REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)* 06-07-2009	**2. REPORT TYPE** Final Report	**3. DATES COVERED** *(From – To)* 16 June 2008 - 16-Jun-09

4. TITLE AND SUBTITLE Vehicle Tracking in UAV Data via Adaptive Weighting of Visual Features	**5a. CONTRACT NUMBER** FA8655-08-1-3028
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Dr. Noel O'Connor	**5d. PROJECT NUMBER**
	5d. TASK NUMBER
	5e. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Dublin City University Glasnevin Dublin 9 Ireland	**8. PERFORMING ORGANIZATION REPORT NUMBER** N/A
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) EOARD Unit 4515 BOX 14 APO AE 09421	**10. SPONSOR/MONITOR'S ACRONYM(S)**
	11. SPONSOR/MONITOR'S REPORT NUMBER(S) Grant 08-3028

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

In our previous work we have investigated data fusion frameworks for object detection and tracking. In particular, we proposed the use of the spatiogram, as opposed to the more commonly used histogram, for object tracking. The motivation for this stems from the fact that a spatiogram allows coarse encoding of an object's spatial information. In this way it is more robust to changes in object appearance than a histogram that carries no spatial information, whilst not suffering the constraints of template-based matching techniques that encode full object spatial information. In addition, we proposed an improved similarity measure for matching spatiograms from one frame to the next. Furthermore, we embedded this measure in an innovative efficient mean-shift search process.

Using this as a basis, we were able to propose a multi-feature tracking framework that we term a spatiogram bank that leverages earlier work on optimal feature fusion. We proved that the spatiogram is particularly suited to this framework as it does not suffer from the curse of dimensionality as new features are added, leading to a very flexible multi-feature tracking framework. We demonstrated the benefits of this framework in a variety of different application scenarios that use visible spectrum and infrared visual data sources. In particular, we showed how this tracking framework can be embedded in a system for detection and tracking of ground-based vehicles in UAV video footage.

15. SUBJECT TERMS
EOARD, Tracking, target identification, UAVs, Image Processing

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18, NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT UNCLAS	**b. ABSTRACT** UNCLAS	**c. THIS PAGE** UNCLAS	UL	25	BARRETT A. FLAKE
					19b. TELEPHONE NUMBER *(Include area code)* +44 (0)1895 616144

Vehicle Tracking in UAV-Captured Video Data Incorporating Adaptive Weighting of Features

Final Report – July 2009

David Sadlier*, Noel O’Connor
CLARITY: Centre for Sensor Web Technologies, Dublin City University, Ireland
{* sadlierd@eeng.dcu.ie}

Introduction

This report constitutes deliverable *D3 (month 12)* as prescribed in the project specification document, and accompanies the delivery of *D2* (the final version of the system for vehicle detection and tracking in UAV-captured video data). The aim of this report is to explain the internal workings of the system developed, providing a detailed description of the algorithms involved, as well as describing and validating its performance based on quantitative evaluations performed.

The report is broken down as follows: Section 1 presents some background information, aimed at providing a level of context to the reader. Section 2 introduces the issues surrounding the problem addressed, provides an outline of the solution proposed, and describes the dataset used in experiments Section 3 details how the algorithms were implemented. Finally, Section 4 describes the experiments performed, discusses the results obtained, and expounds on conclusions drawn.

1. BACKGROUND

1.1. Review of Project Objectives

For the purposes of context, this report begins with a brief review of the objectives, and targeted outcomes of the project.

The goal of this work is the development of a software solution that provides continuous robust localisation (i.e. ‘tracking’) of vehicle-type objects throughout the scenes of aerial video footage captured by Unmanned Aerial Vehicles (UAVs). The scientific field of object-tracking is well-studied in the field of computer vision, with many competing solutions offered. However, one of the interesting aspects of this work is the opportunity to investigate the benefits of a proposed adaptive feature weighting technique, which we envisage should increase the robustness of typical vehicle tracking solutions in the face of common pitfalls (e.g. relating to changes in lighting, poor vehicle-road contrast, etc.).

The targeted outcome of this work is a MATLAB[1] based prototype software simulation, that allows the operator to select a video file for analysis, and to subsequently view the processed video footage (in video playback mode), with the vehicles in the scene automatically ‘highlighted’ against their surroundings. The system was to be developed, tested, and evaluated using the DTO VACE [2] 2005 video dataset, which is a suite of multi-spectral (visible and thermal infrared) MPEG-2 video files captured over a military training base, provided for experimentation purposes by the Air Force Research Laboratory.

1.2. Chronicle of Work

This project was a 12-month undertaking, with three deliverables specified. Deliverable D1 - which consisted of an interim version of the system coupled with an associated report [2] describing the achievements and progress made, was transferred to EOARD at the six-month point (January 2009). As described in [2], the progress made at that time consisted of background research, data formatting, and the development of a working system framework based on preliminary versions of the various components. Since then, the work undertaken has mainly involved activities relating to (i) the refinement of the system components, (ii) the development of additional functionality, (iii) ground-truth creation, (iv) performance evaluation, and (v) document writing. For completeness, a list of relevant milestone achievements is presented in timeline format in Appendix A.

2. PROBLEM & SOLUTION

2.1. Problem Discussion

The task of developing an automated system for the detection and tracking of vehicles in UAV-captured video is non-trivial. To introduce it, we consider its two main constituent parts, i.e. vehicle detection and region tracking.

2.1.1. Vehicle Detection

2.1.1.1. Frame Differencing

The most obvious approach to the detection of moving vehicles within the scenes of a video sequence corresponds to *image-differencing*, whereby the pixels of two video frames are subtracted from each other, yielding so-called ‘frame-differenced images’. The idea is that, assuming a static camera scenario (i.e. an unchanging background), for an appropriately chosen interval over which the differencing is applied, the pixels constituting the frames will differ only in regions of moving foreground objects, and therefore the locations of moving objects will be highlighted in the ‘images’ resulting from the frame-differencing procedure. This technique has been shown to work quite well in a variety of applications featuring static camera scenarios.

2.1.1.2. Camera Modelling

However, the scenario of UAV-captured video data is quite different from the above-described. In short, the aerial video footage captured from an overhead UAV does not conform to the static background description, since the overall scene (including the background) continuously changes as the UAV flies over land. The consequence is that, when trying to detect vehicle locations in this data, simply applying the frame-differencing technique would result in highly irregular results, because, not only are the foreground objects shifting position from frame-to-frame, but the objects constituting the background scene are displaced too. Hence, frame-differencing applied to UAV video footage would result in difference-images exhibiting a multitude of highlighted regions, on the basis of which, the ability to discern foreground objects would be severely compromised. Hence, to overcome this problem, the video sequence needs to be first analysed in terms of camera movement. That is, it is required to mathematically characterise or ‘model’ the camera motion that occurs throughout the sequence. Armed with such a model, it can then be used to process the video sequence towards essentially undoing (‘compensating’) for the frame-to-frame camera motion, to the extent that vehicle detection may then be performed in ‘camera-motion compensated’ versions of the video images - within which (assuming an ideal model) foreground object movement is retained and can be discerned (e.g. as outlined above). Hence, obtaining an accurate camera model that exhibits fidelity to the actual frame-to-frame scene displacement observed throughout the video sequence, is an important component of this approach.

2.1.1.3. Intervals

One of the main issues in relation to vehicle detection by frame-differencing concerns the intervals chosen over which the differencing is to be performed. Clearly, it is crucial that the intervals chosen are long enough such that a detectable amount of vehicle movement is observed within the scene. That is, if the interval over which the frames are differenced is too small, the vehicles may have only moved a very small amount from the first image to the next, resulting in very small highlighted regions in the frame-differenced images may not be discernable from the noise floor. However, different video sequences are obviously also captured from a variety of different altitudes, meaning the vehicles appearing in the video images vary in size, and hence the interval required to represent discernable vehicle movement within the scenes varies from sequence-to-sequence. Hence, no ‘one-size-fits-all’ interval value would be suitable for all sequences. Further still, different video sequences will typically feature a variety of vehicles moving at different speeds through the scenes, resulting in inconsistent levels of vehicle displacement, which suggests that no single interval value would even suffice for a given sequence. Hence the requirement that some level of flexibility be incorporated into the interval setting in a frame-differencing based vehicle detection solution.

