature.

Deletion – A track becomes eligible for deletion when no observations are made for a *significant* period of time. For a track missing an observation the confidence of the track is decreased as $N_0 \exp(-\lambda t)$ where N_0 is the confidence value from when the track was previously observed. When the track confidence decreases below a pre-set value the track is deemed a ghost and removed from the track list after a fixed period of time has elapsed (~ 1 second). In the special case that a track is not mature and not observed (i.e. a new track with short-lived observations) then the track is terminated after a shorter period of missing observations (~ 0.2 seconds).

Merging – Due to 2-D detection, tracking and location errors the observations for larger objects often do not fuse correctly. A merging strategy will be incorporated into the data fusion module for objects that have a significant depth away from the camera (i.e. vehicle categories) such that if two tracks exhibit a similar size and velocity and are within an extended validation gate then these tracks are merged into a single track. For people merging of tracks may prevent the accurate tracking of groups due to the apparent proximity of the measurements (which may be single measurements from certain cameras, depending on

the view point). The feedback of individual object information into the 2D tracking will allow more reasoned splitting of merged measurements if that measurement represents more than one object.

Splitting – Currently there is no mechanism for splitting a track into two or more separate tracks.

4.7 TRACK ATTRIBUTE ESTIMATION

The tracks found during the data fusion process have a number of attributes that require estimation included location, speed etc. These attributes are estimated as follows:

- Location : Estimated using Kalman filter. The per frame fused observation is made by averaging the per-camera object location estimates, with each location weighted by w_i (i.e. the confidence of the measurements)
- Velocity : Estimated using Kalman filter (i.e. filtered from location estimates)
- Category : Estimated using an IIR rolling average filter. The per frame fused category estimate is made by averaging the per camera category estimates. Like the location, this is also weighted by w_i .
- Orientation : Estimated from the track velocity i.e. direction of travel is orientation of object
- Width/Length/Height : The width and length 3D are estimated by taking the minimum (for the width) and maximum (for the length) of the associated 2D width observations made by the frame to frame trackers. Height is taken as the average of the 2D height measurements. This strategy appears a simple, but non-robust, solution to this problem. Alternatively, a parameter can be specified to use *a priori* estimates once the category is known.

CONFIGURATION PARAMETERS

Table 8.1 below lists the parameters that need to be configured for the data fusion and associated tracking algorithms.

Algorithm	Parameter	Description	Value used
All data fusion	VALIDATION_GATE_CONFIDENCE	Determines the maximum size of *d* for the validation gate computation by specifying the required confidence (as a percentage) of the association. Increasing this value constrains the data association to make the more certain associations.	99.0
All data fusion	STATIONARY_THRESHOLD	Value of speed (m/s) below which tracks are deemed to be stationary	1.0
All data fusion	CONFIDENCE_THRESHOLD	Only output tracks that have a confidence greater than this threshold	0.5
All data fusion	OOB_X_MIN	Minimum valid x value (in world co-ordinates)	-50

All data fusion	OOB_X_MAX	Maximum valid x value (in world co-ordinates)	50
All data fusion	OOB_Y_MIN	Minimum valid y value (in world co-ordinates)	-50
All data fusion	OOB_Y_MAX	Maximum valid y value (in world co-ordinates)	50
Kalman filter	MAX_FRAMES_UNOBSERVED	Track can be unobserved for n frames before termination	25
Kalman filter	PROCESS_NOISE_VELOCITY_PER_SECOND	Process noise for velocity in m/sec	10.0
Kalman filter	MIN_ORIENTATION_SPEED	The minimum object speed when measuring the orientation.	0.1
Kalman filter	USE_KNOWN_SIZE	Replace estimated size of object with a priori knowledge	1

SOME RESULTS

The following Figures demonstrate the results of the data fusion module for sequences S3 (all cameras), S4 (cameras 3 and 6) and S21 (all cameras). The Figures show a frame towards the end of the sequence with all confident tracks marked on the ground plane (confident is empirically deemed to be tracks with a confidence score >0.5). For tracks present in the current frame the red ellipses denote the fused measurements for that frame, blue ellipses denote the Kalman state error, black triangles represent unconfident tracks and green triangles represent confident tracks. A more formal evaluation using ground truth will be completed as part of 'Work Package 6.1 – Scene Tracking Evaluation' [?].

