Unmanned Ground Operations using Semantic Image Segmentation through a Bayesian Network

Matthew Coombes, Will Eaton, Wen-Hua Chen
Department of Automotive and Aeronautical Engineering
Loughborough University, Loughborough, LE11 3TQ UK
Email: {M.J.Coombes@lboro.ac.uk, W.Eaton@lboro.ac.uk ,W.Chen@lboro.ac.uk}

***Abstract*—This paper discusses the machine vision element of a system designed to allow automated taxiing for Unmanned Aerial System (UAS) around civil aerodromes. The purpose of the computer vision system is to provide direct sensor data which can be used to validate vehicle position, in addition to detect potential collision risks. This is achieved through the use of a singular monocular sensor. Untrained clustering is used to segment the visual feed before descriptors of each cluster (primarily colour and texture) are then used to estimate the class. As the competency of each individual estimate can vary based 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 map matching.**

Keywords*—*Unmanned Ground Operations; Semantic Image Segmentation; Bayesian Network; Domain Knowledge

I. Introduction

Over the last few decades, extensive military development has significantly improved the capabilities of unmanned aircraft. Many roles that once required a manned aircraft are now primarily performed by UAS. As UAS are becoming increasingly mature, many potential applications outside of 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.

UAS are extremely similar to manned aircraft and have many of the same requirements. As it would be both impractical and expensive to construct new ground facilities solely for unmanned aircraft, this work is based on the prediction that future civil UAS will operate from existing aerodromes alongside conventional manned aircraft. The 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].

The primary motivation of this work is to ensure that the introduction of UAS includes safe ground operations. Some requirements for automated taxiing, such as path planning or communications, are already mature areas of research and existing algorithms can be implemented [2]. However, other elements will require dedicated research. The most critical areas are those which directly relate to the safety of the UAS and other aerodrome users. To ensure 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.

This work has been undertaken in conjunction with BAE systems who have provided the practical test data used for testing and validation. As a result, the system proposed in this paper is constrained by the availability of sensors on the test-platform. Matching the current BAE UAS hardware, this work assumes that direct sensing capabilities are limited to a single forward facing monocular camera. As such, a machine vision approach is the only feasible method of direct sensing.

There has been previous work in the area of semantic image segmentation in papers such as [3], where an image is segmented into the different classes of objects in the scene for the purposes of image labelling, object detection or scene classification.

The basis of this work is a continuation from the initial research conducted in [4], this paper is an extension to the work from another paper submitted to this conference [5]. [4] lays out an approach to perform image segmentation in order to perform semantic classification on segmented clusters. This is opposed to the more traditional methods such as in [6] where segmentation and classification are performed at the same time in what is called classifier-led-segmentation. Precise region extraction is often difficult when using classifier-led-segmentation. Inaccurate region borders are partially introduced through the use of feature descriptors. Although colour data is stored in each pixel, more complex data, such as texture, requires data from multiple pixels. As both colour, and texture information is to be used, pre-segmentation enables the extraction of the texture data easily from within the borders of the cluster. Accurate region boundaries are required for collision avoidance and navigation, as accurate ranging needs to be extracted to navigation or collision features.

[5] proposes a method that uses Hue-Saturation-Value (HSV) colour, and luminance data from each segmented cluster in a trained Bayesian Network (BN) to classify each cluster. Colour is used as the main classifier, while luminance is used to detect surface line markings. These are fused together probabilistically in a BN where the surface line detection is shown to improve the already good colour classification. Due to the intuitive graph structure of BN they are easily extended, in this paper we aim to extend the BN to include other sources of information to improve classification performance further. We aim to add the texture classifiers from [4] as an extra main classification source, as well as horizon intercept to add further domain knowledge logic to remove misclassifications. The emphasis in this paper is the data fusion potential of the BN, where [5] only briefly introduces this, and concentrates on the colour classification and line detection aspects of the method.

As aerodromes are strictly controlled, they are quite sterile environments. This means there are a low number of unique classes to classify. This can be used to our advantage; if there is an object that is not recognised it will be identified as a collision risk. This is different to the standard methodology where identification of all objects is attempted, which is of course much more complex. The aim of this work is to robustly classify a smaller number of classes. Subsequently by using the probabilistic confidence of that classification from the BN the unknown collision risks can be identified. This is one of the main extensions to [5]. The remainder of this paper is organised as follows; section II discusses image segmentation, and how untrained segmentation is performed using methods explored in a previous work [4]. Section III 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 IV introduces the BN data fusion method proposed in this paper, which is followed by a brief introduction to BNs in Section V. Section VI gives details on the final BN used for classification, including the network's structure, inputs and discretisation. Finally, section VII 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.

