J Intell Robot Syst (2017) 88:527–546
DOI 10.1007/s10846-017-0542-5

Machine Vision for UAS Ground Operations

Using Semantic Segmentation with a Bayesian Network classifier

Matthew Coombes · William Eaton · Wen-Hua Chen

Received: 15 September 2016 / Accepted: 16 March 2017 / Published online: 31 March 2017

Abstract This paper discusses the machine vision element of a system designed to allow Unmanned Aerial System (UAS) to perform automated taxiing around civil aerodromes, with only a monocular camera. The purpose of the computer vision system is to provide direct sensor data which can be used to validate vehicle position, in addition to detecting potential collision risks. In practice, untrained clustering is used to segment the visual feed before descriptors of each cluster (primarily colour and texture) are used to estimate the class. As the competency of each individual estimate can vary dependent on multiple factors (number of pixels, lighting conditions and even surface type). A Bayesian network is used to perform probabilistic data fusion, in order to improve the classification results. This result is shown to perform accurate image segmentation in real-world conditions, providing information viable for localisation and obstacle detection.

M. Coombes (✉) · W. Eaton · W.-H. Chen
Department of Automotive and Aeronautical Engineering, Loughborough University, Loughborough, LE11 3TQ, UK
e-mail: m.j.coombes@lboro.ac.uk

W. Eaton
e-mail: w.h.eaton@lboro.ac.uk

W.-H. Chen
e-mail: w.h.chen@lboro.ac.uk

Keywords Unmanned ground operations · Semantic image segmentation · Bayesian network · Domain knowledge

1 Introduction

Over the last few decades, the capabilities of unmanned aircraft have significantly improved, primarily due to extensive research and development for military use. Many roles that once required a manned aircraft are now primarily performed by UAS. As UAS are becoming increasingly mature, potential application outside of conventional military use are being explored, with much research activity focused on allowing UAS to operate in civil airspace. As military and civil aircraft operations differ significantly, there are many barriers that must first be overcome.

One such barrier is determining how UAS will make use of ground facilities. The current inability to operate in non-segregated aerodromes represents a large barrier to bringing UAS into the National Airspace System (NAS), with automated taxiing and aerodrome operations already identified as a research gap [1]. As it would be both impractical and expensive to construct new aerodromes solely for unmanned aircraft, this work is based on the prediction that future civil UAS will operate from existing runways alongside conventional manned aircraft.

With the potential risks of having automated vehicles manoeuvring around passenger aircraft, the primary motivation of this work is to ensure the safety of the UAS and other aerodrome users. For safe transit, there are two main requirements; ensuring that the aircraft is in the correct position and ensuring that it does not collide with anything during taxiing. Other requirements for automated taxiing, such as efficient route planning or communication protocols, are already mature areas of research and existing algorithms are assumed to be already available [2].

The correct sensors must be chosen which can achieve robust localisation, and obstacle detection. While autonomous taxing has been achieved previously on the Global Hawk aircraft using DGPS, IMU, and highly accurate maps to guide the aircraft around a segregated air force controlled airport [3], this is totally inadequate for a civil aerodrome. Such a system would need constant supervision by the remote pilot as it will not react to obstacles, and will not function without DGPS corrections or in GPS denied environments.

Other sensors such as LIDAR could also be used for robust, accurate localisation and obstacle detection [4]. However the addition of extra sensors is undesirable as they add weight, power drain, and certification difficulties. This is why we also aim to find out if these functions can be provided by a single monocular camera, as almost all UASs will have one. As such, a machine vision approach is the only feasible method of direct sensing.

This work has been undertaken in conjunction with BAE Systems, who have provided the practical data used for testing and validation.

There has been previous work that has similar objectives in the field of Advance Driver Assistance Systems (ADAS) and autonomous driving. For example a colour-based machine vision approach is used for tracking unmarked road lanes in [5] and [6]. While the surface and it's markings are similar for roads and aerodromes, they are both have hugely different environments and vehicles. Which means that they have very different considerations in the image processing meahod used. Aerodromes are more open and controlled environments than roads, with much fewer more predictable objects to classify. Here we aim to use this low number of object classes to more reliably find not only surface markings but also objects like the taxiway and potential obstacles. Ground vehicles can afford the extra weight and power consumption of extra sensors. The complexity and greater payload capacity are why road vehicle ADAS use a range of sensors, and we aim not too.

1.1 Machine Vision Approach

Object detection using 2D cameras is conventionally achieved by seeking specific objects. As aerodromes are strictly controlled, most potential hazards (i.e. large objects) belong to only a few objects types (e.g. aircraft, ground vehicles, pedestrians, buildings). Therefore, an approach could be taken in which collision risks are detected by specifically seeking these object types within the image. However, despite the low probability of other types of object being present on a taxiway or runway, any object in front of an UAS would still pose a collision risk. As such, the machine vision system must be capable of detecting any kind of generic obstacle.

As it would be impractical to attempt to identify all possible objects within an image, a simpler approach is to identify safe terrain features (such as empty runway) and infer collision risks from the remaining regions. Therefore, the aim of this work is to robustly classify a small number of classes, primarily focussing on terrain types such as asphalt and concrete. Subsequently, using the probabilistic confidence of that classification, unknown collision risks can be inferred from regions of the image which are not perceived to be a known class.

Unlike active sensing techniques, such as LIDAR, 2D machine vision works with conventional images which have no inherent depth information. Instead, the distance to objects must be estimated based on features within the image. For techniques which rely on specific object detection, the pose and size of a known object within the image can be used to determine it's position relative to the camera with reasonable accuracy for avoidance. However, for generic risks the size of the object within the image cannot be used to estimate distance. Instead, as the limits of each object are defined, the bottom edge can be assumed to be closest to the camera. As the camera is mounted on the aircraft at a fixed height, an Inverse Perspecive Mapping (IPM) can be used to establish the distance from the camera to the object, a technique which has

already been explored in [7]. An additional benefit of this approach is that by identifying and classifying the terrain, IPM can also be used to determine ground features relative to the aircraft, such that map matching can also be used to assist in vehicle localisation. A similar method is performed in [4], using particle filters.

In order to ensure that all potential collision risks are identified, every pixel captured by the camera must undergo classification. This form of total-image classification is typically referred to as semantic segmentation, in which an image is not only divided into regions but in which regions are also assigned a class based on their contents. There have been a number of previous works in the area of semantic image segmentation, in papers such as [8]. The basis of this work is a continuation from the initial research conducted in [9], as well as an extension and a combination of [10] and [11].

After segmentation is complete, each region within an image should represent a single object or surface type. This allows all data within a region to be used for classification. Classification begins by extracting features from each cluster and compare them to known examples of each class. The best match is taken as the classification estimate and the degree of similarity provides a confidence in the result. As images are extremely data-rich, different types of data can be extracted from each cluster. Some of this data is extremely simplistic, such as the mean colour data for each cluster. By contrast, other data is extremely complex, such as texture data which has no standardised method of simplification. Therefore a final classification is formed by combining the results from multiple feature types together in a meaningful way.

Improving segment classification is the main focus of this paper. In the previous work [9], a naïve data fusion approach was used to simply combine different types of information (such as colour and texture) in order to estimate the class of a segmented image region.

In the paper [11], the authors improved classification performance by probabilistically combining information in a Bayesian Network (BN). BNs have been shown highly suitable for data fusion in computer vision applications, allowing contextual information to be included to improve performance [12] [13]. This is further extended in this paper by the introduction of soft evidence for texture classification. Previously only the final results of a texture classifier could be used as an input into the Bayesian Network (BN). Instead, the probability as determined by the individual classifier for each class is entered with soft evidence, with the intention that this extra information will increase the final classification performance.

The BN framework implemented specificity for an aerodrome environment is the main contribution of this work. A BN is easily extendible, so other inputs can be added without changing the original network. The idea of using uncertainty in classification to find potential obstacles is an original contribution not found in any other literature. Finally the addition of domain knowledge to the network such as horizon intercept, and luminance to improve classification performance in an aerodrome are also original.

1.2 Paper Contents

The remainder of this paper is organised as follows; Section 2 discusses image segmentation, and how untrained segmentation is performed using methods explored in a previous work [9]. Section 3 discusses how current texture based methods are used in semantic image classification; both comparing current techniques and assessing their suitability for the proposed methodology. Section 4 describes how colour can be used to aid image classification. In Section 5 a technique is introduced to will aid classification by detecting surface markings using luminance data. Section 6 describes how horizon information can be used to aid classification. Section 7 introduces the Bayesian Network (BN) data fusion method proposed in this paper, which is followed by a brief introduction to BNs in Section 8. Section 9 gives details on the final BN used for classification, including the network's structure, inputs and discretisation. Finally, Section 10 compares the performance of the current texture-only classification method against the proposed BN methodology. Comparison and results are provided using a test set of labelled aerodrome images.