2.1.2. Region Tracking

Assuming the accurate detection of a given vehicle (current position known), the challenge of tracking as it shifts location from frame-to-frame, would typically be approached by first somehow mathematically characterising its appearance in the image (i.e. based on the pixel values constituting the region), and then based on a similarity to this representative model, attempting to determine its new location in the image pixel grid from one frame to the next. Clearly, the main issues here relate to (i) the nature of the representation extracted, (ii) the definition of similarity, and (iii) the search strategy employed, all of which would have some bearing on the tracking performance attained.

In terms of the former, histogram-style solutions have been shown to be robust and efficient, as they discard spatial information, and are therefore insensitive to small changes in object pose. The idea is that, based (e.g.) on the vehicle’s appearance upon detection, statistics are extracted for the pixel values observed in the corresponding region of the image. The extracted histogram serves as the representative model for the vehicle, against which all potential new instances/positions of it throughout subsequent video frames are then compared. However, there is an inherent problem with histogram-style tracking in that these types of representations do not scale well to higher dimensions [3]. For example, given a four-dimensional feature space (R, G, B, I), quantizing the pixel ranges into 8-bins (typical for standard 0-255 pixel ranges) would lead to the construction of 4096-D histogram representations, which are not very practical from a computation or memory point of view. Another problem is the 'curse of dimensionality' [4], which suggests that we would most likely experience unsatisfactory matching performance in comparing representations at this very fine level of granularity. Clearly, these scale-related issues must be addressed in terms of implementing a practical and effective histogram-based solution.

In terms of similarity metrics, there are a variety of options from (e.g.) distance-based to (e.g.) probabilistic styles, all well studied and all equally advocated.

The choice of search-strategy plays an important role in the speed performance of a tracking system. Clearly, a so-called *exhaustive-search* will tend to give the most accurate result - whereby, in attempting to track a moving object from one frame to the next, all (conforming) pixel locations within the image are considered as potential new positions of the object within the new frame. However, this approach is clearly unfavourable from a computational point of view. Hence search-spaces are typically confined to reflect a reduced number of localised candidates for the object's new position, surrounding the current object position ('*localised search*'). In a further attempt to reduce computation, *coarse-to-fine search* involves first sub-sampling the localised search space, determining the best coarse match, and then performing a more detailed search within a small search space surrounding the best match coarse position. Clearly, these latter approaches represent a trade-off between reducing computation (i.e. speed performance) and finding the correct match within the image.

2.1.3. Approaching the Problem

The above outlines the issues involved in addressing the challenge of vehicle detection/tracking in UAV-captured video. The system developed and described in this document represents an implementation of a solution whereby, in approaching the abovementioned issues, decisions were made based on optimising performance for the particular dataset used for development/testing. For transparency, this dataset is now detailed below.

2.2. The Dataset

Provided for experimentation purposes by the Air Force Research Laboratory, the DTO VACE [5] 2005 dataset is a suite of multi-spectral video files captured by airborne UAVs flying over a military training base. The corpus consists of numerous video *sequences*, with each sequence corresponding to a single unbroken pass of video footage captured from the UAV. Each video sequence is in fact typically instantiated by two separate MPEG-2 format video files; one presenting (640x480, 30fps) visible spectrum (RGB) video data, and the other (320x256, 30fps) thermal infrared (I) video data. The two separate streams result from independent cameras mounted on the underside of the UAV. The motivation for including the infrared channel in the dataset is to complement the colour/luminance-based visible data, in the expectation that if and when visible-based detection/tracking is compromised (e.g. due to poor lighting/contrast, etc.), the I-based analysis may suffice in terms of preventing detection/tracking failure, and vice-versa.

The VACE video dataset is accompanied by corresponding ground-truth dataset, whereby the true locations/positions of vehicles within the scenes of each video sequence have been recorded. Specifically, within this dataset, an annotation of vehicle positions exists for every 12th video frame (*I-frame*[*]) of the corresponding (visible-stream) sequences. The choice of I-frame level annotating follows on from the MPEG structure of the video files, and represents a considered trade-off between a more ideal annotation depth (e.g. frame-level), and the labour involved in manual logging. As mentioned above, the system developed targets optimising performance for this dataset in particular. Hence, this (12 frame) I-frame interval constitutes a key analysis level upon which the analysis/evaluation is based around.

The infrared channel can simply be considered as a fourth channel of information to be processed along with the 3-channel (RGB) visible stream. However, given that the two streams were captured using independent cameras, they tend to (i) exhibit a slight temporal misalignment, and (ii) exhibit a discrepancy in the perspective of the scene captured. Clearly, both of these issues need to be corrected in advance of joint visible/infrared-based analysis. However, this involves a substantial level of manual intervention, the procedures involved in which are outlined in Appendix B.

2.3. Solution Outline

Given a video sequence from the dataset to be processed, this section outlines the solution framework designed for the provision of vehicle detection/tracking throughout the images of that sequence.

2.3.1. Camera Modelling & Vehicle Detection

2.3.1.1 Framework & Intervals

Towards synthesizing a camera-motion compensated domain for the sequence (within which vehicle detection will be performed), the camera motion modelling stage is concerned with characterising the scene displacement that occurs as a consequence of the UAV-housed camera(s) travelling over land as they film (see Fig 2.1). In terms of analysis depth, the camera modelling (and therefore vehicle detection) process is targeted at the I-frame level - reflecting the paradigm of the dataset[#]. So, for every I-frame of the sequence, e.g. iframe_i, the process begins by;

- Modelling camera motion between iframe_i and $\text{iframe}_{i\pm1}$, then…
- Modelling camera motion between iframe_i and $\text{iframe}_{i\pm2}$, then…
- Modelling camera motion between iframe_i and $\text{iframe}_{i\pm3}$

[*] I-frames are one of the three main frame types that constitute an MPEG video stream. They occur at regular intervals within the stream, and represent non-temporally predicted image data, and hence are the frame types representing the highest fidelity to the original scene captured by the camera.

[#] It is envisioned that I-frame level analysis represents a reasonable trade-off between the early detection of newly visible vehicles in the scene, and the computational overhead associated with the vehicle detection process.

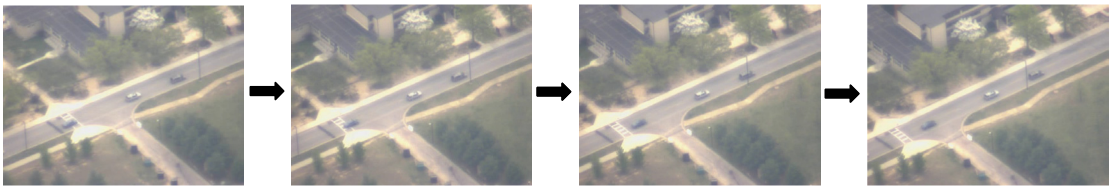

Fig. 2.1. Scene displacement over I-frame intervals due to travelling camera.

The reasons for the bi-directional (±) modelling will become apparent later, but the implementation of the ±1, ±2, and ±3 I-frame intervals represents an attempt to meet the aforementioned requirement of flexibility on the interval underpinning a frame-differencing based vehicle detection solution[#]. It was envisioned that basing the frame-differencing around these three intervals independently, and then merging their individual results, would yield a combined frame-differenced 'image' that would tend to highlight the locations of moving objects largely irrespective of their size/velocity, and would therefore represent an effective, yet computationally moderate, vehicle detection solution.