II. 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, the requirements of this work differ from most other applications. Any unrecognisable regions within an image most likely represent an object that could be collided with. Therefore, accurately segmenting 'unknown' regions within an image is just as important as defining known classes. When an image is segmented using classifier data, there is a tendency to bias results towards the known classes. There are many reasons for this, but the simplest is the size of the feature descriptors used. To extract complex data, such as texture, data must be taken from multiple pixels. If sample data is taken on the boundary between different regions in an image, it draws data from both. If one is unknown, there may still be enough data from the known class to get a match. This can result in inaccurate region borders and small regions becoming absorbed into larger neighbours, reducing the ability to detect unknown collision risks. 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'. As image segmentation techniques existed long before semantic segmentation was possible, there are many segmentation methods available. This work continues to use the 'superpixel' based approach, outlined in [4]. Superpixels are small clusters of pixels, grouped together based on their colour and spatial distance. Within a superpixel, all pixels are very similar to each other, allowing the mean data to represent all the pixels within that superpixel with minimum data loss. As every pixel in the original image is grouped into a superpixel, an image with millions of pixels can be reduced to a meaningful representation of only a few hundred superpixels, dramatically decreasing the computational burden for many applications. As superpixels will not group dissimilar pixels, the edges in the original image are captured by the borders between superpixels, producing highly accurate region borders.

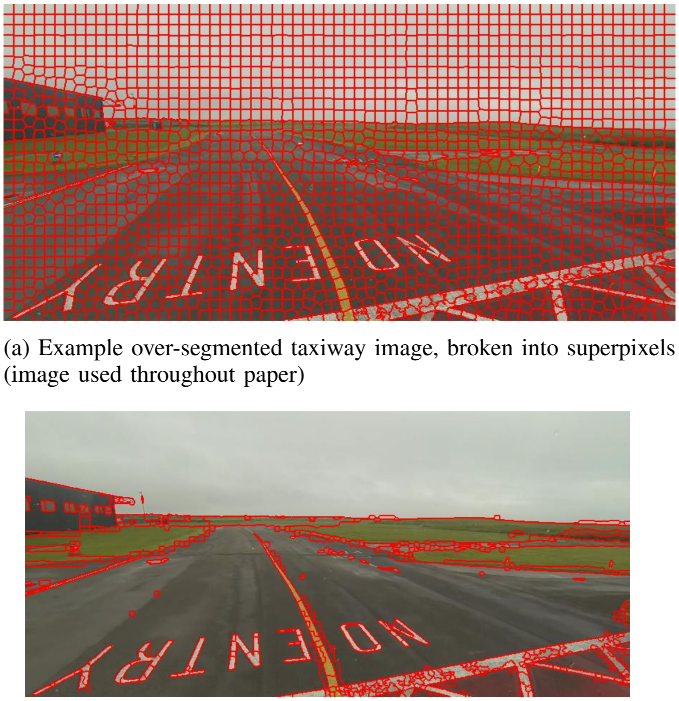
Both [4] and this work use Simple Linear Iterative Clustering (SLIC) to generate the initial superpixel clustering. [7]. The algorithm is computationally efficient, with a standard desktop computer providing sufficient computational power for SLIC to process high resolution images in real time [8]. Fig. 1a is an example of superpixel segmentation achieved using 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. Rather than directly classify each superpixel, the over-segmentation is resolved by a second application of clustering, grouping superpixels into larger, visually similar regions. This reduces the risk of misclassification as each region has more information for the classifier to use. This secondary clustering is achieved using the method outlined in [9], where it is suggested that the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm is a good solution for merging superpixels. DBSCAN is a method of clustering that uses density reachability, so that dissimilar superpixels can be merged if connected neighbours between them are similar enough to form a path. This is essential to overcome the effects of distance on large regions such as taxiways, which introduce colour and texture gradients proportional to physical distance.

The metric used to compare superpixels is colour distance within the CIE *L*a*b** (CIELAB) colourspace. As the original superpixel boundaries formed by SLIC are preserved during DBSCAN clustering, the final result retains the sharp resolution required for 3D estimation. Shown in Fig. 1b is an example of DBSCAN clustering of the super pixels form Fig. 1a. By varying the threshold used by DBSCAN based on empirically inspected results, each output cluster should ideally represent a single object or material type. This makes the scene easier to interpret, as all data within each cluster can be used during classification.