In all test sequences the data association algorithm was the nearest neighbour filter, the data fusion method used was covariance intersection and the track filter was the constant velocity Kalman filter.

8.1S3-A320 results

Figure 4 shows the data fusion result upto and including frame 05500 of sequence S3-A320. It can be clearly seen that many objects (predominantly vehicles) at this frame are fragmented and the association step is not able to find the correct association. Many of these problems are caused by two main factors (a) the poor localisation and representation of vehicles in the association step and (b) misdetection (due to shadows, reflection and occlusions etc) in the frame to frame tracker.

The most prominent missing track is that of the aircraft object, which is not found due to the difficulty in performing data association using only spatial location and uncertainty. This can be alleviated with the addition of more features in the validation gate step, which will make the data association step robust to the problems presented by larger objects when using the standard techniques derived from research in radar-based tracking.

Another related problem affecting this sequence is the recent modification made to the frame to frame tracking module, such that it now outputs stationary tracks for the duration of the observation. The data fusion module has no method for handling these stationary objects, allowing them to be misassociated with adjacent, moving, observations.

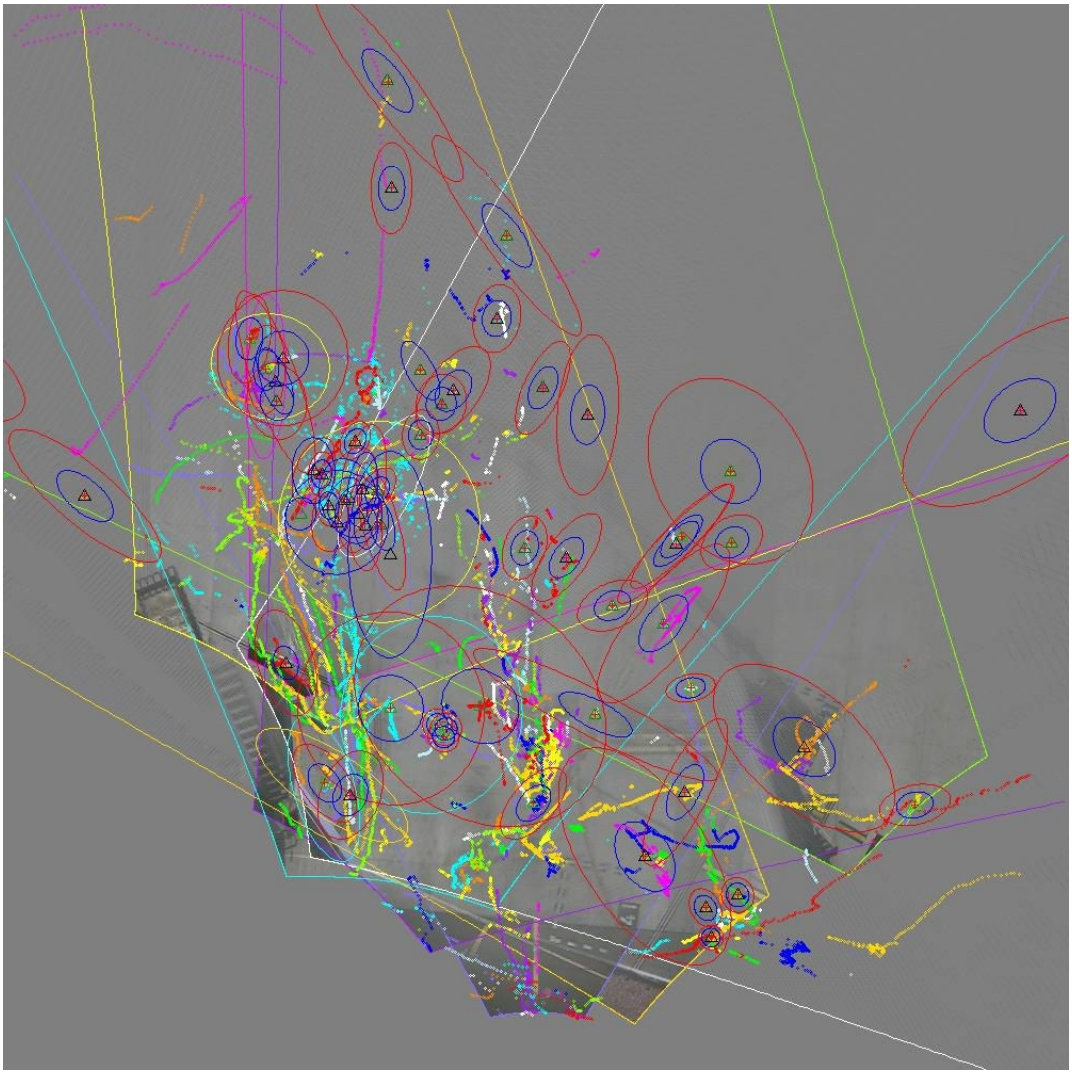

Figure 4 Frame 05500 data fusion result for sequence S3-A320 (all cameras).

8.2S4-A320 results

Figure 5 demonstrates the tracking result for frame 01800 of the sequence S4-A320. The sequence contains a GPU (white track), a services vehicle (light blue track), a person (yellow track) and some chocks deposited on the apron (green track).

It can be seen that the tracks exhibit a reasonable continuity of ID throughout the sequence with a single track ID (denoted by a single colour) detected for each of the four main items in this sequence. The data fusion result for this frame reveals that although the correct quantity of tracks are found, the data from the multiple cameras does not appear to be fused correctly.

Further analysis of the per camera tracking results reveals that this is due to misdetection of shadow and vehicle reflections leading to poor localisation of the objects in the scene. Without a merging strategy the system is unable to recover from this problem, although a localised non-maximal supression strategy is applied to prevent newly created objects becoming confident in the presence of older tracks that have supporting evidence; this is a temporary solution to alleviate some of the problems that result from detection error or congested apron regions.