2 Image Segmentation

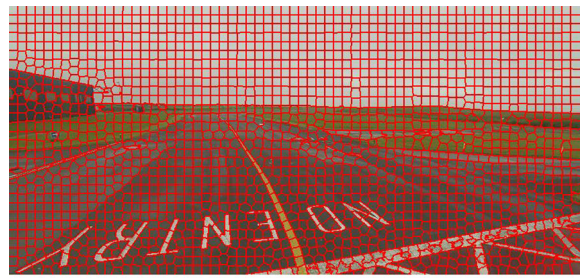
Semantic segmentation can be achieved in different ways. Most contemporary techniques perform

segmentation and classification together, in a single simultaneous process. This is referred to as classifier-led-segmentation. By simply classifying pixels at a low level (either individually or in small groups), larger regions within an image are formed where many neighbouring pixels share the same class. As additional segmentation is not required, data is extracted from each pixel only once, making such techniques highly efficient. When the intention is to divide the image rapidly into expected classes, classifier-led-segmentation is highly appropriate.

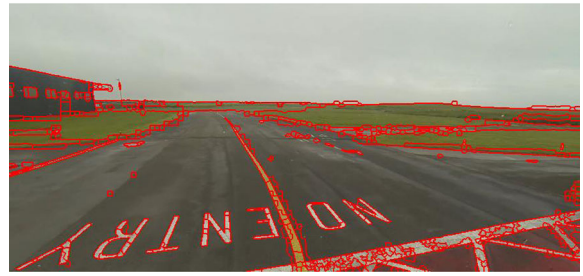
However classifier-led-segmentation can result in inaccurate region borders and small regions becoming absorbed into larger neighbours. As the aim is to localise potential obstacles and objects that can used for navigation, with inaccurate region boundaries this will reduce the localisation accuracy. The methods used in classifier led-segmentation also make implementation more difficult. Due to the complexity of fusing many different information types, non-deterministic approaches, such as Artificial Neural Network (ANN), are commonly used. As non-deterministic methods are difficult to certify for aerospace use, this work has avoided using them, and as such has eliminated the ability to use several common methodologies.

The alternative to classifier-led-segmentation is to perform segmentation and classification separately. Rather than use pixel classification to define regions, segmentation is instead achieved using basic low-level image features. As such, these methods are commonly known as 'untrained segmentation'. This work continues to use the 'superpixel' based approach, outlined in [9]. Superpixels are small clusters of pixels, grouped together based on their colour and spatial distance [14, 15]. Figure 1a is an example of superpixel segmentation achieved using Simple Linear Iterative Clustering (SLIC).

As superpixels are limited in how many pixels they contain, the end result is a significant over-segmentation, introducing many borders which are not present in the original image. The over-segmentation is resolved by a second application of clustering, grouping superpixels into larger, visually similar regions. This secondary clustering is achieved using the method outlined in [16], where it is shows that the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm is a good solution for merging superpixels, shown in Fig. 1b.



(a) Example over-segmented taxiway image, broken into superpixels (image used throughout paper)



(b) Similarly coloured superpixels clustered together

Fig. 1 Untrained segmentation performed on example runway image

3 Texture Based Classification

A texture based classification technique is presented in [9] that will be used here to give an estimation of the class of each cluster, it has been extended to also also give a certainty about its classification. This is in order to further improve the classification performance of the BN.

Texture information is most easily stored by using a texture descriptor; a consistent function which can be applied to any image (or image region) to produce comparable results. As classification is taking place after segmentation, regions are to be classified individually. As such, the texture descriptors must produce spatially cohesive results, which only sample from within each region. For this reason, local area based descriptors are used.

3.1 Texture Extraction

In addition to colour, additional region characterisation is obtained by extracting texture information. Texture transforms can be applied by convolving an input image with a special filter mask. As different masks extract different information, it is common to use

multiple masks at once; with this work using the 38 masks of the Maximum Response Filters (MR8) filter bank [17]. Comparisons are then achieved using a 'Texton' approach [18] to reduce processing time. Due to the size of the masks used in the filterbank, texture data for smaller regions can often be overwhelmed by neighbouring regions. Therefore, in addition to a convolution approach, a secondary feature descriptor is used, namely Local Binary Pattern (LBP). This compares each pixel to its immediate neighbours, providing a simple numeric response based on which neighbours have greater intensity levels. Despite this simplicity, LBP has proven especially capable at small scale texture classification and has the additional benefit of being strongly invariant to global illumination changes.

3.2 Texture Comparison

To classify a cluster its descriptor histogram needs to be compared to typical histograms of of that class. The texture comparison meahod used here is a Support Vector Machine (SVM) based approach as it produced the highest level of accuracy. Rather than use the more traditional pair-wise voting typical of SVM, a Binary Decision Tree (BDT) was used in which all classes are combined into opposing sub-groups based on similarity [19]. After a vote, the subgroups within each group are again split into two groups based on similarity, with voting continuing until only a single class remains. As such, this reduced the number of votes for each pixel-cluster from always requiring 42 to a maximum of just six, significantly decreasing processing time with negligible changes in the final result.

To improve classification performance, the intention is that texture information is entered into the BN as *'soft evidence'*, in which probabilities are used rather than discrete classifications. A such, if a region has two potential texture classes which are similar, this significance is preserved and used to effect the result within the BN.

However, moving to soft evidence introduces difficulties with this approach. The main problem being that to fit within a Bayesian framework, all probabilities must sum to 1, with each potential class receiving accurate data. Using a BDT, multiple classes are grouped together to allow for high speed classification and as such data for each class is not readily available. As alternative texture methods required significantly more processing time, the decision was made to attempt to produce representative probability data based on the incomplete data set. As the BDT is a series of successive votes, an initial approach was to use cumulative probability. Based on the comparative SVM distance between the wining and losing groups within each vote, a Probability Density Function (PDF) specific to that vote was used to convert the distance in a probability. The number of classes within the losing group was then used to normalise the result per class. However, as BDT are not uniformly divided, classes which appear higher within the BDT (i.e. those which are more distinct compared to other classes) would always receive an artificially higher probability, regardless of the actual result.

Instead, the chosen approach was to retain the SVM distance information as an indicator of confidence, allowing comparison between all classes once all voting is complete. Each time a vote occurs, the members of the losing group are all assigned the same SVM distance result (i.e. the distance to the hyperplane which defines the last vote in which the class was eliminated from contention). After the final vote, the distance of each class are 'converted' to likelihood using a Gaussian PDF [20].

$$L(Asphalt) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(D_{Asphalt}-\mu)^2}{2\sigma^2}} \tag{1}$$

where D is the distance to the hyperplane, calculated using the SVM.

To ensure compatibility with the soft BN, a normalising constant is then applied to ensure all the results sum to 1. As such, a representative probability based on SVM distance is available for each class, which is highly indicative of the actual probability, but calculated at a greatly reduced cost. This is then passed to the BN for integration with the other data.

$$P(Asphalt) = \frac{L(Asphalt)}{\sum_{i=1}^{n} L(n)} \tag{2}$$

where the denominator is the normalising constant, which is the sum of all classes likelihoods based on their nearest hyperplane distance.

Figure 2 demonstrates an example BDT and the respective SVM domain, including votes. In this example, the first vote specifies the the texture is either *Asphalt or Paint*, with the second vote determines the winning class to be *Asphalt*. Despite the texture being a closer match to *Building*, both *Building* and *Grass*

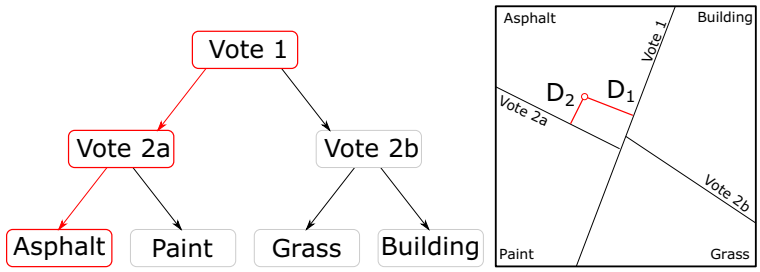


Fig. 2 A simplistic example of BDT, an it's corresponding 2D SVM

receive the same low probability, as the BDT avoids an additional vote to determine between them. This has minimal impact on the final result as the early exclusion of these classes would assign both a low probability, even if using pair-wise voting.