III. Texture based Classification

The intention of this work is to pass data about each cluster to the BN for classification. For simplistic data, such as colour, the raw data can be incorporated into the BN directly. For other data types, such as texture, this is more difficult. Texture is inherently hard to define. Unlike colour,



(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

which is simply stored within each pixel, texture is the way colour changes over multiple pixels. The number of pixels, their relationship to each other and the method of defining colour variance can all vary. Therefore, 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. As descriptors have strengths and weaknesses, two descriptors have been used; a traditional Gabor filter approach and a Local Binary Pattern (LBP) feature extractor.

A. Texture Extraction

The Gabor filter approach specifically makes use of the Maximum Response Filters (MR8) filter bank [10], designed for texture classification. To extract texture data, each of the 38 filters within the filter bank are individually applied to the original image, producing 38 'filter response' images. Stacking these responses similar to how colour channels are stacked in colour images, each pixel in the original image is replaced by a 38-element vector, representing the texture response at that point.

Assuming the response is stored in the same format as the original image, this allows for 255^{38} possible responses. As this represents an enormous amount of variability, K-means clustering is used to simplify the results. Using 120 cluster centres based on typical feature responses for all image types, commonly referred to as 'textons' [11], the texture data is reduced down into a single indexed response image. The texture data for each cluster is a vector of textons, replacing the original pixels. As the filters within the MR8 filter bank are large, they gather much data for each pixel response, providing good results over large areas. However, large filter descriptors can capture results from multiple clusters when clusters are small, reducing the accuracy of small cluster identification.

For this reason, the secondary feature descriptor used is LBP. This compares each pixel to it's immediate neighbours, providing a simple numeric response based on which neighbours have greater intensity levels. Despite it's simplicity, it is widely used and has proven especially capable at small scale texture classification. In addition, as LBP is based on relative intensity, a total brightness changes will shift the intensity of all pixels together making the descriptor light level invariant. In contrast to the MR8 filter bank results, the very small sample size of LBP makes it highly localised, providing poor results on texture types that consist of large numbers of pixels.

B. Texture Comparison

As the output from both MR8 and LBP consists of discrete responses, the texture data can be represented by the total number of each response within that cluster. The limited amount of possible responses allows histographic methods to be used for direct comparison to other known examples of each class. Several methods of comparing histograms have been investigated, including simple distance based estimators such as nearest neighbour, K-nearest-neighbour and mean 10-nearest neighbour. In each case, multiple distance functions have also been used, including Euclidean, L1 and X^2. Better results were obtained through the use of a Support Vector Machine (SVM) based classification approach. As SVM is a binary classifier, a Binary Decision Tree (BDT) structure was implemented for fast classification [12].

The final result is simply the most likely class for each cluster, based on texture information only. This has been found to produce a fairly accurate texture only classification, for both MR8 and LBP feature descriptors. However, neither result is perfect, improvement is still possible. Therefore, this data is then passed to the BN for data fusion with other types of information.

IV. Classification through Data Fusion

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. In order to provide the best result, a two stage classification process is used.

The first stage is to extract individual 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.

The second stage is combining the results from multiple feature types together in a meaningful way, to form a final estimation.

This work aims to improve the classification result by using a Bayesian Network (BN) based approach. Rather than discrete information, a probabilistic approach is used, allowing 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 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, loosing 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.

V. 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. For example, a BN could be used to model the relationship between weather patterns and crop yields. Given the crop yields for a year, the BN could be used to estimate the probable weather, and more usefully, vice versa.

The use of BN is not a new concept in image classification. For example [13] presents a BN framework for combining low level features to detect the most significant object within an image. Another example is [14], 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.

A. Components

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.

A DAG represents the structure of the BN, formed of nodes and directed edges which connect one node to another. As the name suggests, the graph must be constructed without cycles; i.e. later nodes cannot provide feedback. Each node can represent various things (such as individual variables, observable data or hypotheses) while the edges represent the conditional relationships between the nodes.

A major advantage of BN is the ability to represent the entire structure within an intuitive graphical model. The input criteria (also known as evidence) are variables which can be directly observed and are used to create the starting nodes of the network. The other variables within the network are then inferred from the observed variables, based on the conditional dependencies. By using inference techniques, the network can effectively be asked questions about the probability of something happening, given evidence regarding a number of causal factors.

If the structure of the network is known as well as it's CPDs, a full joint probability distribution can be calculated. A joint probability distribution is the probability of the occurrence of every possible combination of states, of all the random variables. This can be calculated using the chain rule shown below

$$P(X_1 ...., X_n) = \prod_{j=1}^{n} P(X_j | parents(X_j)) \tag{1}$$

where X_j represents a variable, and n is the total number of variables.

