AVITRACK

Contract n° AST3-CT-2003-502818

D. 3.3
Object Categorisation / Recognition

Version 1.0– Draft 1
Reference DL_AVI_2_015

Partially funded by the European Commission under the 6th Framework Program

Contract Number :	AST3-CT-2003-502818
Document number :	D 3.3
Document Title :	Object Categorisation / Recognition
Document version :	1.0
Document status :	Draft
Date :	1/8/2005
WP contributing to the deliverable :	WP 3
Availability :	
Authors :	N.Carter, M.Borg, D.Thirde, J.Ferryman (UoR).
Approved by :	

Abstract

This report describes the work performed on Object Categorisation and Recognition, as part of task 3.3 in WP3 of the AVITRACK project. Bottom-up and Top-down classifiers are described, and how these approaches were applied to the apron environment.

Keyword List

Classification, Categorisation, AVITRACK, airport apron, Vehicle Recognition, Bottom-up classifiers, Top-down classifiers, 3D Model-based, iconic evaluation, edge-based, appearance evaluation, SIMPLEX, Simulated Annealing, descriptors, statistical descriptors, invariant, non-invariant, image sub-sections, point of interest operators, Harris operator, affine invariant descriptors, extremal regions, SIFT, distance classifiers, Bayes probability, salient image sections, normalisation, euclidean distance, Mahalanobis distance, KNN, linear discriminate analysis, neural network, eigen decomposition, eigen windows, hierarchical classification.

DOCUMENT CHANGE LOG

Document Issue.	Date	Reasons for change
v0.1	10/02/05	Initial draft version.
v1.0	12/09/05	First draft external release

APPLICABLE AND REFERENCE DOCUMENTS (A/R)

A/R	Reference	Title
	Please refer to the Reference Section at the end of this document.	

Table of contents

1. INTRODUCTION

This document describes the work performed on Object Categorisation for the AVITRACK project. This work forms part of work package WP3, task 3.3 [1].

The next section (Section 2), gives a brief overview of object categorisation and introduces the general characteristics of Multi-Camera Tracking Systems that directly influence the categorisation process, together with issues and problems that need to be addressed. The next sections will then introduce the work performed for AVITRACK, describe the issues and the different algorithms used for categorisation of people and the recognition of vehicle types: Section 3 deals with top-down model-based recognition used for classification, while Sections 4 to 7 describes bottom-up approaches to classification; finally Section 8 describes how top-down and bottom-up approaches can be combined together. Section 9 concludes the report with a brief description of the results obtained and the issues identified when applying categorisation to the apron environment, and possible future work is listed.

2. MULTI-CAMERA SYSTEMS AND CATEGORISATION – OVERVIEW & ISSUES

Object classification is the process of assigning class ownership to a set of image values (pixels). Object classification constitutes an important part of the AVITRACK project as a whole, without which, behavioural analysis would have no firm basis upon which to act, as object behaviour can only be ascertained once the object type is known [2].

There is currently no available system capable of perfect computational object recognition / classification; however, there are numerous techniques, which provide good recognition results in constrained circumstances. This task is also a very heavily researched area of computational vision, with improvements and new techniques surfacing regularly.

Currently, classification is based on two-dimensional images taken from up to eight different camera locations. These images are automatically segmented into probable object areas described by bounding boxes. The object may not consistently be in the same location within this bounding box and may change in appearance dependant upon outdoor conditions and the objects orientation and distance from the camera (among other things). These issues are further outlined below.

2.1. ISSUES

There are many issues associated with object classification or recognition tasks, some of the most pronounced include [2]:

Perspective – the AVITRACK project uses normal CCD cameras to capture images of the airport apron. These cameras, like others, are affected by perspective. That is to say, objects, which are further away, appear smaller than objects in the foreground, even if the object further away is in fact larger. This adds distortion and makes it more difficult to use size as a classification or hypothesis method. This problem is somewhat alleviated in this project as most of the objects move and interact in the camera foreground. Object size, concerning the classification stage, can also be somewhat compensated for through the use of re-sampling algorithms.

Lighting – as the objects to be classified are often identified through techniques based on object appearance, lighting can be very influential. A change in lighting conditions, especially prevalent in outdoor conditions such as in the AVITRACK situation, possibly caused by weather or the camera itself, can drastically alter the perceived appearance of an object. This problem can be somewhat reduced through techniques such as histogram equalisation.

Shadows – shadows constitute a major problem for all outdoor, and to a lesser extent indoor classification tasks. Shadows change the size of the tracked object and add extra regions to classify, which in fact hold little usable data. Shadows are also hard to suppress, as they do not hold a uniform shape or size. Efforts are being made within the AVITRACK project to minimise these problems.

Occlusion – when an object passes in front of any tracked object, the tracked object is said to be occluded. The occluding object may be stationary or in motion, and may take any form. When this occurs, the overall appearance of the tracked object changes in very unpredictable ways. This problem can be minimised through the recognition of subsections of the image to be classified instead of the image as a whole.

Image Transformations – this concerns translation and rotations to the image. Images can differ greatly dependant on the rotation and position within the objects bounding box. The global position of the object is resolved by the current implementation of the system, however, the objects bounding box returned by the system may only have part of the object – in the case of an object entering the scene – and may present the object at any rotation.

Availability of Exemplar Objects – it is often difficult to find objects, which exemplify their respective class, especially within object classes that can deform, for example people, articulated vehicles and vehicles with external moving parts. The objects are also, as mentioned above, often miss-tracked or the bounding boxes contain large amounts of unwanted background pixels.

2.2. CHALLENGES IN APRON ENVIRONMENT

In addition to the the generic issues mentioned above, the apron environment of AVITRACK presents further challenges to object classification.

Objects in the AVITRACK sequences fall into 28 different categories: 2 people categories (singular or group), 1 aircraft category, 3 equipment categories, and 22 ground vehicle categories. (Refer to [78] for an illustration of some of these objects that need to be classified). The challenges for AVITRACK are in the quantity and the similarity of objects to be categorised. For example, most of the ground vehicles have the same colour (white) and are approximately of the same size. Therefore, the simple descriptors used in many visual surveillance systems are likely to fail.

2.3. CLASSIFICATION – AN OVERVIEW

The work detailed within this report was inspired and is based, to certain degrees, upon work undertaken by other researchers within the computer vision & object classification field. Within this vast field, numerous approaches to computerised object classification become evident; these can be loosely grouped as classifications based upon:

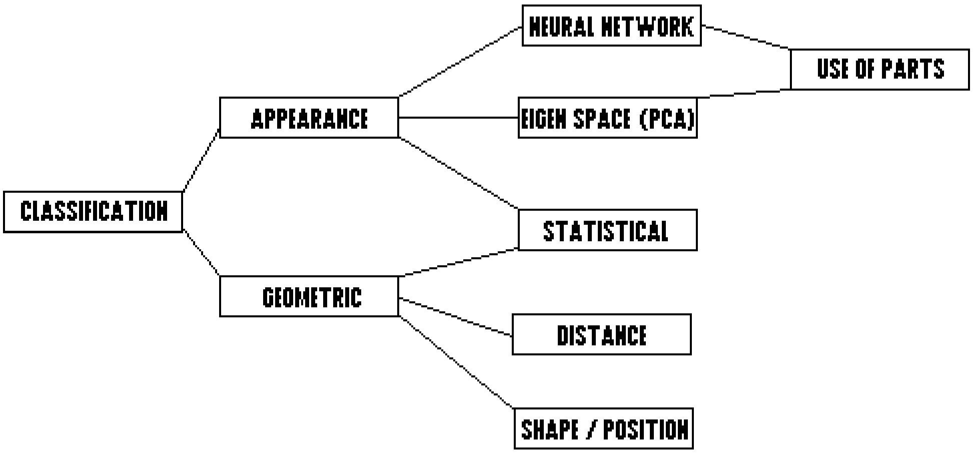


Figure 2.1: Different Classification Methods

Although many statistical approaches exist, the approaches mainly focus on probability density functions [5], maximum a posteriori estimation [5&6] and Bayesian decision networks, which may come in useful if efforts are made to bias the object type within the project based upon object location in the global scene.

Probability density functions may prove useful for some of the more advanced distance / size classifiers (for example, Mahalanobis distance classifiers).

Coots and Taylor [15] have adopted a statistical approach (T.F. Coots & C.J. Taylor 2004) where statistical shape models were used to locate face objects. Sub regions of the image were land marked (areas of maximum curvature were located), then a statistical model of the face was built, incorporating certain modes of variability. This approach produced good results, although this system may prove difficult to use in the AVITRACK situation as the objects can be highly variable and occlusion may hamper recognition.

Pham and Smeulders [6] (T.V. Pham & A.W.M. Smeulders) also used a statistical approach to object recognition and classification, however, unlike Coots & Taylor, parts of objects, rather than the whole object where used as classifiers. This allowed their system a certain degree of occlusion tolerance. This may be a good technique to implement in the final system, as the statistical nature, coupled with the occlusion tolerance allows accurate and robust classification.

This use of subsections of tracked image allowed Sali and Ullman [9] (E. Sali & S. Ullman) to use combinations of common sub-structures, which they termed “fragments” as a classification tool. The stored fragments were compared to sections of candidate images, then a similarity metric was used to classify the object overall.

Nelson and Selinger [11] (R.C. Nelson & A. Selinger) took the idea of fragments one stage further by comparing them with ideas found in the cubist art movement, arguing that objects can be represented by a few key fragments for computerised classification, and suggesting that human perception of objects may function in a similar way.

Moving away from the idea of fragments and image patches, Belongie, Malik & Puzicha [7] use a shape matching technique based on shape contexts. They present a novel approach to measuring similarity between shapes, which involves solving correspondences between points on two shapes, then using these points to create an aligning transform. This system works very well for tasks with uniform backgrounds, however, as soon as background clutter is introduced the speed and accuracy of the system decreases, as extra, erroneous correspondences are solved.

As well as shape, object position can be important in ascertaining object class. Bose and Grimson [8] presented a paper discussing the importance of scene context, as well as the use of object variables. Scene context concerns the position and orientation of objects within a scene. This has definite application within the AVITRACK project, as certain objects have a greater a priori probability of occurring in set areas. The main drawback to this system is that most objects *can* appear in almost any location.

Eigen space classification has become very popular [13, 14 & 16]. Ohba and Ikeuchi present an Eigen based system using subsections of the image, which recovers object pose and type on a uniform background.

Black and Jepson [16] (1996) use and Eigen Window approach to locate objects (drinks cans) in cluttered background scenes, which demonstrates the power of this technique and its applicability to the AVITRACK problem domain.

The creation of hyper planes for high dimensional data and the use of nearest neighbour classifiers is shown to be effective by [17, 18 & 19], and efforts are being made to speed up the entire process [20]. Hyper planes and associated strategies are used to define decision boundaries between different object types efficiently. Other techniques achieve similar aims in lower dimensional space, such as Euclidean or Mahalanobis distance classifiers, and the perhaps more applicable Eigenspace technique.

Most of the methods reviewed so far, can be classified as bottom-up techniques, in that no high-level scene information is used in the classification process; instead they look at inherent structure in the video signal.

Top-down approaches have also been used for object classification/object recognition tasks, such as [76, 77 & 79], where high-level information, 3D geometric models, of the objects of interest are used. Having knowledge of these models, the classification problem is then reduced to a search problem: identifying and locating a specific object, through the use of its model, in the scene. And if found, the object is then assigned the class ID of the model. The work performed by [76 & 77], used both wire-frame (edge-based) 3D models as well as textured 3D models. While in [79], geometric invariants are extracted from a 3D model, from which algebraic relations between the 3D invariants and the 2D scene can be derived, and used to locate the object in the scene.

3. TOP-DOWN MODEL BASED RECOGNITION

Top-down classifiers use high-level information about the scene to help them identify the category of objects. One such type of high-level information that can be used is 3D geometric models. Relying heavily on previous work performed by the CVG group at The University of Reading on model-based tracking [76, 77], it was decided to try and apply this work to the problem of categorisation in the AVITRACK project. Although not as fast as bottom-up classifiers, 3D model-based classification/recognition can provide accurate results when successful. It was also envisioned that a mixture of bottom-up and top-down techniques will be investigated to gain the benefits of both approaches – accuracy and computational efficiency.

3.1. 3D MODEL FORMALISM

The 3D model formalism adopted for AVITRACK is described in more detail in Deliverable D1.3A [75]. In brief, the 3D model consists of a geometric model composed of points, lines, and planar facets (resembling a wireframe model), and a few geometric constraints. The model is specified in a syntax (and file format) called the *primitive* file format (or *prim* for short). Different models can be combined in a hierarchical fashion to allow more complex models to be constructed, with the top node representing the scene (the world). Each node (model) in this hierarchical tree has its own frame of reference, and affine transformations between the nodes allows objects to be moved in the scene with respect to each other. In [77], the *prim* format was extended to add appearance-based information to the geometric information – an illumination texture map is used for each facet to store this appearance information, sampled regularly along the facet's surface. An illustration of the model is given in Figure 3.1 below:

(a) 3D facet (wireframe) model for the Transporter vehicle;

(b) 3D model with appearance information.

Figure 3.1: Model formalisms

3.2. CLASSIFICATION PROCESS

The model-based classification (recognition) process consists of:

- the object model is placed at the world position being tested, through a transformation from the object's frame of reference to the world frame of reference.
- the model is transformed from the world frame to the given camera's frame of reference, and a hidden-line removal algorithm is run to find the lines and facets which are visible from the camera.
- the visible part of the model is back-projected on to the image and evaluated against the image data, resulting in a match score.

An exhaustive search would require running the above process at all possible world points and all possible orientations $(x,y,z,\varphi_x,\varphi_y,\varphi_z)$, to get the point of best fit. In systems like AVITRACK, the *ground plane constraint* can be used to restrict the search range to (x,y,φ_z), where (x,y) is the position on the ground-plane $z=0$ and φ_z represents the rotation of the object around its vertical axis – collectively (x,y,φ_z) are called the *pose* of the model. (While the ground-plane constraint is not completely true all the time – e.g. a container object is elevated above the ground by the loader – it is valid for most of the time; additional context information may be used to handle above-ground objects).

The evaluation function used to test the model normally gives rise to a uniform, continuous evaluation surface with a global maximum, upon which numerical search optimisation techniques can be applied, e.g. the SIMPLEX algorithm [80]. Given an initial estimated position (seed position), the search algorithm will attempt to locate the global maximum in the search space.

3.3. EDGE-BASED (ICONIC) EVALUATION

Edge-based or iconic evaluation is based on the idea that the edges of a model's facets, line markings in facets and extremal edges, should all create illumination discontinuities in an image viewed by a camera. The evaluator projects these model lines on to an image, and searches the surrounding area for evidence of discontinuities by analysing image gradient. For each model line (see Figure 3.2 below), the *k* normals at regular intervals of *f_int* along the line's length *f_l,* are found. Each normal is then sampled at regular intervals of *n_int* (shown as blue points in Figure 3.2), with *n_hn* being the number of samples along each half normal. The evaluation score for each model line is then given by [81]:

$$e_i = \sum \left(dI \left(v \right) \right) w \left(j \right)$$

where *dI(v)* is the discrete directional derivative of the greylevels along the normal, and *w(j)* is a triangular weighting function that monotonically decreases away from the origin.

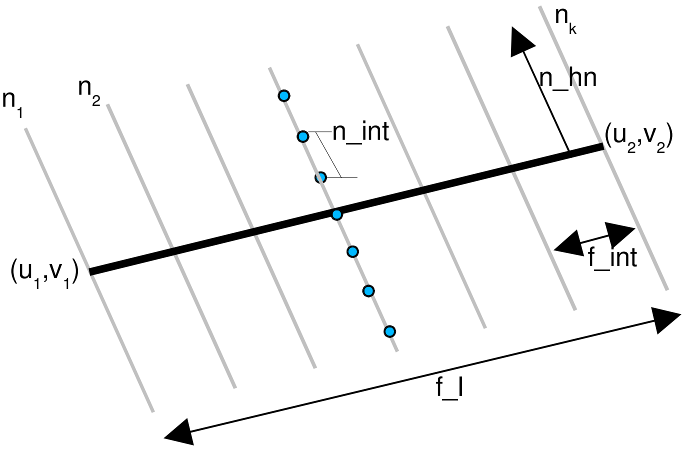


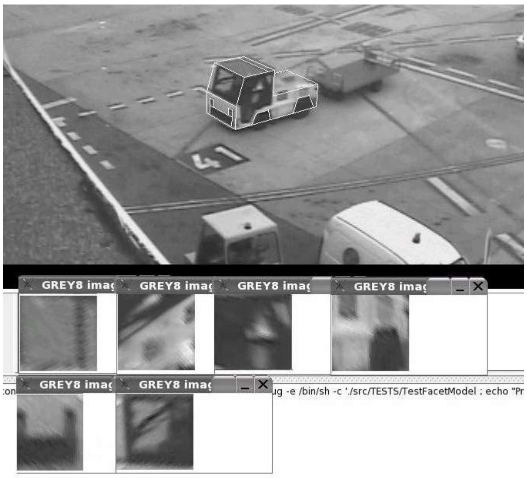
Figure 3.2: Iconic evaluation of a model line (from [81]).

A match probability value is then generated from the score e_i, taking into account the probability that the line segment does not originate from the model (the *null hypothesis*), i.e. it occurs in the background (empty) scene. To calculate this, an offline process is required that creates probability tables for the occurrence of line segments in the empty scene for a set of arbitrary line lengths. For AVITRACK, the process performing this is called `ModelEvaluationInit` and given an image of the empty scene viewed by a camera, it projects lines of arbitrary length randomly in the image and generates the required tables.

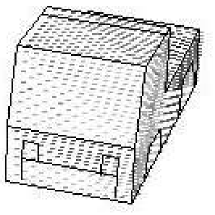
3.4. APPEARANCE-BASED EVALUATION

Appearance-based evaluation, unlike edge-based/iconic evaluation, uses more of the information content of an image. For each facet of a 3D model, the appearance information is represented by a *point model* – this is obtained by sampling the facet at regular intervals in a grid-like fashion and storing the brightness at each point. The sampling rate is determined by the size of the facet and a user-configurable parameter *ppm* which specifies the points per metre.

Learning of the appearance models in AVITRACK is performed by running an offline utility called `ModelEvaluationInit`. This allows the user to browse through pre-recorded video sequences for good examples (good views) of a certain object, to load the 3D model from the prim file, and after manually positioning the model in the image, start the learning phase.



```
ug -e /bin/sh -c './src/TESTS/TestFacetModel ; echo "Pr
```



(Left) learning the appearance model from a manually-fitted model on a test image; texture maps of facets displayed in small windows underneath. (Top) The points sampled on the facets with ppm=30 (low resolution).