2.3.1.2. Vehicle Detection

Following the camera modelling analysis, each I-frame of the sequence, e.g. iframe_{i}, has associated with it; models of the camera motion existing between itself and its three neighbouring I-frames (in both a forward and backward direction) - i.e. I-frames i±1, i±2, and i±3. So, prior to image-differencing, for each I-frame (e.g. iframe_{i}), these models are used to spatially translate ('*warp*') $\text{iframe}_{i\pm1,}$ $\text{iframe}_{i\pm2,\ \text{and}}$ $\text{iframe}_{i\pm3}$ so that their respective scenes become spatially aligned with that of iframe_{i} (this is akin to 'undoing' the scene displacement in each case represented in the respective camera models). The result is that the 'background' scenes of $\text{iframe}_{i\pm1,}$ $\text{iframe}_{i\pm2,}$ and $\text{iframe}_{i\pm3}$ should match that of iframe_{i}, and any residual differences between them should then correspond to foreground objects (e.g. vehicles) moving independent of the camera. Assuming accurate results, image-differencing should then allow for reliable identification of the locations of these objects. Hence, in terms of iframe_{i}, the process proceeds as follows (where w indicates the warped version of a particular I-frame image);

- iframe_{i} differenced from $\text{iframe}^{w}_{i\pm1}$, then…
- iframe_{i} differenced from $\text{iframe}^{w}_{i\pm2}$, then…
- iframe_{i} differenced from $\text{iframe}^{w}_{i\pm3}$

The highlighted regions identified from each individual image differencing process (i.e. over I-frame intervals i±1, i±2, and i±3) are then merged, yielding a finalised set of candidate vehicle locations for iframe_{i}.

[#] Considering the framerate/structure of the MPEG-2 format VACE videos, the intervals between (i) iframe_{i} and $\text{iframe}_{i\pm1}$ approximately correspond to a period of 0.4s, (ii) iframe_{i} and $\text{iframe}_{i\pm2}$ approximately correspond to a 0.8s period, and (iii) iframe_{i} and $\text{iframe}_{i\pm3}$ approximately correspond to an interval of 1.2s.

2.3.2. Vehicle Tracking

2.3.2.1. Spatiogram-Bank

For each detected vehicle, a *spatiogram*[6]-based representation is then extracted - based on its corresponding region of image pixels. Spatiograms are similar to histograms (in that they contain information on colour distribution), but also include some coarse spatial information*. In fact, in attempting to combat the problems of scale surrounding histogram-style solutions (as alluded to in Section 2.1.2), as a compromise, each object is represented by multiple independent spatiograms (i.e. one for each feature) – a so-called *spatiogram-bank*#. The multi-band spatiogram-bank serves as a static representation of each detected vehicle upon which the tracking process involving it is based.

2.3.2.2. Tracking

For a given detected vehicle (e.g. vehicle_v), it's new location in the subsequent frame of the video is then determined by comparing its representative spatiogram-bank with that of (a selective list of) potential new pixel locations within the new image, and then presuming the true new location to be that exhibiting least mathematical distance between them. More explicitly, for a potential new location of vehicle_v (e.g. in the frame subsequent to iframe_i), a spatiogram-bank is extracted for its corresponding pixels (i.e. independent spatiograms are extracted for each image feature), and then the individual spatiograms (corresponding to the vehicle and its potential new location) are compared against each other on a feature-by-feature basis. Then, armed with a similarity score for each feature spatiogram, an overall ('*combined*') spatiogram-bank similarity score is generated via product fusion of these. Formally, an expression for the combined similarity of vehicle_V (ρ_V) at (e.g.) position x, may be written as follows (assuming K features):

$$\rho_V(x) = \prod \rho_k(x) \qquad \forall\ k = \{1,\ldots K\} \qquad (1)$$

This outlines the general framework governing the basis for which a detected vehicle (vehicle_v) is tracked throughout the successive frames of the input video sequence. However, further to this is the proposal of adaptively weighting the features in the calculation of the fused score – see below.

2.3.2.3. Adaptive Weighting

Central to this work concerns an investigation into the benefits of adaptive weighting of features based on the dynamics (throughout the sequence) of their relative object-discrimination ability. That is, depending on the variance of the conditions observed (e.g. lighting, contrast, etc.), of the bank of features upon which the tracking is to be based (e.g. R, G, B, I, etc.), certain features may temporarily exhibit an ability to outperform others in discriminating the target object from its background environment. To exploit this towards more robust tracking, it was proposed that the tracking system should rely more heavily on the better performing features at the expense of those considered less reliable. It was envisaged that emphasising the best performing features in this way should provide for more robust object tracking in the face of potential distractions. To this end, it was proposed to *weight* (on a frame-by-

[*] Essentially, spatiograms represent a trade-off between the efficiency of pure histogram-based representations and the accuracy offered by extracting rigid pixel-level object templates.

[#] The spatiogram-bank approach has the benefit of making the matching task more efficient, as well as more suitable to multi-modal data fusion, by allowing additional features to be easily integrated/removed into/from the tracking system.

frame basis) the contribution of the individual spatiogram similarity scores to the combined similarity score, on the basis of their relative recent ability in object discrimination. This corresponds to an augmentation of the aforementioned formal expression for spatiogram-bank similarity as follows, where w_k is the weight currently assigned to feature k.

$$\rho_V(x) = \prod w_k.\rho_k(x) \quad \forall\ k = \{1,\ldots K\} : \Sigma w_k = 1 \quad (2)$$

Clearly the challenge is to find the appropriate weights w_k in each case. The method proposed for doing so was based on measuring how well the individual features separate the object's true location from other potential background 'distractors' - defined as regions of close similarity to the object, in the vicinity of the object. The idea is that the weights are chosen on the basis of assigning greatest influence to the feature that offers the greatest separation between the true object appearance and that of the distractor (which closely resembles the object, but which we want to suppress as a viable match, as it is erroneous).

This concludes Section 2.3 describing the solution framework governing the detection and tracking of vehicles in UAV video. The next section describes the implementation of the algorithms underpinning each component and sub-component as outlined here.

3. ALGORITHM IMPLEMENTATION

3.1. Camera Modelling

This section describes the implementation of the camera modelling solution deployed in the system.

Modelling the camera motion between e.g. two successive I-frames is performed in the visible (RGB) feature space and begins with the process of *corner detection*#. Applying a standard corner point detector (e.g. Harris [8]) to both images results in a bag of corner points for each image. Armed with these, the next step involves *matching* corner points between the two images, which is performed by extracting small representative regions of pixels surrounding each corner point and comparing their values. So for example, for a given corner point in the first frame, a small region of RGB pixels surrounding the point is extracted and their values compared with those corresponding to the same-sized regions surrounding the corner points of the second image. This process is repeated for all corner points in the first image, until the majority have been paired. The expectation is that the set of matched corner points should then represent the spatial displacement of the most salient points in the scene between the two frames, resulting from the camera motion occurring during the interval between them. Armed with a set of matched corner points, these are fed into an optimisation algorithm [9], which attempts to derive a geometric transform, i.e. a 3x3 *planar homography* matrix [10], which is a standard projective transformation for mapping points from one plane (scene) to another plane (scene). It is the homography (3x3 matrix) that represents the camera motion modelled between

Corner points are typically well defined, and are thus considered a sound basis for modelling the spatial characteristics of images. In fact, the topic of corner-point detection has been well researched in the field of image analysis, with a variety of different approaches advocated [7].

the two images. Once again, the camera modelling process outlined above is performed in the visible domain only, since this exhibits higher resolution images and captures considerably more spatial area than the infrared stream, and therefore offers more scope for scene-to-scene matching. Note, inaccurate or unsuccessful camera model extraction may result from (i) insufficient corner point detections, (ii) insufficient corner point matches, (iii) the optimisation algorithm settling at local minima.

3.2. Vehicle Detection

The following describes the implementation of the vehicle detection algorithm deployed in the system.