The full joint probability distribution is quite complex, becoming difficult to manage for larger networks. Instead, more useful information is the total probability of each variable. This is calculated through 'marginalisation', which is the process of determining a probability distribution of a subset of variables from the larger joint probability distribution. To remove (marginalise) unwanted variables, the probabilities for each combination of these variables are summed together.

$$P(X_j) = \sum_i P(X_j | parents(X_j)) P(parents(X_j)) \tag{2}$$

Given an extremely large BN, information from the original observed variables can be passed through many layers to produce a final useful output. This ability to 'infer' data from one node to another is the main feature of Bayesian networks. Although other graphical models use inference to produce estimations, this is commonly done in the form of 'causal reasoning' (top to bottom). For example, given an observable disease, likely symptoms can be estimated. BN are capable of 'diagnostic reasoning' (bottom to top) which uses probability to asses the probable cause. For example, given observable symptoms, a BN could be used to estimate the disease. The ability to use probabilistic inference is a result of using Bayes Rule shown in Eq. (3), where $P(X_2|X_1)$ is the likelihood which will be obtained from the CPD of X_2, $P(X_1)$ is the prior, and $P(X_2)$ is the marginal likelihood, which is used to normalise the probability.

$$P(X_1|X_2) = \frac{P(X_2|X_1)P(X_1)}{P(X_2)} \tag{3}$$

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	$Dist$

Directly using these equations for conducting inference will only work on a small network where the joint distribution is small. A full network can have many more nodes, each with multiple states. For example, a network with 15 nodes and 4 states per node will have 4^{15} entries in the joint distribution, making direct application difficult.

Methods such as belief propagation [15] or junction trees [16] are used to speed up calculating an exact solution, therefore brute force (direct marginalisation of the joint distribution) is not required. For an approximate solution for extremely large Bayesian networks, various statistical sampling techniques can be used; such as Markov chain Monte Carlo sampling.

VI. Bayesian Network structure

The DAG for the proposed BN is shown in the lower section of Fig. 2. There are four distinctive sub networks, which include texture classification (Section VI-A), horizon intercept (Section VI-C), colour classification and line detection. The colour classification and line detection laid out and discussed in detail in [5]. 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. For the texture classifiers, this is simply the winning class for each cluster. However, $Dist$, NRL, Hoz, H, S and 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

$$c_i = \arg\max_{Class_i} P(Class_i | H_i, S_i, V_i, MR8_i, LBP_i, Hoz_i, Dist_i, NRL_i) \tag{4}$$

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.

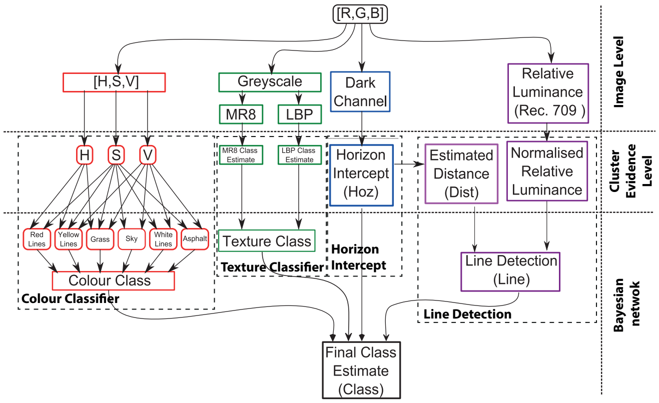


Fig. 2: Whole system showing the processing done on raw RBG image, and displaying the Bayesian Network structure

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 too many too 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.
- Clusters which have a $True$ Line state 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.

As this BN is not overly complex, 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 [17]. The application of MLE to Bayesian networks is explained in detail in [18].

A. Texture

The two texture classifiers that use the MR8 and LBP descriptors will be combined into a single texture classifier node using the simple BN structure seen in Fig. 2. In order to simplify the network and enabling subsections of the

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

Description of CPD entry	Colour Class	Tex	Hoz	Line	Asphalt	Grass	Sky	White	Yellow	Red	Building
Ground non-line cluster classifiers in agreement/non-agreement	Asphalt	Asphalt	Below	F	1	0	0	0	0	0	0
	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

network to be assessed and trained separately. Each node has seven states, which represent each of the seven different classes.

As the number of pixels used in the descriptors are different their classification performance will differ between classes, it has been observed that LBP is better at classifying lines, whereas MR8 performs better on classes such as asphalt and grass. This makes sense as LBP descriptors are smaller, and the clusters for lines are smaller. The larger MR8 descriptors will make classification on the larger clusters for grass and asphalt more accurate.