Figure 3.3: Appearance Model Learning

For evaluating the appearance information during categorisation, one of two similarity measures are used:

1. Sum of Squared Differences (SSD):

For each facet, the SSD measure *e* is given by:

$$e=\sum\left[f(p')-m(p)\right] \tag{3.1}$$

where for each point *p* of the point model, *p'* is the corresponding image point found by back-projecting *p* on to the image; *m(p)* is the brightness determined from the appearance model, and *f(p')* is the image brightness.

2. Normalised Cross-Correlation (NCC):

NCC is more robust to illumination changes than SSD:

$$e=\frac{\sum\left(f(p')-\bar{f}\right)\left(m(p)-\bar{m}\right)}{\sqrt{\sum\left(f(p')-\bar{f}\right)^2\sum\left(m(p)-\bar{m}\right)^2}} \tag{3.2}$$

where $\bar{m}$ is the average brightness of the points the point model; and $\bar{f}$ is the average of the corresponding image points.

Combining the scores:

The above SSD and NCC error measures are calculated separately for each facet. Combining the facet scores, must take into account the fact that as a 3D model rotates, new facets come into view and others disappear (are self-occluded). A simple combination such as averaging would introduce discontinuities in the evaluation surface at the point a facet suddenly disappears/appears. This would make life very difficult for the search algorithms. The method adopted for AVITRACK, weights each facet score by:

1. the angle the facet's normal makes with the camera's optical axis (ranging from 1.0 for facets seen face-on, to 0.0 for facets seen edge-on or if hidden). This is given by w_a in the equations below.
2. the facet's size over the total visible surface area of the model (as projected on to the image plane). This is given by w_b in the equations below.

The final model score is then:

$$e=\frac{\sum_{facet\,j}\left[e(j)\,w_a(j)\,w_b(j)\right]}{\sum_{facet\,j} w_a(j)\,w_b(j)} \qquad (3.3)$$

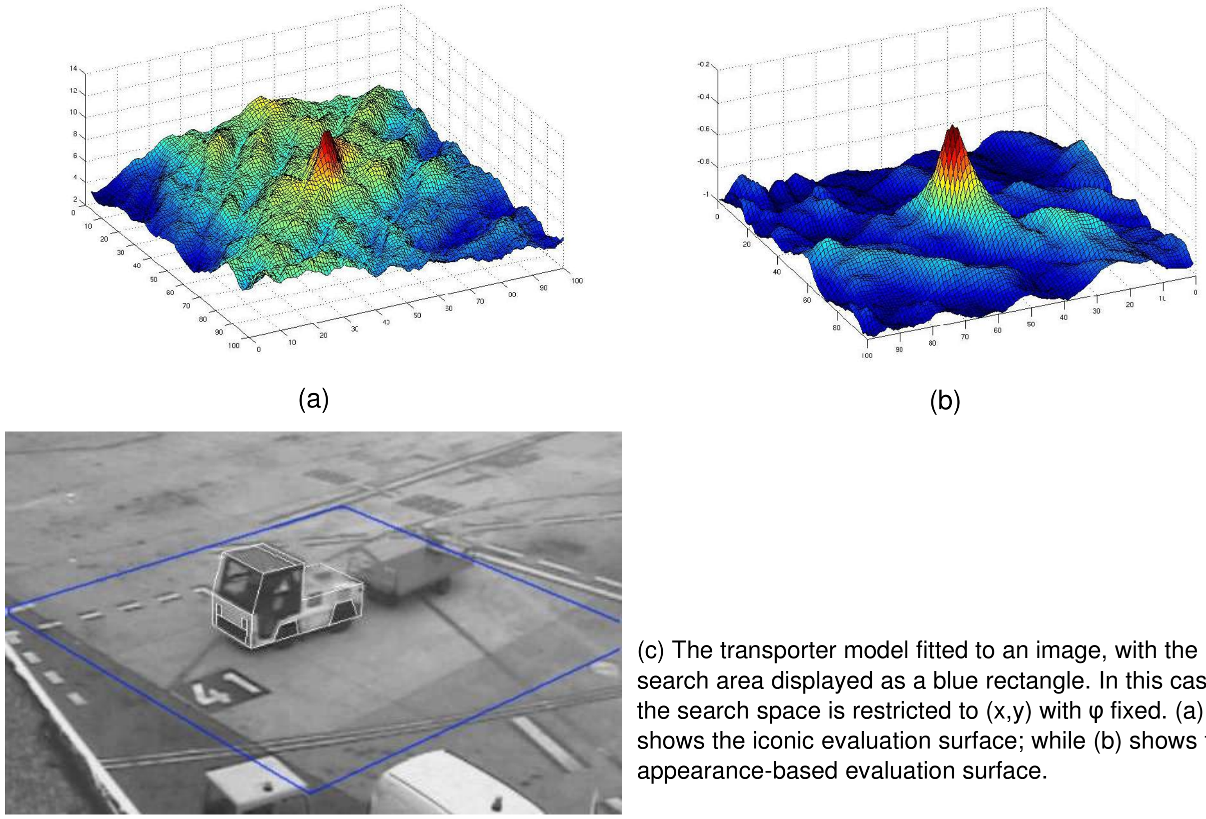
where, the weights are:

$$w_a=\left[\sin^{-1}\left(\frac{\bar{fn}\cdot\bar{ca}}{|\bar{fn}||\bar{ca}|}\right)\frac{2}{\pi}\right]^{11} \quad , \quad w_b=\frac{\sum_{\forall p} visible(p)=1}{\sum_{\forall p} 1} \qquad (3.4)$$

and *fn* is the facet normal, *ca* is the camera axis vector, and *p* is a point in the point model of the facet.

3.5. SEARCH ALGORITHMS

Using the iconic or appearance based evaluation measures mentioned in the previous section, finding the best pose (x,y,φ) of a model is equivalent to locating the global maximum (or minimum) in the evaluation surface. Performing an exhaustive search is costly, and in the case of AVITRACK, because of the large number of models, is also impractical. Therefore optimised search techniques such as the SIMPLEX algorithm are used. Figure 3.4 below shows an example:



(a)

(b)

(c)

(c) The transporter model fitted to an image, with the search area displayed as a blue rectangle. In this case, the search space is restricted to (x,y) with φ fixed. (a) shows the iconic evaluation surface; while (b) shows the appearance-based evaluation surface.

Figure 3.4: iconic & appearance based evaluation

As can be seen from Figure 3.4 above, the appearance-based evaluation surface, because more information content from the image is used, has a stronger peak and a smoother surface. Iconic evaluation also suffers from distraction by background edges. Another issue that affects both evaluation surfaces, is that apart from the global peak, there are also a lot of local maxima. Optimised search routines can easily get stuck in one of these local maxima, instead of finding the global one. Because of this, they are also very sensitive to the initial model pose (seed pose).

The field of optimised search algorithms (also known as the problem of maximisation/minimisation of functions, or simply, optimisation) is a heavily studied area of mathematics. For multi-dimensional functions, search algorithms can be grouped into 2 main types [80]: those that are gradient based and those that are not. Gradient-based search methods tend to be more powerful but at the expense of higher computational cost. In some cases, it might also not be possible to determine the gradient. The most widely used and successful search methods are listed below, and broadly classified by their type [80]:

Non-gradient based methods:	● SIMPLEX algorithm ● Powell's algorithm
Gradient-based methods:	● Fletcher-Reeves algorithm ● Polak-Ribiere algorithm ● Davidon-Fletcher-Powell algorithm ● Gauss-Newton method ● Levenberg-Marquardt method
	● Simulated Annealing method

In most of the previous work performed on the CVG's facet model system [76, 77, & 81], the SIMPLEX method was used because of its simplicity, computational efficiency, and was found to be fairly robust. In a later work, the use of Newton's method, a gradient-based approach, was also evaluated [82]. For AVITRACK, the same approach was taken, with the SIMPLEX algorithm being the main one to be evaluated. More detail about the algorithms used for AVITRACK is given in the following sections.

3.6. ESTIMATING THE SEED POSE

One issue that affects the behaviour of all search algorithms is how the initial position of the model, called the seed model pose, is estimated. This is especially important in the case where in addition to the global minimum, several local minima are also present in the evaluation surface.

In AVITRACK, categorisation is performed on objects that have been successfully tracked for a number of frames. Given such an object, its 2D bounding box in the image is back-projected on to the ground plane in the apron. A search area is then defined, aligned with the X and Y world axes that contains the projected bounding box. The initial model position is taken to be the centroid of this search area.

For the initial model orientation, this is estimated using the object's direction of motion as calculated by the tracker over a window of 5 frames (approx. half a second). The tracker calculates the direction of motion in 2D on the image plane, with an approximately exponential weighting over the time window:

$$\varphi=0.35\varphi_t+0.25\varphi_{t-1}+0.2\varphi_{t-2}+0.1\varphi_{t-3}+0.1\varphi_{t-4} \tag{3.5}$$

This is then back-projected on to the ground plane to get an estimate of the orientation angle of the model. Figure 3.5 below, illustrates the search area (in blue) and initial model position (shown as a wireframe model) for the loader object in S28-A320 Camera 5.

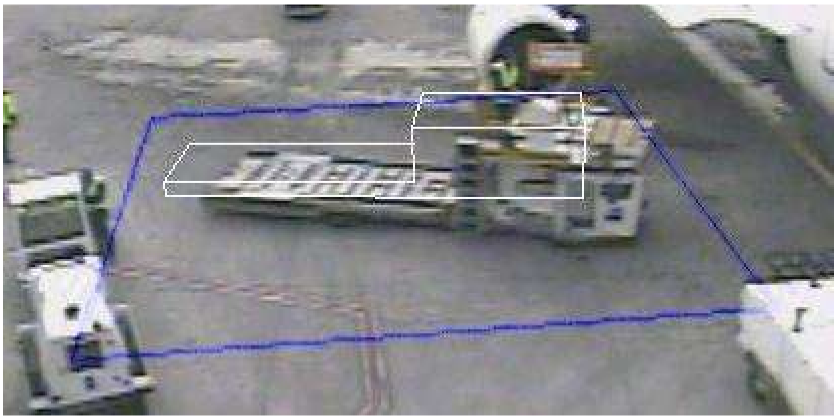
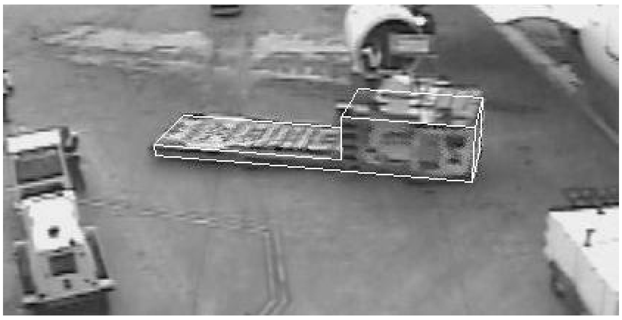


Figure 3.5: Estimating the search area and initial model pose

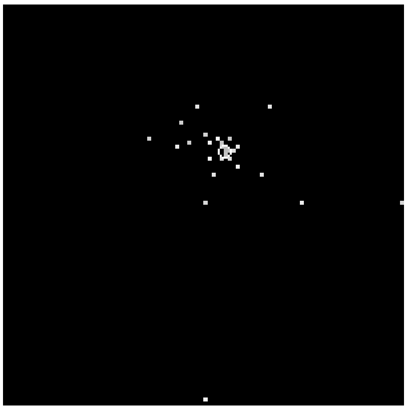
3.7. THE SIMPLEX ALGORITHM

As mentioned previously, the SIMPLEX algorithm is computationally efficient, requires only function evaluation (no calculation of gradient), simple in nature, and given a good initial model pose, quite robust. The SIMPLEX method, also called Downhill SIMPLEX, was first formulated by Nelder and Mead [80] in 1965, and is a geometrically-based method for function minimisation. The word *simplex* stands for a generalised triangle in N dimensions; e.g., for a 3D function, the simplex is a tetrahedron. The function to be minimised is evaluated at the vertices of the simplex, and depending on which vertex gives the best result, the simplex is modified by one or more operations to find better points. The operations that modify the simplex are: reflection about an edge, expansion or contraction. These better points are then used as the new vertices of the simplex, and the process is repeated until the convergence criterion is met. A more detailed description can be found in [80].

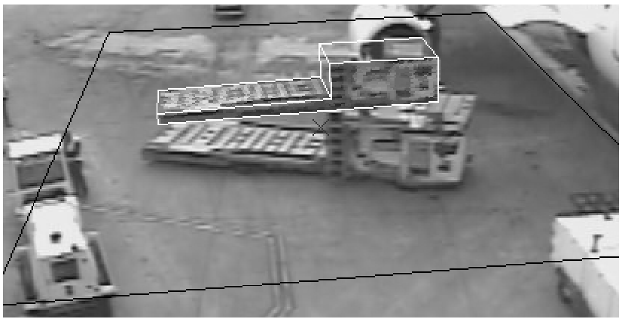
This algorithm was found to work well for AVITRACK if the estimate of the initial position is quite good and close to the true position. But if the initial position is not accurate, then the SIMPLEX algorithm has a tendency to get stuck in a local minimum. It was also found out that the search appears to be very sensitive to orientation – more local minima tend to occur in the X- φ or Y- φ parts of the 3D evaluation surface than in the XY-part. This might be related to the issue that as the model rotates, the facets change their orientation with respect to the camera, causing more variability in the evaluation surface. Figure 3.6 below illustrates some model fit results obtained by the SIMPLEX algorithm.



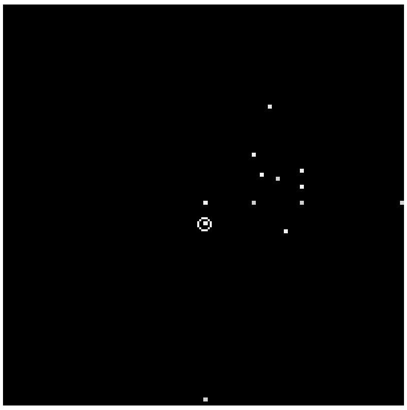
Model fit obtained by SIMPLEX for S28 C5 #449



The (X,Y,φ) search space projected on to the X-Y plane, showing the points evaluated by the SIMPLEX algorithm until convergence is reached.



Model not fitted correctly, when a different starting position is estimated.



Same X-Y projection as above showing the different convergence path taken by SIMPLEX for a different seed pose, causing a local minimum to be found.

Figure 3.6: Model Fitting by using the SIIMPLEX search algorithm

3.8. SIMULATED ANNEALING & SIMPLEX

As mentioned above, although the SIMPLEX algorithm has several advantages, it fares quite poorly in the presence of local minima. If the initial position is not accurate or not close to the global minimum, and SIMPLEX moves close to a local minimum, it will converge to the local minimum. In [80], a method combining SIMPLEX with Simulated Annealing is described.

Simulated Annealing (SA) is a technique used to find a global minimum in the presence of many local minima and when the search space is quite large [80]. The idea behind SA is similar to the physical process of annealing in metallurgy, where a material is heated and cooled slowly in a controlled way to increase the size of its crystals and reduce defects in them. This heating causes the atoms of the material to 'unstuck' themselves from their initial positions (a local minimum of the internal energy of the material) and wander randomly through states of higher energy. Then during the slow cooling, the atoms have a better chance of finding a configuration with lower internal energy than the original one.

In certain cases (e.g. when near the end of convergence), the random movements performed by SA to 'unstuck' itself from a local minimum can be inefficient if many of these random movements result in a worse value than the original position. By combining SIMPLEX with SA, it is claimed by [80] that this inefficiency is reduced (because SIMPLEX always moves downhill) while still maintaining the advantage of random movements.

For AVITRACK, the next steps in using the facet model for categorisation will involve combining Simulated Annealing with SIMPLEX and evaluating this search method.

3.9. CONSTRAINTS ON MODEL SEARCH

Apart from the ground plane constraint, mentioned in section 3.2, other constraints have been used to further reduce the size of the search space, as well as eliminate potential false positives caused by the SIMPLEX algorithm converging to a local minimum. For a tracked object O, and a 3D model M, these constraints are:

- Objects are categorised if their centroid lies within a region of interest on the ground-plane (world bounds).
- If the object's width (the width of the object's blob/connected component(s) in the image) when projected on to the ground plane, is less than half the minimum cross-section width of the model, then model fitting is not done.
- If the object's height is less than the half the height of the model, then model fitting is not done. (This and the previous constraint help to skip trying to fit a tanker to a person).
- If the ratio of motion (foreground) pixels of object O that are covered (explained) by the 3D model, to the total number of motion pixels of O, is below a certain threshold, then the fitted model hypothesis is not accepted. The threshold is defaulted to 0.3, i.e. a model fit is accepted if the model can account for a third or more of the object's foreground pixels.
- If the model's centroid, when projected on to the image plane, is outside the object's blob, then the model fit is not accepted.

In future work, it is envisioned that other constraints, such as context-based constraints will also be used. These will consist of static scene context (e.g. the jet bridge can only be in a limited set of positions on the apron) as well as dynamic context (e.g. in the case of articulated vehicles).
Annealing with SIMPLEX and evaluating this search method.

3.10. USING 3D MODEL-BASED CATEGORISATION WITHIN A REAL-TIME SYSTEM

For AVITRACK, the 3D model-based categorisation method has to be integrated within a real-time system running at 12.5 frames per second. Categorisation is performed independently for each of the 8 cameras in the Frame-to-frame tracking module [83]. A total of 22 different vehicle types need to be categorised. This number may be higher in practice: one or more of these vehicle classes, e.g. fuel tankers, exhibit major differences within the class, requiring different 3D models.

On average it was found that the facet model classifier takes between 0.2 to 0.5 second to classify one model. Therefore it is not possible to run the classifier in real-time on each and every frame of the video sequence. But at least, for the Scene Understanding module, the category of the object only becomes important when it enters its zone of operations near the aircraft. And so it is acceptable for the classifier to start working on classifying objects in a video frame and report the results a number of frames later.

The model-based classifier runs in a background thread and communicates with the frame-to-frame tracker via a processing queue. At the end of each tracking loop, the tracker looks for objects which have not been classified and are not already on the processing queue, and places these on the queue. If there are objects on the queue which have been marked as 'classified', the tracker will remove them from the queue and set the object's type to be that of the returned results. The classifier will in turn take the first item from the queue, classify it and place the result back on the queue and mark the object as 'classified'.

Two things can happen. The queue can become full, in which case the tracker stops putting objects on the queue, resulting in some periods in which no object is classified. An object that is on the processing queue waiting to be classified, dies in a later frame (not tracked any longer); the tracker takes care of cleaning these from the queue to make space for newer objects.