Recall from Section 2.3.1, the approach to detecting a newly appearing vehicle in an I-frame of a UAV-captured video sequence (e.g. iframe_i) exploits corresponding I-frames $\text{iframe}^{w}_{i\pm x}$ (x=1,2,3), which are warped versions of the original images that have been spatially aligned to iframe_i using their respective camera models. In fact, all I-frames concerned are first converted to single channel luminance images (based on a standard RGB-to-greyscale formula), following which the warping takes place (i.e. in the luminance domain). Ultimately, this results in a greyscale version of iframe_i and, greyscale versions of $\text{iframe}^{w}_{i\pm x}$ for each interval x (=1,2,3).

The next step in the process relates to image differencing (see Section 2.3.2.). That is, for $\text{iframe}^{w}_{i\pm x}$ its respective (luminance) pixels values are differenced from those of iframe_i, resulting in two separate (single channel) ‘difference images’, where the regions of high pixel intensity correspond to the regions of disparity between the two images. As described earlier, the expectation is that these should correspond to areas of the scene affected by moving vehicles. To explicitly demarcate these regions (separate them from the noise floor), the difference images are then thresholded[#], resulting in binary images where only the most intense difference regions are highlighted and the noise floor suppressed. Fig. 3.1 illustrates a sample result corresponding to the movement of a single vehicle, where threshDiffImage1 represents the thresholded difference between iframe_i and iframe^{w}_{i-x}, and threshDiffImage2 represents same between iframe_i and iframe^{w}_{i+x}. Note that in threshDiffImage1 there are in fact two distinct highlighted regions, corresponding to two separate regions of disparity between iframe_i and iframe^{w}_{i-x}. These relate to regions of pixel difference between iframe_i and iframe^{w}_{i-x} that correspond to the *previous* (P1) and *new* (N1) positions of the vehicle. Similarly, in threshDiffImage2, there are two separate regions of pixel disparity existing due to previous (P2) and new (N2) locations relating to the movement of the vehicle within the interval between iframe_i and iframe^{w}_{i+x}. Note, the true location (L) of the vehicle in iframe_i is clearly the common highlighted area between threshDiffImage1 and threshDiffImage2, and may be determined by a logical AND between the two images, as shown in Fig. 3.1. Hence the reason for the bi-directional differencing, i.e. to enable the elimination of ‘ghost’ difference regions associated with changes in pixel values corresponding to the previous/new positions of the vehicle in the scene.

The approach outlined above for iframe_i and $\text{iframe}^{w}_{i\pm x}$ is performed for x=1,2, and 3, resulting in three separate images highlighting the locations of detected vehicles estimated across the intervals specified. As alluded to in Section 2.3.2, these

[#] Using the adaptive Kapur threshold [11]

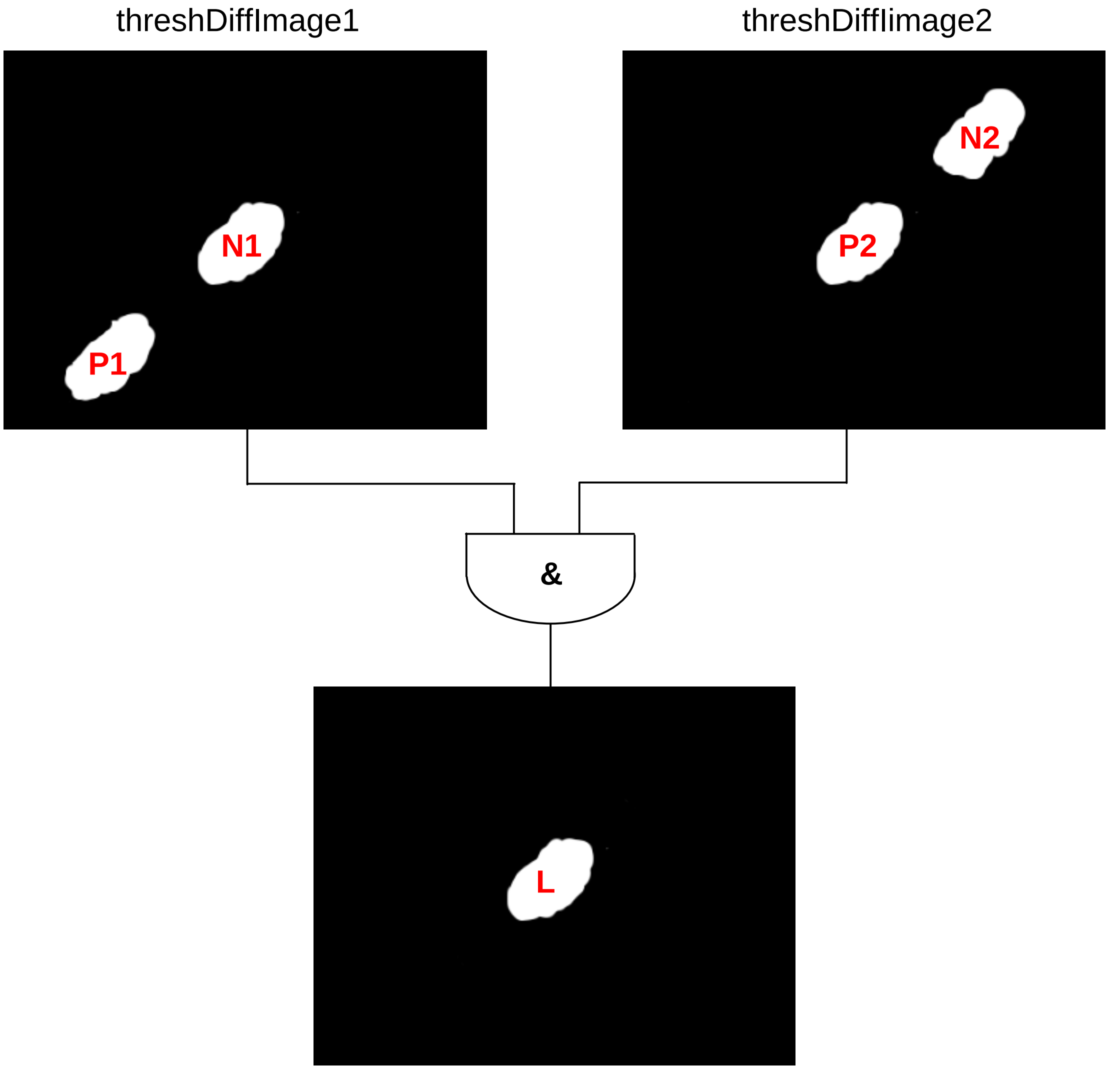

Fig. 3.1. Bi-directional thresholded difference images. Logical AND operation isolates true vehicle location (L).

individual results for each interval are merged (using a logical-OR operation), culminating in a single combined detected-region result for iframe_i, thus maximising the potential for detection across various altitudes and/or vehicle velocities[#].

Once the set of candidate detections have been finalised for iframe_i, they are filtered such that only those conforming to a set of vehicle-style criteria are retained. These criteria are as follows:

(i) **Size**: Candidate regions must exhibit an overall pixel area that lies within the bounds specified[*].

[#] Note, if RGBI-based analysis is invoked, the vehicle detection process outlined for the visible-luminance (L) space is repeated verbatim in the I-feature space. The individual L and I results are then merged via a logical OR operation.

[*] Maximum/minimum candidate region sizes are specified manually on a sequence-by-sequence basis, however, subject to further system development, perhaps these parameters could be automatically self-tuned based on access to real-time data pertaining to UAV altitude and camera zoom length.

(ii) **Solidity**: Regions must exhibit a solidity factor (scalar representing the proportion of the pixels in the *convex hull* that are also in the region) of at least 0.65.

(iii) **Aspect ratio**: Regions must exhibit an aspect ratio (major-axis length ÷ minor-axis length) between 1.0 and 3.5.

All candidate detections satisfying these criteria are considered detected vehicle regions for iframe_i.