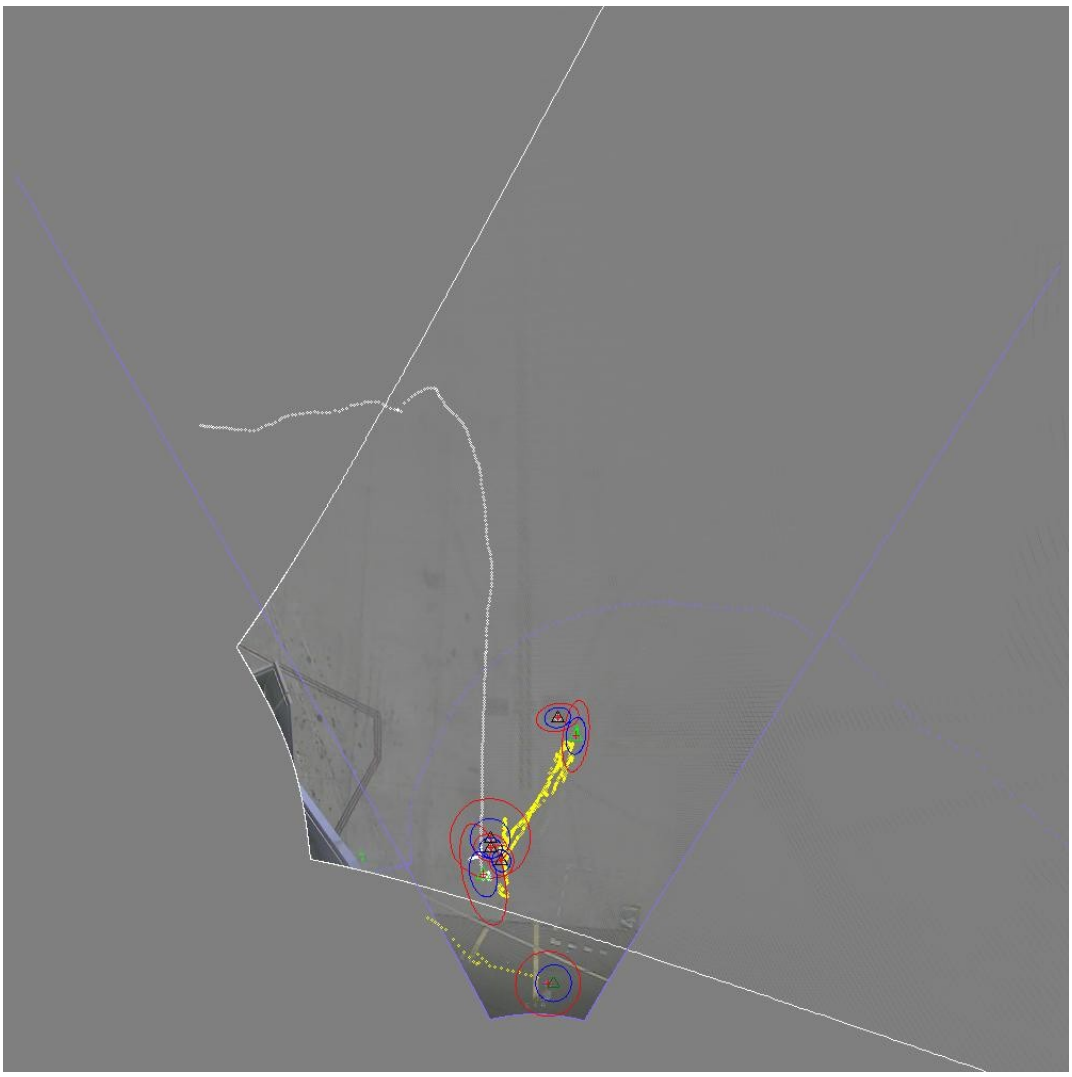

Figure 5 Frame 01800 data fusion result for sequence S4-A320 (cameras 3 and 6 only).