Through this mechanism, a compromise is reached between maintaining real-time tracking and performing categorisation when possible. Section 8 describes how a bottom-up classifier was integrated with the model-based classifier to further improve computational cost of model-based classification.

4. BOTTOM-UP CLASSIFIER BASED RECOGNITION

Bottom-up classification based recognition can be achieved in both supervised and un-supervised manner. The supervised approaches require an operator to manually label training data such that these exemplars can be used to train a classifier. It is common to use the labelled training data to evaluate the performance of the classifier and hence improve the classifier formed from such data. Unsupervised methods (e.g. clustering) have the advantage that no operator training is required and therefore such algorithms could be considered automatic. The weakness of unsupervised methods is that they generally require an expensive optimisation period at initialisation and that the detected objects make have subtle differences not well fitted by the parametric form of the unsupervised model. In the AVITRACK project we have subsequently only evaluated supervised methods, it is common sense to attempt supervised classification first before trying unsupervised classification. All classification methods take input in the form of *feature vectors;* these are conversions of the initial image (commonly RGB) into linear and non-linear measurements that improve the categorisation between object classes --- this feature space is dependent on the application. In this section we detail the methods applied to the problem of people and vehicles categorisation.

In Section 5, the possible representations for the visual objects within AVITRACK are examined: both statistical representations and appearance based object representations. This is followed by Section 6 describing the use of these representations and how objects can be classified via distance metrics, neural networks, Eigen decomposition, and sets of images. Implementation issues specific to the AVITRACK project are discussed in these sections, and results of these various bottom-up classification methods are given in section 7. (The work presented in Sections 5 to 7 is taken from [2],[85]).

5. OBJECT REPRESENTATIONS FOR BOTTOM-UP CLASSIFICATION

5.1. STATISTICAL DESCRIPTORS

Statistical object representations describe an object or class of objects using statistical reasoning. This allows for common class features to be extracted and for collection and representation of class defining measurements. Although many statistical methods exist for describing visual objects, this section separates the descriptors into two categories, non-invariant and invariant descriptors. The majority of the work in this section is focused on the latter. Non-invariant descriptors are covered here due to their readily extractable nature, and as they provide a clue as to the effectiveness of the invariant methods. In the early stages of research, this provides an insight into the type of invariant descriptors that may work well, and shows basic class separation, inter-class clustering and intra-class distances.

5.1.1. NON INVARIANT DESCRIPTORS

As a first step towards using statistical information to discern object class ownership, a small subset (ca. 5) visual objects of the entire set to be classified was considered and the following information extracted for each class, based on its appearance (pixel values):

1. Mean
2. Variance
3. Standard deviation
4. Skew
5. Kurtosis

From these values, an average value per class can be extrapolated. Standard Euclidean distance can then be employed to perform the matching and classification.

5.1.1.1 Height and Width in the Image Plane

Beyond the very basic statistical measurements described in Section 5.1, the measurement of the height and width of objects in the image plane provides meaningful information, which is less affected by lighting conditions and more accurately represents class types. Unfortunately, as the height and width are measured in the image plane, affine distortion is more prevalent. This distortion, due to perspective, makes the height and width measurements more prone to inaccuracy. This technique is included because it allows a simple postulation to be made as to the likely degree of grouping of object classes, and therefore the prospective usefulness of more invariant statistical descriptors, such as those detailed in Section 5.1.2.

In order to ascertain the degree of grouping this type of size or shape based technique may hold, the height and width in the image plane can be plotted as a graph and the clustering observed. From this, a set of group centroids are created, which represent an idealised height to width ratio for a set object type. Centroids are used to reduce the semi-relevant information content of the system, so that only one point is required for matching. This point retains most of the information content of the original clustering, as it is the mean point of the group. This technique also allows for the easy creation of decision boundaries. The clustering information is not completely disregarded, however, as some distance measures use this information to realise better classification results. The Mahalanobis distance measure is a good example of this (see Section 6.1.2).

5.1.2. INVARIANT DESCRIPTORS

Owing to inaccuracies in ascertaining height and width measurements in the image plane, invariant descriptors are now introduced. These descriptors allow measurements to be made that are more capable of dealing with the effects of affine deformations and perspective distortion.

5.1.2.1 3D Height / Width

Building upon the height and width measurements taken in the image plane, height and width measurements are now taken incorporating known camera geometry. This method allows known camera positions relative to object positions in the scene to compute very accurate object height and width measurements. These measurements are invariant to perspective or affine distortion, but not to occlusion. It is important to note, that although the measurements themselves are invariant to 3D rotation within the scene, providing accurate measurements of the object irrespective of their position, this rotation will alter the size of the object under observation. For example, a car viewed side on will be a different width to a car viewed head on. This change can be compensated for within the model created, as the centroid positions will consider these variations. However, this may reduce the accuracy of the final classification, as some classes may overlap. This overlap may mar the decision boundaries.

5.1.2.2. Moments

Region moment representations interpret a normalised gray-level image function as a probability density of a 2D random variable. Properties of this random variable can be described using statistical characteristics - moments [21]. Moments allow an object to be described using standard statistics, for example, the objects area (m_{00}). The use of several statistical descriptions of an object allows a specific object's type (classification) to be postulated. This system has been used to great effect in object and letter classification by Flusser and Suk [23].

In digitised images, moments are created such that:

$$m_{pq}=\sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty} i^p j^q f(i,j) \qquad (5.1)$$

Where i, j are the pixel co-ordinates and p,q describe the moments to create (for example, to create m_{00} , p and q = 0).

It is advantageous for moments to be as invariant as possible as in Section 5.1.2.1., where the heights to width measurements are made invariant to perspective projection. Invariance in moments allows for a more robust and flexible object representation.

Translation invariance (i.e. moments that appear the same at any position within the image) can be achieved if image central moments are used [22], through the generalised formula:

$$\mu_{pq}=\sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty} (i-x_c)^p (j-y_c)^q f(i,j) \qquad (5.2)$$

Where x_c, y_c are the co-ordinates of the regions centre of gravity (centroid), such that:

$$x_c=\frac{m_{10}}{m_{00}} \qquad y_c=\frac{m_{01}}{m_{00}} \qquad (5.3)$$

Further invariance to scale can be obtained using scaled central moments and normalised un-scaled central moments; the later will not be discussed here, however, the former is used within the Hu Invariant Set.

Scaled central moments can be derived through:

$$\eta_{pq}=\frac{\mu'_{pq}}{(\mu'_{00})^{\gamma}} \qquad (5.4)$$

Where,

$$\gamma=\frac{p+q}{2}+1 \qquad (5.5)$$

And,

$$\mu'_{pq} = \frac{\mu_{pq}}{\alpha(p+q+2)} \qquad (5.6)$$

Hu Invariant Set

The Hu invariant set comprises seven invariant measurements, based on scaled central moments, which describe an object, and can be easily built into a standard distance classifier system through treating each measurement as a dimension in a seven dimensional space. The Hu set aims to further reduce the variance inherent within statistical object measurements and thus create a more robust measurement set.

The measurements are created in the following way:

$$\begin{aligned}
&I_1=\eta_{20}+\eta_{02}\\
&I_2=(\eta_{20}+\eta_{02})^2+4\eta_{11}{}^2\\
&I_3=(\eta_{30}-3\eta_{12})^2+(3\eta_{21}-\eta_{03})^2\\
&I_4=(\eta_{30}+\eta_{12})^2+(\eta_{21}+\eta_{03})^2\\
&I_5=(\eta_{30}-3\eta_{12})(\eta_{30}+\eta_{12})[(\eta_{30}+\eta_{12})^2-3(\eta_{21}+\eta_{03})^2]+\\
&(3\eta_{21}-\eta_{03})(\eta_{21}+\eta_{03})[3(\eta_{30}+\eta_{12})^2-(\eta_{21}+\eta_{03})^2]\\
&I_6=(\eta_{20}-\eta_{02})[(\eta_{30}+\eta_{12})^2-(\eta_{21}+\eta_{03})^2+4\eta_{11}(\eta_{30}+\eta_{12})(\eta_{21}+\eta_{03})]\\
&I_7=(3\eta_{21}-\eta_{03})(\eta_{30}+\eta_{12}[(\eta_{30}+\eta_{12})^2-3(\eta_{21}+\eta_{03})^2]+(\eta_{30}-3\eta_{12})(\eta_{21}+\eta_{03})[3(\eta_{30}+\eta_{12})^2\\
&-(\eta_{21}+\eta_{03})^2]
\end{aligned} \qquad (5.7)$$

This set can be used for scale, position and rotation invariant pattern identification [24], and is therefore a good candidate for use within a far field classification context. The generalisation capabilities available with this type of set become apparent when the individual descriptors of the set are analysed. For example, I_7 is a skew invariant descriptor, which can help to distinguish mirror images. Therefore, using this measurement, a camera on the left of an object within a scene (i.e. having a view of the objects right hand side), will return the same object type as a camera positioned on the right of an object in the scene (i.e. having a view of the objects left hand side).

The Hu set, as detailed above does not contain a full set of affine invariant moment measurements, and is therefore still susceptible to some affine distortion. This problem has been rectified by other researchers, for example Flusser and Suk [23].

Affine Invariant Set

Balakrishnama and Ganapathiraju, and Ullman, Sali and Naquet [25 & 26] have all created algorithms for fast computation of translation, rotation and scale invariant moments, however, Flusser and Suk [23] provide descriptors, which are invariant under all affine transformations.

The affine invariant set proposed by Flusser and Suk [23] proffers a set of descriptors derived from second and third order central moments. These descriptors are created by:

$$
\begin{aligned}
I_1 &= \frac{\mu_{20}\mu_{02}-\mu_{11}{}^{2}}{\mu_{00}{}^{4}} \\
I_2 &= \frac{\mu_{30}^{2}\mu_{03}^{2}-6\mu_{30}\mu_{21}\mu_{12}\mu_{03}+4\mu_{30}\mu_{12}^{3}+4\mu_{21}^{3}\mu_{03}-3\mu_{21}^{2}\mu_{12}^{2}}{\mu_{00}^{10}} \\
I_3 &= \frac{\mu_{20}(\mu_{21}\mu_{03}-\mu_{12}^{2})-\mu_{11}(\mu_{30}\mu_{03}-\mu_{21}\mu_{12})+\mu_{02}(\mu_{30}\mu_{12}-\mu_{21}^{2})}{\mu_{00}^{7}} \\
I_4 &= (\mu_{20}^{3}\mu_{03}^{2}-6\mu_{20}^{2}\mu_{11}\mu_{12}\mu_{03}-6\mu_{20}^{2}\mu_{02}\mu_{21}\mu_{03}+9\mu_{20}^{2}\mu_{02}\mu_{12}^{2}+12\mu_{20}\mu_{11}^{2}\mu_{21}\mu_{03}+6\mu_{20}\mu_{11}\mu_{02}\mu_{30} \\
&\quad \mu_{03}-18\mu_{20}\mu_{11}\mu_{02}\mu_{21}\mu_{12}-8\mu_{11}^{3}\mu_{30}\mu_{03}-6\mu_{20}\mu_{20}^{2}\mu_{30}\mu_{12}+9\mu_{20}\mu_{02}^{2}\mu_{21}^{2}+12\mu_{11}^{2}\mu_{02}\mu_{30}\mu_{12}-6\mu_{11} \\
&\quad \mu_{02}^{2}\mu_{30}\mu_{21}+\mu_{02}^{3}\mu_{30}^{2})/\mu_{00}^{11}
\end{aligned}
\qquad (5.8)
$$

5.2. APPEARANCE BASED METHODS

The methods discussed in Section 5 are fully concerned with the statistical representation of object classes. Although this is a well-established and mature way of representing visual objects, without considering the appearance of an object, many visual clues as to an objects class may be overlooked.

The main drawback to the use of appearance-based representations stems from the additional difficulties inherent during the normalisation process. For example, lighting conditions and weather (in outdoor classification tasks), may play a large role in the classification process as image appearance may change dependant on these. There are normalisation techniques, which deal to certain degrees with these problems, though no perfect system exists.

Within this section, numerous non-invariant and invariant image representations are discussed. These representations overcome some of the problems of variability within appearance-based techniques and therefore simplify the matching process.

Importantly, the use of appearance-based classification techniques can more easily deal with the problems of occlusion and shadowing, as sub-sections of an image, rather than the entire image can be used for classification. As occlusion is prevalent within many classification tasks, this is a definite advantage. The use of sub-sections for classification purposes are further validated in Section 5.2.1.

5.2.1. RATIONALE FOR THE USE OF IMAGE SUB-SECTIONS

Before more advanced object recognition or classification methods can be investigated, it is important to establish exactly what information is to be matched. This decision is as important, if not more so, than the choice of classification techniques. Classifiers are simply techniques, which at their core match one set of data with another, based on some criterion. This is of course is meaningless if the data to be matched does not represent the class to be classified in a robust way.

There are many good reasons for the use of image sections, as opposed to the entire image, within the object classification domain, including:

1. Whole images often contain background information not conducive to image classification. This adds noise to the classification problem and exacerbates an already complex task.

2. Whole objects may not be easily recognisable as a member of their object type (from a computer vision standpoint). Indeed, in many cases, an objects overall appearance may vary drastically, although certain sections remain the same. As an example of this, take motorcycles; the body of a motorcycle often varies wildly from manufacturer to manufacturer, however, certain parts (e.g. wheels, handlebars) remain the same or similar and can therefore be tracked.
3. In real world environments, objects are often partially occluded. This means that classification based on whole image views can fail due to other objects within the scene. The occluding objects may distort the object to be classified in very unpredictable ways. Patch based systems have greater tolerance to this problem as sections of the object are often available for classification in partially occluded circumstances.

If any patch (image sub-section) were not centred on, for example, the aeroplane in Figure 5.1 the outcome would obviously be quite erratic; however, even patches, which are centred on the aeroplane, may not hold useful information. Take for example, the patch that captures the intake of the right-hand engine. This is simply a pure black patch, holding little usable information.

To rectify this problem, image subsections should be chosen based on a set of criterion. These criteria may be based on capturing image structure, important colour or gray-level changes, or upon repeatability, i.e. how often a patch is associated with an object class and how reliably these matching points can be found.

The main drawback to using object sub sections for this type of task is that there is a computational cost in finding and using them, as well as normalisation if using an invariant image patch (see Section 5.2.4). As this cost is not prohibitively high, and the benefits of increased classification accuracy far outweigh the computational costs, these techniques are further discussed in detail.

5.2.2. SUB-SECTION VARIATION

As can be seen from comparing Figures 5.1 and 5.2, the advantages of using discriminate patches for object representation are quite clear. Discriminate patches allow important points in the image (in this case corner points) to be isolated, and the role played by background pixels is reduced. These points are more likely to contain usable appearance information than non-discriminant patches.

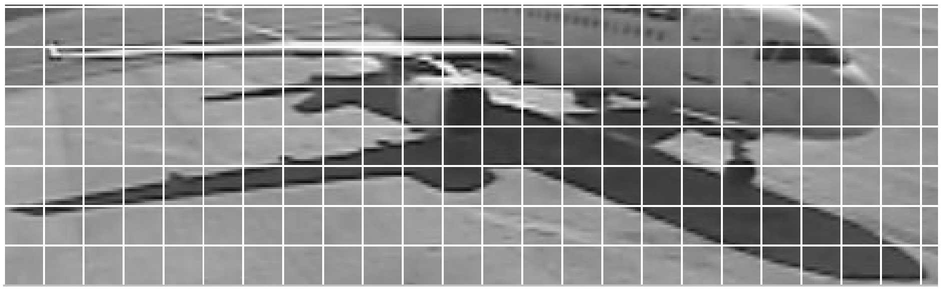


Figure 5.1– Example of Non-Discriminate Images Patches

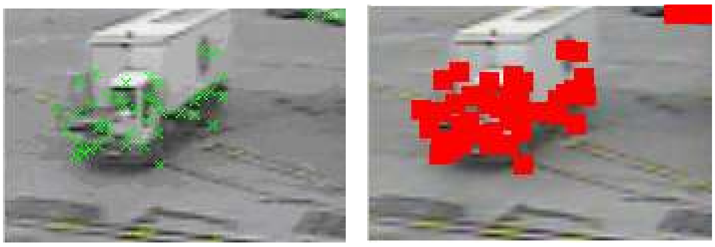


Figure 5.2– Example of Discriminate Image Patches (right) Found
From Harris Corner Points (Section 5.2.4) (left)

It is in fact possible, using neural network classifiers, to apply the Harris corner point image patches in a reasonably invariant manner as they are. This invariance can be trained into the neural network weights file by resizing and rotating images, then presenting them to the network as correct samples (see Section 6.3). Other techniques for creating invariant image segments without neural network training are given in the following (see Section 5.2.5).

5.2.3. POINT OF INTEREST OPERATORS

Point of interest operators (POI) allow important or interesting features, often sections of the image used in tracking (such as in optical flow) or stereo image matching and classification, to be extracted from the main image. These sections are usually more robustly locatable between two or more images of the same object and can act as reference points from which more complex tasks, such as wide baseline matching or, in this case, object classification, may be based. In [73], Shi and Tomasi discuss feature tracking. They conclude that tracking features provide a good means of tracking overall objects. They also state that “tracking” points must be stable over linear affine deformations. Though in their paper they use a “dissimilarity” measure to ascertain the quality of the tracked point, here we will use scale and affine invariant points to effect the same ends.

5.2.4. HARRIS OPERATOR

The Harris-Stephens operator [74] (often simply called the Harris operator) locates informative image structure (corner type areas), within an image. This operator has been shown to be robust, and can return highly repeatable image sections [33]. The idea of repeatability is an important one, as only segments, which appear in numerous images of an object class, should be used in the matching process.

The Harris operator functions by analysing the auto-correlation or second moment matrix of a set region, in the following way:
Representing the spatial image gradient as $[E_x, E_y]^T$, where E_x is created through partial differentiation, such that $E_x = \frac{\partial E}{\partial x}$. A matrix C, which is the auto-correlation matrix of the

region Q, can be formed using the formula:

$$C = \begin{bmatrix} \sum E_x^2 & \sum E_x E_y \\ \sum E_x E_y & \sum E_y^2 \end{bmatrix} \tag{5.9}$$

Q may be of any size, but is often between 5 and 21 pixels per side [32].

To find areas of interest using the Harris operator, the Eigenvalues of the matrix C are then found. Once obtained, these values can be used to locate corner points. If a corner is present, the lowest eigenvalue should be above a pre-defined threshold. In this situation, the eigenvalues represent the strength of the corner detected, whilst the eigenvectors describe the direction of the corner found. As the direction of the corners found are not usually of interest, only the eigenvalues need be monitored.