N.B. Unsuccessful vehicle detection may result from (i) poor (visible) contrast between foreground object and background resulting in weak pixel difference intensity (which will get suppressed by thresholding), (ii) inaccurate camera modelling resulting in poor spatial alignment of I-frames, (iii) stationary or very slowly moving vehicles. In addition, false detections do feature, and whilst naturally safeguarded against, may arise due to (i) inaccurate camera modelling (poor spatial alignment of I-frames) resulting in multiple difference regions unrelated to vehicle positions, (ii) other non-vehicle objects moving within the scene.

3.3. Vehicle Tracking

The following outlines the implementation of the vehicle-tracking algorithm (incorporating adaptive feature weighting) as deployed in the system.

Given $\mathrm{vehicle}_V$ detected in frame_f (position known), a rectangular *bounding-box* is fitted around the corresponding pixel 'region' (region_V), and an 8-bin spatiogram-bank (S_V) is extracted based on the corresponding pixel values. Note, S_V is either of dimension 3 or 4, depending on whether purely visible (RGB-based) analysis or combined visible-infrared (RGBI-based) analysis is invoked. Either way, a corresponding n-dimensional feature weight vector ($\mathrm{weights}_V$) is then initialised with equal weights in each dimension that sum to unity (i.e. for $n=4$; $\mathrm{weights}_V = [0.25, 0.25, 0.25, 0.25]$, for $n=3$; $\mathrm{weights}_V = [0.33, 0.33, 0.33]$).

Tracking $\mathrm{vehicle}_V$ from frame_f to frame_{f+1} proceeds as follows. Starting at the frame_f position of region_V in frame_{f+1}, a matching process is initiated, which (using $\mathrm{weights}_V$ to bias the feature contribution to the overall similarity score – see Eq. 2) compares S_V to the spatiogram-banks extracted for the set of candidate clips corresponding to a shift in the position of region_V of ±15 pixels in both the horizontal and vertical directions[*]. Depending on the magnitude of the best match similarity score, the object is then (i) deemed to have been either successfully tracked to a new location in frame_{f+1}, or (ii) deemed to have departed from the scene (in which case the tracker is expunged)[#].

If adaptive feature weighting is activated, the constituent values of the $\mathrm{weights}_V$ vector are then updated according to how well the individual features separate the object's true location from other potential background 'distractors' – see Section 2.3.2.3. This is done by computing feature-wise *object-to-distractor ratios*, a process that is outlined in Appendix C. Note, if the mechanism is deactivated the values in $\mathrm{weights}_V$ remain unchanged from their initial uniform values.

[*] Coarse-to-fine search, where ±15 pixels represents a trade-off between processing intensity and the likely bound on the potential frame-to-frame displacement of $\mathrm{vehicle}_V$.

[#] i.e. a threshold of 0.85 is set - a value which the combined spatiogram similarity score must reach in order for the object to be deemed re-located.

Subsequent tracking of vehicle_V from frame_{f+1} to frame_{f+2} proceeds in a similar fashion, i.e. by centring a best-match spatiogram-bank search around the region in frame_{f+2} that corresponds to the frame_{f+1} position of vehicle_V. Once tracked, if adaptive weighting is activated, the weights_V vector is then again updated prior to subsequent tracking, based on the object-to-distractor ratios associated with the frame_{f+2} tracking result. The process then proceeds in a similar fashion for tracking vehicle_V in subsequent frames.

4. EXPERIMENTS & RESULTS

4.1. Experimental Corpus

Towards evaluating the performance of the system developed, twelve appropriately selected, ground-truthed sequences were selected from the DTO VACE 2005 dataset. The sequences were chosen such that they (i) exhibited significant levels of multi-vehicle movement, (ii) exhibited significant levels of camera movement, (iii) were captured from various levels of UAV altitude and/or camera zoom, and (iv) exhibited (intra-sequence) road-surface variance. The video files corresponding to each *Sequence ID* (given) are listed in Table 4.1.

Table 4.1. Sequence IDs and corresponding video files.

Sequence ID	**Video Files**
03_019_07	V4V10003_019.mpg, V4V30003_019.mpg
05_020_18	V4V10005_020.mpg, V4V30005_020.mpg
05_023_20	V4V10005_023.mpg, V4V30005_023.mpg
07_004_26	V4V10007_004.mpg, V4V30007_004.mpg
07_006_28	V4V10007_006.mpg, V4V30007_006.mpg
07_006_29	V4V10007_006.mpg, V4V30007_006.mpg
07_007_24	V4V10007_007.mpg, V4V30007_007.mpg
07_007_25	V4V10007_007.mpg, V4V30007_007.mpg
07_017_33	V4V10007_017.mpg, V4V30007_017.mpg
12_046_38	V4V10012_046.mpg, V4V30012_046.mpg
13_053_44	V4V10013_053.mpg, V4V30013_053.mpg
14_060_47	V4V10014_060.mpg, V4V30014_060.mpg

The ground-truths corresponding to these sequences dictate the entry/exit points (I-frame numbers) of the analysis in each case. These are listed in Table 4.2, alongside their corresponding frame-span counts. Note, the total number of frames involved in the experiments amounted to 11’520, corresponding to approximately seven minutes of UAV-captured footage analysed[*].

[*] While this may seem meagre, the limits imposed on the experimental dataset size relate to the enormous manual effort involved in ground-truth creation.

Table 4.2. Analysis entry/exit point (& corresponding frame-span) for each sequence.

Sequence ID	**Analysis Entry/Exit (I-frames)**	**Span (frames)**
03_019_07	0 – 1776	1776
05_020_18	3588 – 3780	192
05_023_20	4644 – 4836	192
07_004_26	3144 – 3336	192
07_006_28	1788 – 2664	876
07_006_29	5688 – 5880	192
07_007_24	3600 – 4932	1332
07_007_25	5388 – 7140	1752
07_017_33	0 – 180	180
12_046_38	6300 – 7200	900
13_053_44	5004 – 6840	1836
14_060_47	0 – 2100	2100
		Total: 11520

4.2. Performance Metrics & Results

During the analysis of each sequence, for each I-frame, the locations of each tracked vehicle were compared against the actual vehicle locations contained in the ground-truth. In each case, Precision and Recall statistics were calculated corresponding to the extent of the agreement/disagreement. Specifically, for vehicle detection in a given image, the Recall (R) measure was implemented as the number of true-positive pixels expressed as a fraction of the total number of true pixels in the ground-truth, whilst the Precision (P) measure was implemented as the ratio of true-positive pixels to the total number of (true and false) positive pixels detected.

$$R = \#(\text{true-positive pixels}) \div \#(\text{true pixels})$$

$$P = \#(\text{true-positive pixels}) \div \#(\text{positive pixels})$$

The two separate statistics are complementary in expressing the performance of a retrieval system, i.e. whilst Recall measures the fraction of the relevant data retrieved, Precision measures the fidelity of that retrieval to the actual true data.

Table 4.3 presents the average (I-frame) P & R values for each RGB-analysed sequence, with adaptive feature weighting disabled, alongside their respective F-measures[#]. Table 4.4 presents the corresponding results for the same analysis, but with adaptive feature weighting enabled.

For the case of seven of the twelve test sequences listed in Table 4.1, infrared data was processed and included in the analysis, i.e. the experiments were repeated with the infrared data constituting a fourth data channel. Table 4.5 presents the results for the sequences in question.

[#] Usually, Precision and Recall scores are not discussed in isolation. Instead, either values for one measure are compared for a fixed level at the other measure (e.g. Precision at a Recall level of 0.75) or both are combined into a single measure, such as the **F-measure**, which is defined as the *weighted harmonic mean* of Precision and Recall.

Table 4.3. Sequence-average Precision and Recall statistics for RGB-based analysis with adaptive feature weighting disabled.