This correlation could be manually included in the CPD of the combined texture class node Tex by making the probability of the classification for line classes greater for LBP classifications of line classes and similar for MR8 with classifications of asphalt and grass. However these differences in performance can be captured more accurately by performing parameter learning by using training data to learn the CPD $P(Tex|MR8, LBP)$ using MLE from a training set of 100 training images.

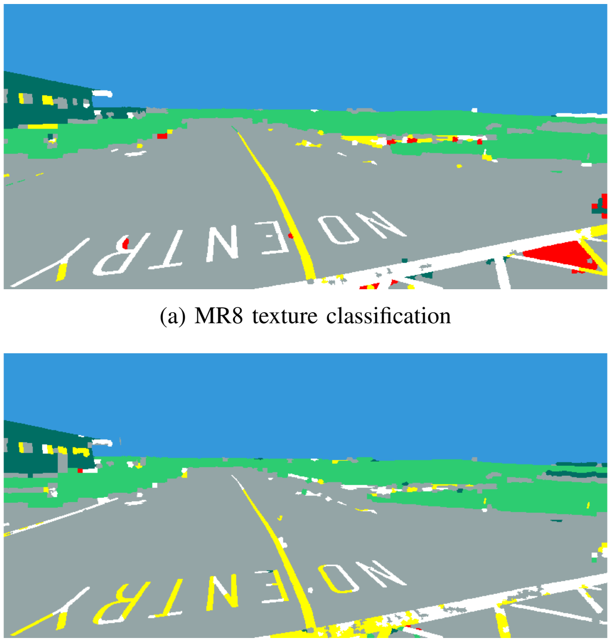
The texture classifiers have a winning class for each cluster, these discrete classes are entered into the $MR8$ and LBP nodes as evidence.

Shown in Fig. 3 are the LBP and MR8 classifications on the example image. The combined texture classification image where the winning Tex class for each cluster is determined by $\arg\max_{Tex} P(Tex|MR8_i, LBP_i)$ is shown in Fig. 4.

B. Bayesian Colour Classification

The subsection of the BN that handles the cluster classification based on colour is shown in Fig. 2. These six classes are represented by individual nodes which all only have two discrete states: true (T) and false (F). They are split like this as not all these classes are conditioned on all H, S and V.

Each of the colour channels is discretised into bands. Hue is discretised into 24 discrete states for 0 - 360° in 15° increments, saturation and value both have 10 states from 0 - 1 in increments of 0.1. This produces a colourspace with 2161 discrete colours in total, which is the reason this CPD is trained as this would not be easy for an expert to fill out. This subsection is detailed in full in [5].


(a) MR8 texture classification

(b) LBP texture classification

Fig. 3: Example of an aerodrome taxiway image, texture classifications

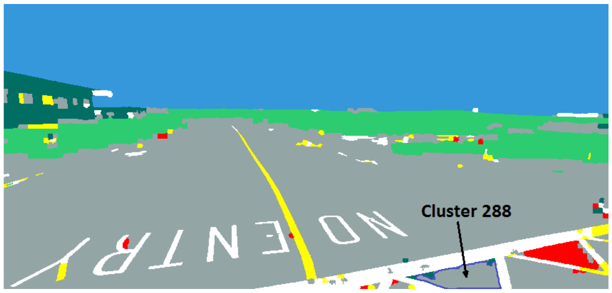


Fig. 4: Combined texture classification on example image

C. Relative Horizon Position

Clusters which are wholly below the horizon line can be considered on the ground, and further processing is required to identify whether they pose a collision risk. 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 sky. 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. Therefore, horizon intercept represents a simple method of detecting collision risks. As only the position of the cluster within the image is used, it requires minimum processing.

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 [19], 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 UAS is facing a large object (such as a building) and therefore a visually derived horizon line will not be accurate.

The three states for horizon intercept are listed in Table 3. 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.

TABLE 3: Horizon intersect discrete states

Normalised Relative Luminance (NRL) states
Above
Below
Above/Below

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

If Hoz state is $Above$ the probability of that cluster being any class other than Sky is drastically reduced. This stops false classifications of sky on the ground. If Hoz state is $Below$ the probability of that cluster being Sky or $Building$ is reduced. The probability of a cluster being $Building$ if Hoz state is $Above/Below$ is increased, whereas all non collision risk classes probability are decreased. This will have the huge benefit when a potential collision risk's clusters are $Above/Below$ and as it will be misclassified it will have a much lower confidence in this misclassification making much more likely to be flagged as a potential collision risk.