It is important that we here introduce the idea of using multi scale Gaussians in conjunction with the second moment matrix used in corner point detection, as this idea will be used in later sections when scale and affine invariance is added to the basic detector. The use of multi scale Gaussians in this way is an important idea, as this allows for the extraction of different size structure from an image, without spurious readings. This is achieved by blurring images using the Gaussian blur, while increasing the size of the corner region (Q) under scrutiny. The blurring effectively removes the smaller corner features, relative to the corner region size, thereby suppressing smaller corners in larger regions and extracting only the larger features. In smaller windows, the Gaussian blur is reduced and smaller corner features are available for extraction.

In this case, two scales, the derivation scale σ_d and the integration scale σ_i, define the second moment matrix. In the case of an infinite Gaussian window centred on point X, [33] shows the formula for computing the second moment matrix is given by:

$$M(x,\sigma_i,\sigma_d)=\sigma_d^2 G(\sigma_i)*\begin{bmatrix} I_x^2(x,\sigma_d), I_x I_y(x,\sigma_d) \\ I_x I_y(x,\sigma_d), I_y^2(x,\sigma_d) \end{bmatrix} \qquad (5.10)$$

Where M represents the second moment matrix, G is a Gaussian, which determines the effective window size, and I in its various forms represents a smoothed image gradient found by:

$$I_x(x,\sigma_d)=\frac{\partial}{\partial x}G(\sigma_d)*I(x) \qquad (5.11)$$

Which is the first derivative of the Gaussian kernel. The measure at each corner point can then be found by:

$$F(x)=\det(M)-ht^2(M) \qquad (5.12)$$

Where h is a value, for example 0.04 proposed by Harris, and t represents the matrix trace.

5.2.5. AFFINE INVARIANT DESCRIPTORS

In order for moveable objects to be reliably identified in images under the effect of perspective projection, irrespective of the matching technique used, it is important that the image to be matched, and the template image, be normalised in some way to be invariant to the affine distortions levied upon them. This requires that some sort of affine invariant image representation be found for both the matching template and the image under scrutiny.

Some detection methods have intrinsic invariance to limited affine manipulations. For example, the Harris point detector has 2D rotation invariance, as a corner, once rotated, remains a corner; however, this method is not invariant to scale (in its basic form, using Equations 5.9 and 5.12) or 3D rotation.

If a representation can be used, which is invariant to all affine transformations, then matching may be performed on objects within the image, which would otherwise fail due to its rotation or distance from the camera (i.e. scale). This allows for extra freedom and an enhancement in the classification accuracy.

5.2.5.1 Scale Invariant Harris Points

In this section, the notion of Harris points will be extended to locate scale invariant points.

To do this, we first create interest points in Gaussian scale space. The scale space is obtained by scaling the derivation scale (i.e. altering the Gaussian size), over a range of scales. The convolved image at scale s is therefore:

$$I(x, s\sigma_d) = I(x) * G(x, s\sigma_d) \qquad (5.13)$$

To find the second moment matrix in Gaussian scale space, $s\sigma_d$ need only be substituted into Equation 5.10 to form Equation 5.14:

$$M(x, s\sigma_i, s\sigma_d) = (s\sigma_d)^2 G(s\sigma_i) * \begin{bmatrix} I_x^2(x, s\sigma_d), I_x I_y(x, s\sigma_d) \\ I_x I_y(x, s\sigma_d), I_y^2(x, s\sigma_d) \end{bmatrix} \qquad (5.14)$$

Once the corner points have been located in scale space, an analysis can be undertaken to find the subset of points for which a local measure is at a maximum over scale. This scale, determined for each point, is known as the characteristic scale. The idea of characteristic scale has been extensively studied by Lindeberg [34].

It is most common for a search to be conducted in the local 3D scale space of an object (x,y,s), to find the maxima of a point. If the point has a value above a set threshold, and is the local maximum of its area in the 3D scale space, then this point is considered a feature point.

Lindeburg [34] proposed searching for extrema of a Laplacian, as well as the magnitude of the gradient, though others have proposed the use of difference of Gaussians. In [35], this idea was developed so that both Harris points and Laplacians are used. This allows the Harris operator to locate interest points in the 2D space, while the Laplacian is used to evaluate Harris points in scale space. The end result of this process being that Harris points are returned, along with their characteristic scales, creating scale invariant Harris points.

The normalised Laplacian used by [35] takes the form:

$$F(x, s_n) = |s_n^2 (I_{xx}(x, s_n) + I_{yy}(x, s_n))| \qquad (5.15)$$

Where I_{xx} and I_{yy} represent the signal measured by the convolution of the image with the Laplacian of Gaussian (LoG) kernel. This kernel takes the form:

$$h(x, y) = \frac{1}{2\pi\sigma^2}\left(\frac{x^2+y^2}{\sigma^4} - \frac{2}{\sigma^2}\right)\exp^{\left(-\frac{x^2+y^2}{2\sigma^2}\right)} \qquad (5.16)$$

Figure 5.3 shows an example of the scale invariance possible when the Harris measure is properly adapted as discussed.

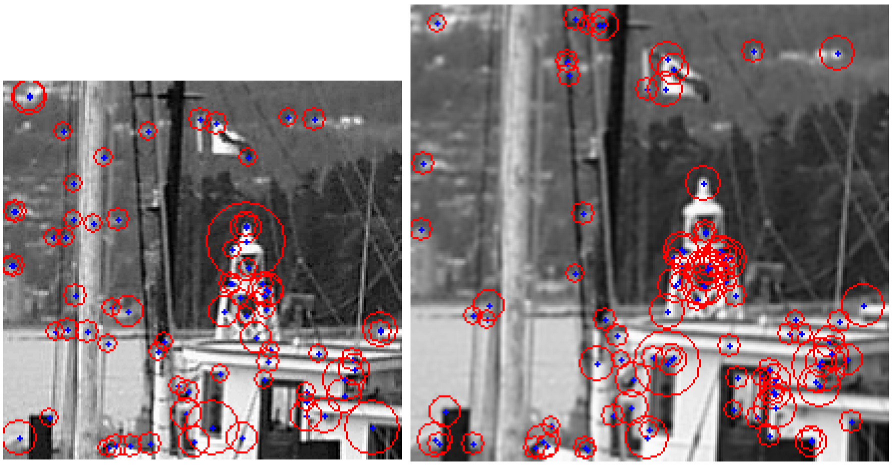


Figure 5.3 – Scale Invariant Harris Points in Two Images With a Scale Difference of 1.2%

5.2.5.2. Affine Invariant Harris Points

The adaptation taking the scale invariant Harris points to affine invariance is non-trivial, however, the ideas continue naturally from previous discussions of Harris measures and characteristic scale. The adaptation proceeds in an iterative cycle, and can be broken into eight steps:

1. Project the image section under scrutiny into a patch centred co-ordinate space using a transformation matrix “U”.
2. Find the characteristic size for the patch (as previously discussed)
3. Find a Gaussian sigma to maximise the ratio of Eigenvalues for this space (from the set [0.5..0.75])
4. Relocate the central point of the transformed space (as it may have moved due to the Gaussian blurring)
5. Find the transformed space second moment matrix (as described in Equation 5.9)
6. Update the “U” matrix with the inverse square root of the second moment matrix from step five, then multiply by the “U” matrix from the last iteration
7. Normalise the “U” matrix so that the maximum eigen value of “U” equals one
8. If the second moment matrix is above a threshold stop, otherwise, go to step one

5.2.6. SALIENT IMAGE SECTIONS

Image saliency, the degree of unpredictability within an image, is a long-standing technique used to locate interesting areas of an image. Researchers experimenting with saliency found that salient regions often contain complex structures, due to their high levels of entropy, which is used to assess saliency. As classification is most interested in locating areas of unique interest, this system has been widely used for locating points of interest within image matching tasks [41].

The idea of entropy, pioneered by Shannon [72], and central to the idea of ascertaining saliency, is found by:

$$H(X) = -\sum_{x\in X} p(x)\log_b p(x) \qquad (5.17)$$

Where p represents the probability of x. In this case, x is considered a discrete random variable of a finite set, i.e. pixel values. "b", represents the base used, in this case equal to 2. The probability of "x" is computed by finding the histogram (probability distribution) of a patch under scrutiny. For each patch, the probability of a certain pixel value will change. Saliency is then found using this probability and the Shannon entropy measure.

Kadir and Brady also introduce the idea of saliency over scale, and use this to find the equivalent of the characteristic scale, used by Schmid and Milkolajczyk [33,35 & 40] for a salient image. A "characteristic" scale is found for a salient region when the magnitude change of the probability distribution over scale reaches a maximum (see Section 5.2.5. for more information on characteristic scale).

5.2.6.1 Affine Invariant Salient Points

Unlike the affine invariant Harris point adaptations, the saliency system's adaptation to affine invariance is very straightforward. The scale invariant salient points are simply transformed by replacing the circles (representing the scale resolved salient points from Section 5.2.6) with ellipses. The scale value is replaced with a vector $s'=(s,p,\theta)$, where p represents the axis ratio, and θ represents the orientation of the ellipse. The major and minor axis of the ellipses can be found by $s/\sqrt{p}$ and $s\sqrt{p}$ respectively.

5.2.7. MAXIMALLY STABLE EXTREMAL REGIONS

This section represents a departure from the more standard types of affine invariant locator, as described in the last sections. Maximally Stable Extremal Regions (MSER) locates detectable regions by an exhaustive thresholding and examination process. Informally, the process can be described through the analogy of watching a film. If it were assumed that all pixels below a set threshold are coloured black, and all above white, then starting from a very low threshold and incrementally increasing the threshold, one would start with a white screen initially, with black regions forming and merging as the threshold was altered. Finally, a black screen would appear. By keeping a record of all the connected black regions over the lifetime of the process, one could ascertain which regions existed for the longest period of threshold values. These regions could be termed Maximally Stable Extremal Regions. This process has a computational cost per image of $O(n\log\log n)$, where n represents the number of pixels within the image.

Once MSER regions have been detected, the areas (which can be of arbitrary shape and size) are represented in an affine invariant manner. This is achieved using complex moment combinations, which acting together can define a region in an invariant way. This is exactly the same process as detailed in Section 5.1.2.2 (Equation 5.8).

5.2.8. APPEARANCE BASED NORMALISATION

In order to store and later match appearance based patches, it is important to normalise the images in a set manner. This allows for detectors, which detect regions of an arbitrary size to store and use images as if they were of fixed size. For example, if MSER is used, an unknown patch size may be returned; this must be stored and matched in a fixed size patch (for efficiency and ease). Furthermore, if a Harris based patch, which returns a fixed size patch but can be at any rotation, is used, some system for normalising rotation and scale must be found. Two such methods for normalisation, which are closely linked, are the Hotelling transform and the Whitening transform.

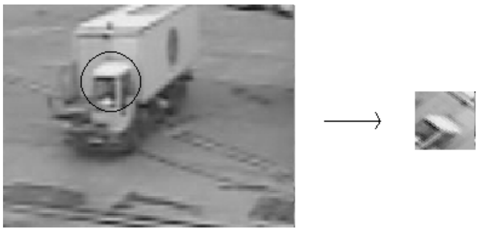


Figure 5.4 – Whitening Transform Used on AVITRACK Data. Circular Covariance Has Been Rotated 45 Degrees to the Right and Scaled, Thus Normalising it.

5.2.8.1 The Hotelling Transform

This transform, also known as Principle Component Analysis, can be used to take a matrix of data to be transformed A and select a standard orientation for it. This means that all images of the same covariance will be orientated in the same manner.

To perform this transform, the covariance (formed as a circle in Figure 5.4, but normally an ellipse) undergoes Eigenvector and Eigenvalue decomposition, to form a ranked Eigenvector and ranked Eigenvalue matrix. The height and width of the transformed image space is now equal to $\sqrt{\lceil E_{Value} \rceil \sigma}$, where E_{Value} represents the largest Eigenvalue and σ represents a sigma capturing the desired number of standard deviations, for example, 3 or 4. To transform the space:

$$Y = AE_{Vector} \tag{5.18}$$

In this equation, Y is the resultant image transform that takes the original image to the correct orientation based on the Eigenvectors, A is a matrix formed by taking the data to be transformed and subtracting the mean of the data set from each value. More formally, $A=(x-Mx)$, where x is a data point within the set and Mx is the mean of the data set.

Due to the effect of multiplying the A matrix with the Eigenvectors, the transform will now send the data to a patch of unit size. To scale this new Y matrix to a set size, a scalar representing the required size must now be post multiplied with the Y matrix values. Once this is done, a patch normalised for rotation and scale is available.

5.2.8.2. The Whitening Transform

The Whitening transform functions in much the same way as the Hotelling transform, however, it incorporates the scaling step within the transform. This is therefore a simpler and more compact transformation. To perform the Whitening transform, the Eigenvalues should be pre-scaled so that:

$$E_{Vector}' = \frac{E_{Vector}\, r}{\sigma^2 \sqrt{E_{Value}}} \tag{5.19}$$

Where r represents the required size of the end patch. Once scaled, the transform may continue as described in Section 5.2.8.1 substituting the newly created E_{Vector}' for E_{Vector} in Equation 5.18.

5.2.9. SCALE INVARIANT FEATURE TRANSFORM (SIFT)

The Hotelling and Whitening transform, as described in Sections 5.2.8.1 and 5.2.8.2, normalise an image taken from a scale- or affine-invariant interest point detector based on pixel values (visual data) and thus facilitates visual based correlation matching. An alternative to this is matching based on measurements from an invariant object representation such as the Scale Invariant Feature Transform (SIFT), which stores an image as a directional histogram of sorts, and matches through Euclidean distance to known objects in Euclidean space.

Once a scale- or affine-invariant feature has been resolved using a scale- or affine-invariant interest operator, a SIFT vector may be used to create a highly distinctive, illumination, perturbation, 3D viewpoint and non-rigid deformation resilient object representation [44].

The SIFT operator is created by examining the resolved image patches from the invariant interest operator and assigning each pixel within it a magnitude and orientation, based on a set formula. To assess the magnitude of a pixel at location x,y:

$$m(x,y)=\sqrt{(L(x+1,y)-L(x-1,y))^2+(L(x,y+1)-L(x,y-1))^2} \quad (5.20)$$

To assess the orientation at a set pixel location:

$$\theta\,(x,y)=\tan^{-1}\frac{((L(x,y+1)-L(x,y-1))}{(L(x+1,y)-L(x-1,y))} \quad (5.21)$$

In the preceding formulas (5.20 & 5.21), “L” represents the image region returned by the invariant interest operator convolved with a Gaussian with a sigma value of 1.5 times the region size.

Once this is done, the data will notionally resemble that shown in Figure 5.5 below.

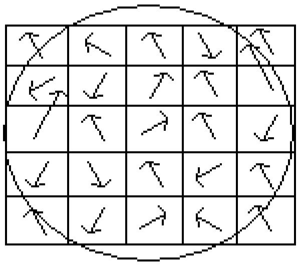


Figure 5.5 – Notional View of Data, “L” in Equation 5.20 & 5.21, After Assignment of Magnitude and Direction per Pixel. The Arrows Used Represent The Magnitude as The Length of The Arrow and The Direction of The Arrow Shows The Notional Direction of The Pixel. The Circle Represents The Gaussian Used (Although This Would Fall Off Smoothly in Reality).

An orientation histogram is formed with 36 bins, covering the full 360-degree range. Once this is done, the pixels are added to this histogram, weighted by their gradient magnitude and the Gaussian imposed upon “L”. The magnitude maxima is located, then pixels with magnitudes within 80% of this maximum are found. These maxima correspond to dominant directions in the local gradients. To increase accuracy, a parabola is fitted to the three histogram values closest to each maxima, or peak, value. This is used to interpolate the peak positions. Once done, a notional view of the descriptor formed is shown in Figure 5.6.

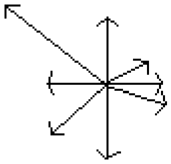


Figure 5.6 – Notional View of Data as a SIFT Vector, Formed Using A Directional Histogram

In the paper presented by Lowe [44], the invariant image patches were sub divided into subsections, and descriptors, such as in Figure 5.6, were made for each subsection. These were then combined into one descriptor of 128 dimensions. The descriptor was then normalised to unit length to reduce the effects of lighting variations.

5.3. IMPLEMENTATION FOR AVITRACK

Non-Invariant Statistical Measurements

Initially, the mean, variance, standard deviation, skew and kurtosis were taken and the object classes were plotted to observe the clustering. These statistical measures were found to be ineffectual for the purposes of classification as the clustering per class was erratic at best. This approach was therefore not pursued further.

Height and Width Measurements

Using the height to width ratio, even in the image plane within the AVITRACK project, produced much improved class clustering as compared to the standard statistical measurements taken in Section 5.1.1. This information was then used to create basic decision boundaries and classification was performed through use of a k-NN scheme.

Although the clustering of classes for height and width in the image plane was encouraging, the classification process itself was still too variable, as the measurements were taken from a variant source. To rectify this, the height and width measurements were taken using known camera geometry. This improved the stability of the measurement process and thereby improved the classification accuracy.

Moments and Moment Combinations

The use of moments within this project allowed different object classes to be modelled and represented by well-established statistical measures. Unfortunately, the objects used were not, in hindsight, well suited to the use of moments, and even the complex combinations of moments, which create semi-variant and invariant measurement sets did not function well.

Use of Image Patches

The use of image sub-sections greatly improves the classification accuracy within the AVITRACK classification task, irrespective of the representation used. This is mainly due to the large amount of occlusion within the AVITRACK data files. This is to be expected within a far field outdoor classification environment. It is important that the image sub-sections used capture informative image regions. To this end, points of interest operators were used. Initially, Harris points were selected for this task. These provided good, repeatable points for tracking and classification, however, these points were not guaranteed to be unique or class specific. Distortion could also corrupt useful points, which then become unusable. To increase the relevance of the points used, and to guard against the problems of affine distortions, invariants were implemented.

Invariants

Invariant descriptors are more appropriate for use in the AVITRACK classification task, as these methods can more easily cope with the perspective / affine distortion so prevalent within the AVITRACK data files.

The cameras used by the AVITRACK project perform an automatic Laplacian type operation upon the images captured, primarily in order to sharpen edges. This operation allows techniques searching for corner sections to operate more effectively, and makes Harris based techniques, especially those with proper affine adaptations, a strong candidate for inclusion within this project.

As a starting point, scale invariant Harris points were used within the project. This was soon super-seeded by affine invariant Harris points. These points allowed affine invariant areas to be represented and stored for classification.

Salient regions were trialled upon the AVITRACK data files; however, the cameras used have very low colour discrimination, which adversely affected the process. The regions located did not have a high repeatability score, so were not useful for classification purposes in this instance.