4 Colour Classification

Colour data is much more simplistic than texture data, typically being represented as a mean or distribution. As multiple objects of different types can share the same colour, which is very common for man-made objects, colour can rarely be used directly for classification directly. However, colour data is a suitable addition to the BN for cases where other data types are insufficient. For example, although texture data can usually provide a good indication of class, colour information can help distinguish classes which have similar texture, or in regions where texture data is limited (such as distant objects). Within an aerodrome environment, a key example is painted surface markings which can be easily identified due to their distinct colour, despite all surface markings sharing similar texture to asphalt.

Colour classification is limited to only the most common aerodrome features, specifically ones that remain the same colour across objects of that class. These are; asphalt (grey), grass (wide-range green), painted surface markings (red, white and yellow) and sky (blue through grey/white, assuming midday conditions). As each of these classes have distinct colours, colour similarity can be used to provide a measure of confidence which can be handled internally by the BN.

For this work, colour classification is primarily achieved within the Hue-Saturation-Value (HSV) colourspace, which is commonly used for image classification [21]. Unlike Red-Green-Blue (RGB), Hue-Saturation-Value (HSV)is designed to make human interpretation easier, separating colour data into channels reflecting how human vision functions. The main benefit is that the image intensity (*Value*) is separated from the colour information. This separation allows colour information to be more robust to changes in lighting.

5 Surface Marking Detection

Combining colour and texture information together can provide a better classification performance over using either data type alone. However due to weather, time of day and variable quality of taxiway surfaces, more inputs are needed to increase robustness. A key requirement is the ability to accurately classify surface markings. Not only are surface markings useful terrain features for localisation through map-matching, but they are also present along most taxiways. As such, incorrect classification of taxiway centrelines could falsely detect an obstacle infront of the UAS at any point during taxiing.

In order to help classify surface markings, a defining feature of this class is required. Fortunately, as surface markings are designed to be visually distinct, they are not only painted in bright colours but also with high reflectance paint. This causes surface markings to have a very high light-intensity, which can be used to distinguish them from other aerodrome classes.

The intensity of light is commonly referred to as *Luminance*. As image data is used, automatic white-balancing and aperture effects prevent the actual luminance value from being used. Instead, relative luminance of all pixels within the image is more common. As clusters of sky pixels will always be brighter than ground pixels [22], the relative luminance within an image is typically based on the sky. As both asphalt and surface markings are found on the ground, this

work suggests the approach of using a horizon detection algorithm to redefine the maximum brightness.

The perceived difference between surface markings and asphalt is increased by re-normalising the luminance values within the image, relative to the maximum luminance of any region on the ground. This new image, which we refer to as Normalised Relative Luminance (NRL), then represents a fairly consistent measure of the brightness of pixels on the ground. A benefit of this approach is that as NRL strongly emphasises what appears bright to human eyes, it is also highly effective in detecting yellow surface markings. Relative luminance can be derived from RGB colourspace using Eq. 3, where R, G and B are the respective mean pixel values in each colour channel per cluster, and Y is the relative luminosity of each cluster i

$$Y_i = 0.2126\overline{R}_i + 0.7152\overline{G}_i + 0.0722\overline{B}_i \tag{3}$$

Normalised Relative Luminance (NRL) can then be calculated below

$$NRL_i = \frac{Y_i}{Y_{max}} \tag{4}$$

As relative luminance is highly sensitive to hue, atmospheric effects can have a large influence in the effectiveness of NRL over great distance. This can be mitigated by combining cluster luminance information with cluster distance to camera information, and is explained in greater detail in Section 9.3 .

6 Relative Horizon Position

To increase the robustness further, and provide better obstacle class detection, each cluster's position relative to the horizon is used as a further input. Clusters which are wholly below the horizon line can be considered on the ground and will have higher probability of being a *'terrain'* class. Conversely, any cluster entirely above the horizon is not considered on the ground. Either this is because the object is airborne or the object represents the sky class. In either case, such clusters are not relevant for ground operations and the probability of being a collision risk is lowered.

Most importantly, if a single cluster extends significantly across the horizon line it can be assumed to be an object that extends up from the ground, and has a greatly increased probability of being an obstacle class. Therefore, horizon intercept represents a simple method of detecting collision risks, and removing errors from misclassification of ground to sky classes and vice versa.

Due to the flexibility of aircraft undercarriage, the horizon line will move in images. Therefore, active detection is required. A dark channel method is used to differentiate sky pixels from ground pixels as used in [23], before a regressive least squares estimate is used to approximate the horizon line. As ground objects can obscure the actual horizon position, the dark channel derived horizon line cannot differ from the attitude derived horizon line too greatly. If it does so, the assumption is made that the Unmanned Aerial System (UAS) is facing a large object (such as a building) and therefore a visually derived horizon line will not be accurate.

7 Classification Through Data Fusion

This work aims to improve the classification performance by using a Bayesian Network (BN) based data fusion approach. This probabilistic approach allows for direct comparison between metrics which are otherwise incomparable (such as colour and texture similarity). Moreover, the probability of the cluster being identified as a certain class is not only dependant on the outcomes of the individual classifiers, but also incorporates knowledge of how successful each classifier is at identifying each class. For example, a Bayesian Network (BN) approach should identify that colour based classification is better at identifying surface markings than texture classification, regardless of the confidence the texture classifier has.

Additional advantages of BNs include the ability to work with full, partial, or uncertain information. If the aircraft camera became defocussed, losing all texture information, the probabilistic approach is flexible enough to allow a result which is only dependant on other data sources, albeit with a less accurate result. The BN approach also gives each cluster a final probability in addition to an estimated class. As this probability incorporates all class information, a simple threshold can be applied to set a level below which all clusters are simply considered unknowns (and therefore potential collision risks). This allows a simplistic

method of tuning the classifier, should a higher degree of confidence be required.

8 Bayesian Networks

Bayesian networks are used to represent knowledge and reasoning under uncertainty. They are built around a probabilistic graphical model, that represents a set of random variables and their conditional dependencies. There are three parts to a BN; a Directed Acyclic Graph (DAG), a set of Conditional Probability Distributions (CPDs) for each node on the DAG, and an inference engine used to solve the network. The input criteria (also known as evidence) are variables which can be directly observed. The probabilities of each discrete variable state within the network can be calculated from the observed variables, based on the conditional dependencies.

The use of BN is not a new concept in image classification. For example [24] presents a BN framework for combining low level features to detect the most significant object within an image. Another example is [25], which uses a simple BN to combine colour and texture data with camera metadata (focal length, exposure time and flash activation) to ascertain whether the photo was taken indoors or outdoors.

This paper aims to use a Bayesian network to perform probabilistic data fusion for classification of a pre-segmented image. The data sources include the aforementioned texture classifiers, in addition to colour, horizon intercept and distance estimation. The BN should not only improve the classification performance but also provide a solution which is more robust to changing conditions. The domain knowledge applied using this technique is unique to this application, so is very dissimilar to previous works. The network parameters are found both manually and from machine learning techniques.

9 Bayesian Network Structure

The DAG for the proposed BN is shown in the lower section of Fig. 3. There are four distinctive sub networks, which include texture classification, horizon intercept, colour classification and surface marking detection. The information from these sub networks is combined in final class estimate node, which provides a more accurate result when compared to any of the individual classifiers alone.