Using the example image, the horizon line is calculated and the clusters assigned horizon intercept states, which is shown in Fig. 5. 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.

D. Normalised Relative Luminance Line Detection

NRL and distance to cluster will be combined in the BN to give a probability that a cluster is a white or yellow line, with the variable $Line$ representing this probability. $Line$ has two states true (T), and false (F). This subsection is also detailed in full in [5].

E. Unknown classes from uncertainty

Not all potential objects that could appear in an aerodrome environment can possibly be classified reliably. The advantage of using this probabilistic method means that objects which do not have a trained class will get misclassified with a low certainty of being that misclassified 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.

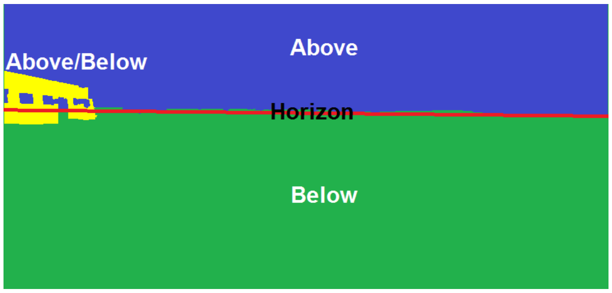
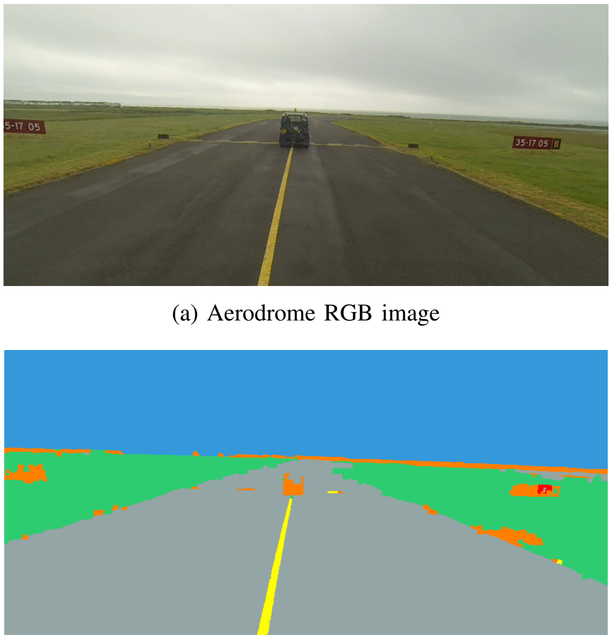


Fig. 5: Example image cluster horizon intersect states



(a) Aerodrome RGB image

(b) Example aerodrome image with vehicle obstacle

Fig. 6: Example aerodrome image with vehicle obstacle

$$\begin{cases} c_i = \text{unknown} & \text{for} \quad \max_{Class_i} P(Class_i) < U \\ c_i \neq \text{unknown} & \text{for} \quad otherwise \end{cases} \tag{5}$$

where cluster (c_i) is in the state $unknown$ which is an unknown collision risk, and U is the probability threshold

An example image (Fig. 6a) has classification performed on it using the full Bayesian network, where U is 0.5 and the clusters in orange are classified as unknown. This processed image is shown in Fig. 6b

It can be seen that a significant portion of the vehicle is 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

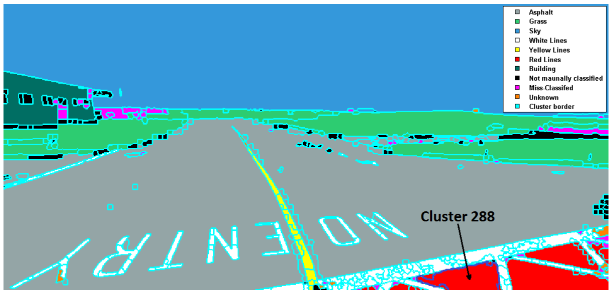


Fig. 7: final classification of test image showing segmentation, misclassifications and unknown clusters

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, also due to bad clustering. As this is not a navigable surface this is not of any consequence, and could be solved by a much larger training set for both the colour and texture classifiers.

VII. Results

Footage from a monocular camera was gathered from a vehicle that was driven around an aerodrome in UK. This data was used to assess the proposed algorithms. The chosen scenario provides a visually challenging 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.

For illustration all clusters in the example image used previously in Section VI-B have been classified using the full BN classification method and displayed in Fig. 7. This single image shows the results to be good with the misclassified clusters shown in Purple.