Sequence ID	**P_{avg}**	**R_{avg}**	**F-measure**
03_019_07	0.21	0.16	0.18
05_020_18	0.37	0.64	0.47
05_023_20	0.68	0.46	0.55
07_004_26	0.68	0.32	0.44
07_006_28	0.56	0.06	0.11
07_006_29	0.41	0.48	0.44
07_007_24	0.65	0.13	0.22
07_007_25	0.48	0.19	0.27
07_017_33	0.57	0.48	0.52
12_046_38	0.72	0.46	0.56
13_053_44	0.60	0.34	0.43
14_060_47	0.78	0.19	0.31
AVERAGE	0.56 (56%)	0.33 (33%)	

Table 4.4. Sequence-average Precision and Recall statistics for RGB-based analysis with adaptive feature weighting enabled.

Sequence ID	**P_{avg}**	**R_{avg}**	**F-measure**
03_019_07	0.22	0.17	0.19
05_020_18	0.37	0.64	0.47
05_023_20	0.69	0.46	0.55
07_004_26	0.68	0.32	0.44
07_006_28	0.56	0.06	0.11
07_006_29	0.42	0.51	0.46
07_007_24	0.58	0.13	0.21
07_007_25	0.50	0.19	0.28
07_017_33	0.57	0.48	0.52
12_046_38	0.72	0.47	0.57
13_053_44	0.60	0.36	0.45
14_060_47	0.78	0.20	0.32
AVERAGE	0.56 (56%)	0.33 (33%)	

Table 4.5. Sequence-average Precision and Recall statistics for RGBI-based analysis with adaptive feature weighting enabled.

Sequence ID	**P_{avg}**	**R_{avg}**	**F-measure**
05_020_18	0.09	0.69	0.16
05_023_20	0.16	0.53	0.25
07_004_26	0.53	0.49	0.51
07_006_28	0.24	0.07	0.11
07_006_29	0.41	0.70	0.52
07_007_24	0.44	0.37	0.41
07_007_25	0.26	0.36	0.30

4.3. Discussion of Results

Considering the results presented in Table 4.3 (RGB-based analysis, adaptive feature weighting disabled), the F-measures indicate considerable variance in performance across the different sequences. Ostensibly, the best performing sequence was 12_046_38 (F-measure of 0.56), and the worst performing was 07_006_28 (F-measure of 0.11). Fig. 4.1 plots the F-measures in order of decreasing sequence length, and the superimposed line-of-best-fit suggests a slight trend towards superior results for the shorter sequences analysed.

In terms of Recall performance, from Tables 4.3 and 4.4, we can see the average of the Recall values corresponds to 0.33 (identical in both cases). This suggests that, in general, the system experienced substantial difficulty in the retrieval of all true vehicle locations throughout the sequences (i.e. on average only managing a 33% accuracy rate in this regard). Further to this, the tables also agree on an average Precision value of 0.56, which suggests that, in addition, a consequence of attaining this 33% true-retrieval rate is the retrieval of (on average) 44% noise, indistinctive from the true-results.

Focussing on Table 4.4 (RGB-based analysis, adaptive feature weighting enabled), we see the effects of adaptive feature weighting. In five cases (05_020_18, 05_023_20, 07_004_26, 07_006_28, 07_017_33), enabling this mechanism had no effect at all (statistics unchanged from Table 4.3). In six cases (03_019_07, 07_006_29, 07_007_25, 12_046_38, 13_053_44, 14_060_47) there was a positive, albeit minimal, effect (i.e. the maximum F-measure increase was 2%, occurring in 07_006_29 and 13_053_44). However, a negative effect was experienced in sequence 07_007_24, where a decrease of 7% Precision contributed to a F-measure decrease of 1%. Hence, in terms of performance accuracy, it seems that, save for this latter outlier, the effects of the adaptive feature weighting on this test set were generally positive, albeit quite moderate.

Turning to Table 4.5, where the statistics are listed for the seven sequences upon which RGBI-based analysis (adaptive feature weighting enabled) was performed, we see the effects of adding the infrared signal to analysis alongside the

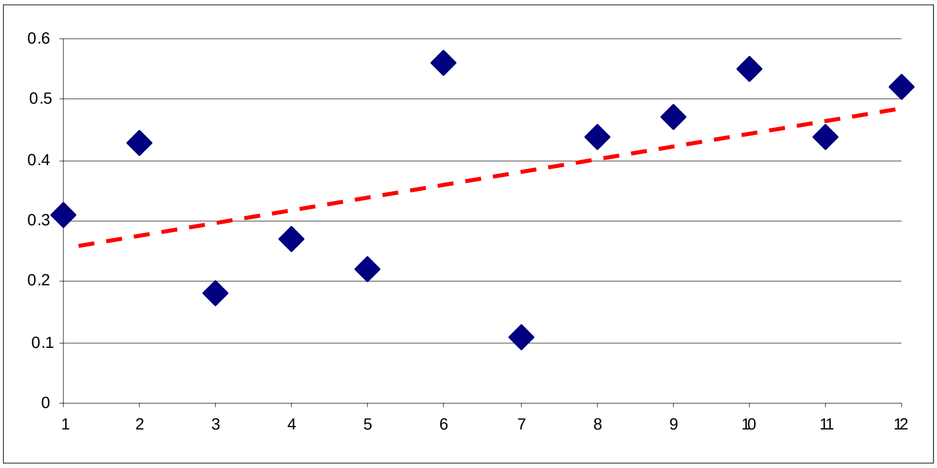

Fig. 4.1. F-measures plotted in order of decreasing sequence length, plus line-of-best-fit.

RGB data. Here we can see a marked variation from the corresponding values in Table 4.4. Explicitly, apart from one sequence where the F-measure remained unchanged (07_006_28), four of the seven sequences analysed exhibited slightly improved performances (07_004_26, 07_006_29, 07_007_24, 07_007_25), whilst the remaining two sequences (05_020_18, 05_023_20) experienced deterioration in performance. Common to all affected sequences was a marked improvement in Recall values, and whether or not this improved Recall figure resulted in an improved F-measure for a sequence depended on the price paid in Precision. That is, clearly some sequences experienced substantial losses in Precision accompanying their Recall increases, whilst for other sequences the Precision loss was not so significant. Hence the disparity in overall perceived performance based on F-measures. In general, it seems that the inclusion of the fourth infrared feature can facilitate better retrieval, but at the potential cost of fidelity in the results retrieved.

Altogether, whilst the results presented above suggest promise in the approaches tested for vehicle detection/tracking in UAV data, the overall performance is far from ideal, and an investigation into the underlying reasons for these imperfections is due.

4.4. Analysis of Results

Given the experimental results discussed above, an exercise in failure analysis was initiated, whereby an attempt was made to identify the main causes of imperfections in the performance of the system.

4.4.1. Recall Performance

Considering first Recall performance, clearly ideal (100%) Recall would correspond to all vehicles within the scenes being first detected, then 'perfectly' tracked (i.e. with detected bounding-boxes exactly matching those specified in the ground-truth).

With an average sequence Recall rate of 33%, the performance of the system is clearly unsatisfactory in this regard, and upon failure analysis, the root causes of Recall deficiency were found to relate to the following:

- Vehicle detection failure.
- Tracking failure due to distractions.
- Tracking failure due to object appearance variation.
- Subjectivity of the ground-truth.

Considering the first of these, recall that robust vehicle detection relies on (i) the integrity of the camera model, (ii) vehicle movement within the scene, and (iii) contrast between moving vehicle and its background. Of these, reliable camera model extraction proved to be the most challenging aspect of the problem. For instance, in many cases the sequences in question exhibited vast scenes of vegetation, which presented difficulties for the corner-based scene analysis approach utilized[#]. Hence reliable camera models were not always computable for some extended periods of the sequences - rendering vehicle detection non-operational, and thus impacting on the

[#] Overall, the camera modelling performance proved most robust when processing images from urban-based scenes.