The inputs into the full network are shown in Table 1. As a discrete BN implementation is used, discretised data is required. $Dist$, NRL, Hoz, H, S and

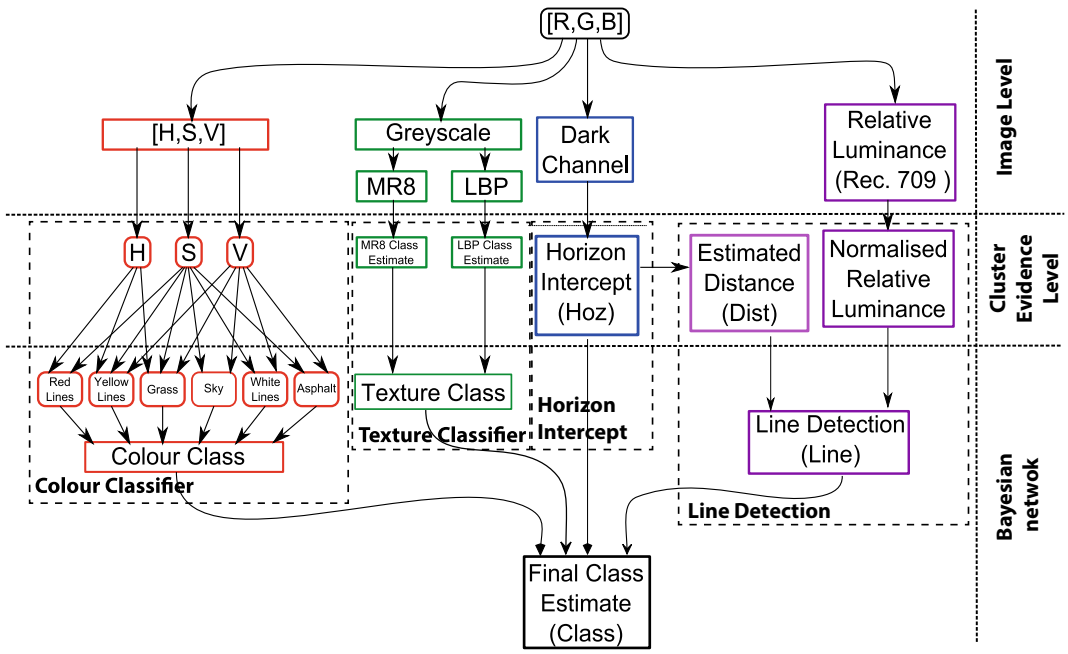


Fig. 3 Whole system showing the processing done on raw RBG image, and displaying the Bayesian Network structure

Table 1 Bayesian network inputs

Input type	Abbreviation
Mean HSV colour data	$[H, S, V]$
Relative horizon position	Hoz
Normalised Relative Luminance	NRL
Texture classification estimates	$[MR8, LBP]$
Estimated 3D distance to cluster	$Dist$

V will need to undergo discretisation. How the data is extracted and discretised is explained in the following sections.

For each cluster in the original image, the network is applied based on the cluster's data. The output is a probability of the cluster belonging to each class. The highest probability indicates the most likely class. Provided that the probability is above a chosen threshold (which separates unknown clusters) the cluster is designated as belonging to that class. This is summarised in

$$c_i = \arg\max_{Class_i} P(Class_i|H_i, S_i, V_i, MR8_i, LBP_i, Hoz_i, Dist_i, NRL_i) \quad (5)$$

where c^i is the class assigned to cluster i.

In order to complete the network, the CPDs need to be determined. Parameter estimation techniques are used to calculate the CPDs of a few key nodes. Where parameter estimation is found to give poor results, human expertise is used to manually define others. In order to minimise complexity, each CPD is trained within it's sub-network. This reduces the number of examples required for each training set.

The final class estimate CPD will be manually defined, as it fuses the four main sections (colour, texture, relative horizon position and line detection) using logic that can be easily applied by an expert. In this case the CPD for *class* has 1764 entries which are to many to display here, so a number of entries are shown in Table 2 which illustrate how the domain knowledge and data fusion logic are applied. The main principles behind the filling out of the CPD are as follows:

- Class probability is mainly based on colour and texture classification, when they agree probability it being that class is 1, if they disagree it will be uniformly distributed between them
- Colour classification performs better than texture on white yellow and red markings
- Clusters which have a horizon intercept state of $Above$ has a probability of being Sky with a probability of 1 regardless of texture or colour classification.
- Clusters which are above and below the horizon ($Above/Below$) have a high probability of being a building and a low probability of being any other class.

Table 2 Example entries from Class CPD $P(Class|ColourClass, Tex, Hoz, Line)$

Description of CPD entry	ColourClass	Tex	Hoz	Line	Asphalt	Grass	Sky	White	Yellow	Red	Building
Ground non-line cluster	Asphalt	Asphalt	Below	F	1	0	0	0	0	0	0
Classifiers in agreement/ non-agreement	Asphalt	Grass	Below	F	0.5	0.5	0	0	0	0	0
Colour classifier preference	Red	Asphalt	Below	F	0.3	0	0	0	0	0.7	0
Above horizon	Sky	Sky	Above	F	0	0	1	0	0	0	0
	Asphalt	Asphalt	Above	F	0.2	0	0.8	0	0	0	0
	Asphalt	Grass	Above	F	0.1	0.1	0.8	0	0	0	0
Ground Line cluster	White	White	Below	T	0	0	0	1	0	0	0
	White	Yellow	Below	T	0	0	0	0.5	0.5	0	0
	White	Red	Below	T	0	0	0	0.8	0	0.2	0
	Asphalt	Grass	Below	T	0.1	0.1	0	0.4	0.4	0	0
Collision risk Cluster	Yellow	Building	Above/Below	F	0	0	0	0	0	0	1
	Yellow	Yellow	Above/Below	F	0	0	0	0	0.1	0	0.9

– Clusters which have a $True$ Line state have high probability of being a White or Yellow line and a low probability of being any other class

Parameter estimation is commonly encountered when designing Bayesian networks. Unlike the intuitive network structure, numerical parameters are harder to elicit from human experts. To this end, a number of methods have been developed to estimate the parameters for both complete and non-complete data.

It is possible to provide manual classification for every cluster in the training set. This allows Maximum Likelihood Estimation (MLE) to be used. MLE selects the set of values of the model parameters that maximizes its likelihood function. MLE parameter estimation is already used for image classification, and has been demonstrated to improve performance in skin detection [26]. The application of MLE to Bayesian networks is explained in detail in [27].

9.1 Texture Sub-Network

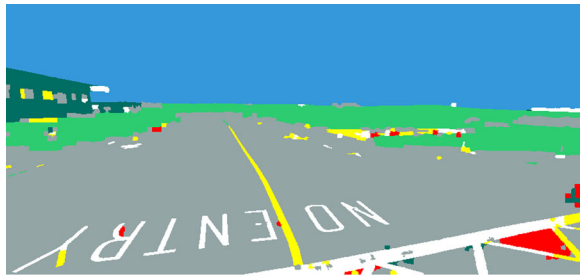
The two independent texture classifiers (i.e. based on the MR8 and Local LBP descriptors) are combined into a single texture classifier node using the simple sub-network structure seen in Fig. 3. This not only simplifies the network, but also allows sub-networks to be assessed and trained separately. Each node has seven states, which represent each of the seven different aerodrome object classes, and an unknown collision risk class.

As stated in Section 3, the two different texture descriptors are used to capture texture information across different scales. As such, the classification performance of these methods differs between classes. As LBP descriptors are small (3×3 pixels) they will typical perform better on clusters with a small area, whilst the larger MR8 descriptors (49×49 pixels) will make classification on the larger clusters more accurate. This has been confirmed through observation, where the LBP classifier has produced better results for surface markings (which are small in the full image), whereas MR8 performs better on classes such as asphalt and grass (which make up the majority of the terrain).

This correlation could be manually included in the CPD of the combined texture class node Tex; biasing the probability that LBP surface-marking classification is better than MR8, and vice-versa with classifications of asphalt and grass. However, determining how to bias the results would also require much manual effort. Instead, the differences in performance can be captured more accurately by using automated parameter learning techniques. Using MLE, a set of 100 manually classified images is used as training data to learn the CPD $P(Tex|MR8, LBP)$.

As previously described, the intention of this work is to enable the use of soft evidence within the BN. Using the probabilistic approximation techniques outlined in Section 3 the texture classifiers provide a probabilistic confidence for each class, rather than a discrete result. As such, this data is simply entered into the BN allowing, the $MR8$ and LBP nodes to function using soft evidence.

Figure 4 shows examples of classification using the independent LBP and MR8 classifiers. Each cluster is coloured to represent the winning class, i.e. the class with highest probability. For comparison, Fig. 5 shows the results from the entire texture sub-network, in which the LBP and MR8 results have been combined. The winning Tex class for each cluster is determined by $\arg\max_{Tex} P(Tex|MR8_i, LBP_i)$.



(a) MR8 texture classification

(b) LBP texture classification

Fig. 4 Example of an aerodrome taxiway image, texture classifications

Fig. 5 Combined texture classification on example image

9.2 Bayesian Colour Classification Sub-Network

The sub-network of the BN that handles the colour-based cluster classification is shown in Fig. 3. Each of the six classes for colour is a suitable metric are represented by individual nodes, all of which only have two discrete states: true (T) and false (F). This arrangement is used as a full HSV approach is not applicable to all classes (e.g. there is little use trying to identify asphalt based on hue) and therefore only relevant information is passed through the sub-network.