To demonstrate the algorithm a particular cluster is chosen from the example image seen throughout this paper, and its classification is discussed. Then the overall classification results are presented from the BN on a 20 image test set of fully manually classified aerodrome images. The output is assessed for total classification accuracy and the accuracy of individual classes. Then is compared to the previous texture only methods.

A. Example Cluster

The cluster labelled 'Cluster 288' shown in both Fig. 4 and Fig. 7 has been misclassified as asphalt in both MR8, LBP and the combined texture classifier. Shown in Table 4 is the posterior marginal distributions for the $ColourClass$, $TextureClass$, and the final $Class$ nodes. On this aerodrome the red surface markings are very old and worn so have a very similar texture to asphalt which is why it has been misclassified as this with a probability of 1. However the colour for this cluster is very distinct and it is very obviously red, which is why the colour classifier has estimated it to be 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, when these are both combined the winning class is Red. Seen in the marginal for $Class$ red has a probability of 0.6714, and asphalt is 0.3286. As this is a ground cluster i.e. a Hoz state of $Below$, there is 0 probability of it being the Sky class. It has marginal $Line$ probability for the state F of 1 which means that is has a very 0 probability of being a yellow or white line.

TABLE 4: Marginal posterior distribution for $Class$, $ColourClass$ and Tex for cluster 288

Class	Colour Class	Tex	Class
Asphalt	0.0408	1	0.3286
Grass	0	0	0
Sky	0	0	0
White Line	0	0	0
Yellow Line	0	0	0
Red Line	0.9592	0	0.6714
Building	N/A	0	0

B. Classification Accuracy

There are effectively two different types of error: misclassification and segmentation error. Segmentation error is when a cluster includes 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 cluster in Fig 7. Misclassification is obviously when the classifier in question incorrectly classifies a cluster. These are shown in purple, in Fig. 7. Depending on the particular error this can be dangerous, for example an obstacle right on an aircraft's taxiing path classified as a navigable surface. However most are inconsequential noise, for example around the edge of the grass taxiway transition where some of the clusters could be misclassified as either grass or asphalt due to the segmentation error just discussed. There are two ways to look at misclassification percentages; per cluster or per pixel. Per cluster is incorrectly classified clusters out of the total clusters in the image. Per pixel is incorrectly classified pixels out of the total pixels in the image.

The previous texture classification methods have been shown to have reasonable performance but don't give adequate performance to be used robustly for navigation or obstacle detection. For the test set, the average percentage error for the classification of each pixel and cluster for the BN and both previous texture classifiers are shown in Table 5. This shows a significant performance increase over texture only with about 5% better pixel classification, but more significantly about a 35.4% improvement in percentage of incorrect clusters.

There is a very large difference between the pixel and cluster error for texture which shows that texture does a much better job of classifying larger clusters than small ones. This is clearly shown in Fig. 3.

It is also useful to compare the classification for individual classes. The breakdown for the BN, LBP and MR8 classifiers are shown in Tables. 8, 6 and 7 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.

TABLE 5: Percentage error for Bayesian network classifier compared to texture only classifiers of test set

Classifier	Mean% pixel misclassification	Mean% cluster misclassification
BN	1.48%	19.82%
LBP	6.29%	52.81%
MR8	5.12%	57.67%

TABLE 6: 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	1.8	0.3	0	0.03
Sky	0.4	0.6	98.9	0.01	0.01	0	0.1
White	4.2	1.7	0	86.5	7.6	0	0.02
Yellow	5.4	1.7	0	17.0	76.0	0	0
Red	86.5	11.2	0	2.0	0.1	0.1	0.1
Building	2.2	0.08	0	0.3	0.4	0	97.1

As mentioned before red surface marking classification is very poor for both MR8 and LBP meaning that the texture is nearly identical between these two classes. It can be seen that MR8 has misclassified red paint pixels as asphalt 86.5% and LBP 86.4% of the time. Both have poor performance for yellow and white lines, due to the occasional very small cluster size. As the smooth painted line texture is similar for both classes, LBP misclassified yellow lines as white line 16% of the time, and for MR8 white line were misclassified as yellow lines for 19% of the pixels. Looking at Table 8 it can be seen by including colour, and line probability in the proposed probabilistic way, this is dramatically improved. Increasing correct classification for white, yellow and red surface markings to 92.3%, 93.3% and 97.0%. Red lines where still misclassified as asphalt for 2.9% of the pixels, this is due to line probability not including red markings as they don't have high NRL values.

Due to line probability being included this has removed many of the misclassifications of non-line classes as lines. For example grass is incorrectly classified as a white line for 1.784% and 0.971% for LBP and MR8 respectively. However for the BN this gets lowered to a tiny 0.01%. This will removed a large proportion of the frame to frame noise. 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.