Recall rate observed. Furthermore, assuming accurate camera modelling, recall that for vehicles to be detected they must exhibit a reasonable level of movement within the scene. However, the ground-truth was indiscriminate between moving or non-moving vehicles within the scene, and hence the non-detection of non-moving vehicles in the scene was another source of penalisation in terms of Recall rate. Finally, recall that the approach deployed for the detection of moving vehicles assumes a level of contrast between the vehicle and its background, which is then exploited via image differencing towards highlighting regions of foreground activity in the scene. In some cases (e.g. RGB-based analysis) there was little or no contrast compared to the level required, thus inhibiting vehicle detection in some cases, and therefore again compromising Recall. Note, this latter problem was alleviated somewhat by the inclusion of a infrared signal (RGBI-based analysis), but associated with this was a typical penalty in Precision.

The above analysis strives to illustrate how Recall figures were affected by failures in the vehicle detection module. In terms of the tracking related causes listed, recall that the tracking system is based on extracting a mathematical model (spatiogram) of the object, and then comparing this model against those extracted for various potential new positions as we move through the subsequent images. Distraction is when a tracker (underpinned by such a matching algorithm) loses its true target because it gets 'distracted' by another region within the search-space that, typically by coincidence, exhibits a closer mathematical ('spatiogrammical') match to the original model. This typically results in tracking failure, whereby the moving true target travels outside the search-space, and thus is unable to be re-found, whilst the tracker remains ‘tracking’ the erroneous group of pixels. Such a scenario obviously results in a drop in Recall due to the true target no longer being retrieved (as well as a corresponding drop in Precision due to the existence of a tracker following noise – see below). Closely related to this concept is tracking failure due to changes in object appearance, whereby at some stage during the sequence the object (vehicle) exhibits a substantial change in pose, rendering the spatiogram non-representative to the extent that the search algorithm matches an erroneous block of pixels rather than the those corresponding to the true target, with the same negative consequences for Recall (and Precision). These two phenomena account for a significant loss of performance accuracy in the system.

Finally, the subjectivity of the ground-truth also had some consequence on the Recall rates observed. The ground-truth consisted of hand-drawn bounding-boxes (rectangles) demarcating the true vehicle locations within the sequence images. However, the subjectivity of this process frequently resulted in many bounding-boxes exhibiting a greater pixel area than those returned by the automatic analysis of the system. Hence, even in cases where the detection/tracking accuracy would be classified as close to ‘ideal’ from a human analysis viewpoint, owing to the discrepancy in pixel grid overlap between result and ground-truth, corresponding Recall figures did not reflect this. Hence, a slight flaw in the evaluation process also served to compromise the reflection of the true performance represented via the statistical results.

4.4.2. Precision Performance

Ideal (i.e. 100%) Precision corresponds to the retrieval of zero false positives, irrespective of Recall performance.

With an average Precision rate of 56%, the performance of the system is clearly unsatisfactory in this regard, and upon failure analysis, the root causes of its Precision deficiency were found to relate to the following:

- Camera model inaccuracy.
- Tracking failures

Considering the former, given an inaccurate camera model, the image differencing performed in the camera motion compensated domain can result in multiple highlighted regions that correspond to noise rather than true vehicle locations. Although many of these may be filtered out via the vehicle-styled criteria that the candidate regions characteristics must adhere to (see earlier), many are not, and go on to be considered true vehicle locations - subsequently going on to have corresponding (false) trackers initiated, and thus compromising the Precision performance of the system in doing so.

In addition to the above, Precision is also affected by the appearance of residual distracted trackers, i.e. trackers who have lost their true target, and who are now remain tracking an erroneous group of pixels within the scene. The distracted tracker will generally exist until as long as the tracked ‘object’ remains visible in the scene, penalising Precision performance throughout.

4.5. Conclusions

In aiming to draw some conclusions from the analysis presented above, clearly one of the main sources of imperfection in the system corresponds to lingering distracted/false trackers. Although many safeguards were established to prevent their occurrence, complete eradication of these phenomena proved futile. Furthermore, it was found that, once occurring, such noise was extremely difficult to eradicate, i.e. without detriment to the Recall performance corresponding to the detection/tracking of true targets. Given this, it becomes clear why the statistical performance (F-measure) of each sequence tends to be inversely proportional to sequence length (Fig. 4.1).

One of the main safeguards against the profusion of tracker distraction was the adaptive feature weighting mechanism, which operated by assigning most influence to the best performing feature in a dynamic adaptive fashion. Whilst the average Precision/Recall rates were unaffected by the activation of this mechanism (i.e. unchanged from Table 4.3 to Table 4.4), on an individual sequence basis there is evidence of a slight trend of improvement in performance, suggested by simultaneous increases in both Precision and Recall in the majority of cases. However, whilst the effects are positive, they are quite moderate (i.e. of the order of 1% or 2%), and in situations where processing time is paramount, any delays associated with enabling the mechanism (yet to be determined) may render it redundant.

It was shown that the addition of an infrared signal to the group of visible features (i.e. RGBI-based analysis) served to benefit the performance in terms of Recall, especially owing to increased effectiveness in vehicle detection (whereby, in the main, the true effect of this feature was in overcoming the contrast issues associated with the visible spectrum - see earlier). However, benefits (or otherwise) in terms of overall performance accuracy (as indicated by the F-measure) seemed to be somewhat inconclusive, owing to the spurious fluctuation in Precision typically induced by the inclusion of this additional data source.

In summary, adaptive feature weighting was shown to offer some improvement in performance accuracy, but only minor, whereas the effect of the including an infrared signal into the analysis was shown to be beneficial for the majority of cases, but with more testing required (non-trivial due to the enormous overhead currently associated with manually aligning the visible and infrared streams – see Appendix B).

Overall, towards increased performance accuracy, the recommendation is that future work should concentrate on (i) improving the algorithms for camera modelling (i.e. towards unrestrained scope for vehicle detection, as well as limiting the number of false detections), and (ii) investigating new algorithms for the eradication of false/distracted trackers (i.e. where the challenge relates to not compromising the tracking of true targets).

APPENDIX A: Project Milestones (July’08 – July’09)

- **July’08:** Background research into fields of vehicle tracking and performance evaluation.

- **August’08**: Development of data integrator system, for combining visible and infrared video streams.

- **September’08:** Development of data formatter system, for the appropriate configuration and storage of the dataset.

- **October’08:** Development of preliminary vehicle tracking module.

- **November’08:** Development of preliminary camera modeling system.

- **December’08:** Development of preliminary vehicle detection module.

- **January’09:** Development of preliminary adaptive feature weighting mechanism.

- **February’09:** Refinement of detection module (reduction of false detections).

- **March’09:** Refinement of tracking module (including improved speed performance); Establishment of oriented bounding-box display.

- **April’09:** Refinement of adaptive feature weighting mechanism; Processing and formatting ground-truth; Initial experiments and qualitative evaluations.

- **May’09:** Further refinement of detection and tracking modules.

- **June’09:** Quantitative evaluation of system performance; Writing of reports; Transfer of software-related deliverables.

- **July’09:** Writing of reports. Transfer of final documents.

APPENDIX B: Manual Alignment of Visible-Infrared Video

For a given sequence, the temporal discrepancy between its infrared and visible streams is typically of the order of one or two seconds, but needs to be estimated as accurately as possible (i.e. at frame level) so that the extracted visible-IR video images may be precisely aligned. The approach taken here for each sequence is to manually traverse the video and pick out time signatures for a handful of unambiguous ‘events’ that are evident in both streams. Once this is done, we can estimate the temporal discrepancy between the two streams by comparing the relevant time signatures for the set of events selected. Given this estimate, we then proceed to offset the image set from either the visible or infrared (whichever is relevant) by the particular value, thus producing a set of visible and infrared images that are in temporal agreement. Note, it is assumed that any temporal discrepancy between corresponding visible and infrared streams is a fixed offset, and once determined via the method described, doesn’t vary throughout the duration of the sequence.