To allow discrete states, each of the colour channels is discretised into bands. Hue is modelled as a repeating circular distribution, and is discretised into 24 discrete states for 0–360° in 15° increments. Saturation and value are both discretised into 10 states from 0–1 in increments of 0.1. This produces a colourspace with 2161 discrete colours in total. As this would be difficult for even an expert to manually complete, the CPD is created through training.

The node $ColourClass$ is a hidden node which simply combines the individual true/false probabilities into a single node. This simplifies the network, making it easier to observe the output of the colour classifier, as all classes can be compared in a single node. In addition, this also makes the CPD of the final $Class$ estimate node much simpler, as it will have only a single parent.

An example of the classifier output is shown in Fig. 6. In this figure, the top image shows a typical aerodrome scene which has undergone discretisation in HSV, the lower half depicts the classifier output.. The class probabilities of each cluster are calculated from $P(ColourClass|H, S, V)$. This is the marginal probability distribution of $ColourClass$ with H, S, V entered as evidence. The equation below shows how the colour class estimate is make for each cluster i

$$c_i = \underset{ColourClass_i}{\text{argmax}} \; P(ColourClass_i|H_i, S_i, V_i) \qquad (6)$$

Using only the HSV colour classifier, the percentage of correctly classified pixels for the example image is 95.6%. From Fig. 6 it can be clearly seen that the largest source of error is the misclassification of white surface markings as sky. This is due to the two classes sharing the same discrete colour

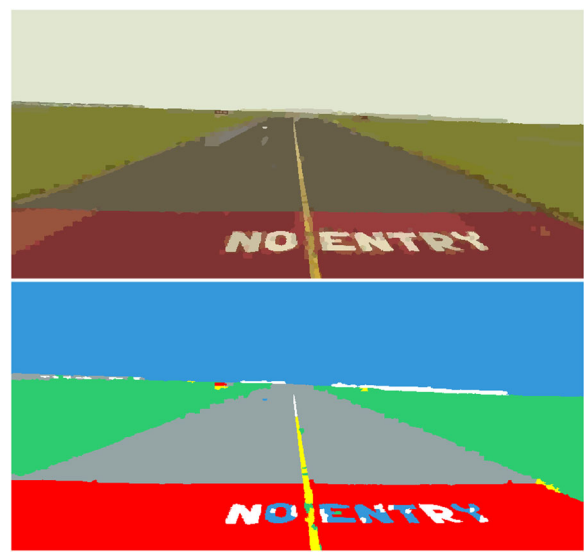


Fig. 6 Discretised HSV colour image and subsequent Bayesian network colour classifier output

and performing classification without any additional context.

9.3 Surface Marking Detection Sub-Network

The second sub-network is specifically intended to provide an additional probability of a cluster being either a white or yellow surface marking. This sub network is shown on the right side of Fig. 3, with the variable $Line$ representing this probability. As with the discrete classes within the HSV sub-network $Line$ has two states, true (T) and false (F).

As mentioned in Section 5, atmospheric effect can affect NRL over distance. On clear days, Rayleigh scattering scatters blue light more than red, lowering the NRL values of objects in the distance. Conversely, on overcast/rainy days, the presence of water droplets in the air scatters all light wavelengths equally. As such, all colour channels increase, with objects in the distance tending to have a higher NRL value. As the test footage used for this paper was taken on a wet, overcast day, NRL values increase at extreme distance. As a result, classes which would have Low NRL states in the foreground gain $Medium$ or $High$ states when far from the camera.

A different example image in Fig. 8a shows the NRL values for each cluster, clearly indicating the NRL for surface marking detection. This figure also shows the atmospheric effects created by rain, as the areas highlighted with red circles have high NRL values despite being grass and asphalt. To mitigate these atmospheric effects, clusters in the distance that have a higher NRL values need to be given a much lower probability of being a line. This is easily achieved using the BN structure shown in Fig. 3, where cluster NRL and cluster distance from camera ($Dist$) are combined together.

As only distant clusters are affected significantly, precise distance estimation is not required. Therefore, for simplicity of concept, distance to cluster is approximated using the pinhole camera model. For any point captured by a pinhole camera, similar triangles can be used to map between the 3D position of the point $P(X, Y, Z)$, and the position of the point within the image $P_c(u, v)$ as shown in Fig. 7. Given the focal length f and the height of the camera above the ground Y_{camera}. The ground position X and Z to the base of the cluster is calculated below. Using simple

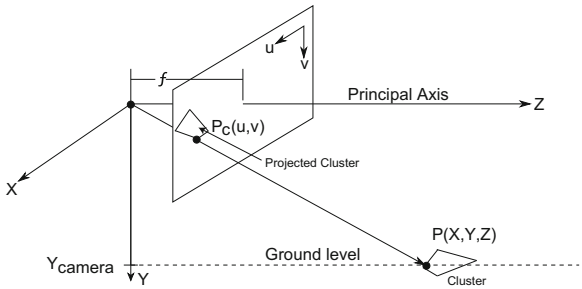


Fig. 7 Pinhole camera model used for depth estimation

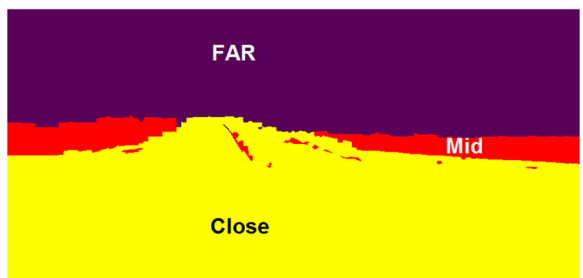
trigonometry the ground distance D_c to the cluster can be calculated.

$$Z = \frac{f.Y_{camera}}{v}$$
$$X = \frac{u.Y_{camera}}{v} \quad (7)$$

To be used within the BN, cluster distance must also be discretised. This is achieved in a similar way to NRL, with $Dist$ discretised in to three states: $Close$, Mid and Far, representing distances of less than 20m, between 20m and 55m, and more than 55m respectively.

For the same example image as in Fig. 8a the discrete states for each cluster are displayed in Fig. 8b. In ellipse 1, there are several distant clusters which

(a) Normalised Relative Luminance



(b) Discrete cluster distance

Fig. 8 Example of an aerodrome taxiway image, processed for NRL and distance

Table 3 Line CPD $P(Line|NRL, Dist)$

NRL state	Dist state	True	False
Low	Close	0	1
Medium	Close	0.75	0.25
High	Close	0.9	0.1
Low	Mid	0	1
Medium	Mid	0.55	0.45
High	Mid	0.8	0.2
Low	Far	0	1
Medium	Far	0.2	0.8
High	Far	0.6	0.4

have high NRL values due to the weather. By introducing distance, these clusters are in the *Far* state, and therefore will no longer have any chance of being misclassified as a line.

Ellipse 2 demonstrates a more difficult result, as the clusters have similar NRL values to actual surface markings at the same distance. Therefore, a *Mid* state simply reduces the probability of being a line. As actual lines generally have far higher NRL values, this is seen as an effective solution. The CPD of the *Line* node is compiled to represent these relationships and is shown in Table 3.

Figure 9 depicts the final probabilities of clusters being considered surface markings, based on the sub-network alone. Actual surface markings are clearly well defined. Although some non-surface marking clusters have non-zero probabilities of being a line, the actual probability remains low at 0.25 which will not affect the final classification unduly.

Table 4 Horizon intersect discrete states

Hoz states
Above
Below
Above/Below

9.4 Relative Horizon Position Sub-Network

The three states for horizon intercept are listed in Table 4. The horizon line is calculated and clusters with 100% of their pixels below or above this line are defined appropriately if the pixels are distributed above and below they get assigned to this state.

The horizon intercept logic is applied to the network in the final class estimate *Class* node's CPD.

Using the example image, the horizon line is calculated and the clusters assigned horizon intercept states, which is shown in Fig. 10. All the sky clusters have been shown to be above the horizon, and the ground below which will easily stop misclassifications between the two. The building in the image is in the *Above*/*Below* state which will give it a much higher probability of being classified as a building.