8.3S21-Vehicles results

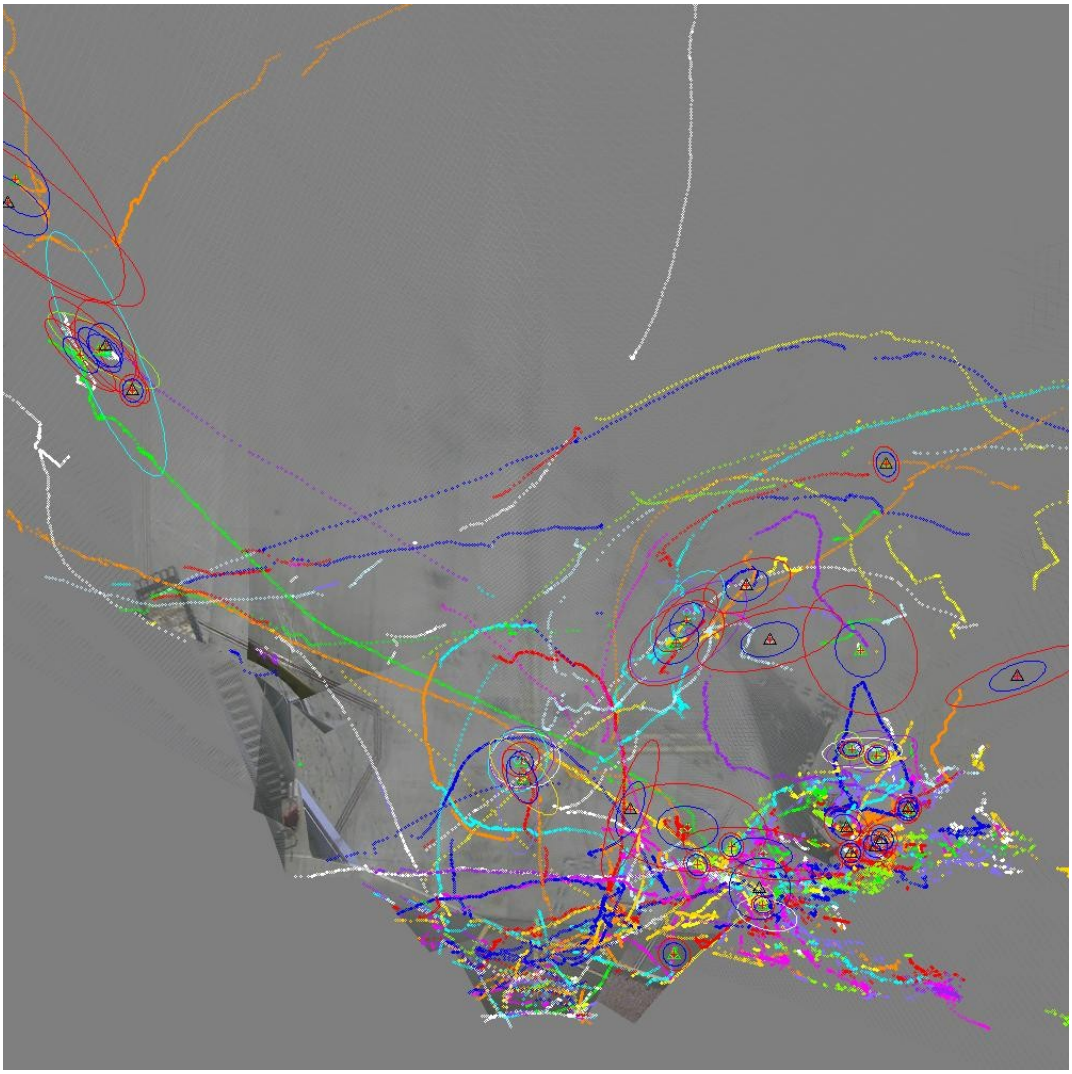

Figure 6 Frame 09100 data fusion result for sequence S21-Vehicles (all cameras).

Figure 6 shows the data fusion result for the sequence S21-Vehicles. In this sequence many of the objects are well separated and there is less predominant shadow; due to this there is reasonable track location and fusion, although errors do occur for the larger vehicles due to inaccurate localisation and inadequate association metric between observations and tracks (as discussed for sequence S3-A320). These errors are evident by a loss of identity for tracks (shown by a change of colour ID) and by the presence of two adjacent tracks with similar trajectories.

The performance of the data fusion module on S21-Vehicles suggests that many of the problems fusing observations on sequences S3-A320 and S4-A320 can be attributed to the misdetection of the observations. With more accurate detection of the observation the localisation strategy appears to be adequate, especially for people and smaller vehicles which are generally fused successfully under such conditions.

CONCLUSION

To conclude, the data fusion module performs adequately given isolated targets correctly detected in the frame tracker. The data fusion module incorporates uncertainty information in the location estimate of the observation and it is often an inaccurate location estimate that results in the failure of the data association step. It is noted that a significant proportion of the localisation problems that occur in the data fusion module can be traced back to the motion detection module i.e. shadow, reflections etc.

The features used in the validation gate appear to be inadequate for handling the association of observations for vehicles in close proximity; the use of purely spatial features has been found to be adequate for tracking isolated targets, but results in misassociation for vehicles of different classes/velocities.