Maximally Stable Extremal Regions (MSER) were also investigated, however, from empirical testing, it became obvious that this technique also suffered from low colour discrimination problems similar to those experienced by the salient region detector. For this reason, low repeatability was realised. Both salient regions and MSER are not considered further for inclusion within the final AVITRACK classification task.

The variant and invariant image patches are all normalised using the Whitening transform, as shown in Section 5.2.8 and later using the SIFT representation.

6. BOTTOM-UP OBJECT CLASSIFICATION

6.1. DISTANCE CLASSIFIERS

In the field of computer vision, measurements are often transferred to a dimensional graph representation. Using this representation, distance can be measured between points. This distance can be used to classify new points (measurements) from unclassified objects. An overview of different methods of obtaining distance measures, and the ways in which these can be built into classification systems is given below.

6.1.1. EUCLIDEAN DISTANCE

$$d(u,v)=\|u-v\| \qquad (6.1)$$

Although this is a very basic type of measure, Euclidean distance – which measures the direct distance from one point in graph space to another in a uniform manner – is widely used and has been built into many classification systems. Its relative simplicity lends it speed and enables easy implementation.

6.1.2. MAHALANOBIS DISTANCE

$$D^{2}{}_{t} = (X - \mathrm{M}x)^{T} \Sigma_{t}{}_{-1} (X - \mathrm{M}x) \qquad (6.2)$$

x = [x,y] ^{T} the values associated with the height (y) and width (x)
Mx = the mean values [Mx,My] ^{T} (centroids) for each class
$\Sigma_{t}{}_{-1}$ = the inverted covariance matrix for class t

This distance metric, unlike Euclidean distance, does not treat the data in a uniform way; instead, the shape or distribution of the class members in graph space are taken into account, allowing more accurate class ownership to be established. This is achieved using a covariance matrix, which describes the way in which all of the data points within a class vary in relation to one another.

6.1.2.1. Use of Covariance Matrix

The covariance matrix used can be created such that the off diagonal values are zero. This, therefore, assumed no correlation with the X and Y-axis. Although this is often untrue, it makes the creation of the covariance matrix inverse mathematically simple. In this case, the diagonal (trace) values of the covariance matrix are created such that:

$$Cx = \frac{1}{N} \sum (X - \mathrm{M}x)(X - \mathrm{M}x)^{T} \qquad (6.3)$$

Where N is the number of points in the data set, X is a point in the data set and Mx is the mean of the data set.

This makes the inversion of the matrix trivial, as:

$$Cx^{-1} = \frac{1}{Cx} \qquad (6.4)$$

However, as the covariance matrix assumes no correlation between the axes, the metric may not function at an optimum level; the Gaussians may not be able to fit the data accurately. To remedy this, the full Mahalanobis distance can be taken with a full covariance matrix (by including the correlation between the x and y-axis) using the inversion such that:

$$Cx^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d, -b \\ -c, a \end{bmatrix} \qquad (6.5)$$

Assuming Cx was first arranged such that:

$$Cx = \begin{bmatrix} a, b \\ c, d \end{bmatrix}$$

This allows the Gaussians to fit the model accurately, as the correlations between the measurements are known.

6.1.3. USE OF BAYES PROBABILITY

Generalised Bayes Probability Function:

$$p(x)=\frac{1}{(2\Pi)\frac{d}{2}|\Sigma|\frac{1}{2}}\exp[-\frac{1}{2}(x-\mathrm{M}x)^{T}\Sigma^{-1}(x-\mathrm{M}x)] \tag{6.6}$$

Where Σ is the determinate of the covariance matrix, d is the dimensionality, Mx is the mean and x is a feature vector.

Use of this probability function allows the output of a system to be represented as a percentage probability of class ownership (e.g. 78% sure this is class X).

To achieve this, the probabilities should be scaled by:

$$P(k|x)=p(x|k)/p(x) \tag{6.7}$$

Where p(x) is the sum of all the classes probability.

No *a priori* probability such as:

$$P(k|x)=p(x|k)P(k)/p(x) \tag{6.8}$$

May be assumed, unless it is known that a certain class of object has a specific probability of occurring in a set location within an image. In some applications, it may be prudent to include this kind of information. If the time is taken to define *a priori* probabilities for set classes, this added information may well affect an increase in recognition rates.

6.1.4. K-NEAREST NEIGHBOUR

K-Nearest neighbour (k-NN) is a classification decision system, which makes use of distance measures within an n-dimensional space. The system finds matches for new points within the search space by consulting neighbouring points, assessing the neighbouring point's class and assigning the new point a class based on this. "K", in the case of k-NN refers to the number of points to be consulted before a classification is determined.

The equation for finding a basic k-NN based classification is as follows:

$$F(x_q)=\mathrm{argmax}\sum_{i=1}^{k}\delta\ (v,f(x_i)) \tag{6.9}$$

This classification method is simple, robust and highly effective for many real world tasks, however it is not optimal in terms of memory use and search speed [31]. Other systems, for example Support Vector Machines, have been proposed which, although significantly more complex than the k-NN algorithm, are more optimal in terms of search speed and efficiency.

6.1.5. WEIGHTED K-NEAREST NEIGHBOUR

Created such that:

$$F(x_q) = \text{argmax} \sum_{i=1}^{k} w_i \delta(v, f(x_i)) \tag{6.10}$$

Where w_i is formulated by:

$$w_i = \frac{1}{d(x_q, x_i)^2} \tag{6.11}$$

Where $d(x_q, x_i)$ is the Euclidean distance from the test point to the i^{th} reference point.

The adaptation to the basic k-NN algorithm, weighted k-NN, allows points at differing distances to contribute with differing degrees to the final classification result. For example, were k to be set to 10, the 10^{th} nearest neighbour to the test point (the point to be classified) may in fact be a significant distance from the test point and may therefore, not contribute in an accurate way towards the classification result. To rectify this situation, the greater the distance a reference point is from the test point, the less onus is placed upon the classification it returns. This is a fairer and more accurate way to classify unknown objects.

Through use of a weighting function, it is possible to use all the training data within the set, not simply k of them. The obvious disadvantage of this is that the search would then be more costly concerning time required to classify a new point.

6.2. LINEAR DISCRIMINATE ANALYSIS

Linear discriminate analysis (LDA) provides a means by which the optimum separation, concerning the ratio of intra class to inter class distance can be realised. This allows the best possible decision boundaries to be chosen in graph space, thus giving the distance classifiers the best chance of a correct classification.

There are two distinct types of LDA; the first, class-dependant transformation, transforms the data sets independently using L optimising criteria for an L-class problem. The second system, class-independent transformation, maximises the ratio of overall variance to within class variance. This approach uses only one optimising criteria, thus the entire set is transformed uniformly.

To perform LDA, irrespective of the type as before mentioned, firstly the mean (centroid) must be calculated for each data set. The mean of the entire set must then be found; in the case of a two-class example, this is found by:

$$\mu_3 = p_1 \times \mu_1 + p_2 \times \mu_2 \tag{6.12}$$

With μ_1, μ_2 being the mean values of data set (class) 1 and 2 respectively, and p_1, p_2 being the *a priori* probability of the classes. In this simple case, the probability is assumed to be 0.5 for each class.

In LDA, the intra class and inter class scatter or variance is used to formulate the criteria upon which the separation transform is based. Intra class variation constitutes the covariance of each class, and the final measure is created such that:

$$S_w = \sum_j p_j \times c_j \tag{6.13}$$

Where c_j is the covariance matrix for class j, created in the usual manner (see Equation 6.3 without the scaling factor $\frac{1}{N}$).

The inter class variance is formulated such that:

$$S_b = \sum_j (\mu_j - \mu_3) \times (\mu_j - \mu_3)^T \tag{6.14}$$

So that S_b can be thought of as the covariance of a data set whose members are the centroids of each class.

Having established these measures, the criterion for the class separation can now be formulated. If a class dependant approach is taken, the optimising criterion (of which there are L for an L-class problem) is created such that:

$$OC_{DEP_j} = c_j^{-1} \times S_b \tag{6.15}$$

Whereas the single optimising criterion used in the class independent system is created such that:

$$OC_{IND} = S_w^{-1} \times S_b \tag{6.16}$$

Eigen vector / value pairs are used to find the direction along which there is maximum discrimination information. Depending upon the type of LDA to be performed, inter or intra class transformation, the Eigen vector / value pairs are extracted from either the class dependent or independent optimising criterion.

There will always be L - 1 non-zero Eigen values, owing to the constraints upon the mean vectors during the creation of μ_3 [25]. Only non-zero Eigen values, and their corresponding eigen vectors should be used for the transform.

Having created the transform from the Eigen vectors, the data set can now be transformed in the following way, either for a class dependent approach:

$$Trans_{Set_j} = Transform_j^T \times Set_j \tag{6.17}$$

Or, for a class independent approach:

$$Trans_{Set} = Transform^T \times Entire_{Set^T} \tag{6.18}$$

The test vectors are similarly transformed before a distance measurement (e.g. Euclidean measure between the test point and the class means) are taken.

6.3. NEURAL NETWORK APPROACH

Neural networks facilitate the modelling of multi-variant and non-linear data [64]. This ability endears these types of approaches to many researchers. Neural networks (in the case of image recognition and classification tasks) function by assigning input nodes to pixel values within the image, then processing the image using transfer functions (for example sigmoid functions) and weighting values within the hidden layer (s). The output layer nodes are then binary coded to give the desired classification results. Figure 6.1 shows the basic structure of a feed-forward neural network.

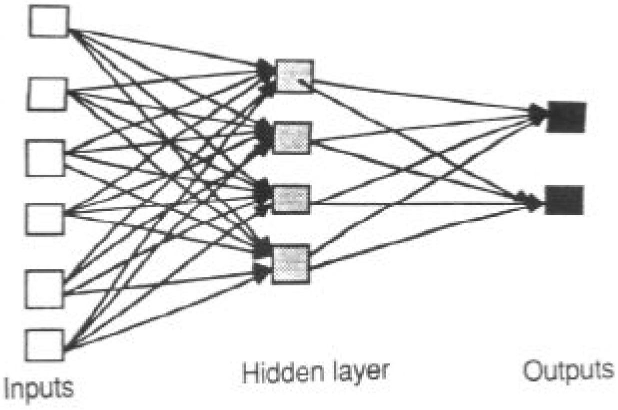


Figure 6.1 – Basic Structure of a Forward Feeding Neural Network

6.3.1. FORWARD FEEDING, BACK PROPAGATING NEURAL NETWORK WITH SIGMOID TRANSFER FUNCTION

Most neural networks use the forward feeding network topology, and many use the backwards propagation system for setting and updating the network weights (especially for classification), as this technique gives accurate results and is easy to implement. It is often important for a network to express certainty as to the classification it produces; this can be achieved through using a sigmoid transfer function.

6.3.1.1. Forward Pass

The “forward pass” of a neural network refers to the theoretical concept of data passing (in the case of Figure 6.1) from left to right, starting with an image and ending with a classification decision. The idea of a forward-feeding network also stems from this idea. The network could be thought of as being fed data initially, and then delivering a result.

In a forward feeding network, each node is used in delivering a classification result, as well as the weighting values theoretically thought of as being between these nodes. The nodes themselves are in actuality, normally represented by multi-dimensional array positions in computer simulations. If the network has been correctly trained in the back propagation stages, the output value from the forward pass (originally pixel values) will have been transformed to a value between zero and one, representing the classification certainty at a binary coded output node (when a sigmoid function is used).

6.3.1.2. Backward Pass

During the backward pass of the network (only done during training), the network error is calculated and the weightings between nodes are adjusted to create the desired output (or a close approximation of same) from the output nodes.

To ascertain the error of the output nodes and thereby correct them, the following equation is used:

$$E_k = O_k(1 - O_k)(T_k - O_k) \tag{6.19}$$

Where E represents the resultant error value, O is the network output at node k, and T is the target output value for node k.

If a sigmoid transfer function is not used (for example with a binary transfer function), the section $O_k(1 - O_k)$ need not be included. The alteration of the weighting values per node can be calculated as follows:

$$W'_{ij} = W_{ij} + E_{k-1} O_k \tag{6.20}$$

Where W' represents the new weighting value for a set node (at position ij), W is the old weighting value, E_{k-1} represents the error of the previous layer node in the network and O is the output of this node.

Now that the output nodes error is known, and the weightings for these nodes can be altered, the error of the hidden layer nodes must be found. This process (working right to left in Figure 6.1) allows the error in the known nodes to back-propagate and solve the error values for nodes at each previous layer. The equation to perform this is (with reference to Figure 6.2):

$$E_A = O_A(1 - O_A)(E_B W_{AB} + E_C W_{AC}) \tag{6.21}$$

E_A is the error value at node A, O_A is the Output of node A, W represents the weighting values between nodes (defined by its letters).

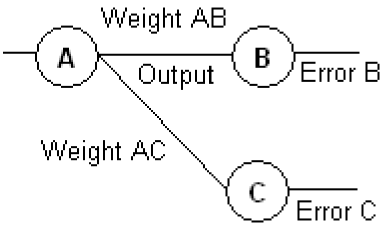


Figure 6.2 – Partial View of a Neural Network. “A” Represents a Hidden Layer Node, With “B” and “C” Being Output Layer Nodes.

6.3.1.3. Sigmoid Function

The sigmoid function provides a method for changing standard binary (Boolean) decisions into probability measures. This has the result of changing a standard “yes” or “no” class ownership statement into a percent certainty of class ownership, which allows the network to express doubt. The Sigmoid function takes the form shown in Figure 6.3, and is formulated as shown in Equation 6.22.

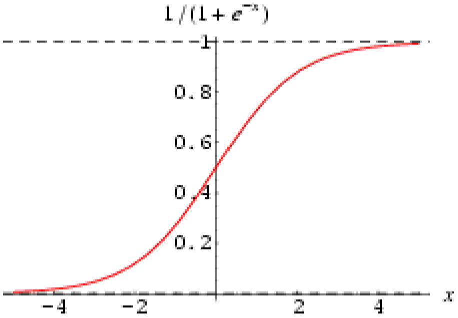


Figure 6.3 – Graphical Representation of a Sigmoid Function

$$S_k = 1/(1+\exp(-x)) \qquad (6.22)$$

Where S is the resultant sigmoid value and x is the output of the node, passed into the function.

This function is used at the transition between each layer of the neural network. The final output is then a scaled percentage probability between zero and one.

6.4. EIGEN DECOMPOSITION

It is possible to plot an image in high dimensional space, which for a standard 200*200 image, would have roughly 10.25 million possible points. This image plot could then be used to match other candidate images, and from this establish class ownership; however, this is a slow (due to the high dimensional nature of the image projection) and inaccurate system (due to the projection of noise, e.g. unwanted image sections). To minimise these problems, an Eigen space approach is taken. This approach finds the most important (principle) components of an object or set of objects and discards the noise. This system also reduces the dimensionality of the problem, thereby improving the accuracy and speed of the technique. In effect, the image is compressed, and then the compressed image is used for matching.

In this case, and for the reasons given in Section 5.2, small sections (image sub-sections) may be used to classify the objects. This approach is known as an Eigen-Window approach. To understand the eigen-window approach, it is useful to understand the founding Eigenspace approach. This approach aims to reduce the dimensionality of high dimensional information using principle components analysis (PCA).

6.4.1. EIGEN SPACE APPROACH

The simplest way of performing PCA is to create a covariance matrix and find the Eigen vectors and corresponding Eigen values:

$$Z=[z_1-c, z_2-c \ldots z_m-c] \qquad (40)$$

$$Q=ZZ^T \qquad (41)$$

Where “m” is the number of training images, $z_1, z_2 \cdots z_m$ are the training image’s intensity values of size $N \times N$, which form a matrix Z, shown above, of size N^2 X N^2 , c represents the average intensity values over all the training images, and Q is the covariance matrix.

The Eigen values and Eigen vectors are then extracted from the covariance matrix Q. In practice, these eigenvector / value pairs are found through Singular Value Decomposition (SVD) as other techniques are too slow for very large covariance matrices such as these. This system requires only minor alteration to serve our purposes, as the eigenvectors can be retrieved from the SVD process by reading either the U or V matrices created (Eigenvectors), then taking the singular values, which represent the eigenvalues.

The next stage is to find the important eigenvector / value pairs, which should capture a high percentage of the variation in the images, by ranking them. Once done, future images can be projected into the same space by simply:

$$g_i = E^T (z_i - c) \tag{6.23}$$

Where,

$$E = [e_1, e_2 \ldots e_k] \tag{6.24}$$

“E” constitutes the important Eigen vectors taken from the SVD.

$$g_i = \dim(k)$$

Where k is an arbitrary value, which captures a significant amount of variation.

A simple voting strategy, based on the distance in Eigenspace between the Eigen point formed from the candidate image passed through the above transform, and the points made from training images can then be used to generate a classification, or as in this case, a weighted k-NN system could be employed. When a large amount of training data is used, more sophisticated searching methods may be required in order to reduce search time.

6.4.1.1. Advantages

Eigen space decomposition allows high dimensional data such as images to be represented by there principle, or most important, qualities. Using this kind of transformation, the image data can be reduced from a very high dimensional state, to a much lower one. This provides a much more accessible representation for statistical analysis, searching algorithms and matching strategies.

As well as the lower dimensionality of the problem domain once transformed, extra mathematical operations can be more easily performed upon the data once taken to Eigen space, for example, threshold of image distance for image matching (equivalent to normalised cross correlation in the spatial domain), can be performed very simply.

6.4.1.2. Disadvantages

Although there are numerous advantages to the use of Eigen space, there are a few important disadvantages, which should be taken into account. Firstly, there are no guarantees that the principle components returned by the principle component analysis (PCA) are optimal for the classification task. The dimensions selected by the PCA may in fact be dimensions capturing lighting variations in the training sequence. This can however be reduced through careful selection of training data and image pre-processing before PCA is used.

6.4.2. EIGEN WINDOWS

The Eigen window approach is almost the same as the general Eigen space technique. Using Eigen windows, widows (often found from point of interest operators) are substituted for whole images. The process then continues as discussed. The Eigen window work undertaken in this project was initially based on other work by [13] [14] and [16].

Some windows may be rejected based on the low amount of variation they capture, found by taking the average change in pixel intensity value throughout the window, then using a suitable threshold value.

Once the projection data is obtained, training points are projected to create the Eigen space. Global Eigen point culling is then performed so that the optimum Eigen space for classification is created [13].