9.5 Unknown Classes From Uncertainty

As stated in Section 1, a key motivator of this work is that the system is capable of detecting generic collision risks (i.e. objects) within the image. As the BN must assign every cluster a known class, objects which are of an unknown class will get misclassified. However, based on the input probabilities the final result will include a low certainty of being that misclassified

Fig. 9 Example of an aerodrome taxiway image white or yellow line probability

Fig. 10 Example image cluster horizon intersect states

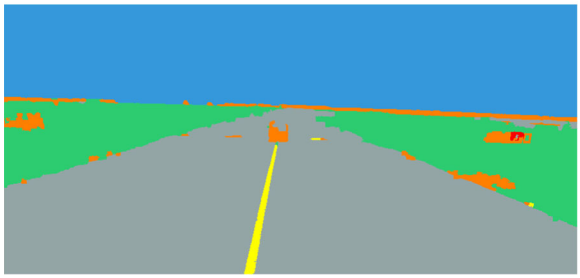
class. If a threshold for certainty is set that if the winning class probability $\arg\max_{Class_i} P(Class_i) < U$ for a cluster is below this threshold it will be classified as unknown and could be an obstacle. As such, the detection of 'unknown' objects is simply achieved through the final confidence.

$$\begin{cases} c_i = \text{unknown for } \max_{Class_i} P(Class_i) < U \\ c_i \neq \text{unknown for } otherwise \end{cases} \quad (8)$$

where *unknown* is an unknown collision risk state, and U is the probability threshold.

Figure 11b shows the results of classification using the full BN, including the addition of an 'unknown' class where U is 0.5. Clusters depicted in orange are those classified as unknown. From this figure it can be seen that a significant portion of an obstacle (i.e. a ground-vehicle) is classified as unknown and would be considered an obstacle. The marking boards are also unknown classes so would rightly be considered an obstacle. There are however some other clusters of grass around the edge of the taxi way that have a low enough certainty to be classified as unknown. This is due to the inconsistent colour of the grass around the edges, in addition to imprecise cluster borders. As grass is not a navigable surface, this result would have minimal impact on the performance during taxiing.

(a) Aerodrome RGB image



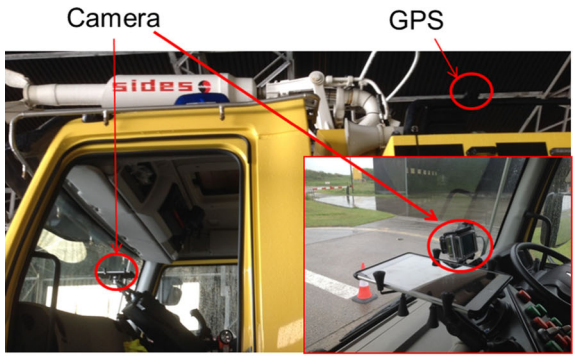
(b) Example aerodrome image with vehicle obstacle

Fig. 11 Example aerodrome image with vehicle obstacle

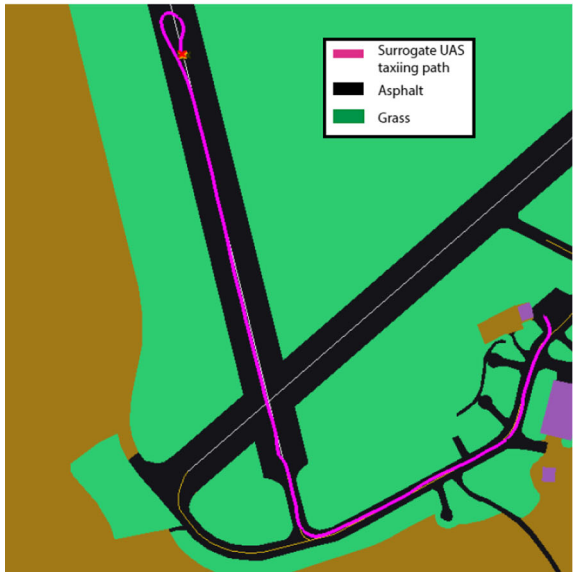
10 Results

Test footage was captured using a monocular camera mounted on a test vehicle so as to represent the view from a UAS camera system shown in Fig. 12a. The vehicle was driven around an aerodrome in the UK, driving multiple typical taxiing paths around the aerodrome, an example path shown in Fig. 12b. Obstacles, such as other vehicles, were positioned at various places to enable the testing of the obstacle detection algorithm. This scenario provides a visually realistic scene, both in terms of lighting and surface conditions. For most of the footage the weather is overcast, limiting the colour range available. In addition, the aerodrome asphalt surface is aged and worn, with inconsistent surface textures where repairs have been made. This makes this a highly challenging and realistic data set.

As this experiment was to generate results in post process, the code is not optimised. It runs at around 0.1 Hz on an I7 desktop PC. However the BN can be ran at 2 Hz on the same computer, it is the texture classifier, and the pre-segmentation that increases computational time. However as each classification is performed on



(a) UAS surrogate test vehicle to be driven around the aerodrome taxiways to collect taxiing imagery and GPS measurements.



(b) An example UAS surrogate taxiing path around Walney Aerodrome

Fig. 12 Example aerodrome image with vehicle obstacle

each cluster individually, this lends itself to parellelisation using multi-core CPU, or CUDA to enable GPU acceleration. These would enable it to run in real time.

For illustration all clusters in the example image used previously in Section 9.1 have been classified using the full BN classification method and displayed in Fig. 13. This single image shows the results to be good with the misclassified clusters shown in Purple.

From the multiple taxiing paths through the aerodrome, 100 images were selected at random. For each image, every pixel underwent manual classification; being labelled as a known class or left as an unknown. 80 of the images were then used to train the texture and BN classifiers, whilst the remaining 20 images were used for testing. Comparison between the manual classification and BN output were then used to determine the accuracy of the approach. Further results for individual BN sub-networks and the previous 'texture only' approach were also created for comparison. To aid in discussion, a particular cluster has been selected from the example image seen throughout this paper, with the classification results reviewed below.

To aid in discussion, a particular cluster has been selected from the example image seen throughout this paper and it's classification is reviewed below.

10.1 Example Cluster

The selected example cluster is labelled as 'Cluster 288' and is shown in both Figs. 5 and 13. In both MR8, LBP and the combined texture classifier, this cluster has been misclassified as asphalt whereas it should be classified as 'red-surface marking'. Table 5 shows the posterior marginal distributions for the $ColourClass$, $TextureClass$, and the final $Class$ nodes. For this aerodrome, the red surface markings are very old and worn so have a very similar texture to asphalt, resulting in the texture misclassification. Despite this, the colour for this cluster remains distinct, with the colour classifier estimating a red surface marking with a very high probability of 0.9592.

As the CPD for $Class$ gives a greater weighting on the colour classifier for surface markings, the combined result of both colour and texture alters the result such that the overall winning class is Red. Seen in the marginal for $Class$, *red* has a now winning probability of 0.7604, whereas asphalt is only 0.1905. As this is a ground cluster (i.e. a Hoz state of $Below$), there is a zero probability of it being the Sky class. It has marginal $Line$ probability for the state F of 1 which means that is has a also has a near zero probability of being a yellow or white line.

The benefit of using soft evidence from the texture classifiers can clearly be seen. Relying on hard evidence alone (as in [11]), asphalt would have received a texture probability of 1, whilst all other classes would have received zero probability based on texture. By comparison, using soft evidence the confidence in asphalt was only 0.5292. This raises the final probability for a red line estimate from 0.6714 to 0.7604, correctly identifying the class.