With correct data association the mechanisms for data fusion and object tracking algorithms appear to be adequate for the AVITRACK project, therefore we propose that the majority of the remaining work on the data fusion module is focussed on the object localisation and data association problems.

FUTURE WORK

Future work in the data fusion module consists of:

(1) Evaluate the JPDAF algorithm and compare the probabilistic association method with the deterministic nearest neighbour method.

(2) Add rules for merging and splitting of tracks to make the tracker robust to data association failures.

(3) Improve the association metric, currently this is a gating method using the location and uncertainty of the observations and the confidence has previously been incorporated to weight the fusion in favour of more confident tracks (although the confidence in the frame tracker is somewhat arbitrary). The use of these features has been found to be adequate for tracking isolated targets; we propose that the location alone will be insufficient to allow modelling of more complex scenarios involving many classes of object interacting with each other.

The suggested solution to this problem is to improve the association feature space to take into account the status of the object (i.e. occluded, stationary or moving etc), the classification score , the location and the velocity etc. Improving the data association matching score by expanding the features applied will increase the robustness of the data fusion for objects in close proximity or objects that are misdetected in the per camera frame trackers.

(4) Allow per camera classification results with more confident matches (especially for sub-types) to be used as the fused classification result since some cameras may have unreliable recognition of objects which decreases the reliability of the data fusion classification result under the current system.