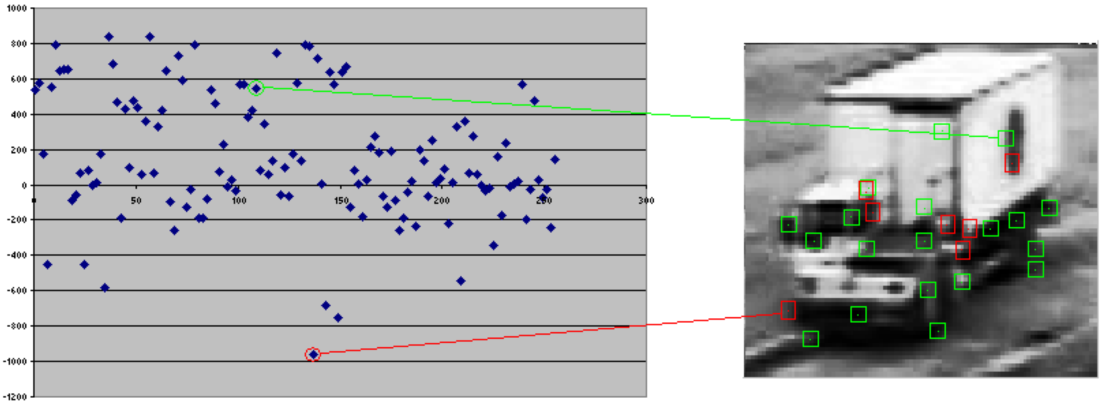


Figure 6.4 – Eigen Window Patch Matching: Green Patches Represent Correctly Classified Patch Sections, Red Represent Incorrect Points

6.4.3. IMPROVING EIGEN WINDOW SELECTION

To assess the suitability of fragments (Eigen windows) used in the last section, metrics may be used. One such metric, based on mutual information, is proposed here. This metric, used by Ullman, Sali and Vidal-Naquet [26], describes the amount of information a specific fragment or patch contains about the image in question. This measure is found by:

$$I(C,F)=H(C)-H(C/F) \tag{6.25}$$

Where C represents the class as a whole and F represents the fragment. H(X) represents the entropy of X, and is found by using the standard entropy measure, as given in Equation 5.17.

Once this metric is used, fragments (image patches) with high mutual information content are found. These are then ranked to resolve the patches with the best content. However, based on this metric alone it is not possible to extract patches which are likely to represent their respective class adequately. To locate patches which not only contain high mutual information concerning their class, but also represent their class with respect to classification, and therefore prove useful as a basis for class matching purposes, another metric must be used. This metric assesses the degree (based on probability) to which a certain patch is present within images of its own class as opposed to images of other class types. This is achieved through use of the likelihood ratio:

$$x=\frac{p(F|C)}{p(F|NC)} \tag{6.26}$$

where NC represents all classes other than the class the fragment originated from.

Due to the length of time required for this type of search, pair wise testing may first be used to highlight patches, which were likely to be useful. These patches are then tested on a reduced set of class images to further focus the set of possible image patches, before a full analysis, which gives accurate class representation probability based upon the above likelihood ratio is used.
Once the two measures are combined, patches with high informational content, and which represent their class, can be extracted and used as a basis for classification.

During the matching phase of this technique (used when ascertaining the values of the likelihood ratio for each candidate patch), two types of matching scheme are used [26], namely normalised cross correlation and an ordinal ranking measure. Other measures, such as sum of squared differences (SSD), $SSD=\sum(I_1-I_2)^2$, where I_1, I_2 represent the intensity values in image patch 1 and 2, which is also often used, is not included here as it suffers from an increased sensitivity to illumination conditions [27], which is an important limitation in outdoor visual surveillance tasks. On top of this, the normalised cross correlation technique is equivalent to Eigen space matching; allowing a direct comparison to be made between the results obtained using this method and the results obtained using the Eigen space system. This will therefore show the direct difference made by improving the quality of the patches.

The ordinal measure is included for consideration based on the results and robustness of the metric as shown in [28] by Bhat and Nayar. This measure demonstrates considerable tolerance to minor image differences during the matching process and monotone transformations of intensity (due to their ordinal nature) [28].

Normalised cross correlation uses a correlation coefficient to describe the degree of correlation (match) one image has to another at a set location on the test image. This is achieved by sliding one image, the template, which is smaller than the image it is being matched to, over a matching image while taking a matching score at each point. This is obviously an exhaustive process; however, the best match, based on the coefficient, will always be found.

Normalised cross correlation is defined as:

$$d_{f,t}^2(u,v)=\sum_{x,y}[f(x,y)-t(x-u,y-v]^2 \tag{6.27}$$

Where f is the image with positions x, y, containing the feature t positioned at u, v.

Ordinal representations of image patches allow added resilience to image distortion and allow the arrangement of image values to play a more important part in the matching process than the values themselves.

Ordinal ranking is performed by a simple permutation rank of the values within the patch. For example:

EXAMPLE 1

If an image patch contains the values:

10, 20, 30, 40, 50, 60, 70, 80, 90

then the ordinal rank is the value set:

1, 2, 3, 4, 5, 6, 7, 8, 9

To demonstrate the added robustness gained from using an ordinal ranking of the image patch, suppose the last value in the original set were altered from 90, to any number within the range 81 - 255 (255 being the bounds imposed when using digitised images). In this case, the ordinal rank value set would remain unchanged. Considerable alteration could also be levied upon the other values in the set before the rank value set would need to be reappraised.

To ensure the improved invariance gained from the transform to ordinal rank representation is maximised, a suitably robust ordinal similarity measure should be found. This measure should allow for changes in the order of the rank permutation without unduly influencing other permutation positions; therefore, the chosen measure should report significant local change, while minimising minor global change and its effect on the overall change measure.

Once this distance metric, $d(r_1, r_2)$ is created for ordinal rankings r_1, r_2 of image patches I_1, I_2 a correlation coefficient α can be found through the generalised formula:

$$\alpha = 1 - \frac{2d(r_1, r_2)}{M} \qquad (6.28)$$

Where M is the maximal value of the distance measure $d(r_1, r_2)$.

For this specific task, several possible distance metrics may be used, including Hamming distance, Spearman's ρ and Kendall's τ measure. In the case of Hamming distance, defined as:

$$d_h(r_1, r_2) = \sum (|\mathrm{sgn}(r_1 - r_2)|) \qquad (6.29)$$

Where $\mathrm{sgn}(x) = x/|x|$ if $x \neq 0$, and zero otherwise; the number of rank permutation items which do not exactly agree are returned. This is obviously a simple measuring system and does not add to the invariance or robustness of the ordinal representation, and therefore is not be suitable in this case.

The use of Spearman's ρ and Kendall's τ measures is an improvement upon the simplicity, and intrinsic rigidity of the Hamming distance measure. Spearman's ρ estimates the Euclidean distance between permutations, therefore somewhat minimising the effect of outliers, and Kendall's τ provides a normalised measure of the difference between the number of concordant and discordant pairs. It is clear from this, that Kendall's τ , although better than the Hamming distance measure, is still unsuitable for this task. Indeed, all three measures are quite susceptible to discontinuities caused, not by dissimilar images, but by naturally occurring phenomena such as specular reflection and noisy data. As these are commonplace in outdoor scenes, a measure should be selected which can, if not negate these problems, deal with them in a more robust manner.

A "K" measure, as described in [28] by Bhat and Nayar, and based on the Kolmogorov-Smirnov test statistic, seems well suited to this task. This test metric is unaffected by outlier data and captures the overall correlation between permutations [28]. The measure is created as follows:

if r_1, r_2 are the rank permutations (ordinal rank) of two image patches I_1, I_2 , then S, which is a required intermediate permutation, is the inverse permutation of r_2 with respect to r_1 . More formally:

$S = r_2^k$, where $k = (r_1^{-1})$.

Once S is found, a distance measure can be created that indirectly shows the correspondence or otherwise of the original rank permutations:

$$d_m = i - \sum J(s^i \le i) \qquad (6.30)$$

Where s^i is a subscript of S and J(B) is a function such that J(B) = 1 when true and zero otherwise.

The distance vector d_m estimates the number of preceding elements in S that are out of place. Using the distance vector, d_m , it is then possible to use the generalised correlation coefficient to find k:

$$k(I_1, I_2) = 1 - \frac{2\max(d_m)}{\lfloor \frac{n}{2} \rfloor} \qquad (6.31)$$

Where n equals the number of components in the distance vector.

Although the measure boasts good rates as compared to other matching methods in the literature, there are disadvantages to using this measure. Firstly, it assumes no ties in the data. For image patch analysis, this is a concern as the data is usually tied in some way. Secondly, the coefficient has a discriminatory range of $\left(\frac{n}{2}\right)+1$, therefore, for a small patch size, such as 3×3 patches, there are only five possible coefficient values, although for patch sizes of 7 and above the range of coefficient values becomes reasonable, namely 25 possible values at a patch size of seven. It must be noted that continually increasing the window size will not continually provide for better classification and discrimination, as both the false positive (mismatch rate) and the computational cost will increase.

6.5. IMPLEMENTATION FOR AVITRACK

Within this section, specific concerns are discussed regarding the actual implementation of the techniques previously detailed in Part III of this report. Although all techniques in this part have been implemented, some of the implementations require further detail and specifics to be given relating to the AVITRACK classification task.

Distance

As discussed in Section 6.1, a dimensional representation was used during implementation, in order to more easily take distance scores between measurements. The measurements used were the centroids, or mean values of a set of measurements for a class of visual objects. This simplifies the matching process.

The matching or classification process takes new, unclassified points and measures the distance from these points to known object centroids. The closer a new point is to a known centroid, the higher the confidence that this new object is the same class as the known one.

Initially, Euclidean distance was used to measure the distance between unclassified points and object centroids, however, this was soon replaced by Mahalanobis distance, as this takes the distribution of the training points used to create the centroid into consideration, and creates decision boundaries (threshold distances for object class membership) accordingly.

The Mahalanobis distances' covariance matrix was initially created in a simplistic manner, as described in section 6.1.2. This provided an early indication as to the likely success of the measurement system, as compared to the Euclidean system already used. After promising results, the full Mahalanobis distance covariance matrix was formed. To further enhance the output of this measurement scheme, Bayes probability distribution equations were used to create a scaled percentage probability of class ownership, instead of a distance score. This provides better user feedback, as well as providing better classification results. No prior probability was used in the probability distributions. This assumed that all objects have an equal probability of appearing within the scene, performed by using Equation 6.7. If this was known to be untrue, Equation 6.8 could be used instead.

K-nearest neighbour searches were used initially which uses Euclidean distance measurements, however, this was soon super-seeded by a weighted k-NN system as this proved to be more accurate in practice.

Linear Discriminate Analysis

The class centroids, mentioned earlier within this Section were then used within a Linear Discriminate Analysis transform to separate the measurements used and give the best possible classification accuracy. It is possible to use LDA within an Eigen space type of system to create very distinctive image patches for classification. A system such as this is discussed within the literature review. This type of system is not used here are it does not generalise well, and is therefore more suited to recognition tasks, rather than classification tasks.

Within Section 6, two types of class transform are mentioned, namely class dependant and class independent transforms. Here, class independent transforms were used, as this allows the entire data set to be transformed uniformly in a linear manner, without distorting the set itself, as can happen with class dependent transformations.

Neural Networks

The ability of neural networks to model non-linear, multi-variant data allowed a standard network with sigmoid function to classify variable class images from a learned training set.

Unfortunately, neural networks have the drawback of having a fixed size and structure once trained; therefore, any image presented to the network would have to be of a standard size. Large images in the AVITRACK video files would have a large amount of distortion placed upon them if they were reduced to, for example, 10X10 or 15X15 sizes, and network speed would decrease if larger sizes were used. For use in the AVITRACK classification task, image patches, rather than whole images should be recognised, principally so that images would not have to be resized before classification. Images of less than the standardised size (10X10 or 15X15) are not fed into the network and are classified as background clutter.

Image patches (10X10 image sub-sections initially, later increasing to 15X15) are used to train a network weights file. This file is then used for recognition later. The network was set up so that there were one hundred input nodes (for the 10X10 system) – one node per pixel grey value – three hundred hidden nodes and two output nodes, which were binary-coded to give an output of one of two possible outputs (i.e. two possible classes). The network was then used to discriminate between easily confusable object types (based on the most commonly misclassified objects from the Mahalanobis distance classifier), such as conveyor belts and loaders. The image patches were initially taken from non-discriminate image patches, then, from Harris corner points. Finally, invariant image patches were used for classification.

After classification between two easily confusable objects, the system was trialled on the classification of more object types.

Eigen Decomposition

The Eigen windows were initially chosen through the included corner data found within the AVITRACK data files (using the "Harris" corner detector previously mentioned). A 10X10 window is extracted around these corner points and the covariance matrix and projection data are then extracted as described in section 6.4. The variable Harris points were later super-seeded by invariant point of interest operator points. This gave the process more stability.

Improving Eigen Window Selection

The information given in Section 6.4.3 concerning Eigen window selection was used to ensure the best possible Eigen windows were used in the classification phase. This was a slow and involved process, however, as the results show, improvement has been affected in the classification results due to this.

Cross correlation is slow and inaccurate, and was therefore not used to assess the quality of patch sets. The use of the k-measure provides a considerable speed increase over the cross correlation method; however, for extensive search problems, such as searching an entire object class of patches, the same efficiency dilemma reoccurs, as the images must be converted to ordinal rank representation.

7. BOTTOM-UP CLASSIFICATION – SOME RESULTS & CONCLUSIONS

In this section, some results are presented, and conclusions are drawn as to the suitability of the bottom-up classification techniques as applied to the AVITRACK project, giving their relative merits and drawbacks.

7.1. TECHNIQUE CONCLUSIONS

7.1.1. IMAGE PRE-PROCESSING

Pre-processing plays a large role in determining the classification results obtained within this project when using appearance based techniques for representing or matching images. This is mainly due to the outdoor, far field nature of the AVITRACK project (which adds large amounts of perspective distortion and lighting variations), as well as the low contrast pictures obtained from the cameras used. Image pre-processing allows the effects of perspective distortion to be minimised using scaling algorithms, as well as limiting the effects of lighting variations.

Pre-processing has a computational cost attached to it; however, if a suitable balance can be found between cost and benefits, final recognition rates can be improved without unduly reducing the overall speed. In this project, different rescaling techniques were trialled, with differing computational costs associated with them, from the very low (nearest neighbour interpolation), to the highest (bi-cubic interpolation). Although the cost per use for each interpolation method is quite low overall (as compared to, for example, locating scale invariants), the images within a matching system often requires multiple rescaling before a classification may be decided upon. This therefore increases the time required by a factor equal to the number of images to resize, and which grows with increased image size.

Within this project, it was found experimentally (See [85]), that little difference was evident between bi-linear and bi-cubic rescaling methods, both in visual examination and classification rates. It was therefore decided that the extra smoothing available within the bi-cubic scheme was superfluous to requirements and that to maximise efficiency and classification rates, bi-linear interpolation may safely be used.

Having completed work using Gaussian scale space, it is possible that an alternative to using bi-linear or bi-cubic scaling would have been to use nearest neighbour interpolation then smoothed the image using a Gaussian convolution. This idea is normally referred to as a Gaussian pyramid. The drawbacks to using this type of technique are that it is most efficient when the desired size is a power of two from the original image. This is unlikely to occur within the AVITRACK image files, as the images are often non-square and are of arbitrary size. It should also be noted that the blurring achieved by the Gaussians within a Gaussian pyramid is recreated using pixel averages within the bi-linear and bi-cubic interpolation methods, though not to the same precision, and they do not fall away smoothly. The advantage of this simple approximation of the Gaussian smoothing system is a saving in computational cost.

It has been shown experimentally that the use of histogram equalisation within the AVITRACK project can also increases the final classification rate.

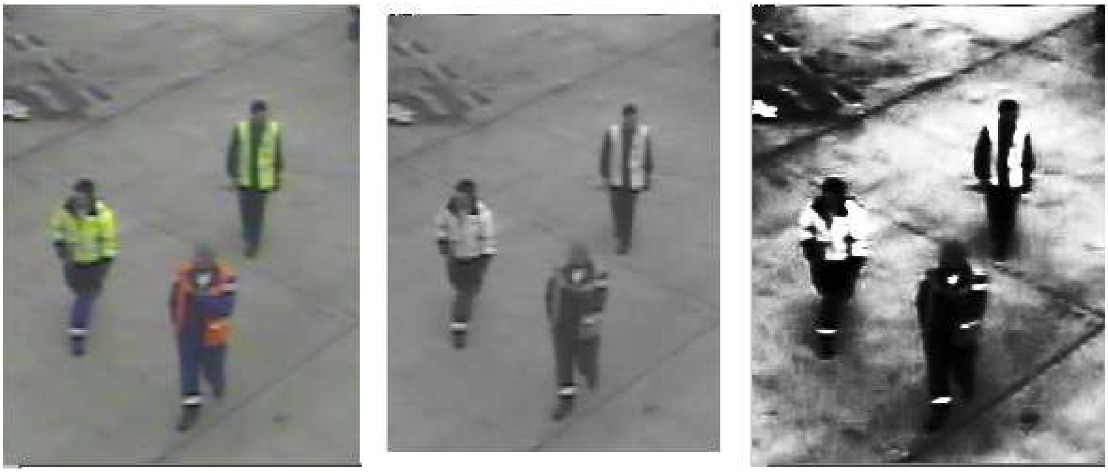


Figure 7.1 – AVITRACK Image as Seen from A Camera (left), Greyscale (middle), and Histogram Equalised (right). The Histogram Equalised Image Outlines Most Objects More Effectively and Draws Attention to Features (In This Case Distinctive Clothing Features).

7.1.2. DISTANCE CLASSIFIERS

Distance classifiers, which use models of data positions in Euclidean space and match based on distance to these positions, are useful for two reasons. Firstly, the process of creating models of the data enables clustering and position of class objects to be observed. Likely success or failure of the classifier can be gleaned from this model, as poor clustering of object types or poor classification boundary creation problems can be observed at an early stage. In the case of poor clustering, the data collection method or measurement type may be changed to obtain a better model. Secondly, there exists many ways of separating and clustering data based on this type of approach, therefore, even data which is poorly clustered my still be used in some instances, as long as a powerful decision boundary technique (for example, support vector machines or linear discriminate analysis) is used.

In this instance, the model used was based on centroid positions for each class, constructed from class measurements of width and height. The use of centroids simplified the model and clearly showed the clustering and inter-class separation. Decision boundaries were then created using both Euclidean based and Gaussian fitting techniques. Other techniques for clustering class measurement data exist, for example k-means clustering. This technique creates clusters from data by altering inter- and intra-class distances until they are optimal. This technique was not used in this case as the class types and juxtapositions are known and should be preserved in order to function as a useful model.

7.1.2.1. Distance Measures

Two main distance measures were introduced in this report, namely Euclidean and Mahalanobis based measures. These measures were used for both classification purposes, and for creating decision boundaries within the height and width based models.