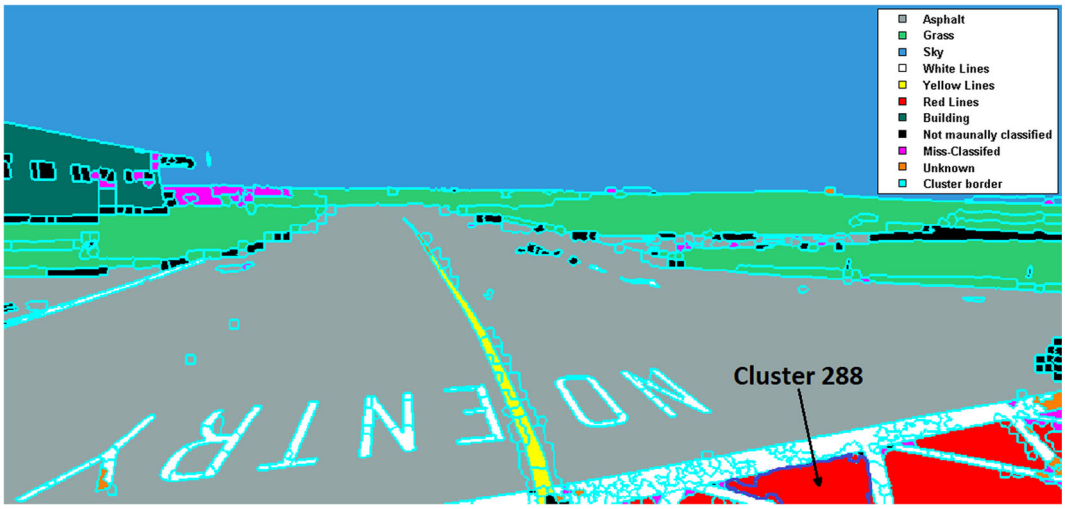


Fig. 13 Final classification of test image showing segmentation, misclassifications and unknown clusters

10.2 Classification Accuracy

For this work, there are effectively two different types of error: misclassification and segmentation error. Segmentation error is when a cluster has erroneously grown to include pixels from two or more classes. This can be due to low image quality, or more commonly superpixels being close to the boundary between two visually similar classes. Using the example image used previously, the clusters which have not been segmented correctly are shown as black clusters in Fig. 13.

By comparison, misclassification is when a classifier incorrectly identifies a cluster. Examples of misclassification are shown in purple, in Fig. 13. There are two ways to statistically represent misclassification error percentages; per cluster or per pixel. Although classification is performed on a per-cluster basis, as clusters can be drastically different in size, large errors in the final result can be lost if only the cluster based result is reviewed. (e.g. if the entire sky is a single cluster, but is misclassified as an obstacle).

The previous texture classification methods have been shown to have reasonable performance but were not considered adequate or robust enough for navigation or obstacle detection. For the test set, the average percentage error for the classification of each pixel and cluster are shown in Table 6, with results for both previous texture classifiers as well as the complete BN.

From the table it can be seen that the BN has a significant performance increase when compared to individual texture classifiers alone. When comparing correct cluster classification, the BN shows a significant improvement of 41.8% fewer incorrect clusters. For perspective, as the relative improvement in pixel accuracy is only 5.5%, it is clear that this primarily relates to improvements in the classification of small clusters. This is to be expected, as larger clusters have more data to form the texture information used for comparison, which is clearly shown in Fig. 4. Instead, the additional information provided through the BN

Table 5 Marginal posterior distribution for $Class$, $ColourClass$ and Tex for cluster 288

Class	Colour Class	Tex	Class
Asphalt	0.0408	0.5292	0.1905
Grass	0	0.0271	0.0008
Sky	0	0.0225	0
White Line	0	0.1043	0.0204
Yellow Line	0	0.1036	0.0203
Red Line	0.9592	0.0932	0.7604
Building	N/A	0.1202	0

Table 6 Percentage error for Bayesian network classifier compared to texture only classifiers of test set

Classifier	% Pixel error	% Cluster error
BN soft evidence	0.7241%	13.4%
BN hard evidence	1.48%	19.82%
LBP	6.29%	52.81%
MR8	5.12%	57.67%

Table 7 Percentage breakdown of LBP Texture only classier of test set

Man \ Auto	Asphalt	Grass	Sky	White	Yellow	Red	Building
Asphalt	95.5	2.7	1.3	0.4	0.1	.003	0.1
Grass	3.5	94.5	0.0	1.8	0.3	0.0	0.03
Sky	0.4	0.6	98.9	0.01	0.01	0.0	0.1
White	4.2	1.7	0.0	86.5	7.6	0.0	0.02
Yellow	5.4	1.7	0.0	17.0	76.0	0.0	0.0
Red	86.5	11.2	0.0	2.0	0.1	0.1	0.1
Building	2.2	0.08	0.0	0.3	0.4	0.0	97.1

and the reduced dominance of texture data through the use of soft evidence are the key reasons for the improved result. As important surface markings are typically smaller clusters, this improvement should significantly improve the ability to identify taxiing markings.

It is also useful to compare the classification for individual classes. The breakdown for the LBP, MR8 and BN classifiers are shown in Tables 7, 8 and 9 respectively. Each row represents the percentage breakdown of the original manually classified class, in terms of the automated segmentation and classification results. The highlighted diagonals represent correct classifications.

As previously stated, texture data for surface markings and asphalt is very similar, leading to poor classification results for both MR8 and LBP approaches. Red surface markings can be especially difficult to identify as they are typically far less bright, more closely resembling aspahlt when compared to yellow or white paint.

It can be seen that MR8 has misclassified red paint pixels as asphalt 86.5% and LBP 86.4% of the time. Both also have poor performance for yellow and white lines, most likely due to the many of the surface marking clusters being very small in size. As the painted line textures are also similar between classes, LBP misclassified yellow lines as white markings 16% of the time, and for MR8 white markings were misclassified as yellow markings for 19% of the pixels. Looking at Table 9 it can be seen by adding other inputs, this is dramatically improved. Increasing correct classification for white, yellow and red surface markings to 95.32, 93.65 and 97.38% respectively. Red surface markings are still misclassified as asphalt for 2.49% of the pixels. This error is likely due to red markings lacking the high NRL values required for this additional data to be useful within the BN. However as NRL can not distinguish between yellow and white lines, yellow lines are still being miss-classified as white lines for 2.05% of their pixels.

With NRL used to improve the classification of surface markings, much of the former misclassification of classes as surface markings has been eliminated. For example, LBP and MR8 texture classifiers incorrectly classified grass as a white line 1.784 and 0.971% respectively. By comparison, using the full BN lowers this misclassification to just 0.01%.

Excluding surface markings, texture was already shown to be sufficient to separate asphalt from other classes, due to asphalt's consistent and mostly uniform texture. As such, using texture classifiers alone LBP is shown to achieve 95.5% accuracy whilst MR8 achieves 99.0% independent of other data. As minimal

Table 8 Percentage breakdown of Maximum Response Filters (MR8) Texture only classier of test set

Man \ Auto	Asphalt	Grass	Sky	White	Yellow	Red	Building
Asphalt	99.0	0.5	0.0	0.2	0.3	.01	.03
Grass	10.1	87.9	0.0	1.0	0.9	0.0	0.2
Sky	0.0	0.0	98.9	1.1	0.0	0.0	0.0
White	3.5	.03	0.0	77.3	19.2	0.0	0.0
Yellow	2.5	0.0	0.0	2.1	95.3	0.0	0.0
Red	86.5	0.0	0.0	0.8	0.9	9.4	2.4
Building	2.2	0.0	0.0	0.0	0.2	0.0	97.6

Table 9 Percentage breakdown of Bayesian network classier of test set

Man \ Auto	Asphalt	Grass	Sky	White	Yellow	Red	Building	Unknown
Asphalt	99.0	0.54	0.00	0.04	0.01	0.00	0.00	0.41
Grass	1.60	98.33	0.00	0.00	0.01	0.00	0.00	0.06
Sky	0.04	0.00	99.55	0.00	0.00	0.00	0.00	0.40
White	0.90	0.00	0.01	95.32	0.77	0.23	0.00	2.78
Yellow	0.06	1.26	0.00	2.05	93.65	0.00	0.00	2.98
Red	2.49	0.10	0.00	0.01	0.00	97.38	0.00	0.01
Building	3.95	0.00	0.00	0.00	0.00	0.00	96.05	0.00
Unknown	53.57	37.71	2.78	1.04	0.54	0.57	0.00	3.80

additional information is provided through the introduction of colour, the final BN result does not significantly increase in accuracy. However, as multiple sources of data combined together provide the same result as the best indicator of class, it is hoped that this approach has made the end result more robust for scenarios where texture data is limited.

Focusing on the classification of grass, the lack of a consistent texture makes texture based classification less successful. Grass clusters will have differing densities throughout due to patchiness and possibly different grass breeds. Therefore the texture only performance is not as favourable at 94.4% for LBP and 87.9% for MR8. This is improved by using the BN classifier to 98.33%. Grass has a distinct colour (which is why hue is included for its classification) so data fusion with the colour classifier has lead to improved results.

As the training and test data sets were created using footage from the same aerodrome, it is likely that an element of over-training has been introduced within the texture classifiers. This is evident in the very high classification performance results of the texture only classifiers when classifying buildings. As there are typically very few buildings along taxiways, the entire class has been trained to identify these specific buildings. As such, when additional information is introduced within the BN, there is around a 1% decrease in accuracy compared to texture alone.