REFERENCES

[1] P. Remagnino, A.I. Shihab, G.A. Jones, *"Distributed Intelligence for Multi-Camera Visual Surveillance"*. In *Pattern Recognition* 37(4): 675-689, 2004.

[2] Y. Bar-Shalom and X. R. Li, *"Multitarget-Multisensor Tracking: Principles and Techniques"*. YBS Publishing, 1995.

[3] Y. Ruan, *"Some Statistical Models and Approaches to Target Tracking and Data Association"*. Ph.D. Thesis, University of Connecticut, 2003.

[4] R. Collins, A. Lipton, H. Fujiyoshi, and T. Kanade, *"Algorithms for Cooperative Multisensor Surveillance"*. In *Proc. of the IEEE*, vol 89, no 10, pp. 1456-1477, Oct 2001.

[5] R. Collins, A. Lipton, T. Kanade, H, Fujiyoshi, D. Duggins, Y. Tsin, et al., *"A System for Video Surveillance and Monitoring"*. Technical Report CMU-RI-TR-00-12, Carnegie Mellon University, PA, 2000.

[6] A. Turolla, L. Marchesotti, and C.S. Regazzoni, *"Multicamera Object Tracking in Video Surveillance Applications"*. Presented at the *IEE Target Tracking: Algorithms and Applications*, March 2004.

[7] D.F. Hsu, D.M. Lyons, C. Usandivarus, and F. Montero, *"RAF: A Dynamic and Efficient Approach to Fusion for Multitarget Tracking in CCTV Surveillance"*. In *IEEE Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems*, Japan, Aug 2003.

[8] K. Nummiaro, E. Koller-Meier, T. Svoboda, D. Roth, and L. Van Gool, *"Color-Based Object Tracking in Multi-Camera Environments"*. DAGM Symposium, 2003, pp. 591-599.

[9] A. Mittal, and L. Davis, *"Unified Multi-camera Detection and Tracking Using Region-Matching"*. In *IEEE Workshop on Multi-Object Tracking*, Vancouver, Canada, July 2001.

[10] L. Snidaro, R. Niu, P.K. Varshney, and G.L. Foresti, *"Automatic Camera Selection and Fusion for Outdoor Surveillance under Changing Weather Conditions"*. In *IEEE Conf. on Advanced Video and Signal Based Surveillance,* Florida, July 2003, pp. 364-369.

[11] J. Ohya, A. Utsumi, and J. Yamato, *"Analyzing Video Sequences of Multiple Humans – Tracking, Posture Estimation and Behaviour Recognition"*. Kluwer Academic Publ., Mar 2002.

[12] J. Black, T. Ellis, and P. Rosin, *"Multi View Image Surveillance and Tracking"*. In *Proc. of the IEEE Workshop on Motion and Video Computing*, pp. 169-174, 2002.

[13] T. Ellis, J. Black, M. Xu, and D. Makris, *"Integrating and Learning Information from Multiple Camera Views"*. In *Visual Surveillance Book*, City University, London.

[14] L. Jiao, G. Wu, Y. Wu, E. Chang, and Y.-F. Wang, *"The Anatomy of A Multi-Camera Video Surveillance System"*. In *ACM Multimedia System Journal Special Issue on Video Surveillance*, May 2004.

[15] S. Dockstader, and A. Murat Tekalp, *"Multiple Camera Tracking of Interacting and Occluded Human Motion"*. In *Proc. of the IEEE*, 89, pp. 1441-1455, 2001.

[16] M. Xu, L. Lowey, and J. Orwell, *"Architecture and Algorithms for Tracking Football Players with Multiple Cameras"*. In *IEE Intelligent Distributed Surveillance Systems*, London, Feb 2004.

[17] J. Kang, I. Cohen, and G. Medioni, *"Tracking Objects From Multiple Staionary And Moving Cameras"*. In *IEE Intelligent Distributed Surveillance Systems*, London, Feb 2004.

[18] J. Black and T.J. Ellis,``Multi Camera Image Measurement and Correspondence." *Measurement - Journal of the International Measurement Confederation* 35(1) July, pp 61--71, 2002.