There are numerous alternative distance measures for use in this type of situation. Some of the more famous include Manhattan distance, Chebyshev distance (computationally cheap but less accurate than Euclidean distance), Minkowski distance (of a set order), Quadratic distance (which is computationally very expensive) and Canberra distance (which is normally only used for non-negative values).

Euclidean distance was introduced in this report based on its simplicity, and ease of use. Comparable techniques, such as Manhattan distance could have been used; however, measures such as these add complexity without adding any significant advantages to the measurement process (Manhattan distance is similar measure to Euclidean distance using absolute values). Beyond this, while Euclidean distance takes the direct (Pythagorean) distance between two points (hypotenuse), Manhattan distance (sometimes known as city block distance) takes the distance of the other sides of the triangle, making it un-usable for model-based classification:

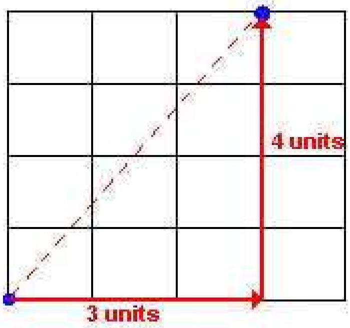


Figure 7.2 – Operation of Manhattan or City Block Measure

It is also known that Minkowski distance of order m = 1 is the same as using the Manhattan distance. If m= 2, the distance returned is the same as using Euclidean distance. As the order increases, the measure tends towards the Chebyshev result. From this, it can be seen that many of the popular distance measures are connected or similar. Therefore, use of the Euclidean measure is adequate for this project.

One of the few measurement techniques that operate on a rather unique system is Mahalanobis distance. Based on covariance matrix values, it allows models to use the clustering information from class measurements, and thus provides a more applicable measurement and decision boundary creation system than many others provide for model based distance matching. This was the main reason for its inclusion within this project.

7.1.2.2. k-Nearest Neighbour

This technique formed the main stay of the distance classification system used whenever the representation involved using a vector space. This technique is widely used by many different researchers to great affect, in many different circumstances.

In this instance, the value of "k" was not resolved through testing. This naturally presents a significant problem when proposing a best-case classification system. The three options available based on the results obtained are to set k equal to one, making this a nearest neighbour classification system. This option negates the use of a k-NN classification system. The second option is to set k to an arbitrary value between one and ten, though probably towards the higher end of the spectrum, as the results do seem to suggest an increase in classification rates at this end. This proposal would not necessarily resolve the best classification rates, and would rely on the position of the underlying data. The third option would be to use a weighted k-nearest neighbour system and simply use all the available reference points within the vector space. Although this may increase classification cost marginally, the problem of the underlying data structures would be effectively removed.

Although the testing data given does not conclusively determine the correct value of k, this classification system will almost certainly be used within this project, as it provides an accurate, fair and computationally cheap system for discerning the class of new objects based on known training points, especially when weighted k-NN is used.

7.1.2.3. Linear Discriminate Analysis

When used within this project, Linear Discriminate Analysis provided poor separation and did not aid the classification process (based upon the classification rates obtained in [85]). This is unfortunate, as this system should have increased accuracy by transforming to a best separation space. The fact that this was not done in a useful way suggests that the easily confusable objects (i.e. those with centroids in close proximity to each other) may not have an adequate best separation decision boundary due to the dominant direction of the data clustering.

This report introduced two types of LDA transform, namely class dependant and class independent; however, only class dependent was used. It can be observed within the literature, that the two differing types of LDA have particular specialities. Class independent transform LDA allows the data set to generalise well, which in this instance is not desired. Class dependant LDA, on the other hand, aids in discrimination as it aims to find the best separation between two classes by linearly separating the classes individually. Figure 7.3 shows this idea:

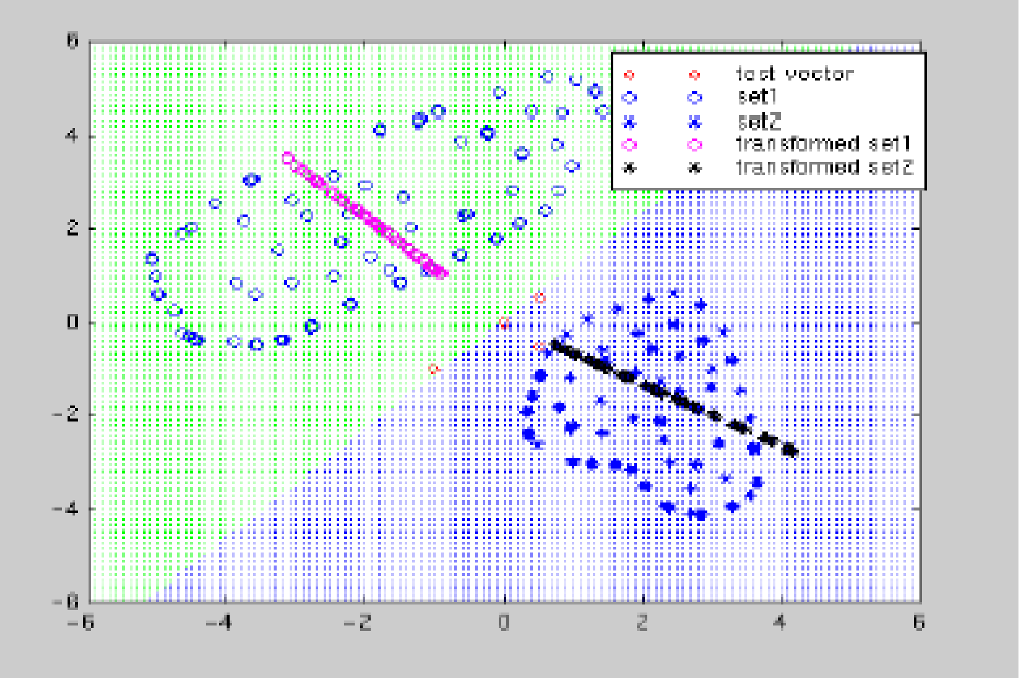
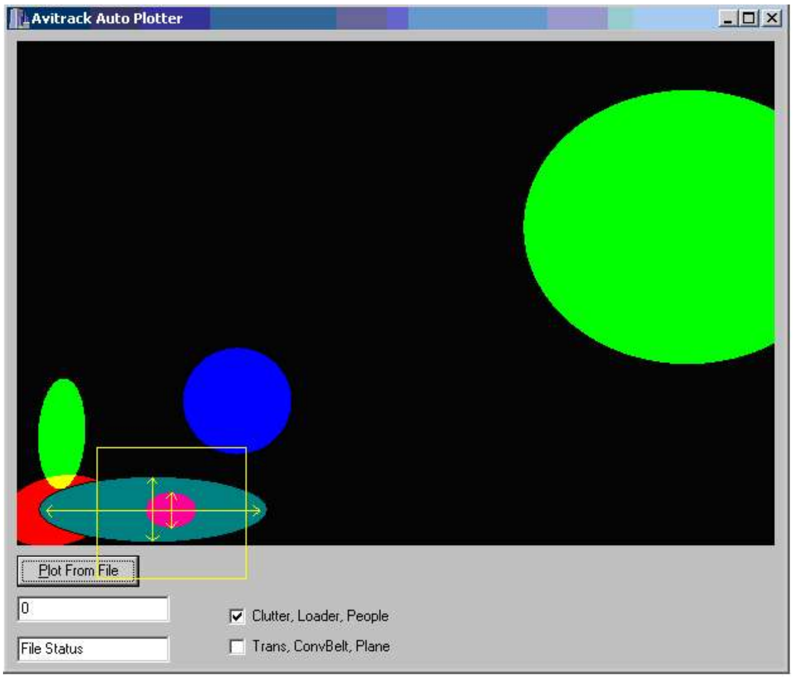


Figure 7.3 – Example of LDA Using Class Dependant Transform Superimposed Upon The Original Data.

In can be seen, using the above figure for reference and the recreated right hand side of Figure 7.4 below, that the confusable objects (within the yellow square), may have no good separation based on LDA. The dominant directions coupled with their position make this kind of separation difficult.



0

File Status

☑ Clutter, Loader, People

□ Trans, ConvBelt, Plane

Figure 7.4 – Recreation of Figure 7.3, With Confusable Objects (Neighbouring Centroid Positions) and Dominant Directions Marked In Yellow.

Due to the underlying model and its class distributions, the confusable objects are not resolved, and other centroids within the model have sufficient separation as to require no further separation. This technique does not improve the data model, and will not be included in the final classification system.

7.1.3. NON INVARIANT STATISTICAL DESCRIPTOR

In Section 5.1.1, the use of non-invariant measurement was argued for, principally so that the applicability of invariants could be assessed before implementation. This was done through observing the clustering of object classes and the separation between these. For this task, non-invariants have been very useful in helping to assessing the prospective effectiveness of invariant methods, ultimately encouraging their creation.

It is important to note that non-invariants were only used here as a preliminary stage. After non-invariants were used and examined, these methods were always upgraded to some sort of invariant. Non-invariants, by their very nature, are unstable and prone to inaccuracy.

7.1.3.1. Height / Width (Image Plane)

From non-invariants using a size-based model, two main points were gleaned. Firstly, the overall clustering using height and width models are adequate for basic hypothesis of class ownership, though not for final classification purposes. Secondly, that based on these models, there are certain objects which exhibit similar shape and size attributes, which are mainly responsible for miss-classifications.

7.1.4. INVARIANT STATISTICAL DESCRIPTORS

Invariants, as the name suggests, are a means to aid the creation or measurement of information while reducing inaccuracies and uncertainties in the creation or measurement process. This means that objects will exhibit similar, if not exactly the same, measurements in differing situations. This greatly improves and simplifies the classification and matching process, especially in far field classification tasks where measurement conditions may vary widely.

It is often the case, that invariants require significant processing (especially true in appearance-based invariants), to resolve and ensure their invariance. This increases computational cost. Having said this, the cost is normally outweighed by the increase in classification accuracy. It is also true that, given current trends, computer systems will become ever more powerful, thereby allowing the increased cost of using invariants.

7.1.4.1. 3D Height / Width

Encouraged by the non-invariant model of height and width (size), an invariant form was created using known camera geometry to increase measurement accuracy and overcome the problems of distance, or perspective, within the data files. This model used the same decision-boundary creation schemes as the non-invariant model, and matched in the same manner, but scores over the non-invariant model in the accuracy of the measurements returned.

Although this invariant model gives better results than its non-invariant form (as can be seen from the results obtained), the classification accuracy is not high enough for outright object class to be determined based on this technique. This is mainly due to the problems of 3D rotation within the invariant model, for although it is invariant to scale, the measurements returned are non-linear (i.e. as an object turns, its class remains the same, but its size alters non-linearly). This type of size-based model cannot easily cope with this problem. The model used can claim to reduce the effects of 3D rotation, which is done using Mahalanobis distance, which takes the distribution (alterations) in rotation into account for each object. Unfortunately, this is obviously not sufficient for the task, however, based on the results listed in [85].

7.1.4.2. Moments

Moments were used within this project to describe the objects within a scene in a purely statistical way. This negated the need for appearance-based normalisation. It is apparent from the testing that, generally, appearance based classification schemes outperformed statistically based classification, however, in terms of speed, statistically systems performed well. This partly explains the dependence upon statistical classification systems in the early days of computer vision, and why appearance based techniques have only recently become much more popular.

7.1.4.3. Moment Combinations

The complex combinations of moments, as discussed in the main body of this report, increases invariance in the measurement process, creating moment invariants. From observing the results obtained when testing these moment invariants, it is obvious that this technique is not suited to a task such as this. The reason for these low classification rates is that the objects (with the exception of people) within the AVITRACK sequences have very similar geometric configurations. This may also explain the general low scoring of statistically based techniques within the project.

Due to the low classification rate when classifying between vehicles, it would be unwise to use this technique for general classification purposes; however, for objects with very different geometric configurations, such as people and vehicles, this type of approach may be much more successful. This would therefore suggest a multi-stage classification system, resolving a people or vehicles decision, and then resolving the vehicle class using another technique, from the results, most likely using and Eigen window based technique.

7.1.4.4. Eigen Windows

Eigen window based patch-matching techniques, are shown to be usable and robust for classification tasks such as these. The classification rates are almost all in the 80 – 90% correct classification range, which although not as good as some techniques published, is a reasonable result considering the low camera resolution and complexity of the task.

The top scoring Eigen window based technique, which used scale invariant patches for matching, robustly and continually produced classification scores in the high 80% and low 90% range. Speed is a concern when using this technique and considerable optimisation may have to be undertaken before this technique as implemented can be used in a frame rate application, especially if a large patch size is to be used. This technique deals with the problems of image noise, which can be a problem for affine invariant techniques, as the system for resolving the characteristic scale uses very robust criteria.

Surprisingly, the next highest scoring Eigen space based technique was the unchanged (simple Harris point) Eigen window. This patch had no tolerance to the different distortions present within the project, and so was not expected to perform well. The main reason for its high score rate in this instance is the similarity between the testing and training sequence. Similar object views were available in both, so the scale and rotation difference did not overly effect the classification. Had a different training or testing sequence been used, this technique may have performed less admirably.

Before testing, the SIFT adaptations to the Harris based Eigen window technique was expected to outperform both the unchanged Eigen window and Whitening normalised Eigen window technique. After testing, it became apparent that although this technique is normally very robust, the reduced clarity of the cameras used and non-linear lighting effects adversely affected the histogram orientation assignment process upon which this technique relies. This is also apparent in techniques such as MSER and salient region detection.

Although many improvement criteria exist, the criteria used here was included to improve the quality of the patches used, while reducing as many non-informative image patches from the matching space as possible. Although the improvement criteria did reduce the overall number of patches within the matching space, thus delivering a large decrease in time requirements per patch classification, many of the patches removed were important in the matching process. This in fact created a decrease in classification accuracy. As classification speed is slightly less important than classification accuracy within this specific project, this “improvement” criterion will not be used.

Although better than most other techniques tested on the AVITRACK test set, the affine adaptations to the Eigen matching system did not perform as well as expected. It was found that the affine selection method was too sensitive to image noise, and therefore could not resolve stable affine invariant regions. This adversely affected the matching process, as spatially similar points located using this technique were dissimilar in appearance.

7.1.4.5. Maximally Stable Extremal Regions and Salient Regions

Maximally Stable Extremal Regions (MSER) and Salient regions appear to offer stable and robust methods for locating points of interest within an image; however, within the AVITRACK project this was not true. This failure is not with the algorithms themselves; rather, it is due to the low colour discrimination of the cameras used, and the algorithms reliance on locating connected components, which exhibit similar stability. As the image colour does not distinguish between image sections adequately, these stable or salient regions may be formed of a mixture of background and foreground pixels, and may change quite arbitrarily, as shown in [85]. It is possible, based on motion data, to restrict the MSER or salient region detection scheme to moving pixels only, however, the same problem reoccurs between moving pixels of different object sections.

7.1.5. APPEARANCE BASED TECHNIQUES

Within this project, a plethora of appearance-based techniques have been investigated and used. This reliance upon appearance-based techniques stems from the amount of information available when using them (often not available when using statistically based techniques), and the ever more reliable normalisation techniques available. Indeed, many state-of-the-art systems now use appearance-based techniques, reflecting a paradigm shift within the research community regarding visual classification. Appearance based techniques’ dominance within this project aimed to reflect this paradigm shift and exploit new normalisation methods within the field, especially state-of-the-art invariant normalisation methods (e.g. SIFT descriptors and affine invariants).

It is often true that many sections of an image contain little useful information for appearance-based classification. This additional, unwanted visual “noise” adversely affects classification accuracy. To counteract these problems, point of interest operators were used. Although systems have been created, which do not rely on point of interest operators (for example, original Eigenspace classification), these operators increase occlusion tolerance (important as AVITRACK is a classification task operating within a busy environment, therefore occlusion is prevalent.) and reduce computational load, as they only need to classify small subsections of an image.

Initially, Harris-Stephens interest operators were proposed. These provided 2D rotation invariance and gave a high repeatability score. The areas returned (normally corner like areas), were also distinctive. This distinctiveness was of great importance, as distinctive areas offer the best chance of correct classification. As can be seen from Figure 5.3, however, corner points do not provide optimum distinctiveness. Harris-Stephens points also suffer from a lack of 3D rotation invariance and are scale variant. This permits 3D rotation of objects within the AVITRACK scene, and the effects of perspective projection, to distort and corrupt the classification process.

Following the inclusion of Harris-Stephens points within the project, a saliency operator was introduced. It was hoped that this operator would return interesting regions of interest from a classification standpoint, as the regions located through their saliency measure would usually be complex structures, due to the nature of the measure. This would therefore improve the distinctiveness lacked by the Harris-Stephens system. Unfortunately, the saliency operator performed poorly due to the low colour discrimination of the camera system used within the AVITRACK project.

As Harris-Stephens points gave good repeatability and resolved interesting image regions, but suffered from reduced distinctiveness, this point of interest operator was improved to become more applicable to the AVITRACK classification task. Firstly, the problems associated with scale invariance were removed through the inclusion of a scale invariant image representation and searching technique. This system allowed images at different scales to be normalised to appear similar during the classification process. As can be seen from the testing data, this approach worked very well. Then, the project aimed to reduce problems associated with 3D rotation using an affine invariant image representation, which operated in much the same way as the scale invariant representation. It was feared that this representation would be extremely sensitive to noise within the image files based on testing results for MSER affine interest points. This fear was well founded. The noise within the image files made this technique highly erratic, and less effective than unchanged Eigen patch matching.

Another technique for locating scale and affine invariant regions, which is not based on Harris-Stephens or saliency detection, called Maximally Stable Extremal Regions, were also trialled upon the AVITRACK data files. This technique has received much acclaim for being a stable and efficient system. When implemented and tested within this project (see [85]), however, the problem with low colour discrimination within the data files reoccurred. This had the effect of detecting unstable or non-repeated tracked regions, reducing the effectiveness of the system to an unusable level.

Due to the colour discrimination problems and based on the testing results given in [85], the most applicable system for use as a point of interest operator is the Harris-Stephens operator. This system returns interesting corner regions which are repeatable and which have been shown, in this project, to function in low-resolution image sequences.

Two completely different normalisation methods have been proposed within this project regarding appearance-based classification using image patches. During the initial stages of development, a normalisation method based on the covariance returned by the invariant interest point technique was used. It would have been possible to simply take the region returned by the invariant operator and resize it using techniques such as bi-linear interpolation. However, had this been done, the objects within this region would have been at an arbitrary rotation, making visual matching difficult. Using the Hotelling transform, then later the Whitening transform due to its duel role in transforming and resizing image patches, the images could all be normalised not only in scale, but also in rotation. This type of normalisation method feeds naturally onto techniques such as correlation and Eigenspace matching.