The loss of accuracy is mainly the result of over-segmentation, with parts of the building isolated fully above the horizon and therefore discarded as not being a potential risk. An example of this can be seen in Fig. 10, which lead to the misclassification of that cluster shown in Fig. 13. However, the relative reduction in classification accuracy is the result of such a small training set. Had more buildings and material types been used to create the building texture class, the accuracy of the texture classifiers would have been somewhat decreased. In this case, the inclusion of the horizon information would have been an improvement, rather than a detriment. Further testing at an airfield with more buildings is therefore required to validate this hypothesis.

As nearly all of the test images are around 50% sky pixels, the extremely large cluster size benefits both training and classification. As such, the sky class has the best classification result, with around 98.9% accuracy for both MR8 and LBP. However some misclassification still occurs when using texture alone. From Table 9, nearly all misclassification of sky clusters has been rectified by applying horizon logic. Only around 0.04% of sky pixels are incorrectly classified as asphalt, again primarily due to errors in the initial segmentation.

The largest misclassification introduced by segmentation error occurs between grass and asphalt, as shown in Table 9 where 53.5% of all segmentation errors are asphalt and 37% are grass. This is due to the most common boundary within taxiway images being the taxiway edge, with these segmentation errors accounting for an average of 2.5% of the pixels across the whole test set. Although this appears significant, as both classes represent terrain types and the clusters are typically very small, there should be minimal effect on the actual taxiing process.

11 Conclusions

In this paper we have presented a method for segmenting images and semantically classifying the resulting regions, using domain knowledge of aerodrome environments. This is to enable automated taxiing of UAS

at non-segregated aerodromes. The accurately segmented and classified images are intended to allow both improved localisation though map-matching, as well as generic obstacle detection.

A probabilistic BN framework is used for fusing multiple sources of information with domain knowledge. This method has been shown to improve classification performance compared to the individual classifiers by 5.5% per pixel and by a large 41.8% per cluster. By entering probabilistic confidence of the texture classification into the network using soft evidence, the classification performance has improved, with a per-pixel performance increase of 0.75% and per cluster of 6.42%.

We have been constrained by only using a single sensor, but have shown if the BN is developed and tuned for a particular environment, promising classification performance can be achieved with only a monocular camera. The intuitive graph structure of the BN allows for extending the network to include other sources of information in the future, such as cluster adjacency.

As the BN process is deterministic and as any node can be marginalised, it can easily be monitored and verified, making this much more appropriate for safety critical aircraft systems. This is in comparison to non-deterministic classification methods such as neural networks, which would not be appropriate.

The data set is limited to the same taxiway, with training and testing videos only being taken a few hours apart, meaning that different weather and lighting conditions where not tested. In future work we aim to extended the BN to take account image illumination, and conduct tests under different lighting and weather conditions.

Acknowledgments This work was supported by the U.K. Engineering and Physical Sciences Research Council (EPSRC) Autonomous and Intelligent Systems programme under the grant number EP/J011525/1 with BAE Systems as the leading industrial partner. The work greatly benefits from the data set collected from an airfield provided by BAE Systems and technical advice provided by the technical officer Rob Buchanan.

References

1. European RPAS Steering Group: Roadmap for the Integration of Civil Remotely-Piloted Aircraft Systems into the European Aviation System, European RPAS Steering Group, Tech. Rep. (2013)
2. Lekkas, A.M.: Guidance and path-planning systems for autonomous vehicles (2014)
3. Loegering, G., Harris, S.: Landing dispersion results global hawk auto-land system in AIAAs 1st Technical Conference and Workshop on Unmanned Aerial Vehicles (2002)
4. Durrie, J., Gerritsen, T., Frew, E.W., Pledgie, S.: Vision-aided inertial navigation on an uncertain map using a particle filter. Proceedings of the IEEE International Conference on Robotics and Automation, pp. 4189–4194 (2009)
5. Sotelo, M.A., Rodriguez, F.J., Magdalena, L., Bergasa, L.M., Boquete, L.: A color vision-based lane tracking system for autonomous driving on unmarked roads. Auton. Robot. **16**(1), 95–116 (2004)
6. Cheng, H.-Y., Jeng, B.-S., Tseng, P.-T., Fan, K.-C.: Lane detection with moving vehicles in the traffic scenes. IEEE Trans. Intell. Transp. Syst. **7**(4), 571–582 (2006)
7. Muad, A.M., Hussain, A., Samad, S.A., Mustaffa, M.M., Majlis, B.Y.: Implementation of inverse perspective mapping algorithm for the development of an automatic lane tracking system. TENCON 2004. 2004 IEEE Region 10 Conference, vol. A, Nov 2004, vol. 1, pp. 207–210
8. Yao, J., Fidler, S., Urtasun, R.: Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pp. 702–709 (June 2012)
9. Eaton, W., Chen, W.-H.: Image segmentation for automated taxiing of unmanned aircraft. Unmanned Aircraft Systems (ICUAS), 2015 International Conference on, June 2015, pp. 1–8
10. Coombes, M., Eaton, W., Chen, W.H.: Colour based semantic image segmentation and classification for unmanned ground operations. 2016 International Conference on Unmanned Aircraft Systems (ICUAS), June 2016, pp. 858–867
11. Coombes, M., Eaton, W., Chen, W.H.: Unmanned ground operations using semantic image segmentation through a bayesian network. 2016 International Conference on Unmanned Aircraft Systems (ICUAS), June 2016, pp. 868–877
12. Liu, F., Xu, D., Yuan, C., Kerwin, W.: Image segmentation based on bayesian network-markov random field model and its application to in vivo plaque composition. Biomedical Imaging: Nano to Macro, 2006. 3rd IEEE International Symposium on, April 2006, pp. 141–144
13. Bouman, C., Shapiro, M.: A multiscale random field model for bayesian image segmentation. IEEE Trans. Image Process. **3**(2), 162–177 (1994)
14. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S.: SLIC Superpixels, EPFL, Tech Rep. (2010)
15. Ren, C.Y., Reid, I.: Gslic: a Real-Time Implementation of Slic Superpixel Segmentation, University of Oxford, Department of Engineering, Technical Report (2011)
16. Kovesi, P.D.: MATLAB and Octave functions for computer vision and image processing, Centre for Exploration Tar-

geting, School of Earth and Environment, The University of Western Australia, available from: <http://www.csse.uwa.edu.au/~pk/research/matlabfns/>
17. Varma, M., Zisserman, A.: Texture classification: Are filter banks necessary?. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, Jun. 2003, pp. 691–698. [Online]. Available: http://www.robots.ox.ac.uk/vgg
18. Julesz, B.: Textons, the elements of texture perception, and their interactions. Nature **290**(5802), 91–97 (1981)
19. Xue, S., Jing, X., Sun, S., Huang, H.: Binary-decision-tree-based multiclass support vector machines. Communications and Information Technologies (ISCIT), 2014 14th International Symposium on, pp. 85–89 (2014)
20. Foresee, F.D., Hagan, M.T.: Gauss-newton approximation to bayesian learning. Neural Networks, 1997., International Conference on, vol. 3, vol. 3, pp. 1930–1935 (1997)
21. Du, C.-J., Sun, D.-W.: Comparison of three methods for classification of pizza topping using different colour space transformations. J. Food Eng. **68**(3), 277–287 (2005)
22. He, K., Sun, J., Tang, X.: Single image haze removal using dark channel prior. IEEE Trans. Pattern Anal. Mach. Intell. **33**(12), 2341–2353 (2011)
23. Yuan, H.-Z., Zhang, X.-Q., Feng, Z.-L.: Horizon detection in foggy aerial image. in Image Analysis and Signal Processing (IASP), 2010 International Conference on, pp. 191–194 (2010)
24. Luo, J., Singhal, A.: A bayesian network-based framework for semantic image understanding. Pattern Recogn. **38**(6), 919–934 (2005). image Understanding for Photographs
25. Boutell, M., Luo, J.: Bayesian fusion of camera metadata cues in semantic scene classification. Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the IEEE Computer Society Conference on, vol. 2, June 2004, pp. II–623–II–630 Vol. 2 (2004)
26. Sebe, N., Cohen, I., Huang, T., Gevers, T.: Skin detection: a bayesian network approach. Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, vol. 2, vol. 2, pp. 903–906 (2004)
27. Koller, D., Friedman, N.: Probabilistic graphical models: principles and techniques. MIT press (2009)