There also tends to be a discrepancy between the perspective of the scene areas captured by the visual and IR streams, a second consequence of the dual camera set-up. Although the cameras are ostensibly positioned side-by-side in the UAV to minimize this discrepancy, any slight difference clearly needs to be accounted for. The approach undertaken here is as follows. Firstly, it is assumed that the spatial discrepancy between the two streams is fixed for each sequence. This is justified on the basis of the assumption that two cameras are in a fixed position relative to each other within the UAV. Given this, we compute a geometric transform (3x3 planar

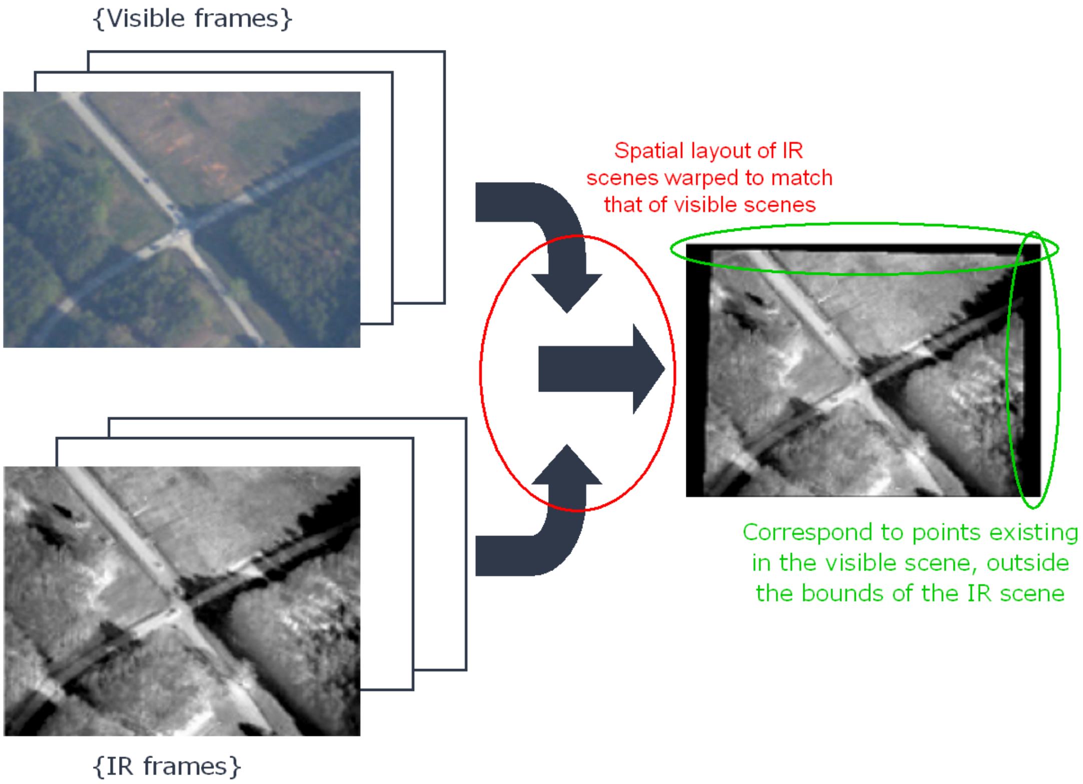
{Visible frames}
Spatial layout of IR scenes warped to match that of visible scenes
Correspond to points existing in the visible scene, outside the bounds of the IR scene
{IR frames}

Fig. B1. Warping of the infrared frames of a sequence so that they are spatially aligned with their visible frame counterparts.

homography matrix) that *maps* the orientation of the spatial area captured in the IR stream frames to that of the visual stream frames. Given the assumption of a fixed discrepancy, there should be one unique set of homography values representing this transform for an entire sequence. The idea is that we can then use this sequence-level homography to *warp* the images of the IR stream such that they become *spatially aligned* with their visible stream counterparts – see Fig.B1.

The method by which the abovementioned homography is computed is an offline manual calibration step and is outlined as follows: A selection of images from the (temporally aligned) visible and infrared streams are juxtaposed on a computer screen display. The user then '*mouse selects*' corresponding points in each image, i.e. a point-of-interest in the infrared scene, followed by the corresponding position of that point in the visible scene. The user generates several dozen of these '*anchor points*' across several frame pairs of the sequence. These are then fed into an optimisation algorithm [9], which attempts to locate the 'homography of best fit', representing the spatial offset between the two streams.

APPENDIX C: Algorithm for Adaptive Feature Weighting

The algorithm for the adaptive weighting of features is based on the computation of *object-to-distractor ratios*, and is outlined below.

Consider the successful tracking of object$_o$ from its previous position in frame$_{i-1}$ to current position in frame$_i$ in (e.g.) a 4-D feature space based on equal weights vector weights$_o$ = [0.25 0.25 0.25 0.25]. It is desired that prior to tracking object$_o$ from frame$_i$ to frame$_{i+1}$, the weights be first modified such that they reflect the (current) relative performance of each feature in discriminating the object from the background. To this end, the area surrounding the current object position is searched[#] for its corresponding '*distractor*', which is defined as the (same sized) region exhibiting the closest 'spatiogrammatical' match to that of the object in question. In other words, the distractor is defined to represent the region of closest similarity to the object (in the vicinity of the object). The distractor is located in the same way the object matching is performed, i.e. via a comparison of combined spatiogram similarity scores.

Once the distractor is located for the object, for each feature, an individual *object-to-distractor ratio* (ODR) is computed by calculating the ratio of its corresponding object similarity score (relating to the successful tracking of object$_o$ from frame$_{i-1}$ to frame$_i$) to that of its corresponding distractor similarity score. This results in four ODR values corresponding to the four features concerned. The ODR values are then normalised such that they sum to unity, upon which they are then considered to constitute the new (updated) weights of weights$_o$. In this way when tracking object$_o$ from frame$_i$ to frame$_{i+2}$ the feature exhibiting the highest ODR will have an increased influence (to the degree to which its ODR value exceeds that of the other features, and so on).

[#] To find a set of background candidates, the area around the current object position is sampled at 16 pre-defined locations corresponding to horizontal and vertical shifts around the true object (region) position of ± 0.5*width, ± 0.75*width, ± 0.5*height, and ± 0.75*height. N.B. These are chosen to give good overall coverage of potential areas of distraction, without overly expending in computation.

REFERENCES

[1] http://www.mathworks.com

[2] D. Sadlier, N. O'Connor, A. Smeaton, 'Vehicle Tracking in UAV-Captured Video Data via Adaptive Weighting of Visual Features: 6-Month Interim Report', 2009.

[3] S. Avidan, Ensemble Tracking, in IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2005.

[4] R.E. Bellman, Adaptive Control Processes: A Guided Tour, Princeton University Press, Princeton, NJ, 1961.

[5] http://www.perceptual-vision.com/vt4ns/vace_brochure.pdf

[6] S.T. Birchfield, S. Rangarajan, Spatiograms versus Histograms for Region-based Tracking, in IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2005, pp. 1158-1163.

[7] C.S. Kenney, M. Zuliani, and B.S. Manjunath, "An Axiomatic Approach to Corner Detection". In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cvpr'05) - Volume 1 - Volume 01 (June 20 - 26, 2005). CVPR. IEEE Computer Society, Washington, DC, 191-197. DOI= http://dx.doi.org/10.1109/CVPR.2005.68

[8] C. Harris and M.J. Stephens. "A Combined Corner and Edge Detector". In Alvey Vision Conference, pages 147–152, 1988.

[9] P.D. Kovesi, A General Purpose Implementation of the RANSAC Algorithm. MATLAB and Octave Functions for Computer Vision and Image Processing. School of Computer Science & Software Engineering, The University of Western Australia.
Available from: http://www.csse.uwa.edu.au/~pk/Research/MatlabFns/

[10] DOI: ftp://ftp.cs.utoronto.ca/pub/jepson/teaching/vision/2503/tutorial2.pdf

[11] J.N. Kapur, P.K. Sahoo, A.K.C. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram". Comput. Vision Graphics Image Process. 29 (3), 273–285.