It is important to note that, although these images are normalised in rotation and scale, the actual information, or pixel values, within the image patches are still subject to lighting variations and camera noise. A better alternative to using the pixel values themselves within a cross correlation based matching system comes in the form of the second normalisation method included within the project, that of the Scale

Invariant Feature Transform descriptor. This descriptor is more able to deal with lighting variations than a pixel based matching system, as it normalises the final descriptor specifically for this reason. The descriptor cannot, however, deal with the problems of camera noise or aid the low contrast nature of the images. These problems are responsible for the low classification rates obtained when using this technique.

7.1.6. NEURAL NETWORKS

Neural networks, though not an area of significant active research at present, were included within this project as they represent an accurate, based upon the literature review, and interesting method for visual object matching, allowing the modelling of non-linear multi-variant information. Neural networks also have the advantage of being able to create object class invariants of sorts. If objects of the same class, but at different rotations (though equally applicable to lighting variations and scales, etc.), are presented to the network during its training phase, all differences between the class objects will be assimilated (with an associated error rating) into the model used, and objects at these different rotations will all be treated as the same class. This naturally requires painstaking effort during the creation of the training set and often requires bootstrapping of the training set (i.e. repeatedly finding and removing contentious images) to resolve an adequate network error rate.

The main drawback to the use of neural networks, which became particularly apparent when implemented within this project, is the time required per classification when using large networks. There appears to be a “catch 22” type situation attached to the training of neural networks. On the one hand, the addition of training examples improves the classification accuracy of the network as a whole; however, the addition of new training examples slows the network during classification and increases the overall network training error making it more difficult and time consuming to train.

An additional drawback of using neural networks is that the weighting file (trained during the original training process), cannot be easily updated without training the entire system for every object class again. This means that new objects (objects not included in the original specification for the project) cannot be simply added to a neural network based classification system without complete retraining.

[85] assesses the suitability of possible image re-scaling and normalisation methods, using the neural network technique, due to its sensitivity to these factors (the differences in re-scaling and normalisation techniques created a 3% difference in results). Though useful for deciding upon pre-processing techniques, this type of sensitivity is not advantageous for object classification within a project such as AVITRACK. This sensitivity to image conditions stretches to include, and become perturbed by, image noise and heretofore-unseen lighting conditions. Lighting conditions and to a lesser extent rotation and scale, may remain a problem even if a well-chosen training set is used, as not all possible lighting conditions can be foreseen, thus invariance within the network will not be achieved. Image noise cannot be trained out of the system in any eventuality.

In the literature, there exist numerous examples of network pruning and general network optimisation techniques. These techniques provide speed and efficiency increases of up to 625 times, when using discrete cosine transformed data as proposed by Pan, Rust and Bolouri [68]. These types of optimisation techniques make neural network approaches usable even within a real time application. These optimisation techniques do not increase the accuracy of the underlying technique. Based on the testing results from [85] and from comparative testing, it is clear that this system does not produce the best classification results. For this reason, it is frivolous to spend time and energy increasing the classification speed of this technique.

7.1.7. EIGEN DECOMPOSITION

Eigen decomposition provides an attractive alternative to cross correlation matching. Although equivalent to normalised cross-correlation, it provides speed and accuracy increases not available with the original correlation co-efficient based technique. Eigen decomposition also enables matching to take place within a specialised matching space, based on the objects to be matched. This reduces the dimensionality of the matching process and effectively compresses the images.

Originally, Eigen space matching focused on matching an entire object to another object. This required that the objects to be matched were complete and not occluded. Within the AVITRACK project this is quite unlikely, and were it to be assumed, many false matches would ensue. This makes the original technique unsuitable for use in this project, however, adaptations and variations upon the theme of Eigen matching, such as that discussed in [85] may be more applicable.

8. COMBINING TOP-DOWN AND BOTTOM-UP CLASSIFIERS

As seen in the previous sections, while the top-down model-based technique can give good results, it is quite computationally costly. Bottom-up techniques have the advantages of being faster, and more generic, but might not perform well when classifying objects that are very similar in appearance (as unfortunately most of the vehicles in AVITRACK are). One way of having the advantages of both approaches is to combine a bottom-up technique with the model-based technique.

For AVITRACK, a hierarchical classifier was developed that uses a bottom-up technique for the broad classification of objects into the main types of objects – 'people', 'vehicles', and 'equipment'. If an object is labelled as 'vehicle', it is then passed on to the model-based classifier to allow it to recognise the sub-type of vehicle, i.e., if it is a 'tanker', 'transporter', etc.

Each of the two classifiers is assigned an *a-priori* confidence value for the main categories of objects. For example, the bottom-up classifier is able to classify people with a high confidence, but the model based classifier will have a low confidence in classifying people. Similarly, while the model based classifier can recognise a particular vehicle type to a high confidence, the bottom-up classifier is only able to recognise the main vehicle category (and perhaps a few distinct vehicles types) to a high confidence.

In this way, it is possible to come up with a final confidence value for an object type, constructed from a weighted sum of the a-priori confidence values of the 2 classifiers as well as the matching/classification results/scores returned by one or both of the classifiers.

Another advantage of using a hierarchical classification scheme is that the bottom-up classifier will filter out many non-vehicle objects from being passed to the model-based classifier.

Comparative results of running the AVITRACK system with hierarchical classification and individual top-down / bottom-up classifiers, is given in the AVITRACK Scene Tracking Evaluation Report [84].

9. CONCLUSIONS

This report has described the work performed on Object Categorisation for the AVITRACK Project. The special characteristics of the airport apron, with the different types of objects (vehicles, people and equipment), have been identified, together with the various challenges that the apron environment provides. In particular, many of the vehicles are very similar in appearance. Both bottom-up and top-down techniques have been implemented and evaluated on AVITRACK sequences. The relative merits of both techniques are discussed together with the results obtained. The issue of supervised versus unsupervised learning has also been addressed. Finally, we looked at combining both top-down and bottom-up techniques to improve classification results.

Performing a detailed and comparative evaluation of the methods is beyond the scope of this report; this is performed in the AVITRACK Scene Tracking Deliverable [84].

10. REFERENCES

[1]	“AVITRACK Technical Annex” – AST3-CT-2003-502818.
[2]	Nicholas Carter and James Ferryman – “Object Classification Report” – AVITRACK Internal Technical Note, Version 1.0 Draft 2, IN_AVI_2_011B, 25 Feb 2005.
[3]	James Ferryman – “People and Vehicle Tracking – Big Brother Looking After You”.
[4]	Nicholas Carter – “Face recognition and classification”.
[5]	David Thirde – “Technical Memorandum DJT/06-2002 Probabilistic Modelling”.
[6]	Thang V. Pham and Arnold W.M. Smeulders – “Statistical Strategy for Object Class Recognition Using Part Detectors” – University of Amsterdam
[7]	S. Belongie, J. Malik, J. Puzicha – “Shape Matching and Object Recognition Using Shape Contexts” – IEEE Transactions on Pattern Matching and Machine intelligence – Vol. 24, No. 24 – April 2002
[8]	B. Bose, E. Grimson – “Learning to Use Scene Contexts for Object Classification in Surveillance” – MIT
[9]	E. Sali & S. Ullman – “Combining Class-Specific Fragments for Object Classification” – Weizmann Institute of Science – Israel
[10]	V. Colin de Verdiere & J.L. Crowley – “Visual Recognition using Local Appearance” – PRIMA lab – France
[11]	R.C. Nelson & A. Selinger – “A Cubist Approach to Object Recognition” – University of Rochester

[12]	P. Viola & M. Jones – “Rapid Object Detection using a Boosted Cascade of Simple Features” – Mitsubishi Research Labs & Compaq CRL
[13]	K. Ohba & K. Ikeuchi – “Recognition of the Multi Specularity Objects Using the Eigen-Window” – Carnegie Mellon University – Feb 1996
[14]	“Detectability, Uniqueness and Reliability of Eigen-Windows for Robust Recognition of Partially Occluded Objects” – IEEE PAMI, Vol. 19, No. 9, 1997
[15]	T.F Cootes & C.J.Taylor – “Statistical Models of Appearance for Computer Vision” – University of Manchester – 2004
[16]	M.J. Black & A.D. Jepson – “EigenTracking : Robust Matching and Tracking of Articulated Objects Using a View-Based Representation” – University of Toronto – 1996
[17]	N.H. Son, N.S. Hoa & A. Skowron – “Searching for Features Defined by Hyper Planes” – Warsaw University
[18]	S.A. Nene & S.K. Nayar – “A Simple Algorithm for Nearest Neighbour Search in High Dimensions” – Columbia University
[19]	A. Hinneburg, C. Aggarwal & D.A. Keim – “What is the Nearest Neighbour in High Dimensional Spaces?” – University of Halle, Germany
[20]	R. Weber & K. Bohm – “Trading Quality for Time with Nearest Neighbour Search” – Institute for Information Systems - Switzerland
[21]	A. Papoulis - "Probability, Random Variables, and Stochastic Processes" - McGraw-Hill - 1991
[22]	"Image Processing, Analysis, and Machine Vision - Second Edition" - Milan Sonka, Vaclav Hlavac & Roger Boyle - PWC Publishing - Page 259-260
[23]	Jan Flusser and Tomas Suk - "Pattern Recognition By Affine Moment Invariants" - Czechoslovak Academy of Sciences - 1993
[24]	"Hu Invariant Set" - homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/SHUTLER3/node8.html
[25]	"Linear Discriminant Analysis - A Brief Tutorial" - S. Balakrishnama and A. Ganapathiraju - Institute for signal and information processing - Mississippi State University
[26]	S. Ullman, E. Sali and M Vidal-Naquet - "A Fragment-Based Approach to Object Representation and Classification" - The Weizmann Institute of Science, Israel.

[27]	http://cmp.felk.cvut.cz/cmp/courses/EZS/Correlation/
[28]	D.N. Bhat and S. Nayar - "Ordinal Measures for Image Correspondence" - IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 415-423, vol. 20, no. 4, April 1998
[29]	D. I. Barnea, H. F. Silverman – ``A class of algorithms for fast digital image registration'' – IEEE Trans. Computers, 21, pp. 179-186, 1972.
[30]	J.P. Lewis - "Fast Template Matching" - Vision Interface - P120-123, 1995
[31]	A. Atramentov & O. Atramentov - "RRT-Compression for KNN classification Algorithm" - Iowa State University
[32]	Emanuele Trucco & Alessandro Verri -"Introductory Techniques for 3-D Computer Vision" - Prentice Hall 1998 - Page 82
[33]	K. Milkolajczyk and C. Schmid - "An affine invariant interest detector" - European Conference on Computer Vision, volume 1, pages 128 - 142 - 2002
[34]	T. Lindeberg - "Feature Detection with Automatic Scale Selection" - International Journal of Computer Vision pages 77-116, October 1998
[35]	K. Milkolajczyk and C. Schmid - "Indexing based on scale invariant interest points" - Computer Vision 2001, ICCV 2001, Eighth IEEE International Conference, vol 1, pages 525-531, 2001
[36]	G.D. Sullivan, K.D. Baker, A.D.Worrall, C.I. Attwood and P.R. Remagnino - "Model-based vehicle detection and classification using orthographic approximations" - The University of Reading
[37]	A.E.C. Pece and A.D.Worrall - "Tracking without feature detection" - University of Reading
[38]	D.J.Moore, I.A.Essa and M.H.Hayes - "Exploiting Human Actions and Object Context for Recognition Tasks" - 7th International Conference on Computer Vision - 1999
[39]	D.L.Swets and J.Weng - "Using Discriminant Eigenfeatures for Image Retrieval"
[40]	K. Milkolajczyk and C. Schmid - "Scale & Affine Invariant Interest Point Detectors" - International Journal of Computer Vision - 2004
[41]	T.Kadir, A.Zisserman and M.Brady - "An affine invariant salient region detector" - University of Oxford, 2002
[42]	J.Matas, O.Chum, M.Urban and T.Pajdla - "Robust wide baseline stereo from maximally stable extremal regions" - University of Surry

[43] T.Tuytelaars and L. Van Gool - "Matching Widely Seperated Views Based on Affine Invariant Regions" - International Journal of Computer Vision - 2004

[44] D.G.Lowe - "Distinctive Image Features from Scale-Invariant keypoints" - University of British Columbia, Canada

[45] C.E. Thomaz and D.F.Gillies - "A maximum Uncertainty LDA-based approach for Limited Size problems - with application to Face Recognition" - Imperial Collage London - 2003

[46] M. Burl & P.Perona - "Recognition of planar object classes" - in Proceedings of CVPR 1996 pp. 223-230

[47] M. Turk, A. Pentland. ”Eigenfaces for Recognition”. Journal of Cognitive Neuroscience. Vol 3, No. 1. 71-86, 1991.

[48] E. Sali and S. Ullman - "Combining class-specific fragments for object classification" - The Weizmann Institute of Science, Israel - pp. 203 - 213 In BMVC '99

[49] J.P.Lewis - "Fast Template Matching" - Vision Interface, pp. 120-123, 1995

[50] R.O. Duda and P.E. Hart - "Pattern Classification and Scene" - New York - Wiley, 1973

[51] R.C. Gonzalez and R.E. Woods - "Digital Image Processing (third edition)" - Reading, Massachusetts - Addison Wesley, 1992

[52] R. Brunelli and T.Poggio - "Face Recognition: Features versus Templates" - IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15, no. 10, pp. 1042-1052, 1993

[53] R. Zabih and J. Woodfill - "Non-parametric local transforms for computing visual correspondence" - Proc. European Conf. Computer Vision, pp. 151-158, 1994

[54] J. Pearl - "Probabalistic reasoning in intelligent systems: networks of plausible inference" - Morgan Kaufmann publishers, inc., San Mateo, California, 1988

[55] J.S. Beis and D. Lowe - "Indexing without invariants in 3D object recognition" - IEEE Trans. on pattern analysis and machine intelligence, vol. 21, no. 10, pp. 1000-1015, 1999

[56] Y. Ke and R. Sukthankar - "PCA-SIFT: A more distinctive representation for local image descriptors" - Carnegie Mellon and Intel Research

[57] D.G. Lowe - "Object recognition from local scale-invariant features" - Univ. British Columbia - 1999

[58] F. Torre and M.J. Black - "Robust principle component analysis for computer vision" - Univ. Ramon LLull, Spain

[59] P. Zhou, J. Austin and J. Kennedy - "A high performance k-NN classifier using a binary correlation matrix memory" - Advances in neural information processing systems, vol. 11 1999

[60] S. Lawrence, I. Burns, A. Back, A. C. Tsoi and C. L. Giles - "Neural network classification and prior class probabilities" - appears in "Tricks of the trade, lecture notes in computer science state-of-the-art surveys", Springer Verlag, pp. 299-314, 1998

[61] E. Skubalska-Rafajlowicz and A. Krzyzak - "Fast k-NN classification rule using metrics on space-filling curves"

[62] G. Demiroz and H. Altay Guvenir - "Genetic algorithms to learn feature weights for the nearest neighbour algorithm" - 1996

[63] J. Laaksonen and E. Oja - "Classification with learning k-nearest neighbours" - Univ. Helsinki, 1996

[64] D. Faraggi - "The maximum likelihood neural network as a statistical classification model" - American national cancer institute

[65] T. Kavzoglu and P. Mather - "Assessing artificial neural network pruning algorithms" - in proc. 24th annual conference and exhibition of the remote sensing society, pp. 603-609, sept. 1998

[66] R. Parekh, J. Yang and V. Honavar - "Constructive neural network learning algorithms for pattern classification" - Oct. 20, 1998

[67] B. Lerner, H. Guterman, M. Aladjem and I. Dinstein - "A comparative study of neural network based feature extraction paradigms" - Pattern recognition letters, vol. 20 pp. 7-14, 1999

[68] Z. Pan, A. G. Rust and H. Bolouri - "Image redundancy reduction for neural network classification using discrete cosine transformation" - 2000

[69] R. Polikar, L. Udpa, S. Udpa and V. Honavar - "Lean++: an incremental learning algorithm for multilayer perceptron networks" - 2000

[70] D. G. Lowe - "Object recognition from local scale-invariant features" - in proc. of international conference on computer vision, pp. 1150-1157, 1999

[71] T. Kadir and J. M. Brady - "Scale, saliency and image description" - International journal of computer vision, pp. 83-105, 2001

[72] C. E. Shannon, ``A mathematical theory of communication,'' Bell System Technical Journal, vol. 27, pp. 379-423 and 623-656, July and October, 1948.

[73] J. Shi & C. Tomasi - "Good Features to Track" - IEEE conference on computer vision and pattern recognition - Seattle, June 1994

[74] C. J. Harris and M. Stephens, "A combined corner and edge detector," In Proc. 4th Alvey Vision Conf., Manchester, pages 147-151, 1988.

[75] M. Borg & J. Ferryman – “Formalisms for Model Representation” – AVITRACK Deliverable D1.3A (DL_AVI_2_001), Version 1.0 Draft 2, 1 June 2004 – The University of Reading.

[76] G.D. Sullivan - “Visual Interpretation of Known Objects in Constrained Scenes” - In Phil. Trans. R. Soc. Lon, vol. B, 337, pp. 361-370, 1992.

[77] J. Ferryman, A.D. Worrall & S. Maybank - “Learning Enhanced 3D Models for Vehicle Tracking” - Proc. of British Machine Vision Conference, 1998.

[78] Laure Bajard - “Glossary” - Internal Technical Note, IN_AVI_1_024, Version 1.0 Draft 1, 17 May 2004.

[79] I.Weiss & M.Ray - “Model-Based Recognition of 3D Objects from Single Images” - In IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.23, no.2, pp.116-128, 2001.

[80] “Numerical Recipes in C”, 2nd Edition, Cambridge University Press.

[81] James Ferryman - “Visual Surveillance using 3D Deformable Models” - PhD thesis, Dec 1999, The University of Reading.

[82] Arthur Pece & Anthony Worrall - ”A Newton method for pose refinement of 3D models” – In Proc. of the 6th Int. Symposium on Intelligent Robotic Systems, 2111-23 July 1998, The University of Edinburgh, UK.

[83] Florent Fusier - ”Architecture for Vision Algorithms. WP1 to WP4” - Internal Technical Note, INRIA Meeting Minutes, IN_AVI_3_001, Version 1.0 Draft 1, July 2004.

[84] Josep Aguilera & Gustavo Fernandez - ”Prototype Scenes Tracking Evaluation” - AVITRACK Deliverable D6.1B, Version 1.2, July 2005.

[85] Nicholas Carter – ”Research Report – M.Sc. Engineering and Information Science” - The University of Reading.