The Texture only classifiers distinguishes between asphalt and other classes well, with MR8 performing better at 98.9% compared to 95.5% for LBP. This due to asphalt's consistent and mostly uniform texture. As the colour classifiers only use saturation and value to classify asphalt, and texture only is quite good anyway, combining them together in the BN has yielded a minor decrease in performance of 0.02% compared to the MR8. This is because it is combined with LBP which performs more poorly for asphalt. However this combination has shown to make the classifier more robust, which makes a decrease of 0.02% totally insignificant.

TABLE 7: Percentage breakdown of MR8 Texture only classier of test set

Man \ Auto	Asphalt	Grass	Sky	White	Yellow	Red	Building
Asphalt	99.0	0.5	0	0.2	0.3	.01	.03
Grass	10.1	87.9	0	1.0	0.9	0	0.2
Sky	0	0	98.9	1.1	0	0	0
White	3.5	.03	0	77.3	19.2	0	0
Yellow	2.5	0	0	2.1	95.3	0	0
Red	86.5	0	0	0.8	0.9	9.4	2.4
Building	2.2	0	0	0	0.2	0	97.6

Looking at grass, its texture tends to be more inconsistent, especially around the edges. 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 96.15%. Grass has a distinct colour (which is why hue is included for its classification) so data fusion with the colour classifier has given improved results.

For a huge variety of buildings texture would give much worse results and the BN would increase this performance by using the horizon information. In the training and test set of aerodrome images, there are very few building, and the ones present are very similar, this has led to over training. This can be seen by the very high classification performance results of the texture only classifiers. The BN classifier has lost about 1% performance compared to these due to this over training. There where a few segmentation errors which caused the Hoz state of a few small clusters which are a part of a building to not be $Above/Below$ which has lead to the drop in performance. An example of this can be seen in Fig. 5, which lead to the misclassification of that cluster shown in Fig. 7. However if a number of different building types where present, the texture performance would drop and the robustness added by including horizon information would have the BN performing much better. Sky is a large cluster so has the best correct classification at around 98.9% for both MR8 and LBP. However some sky clusters have still been misclassified as asphalt, grass, white, and yellow lines for LBP and just white line for MR8. From Table 8, these incorrect sky classifications have been completely removed by applying horizon logic, apart from the inconsequential 0.03% of sky pixels being classified as asphalt, which are due to a few automatic segmentation errors.

Automating segmentation has been shown to be quite effective; however as discussed can fail and cover two or more classes. As can be seen from Fig. 8 asphalt and grass are the most common classes to be manually classified as unknown at 53.5% grass and 37%. this is because they are the two most common classes to be on boundaries adjacent with one other. These segmentation errors account for an average of 2.5% of the pixels across the whole test set. While this sounds high, as shown it mostly only affects grass and asphalt this will not cause a dangerous misclassification just slight noise around major class boundaries.

VIII. 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 autonomous taxiing of UAS at non-segregated aerodromes. The accurately segmented and classified images will enable both navigation and collision avoidance.

A probabilistic BN framework was used for fusing multiple sources of information with domain knowledge. This

TABLE 8: Percentage breakdown of Bayesian network classier of test set

Man \ Auto	Asphalt	Grass	Sky	White	Yellow	Red	Building	Unknown
Asphalt	98.9	0.97	0.00	0.04	0.07	0.00	0.00	0.03
Grass	3.83	96.2	0.00	0.01	0.00	0.00	0.00	0.01
Sky	0.30	0.53	99.2	0.00	0.00	0.00	0.00	0.01
White	1.40	0.00	0.01	92.3	1.32	0.23	0.00	4.78
Yellow	0.36	1.22	0.00	2.05	93.3	0.00	0.00	3.04
Red	2.93	0.04	0.00	0.04	0.00	97.0	0.00	0.01
Building	3.61	0.00	0.00	0.00	0.00	0.00	96.4	0.00
Unknown	62.60	30.34	3.21	0.97	0.51	0.57	0.00	1.80

method has shown to give improved classification performance compared to the individual classifiers by 5% per pixel and by a large 35% per cluster.

The use of a BN also has the advantage of giving an intuitive graph structure for extending the network to include other sources of information such as adjacency. It 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. Whereas other deterministic classification methods such as neural networks, would not be appropriate.

By using a BN it has been shown that unknown collision risks can be determined. This is achieved by the BN giving a certainty on each of its cluster classifications. This is important as classification of every type of collision risk is not possible.

A fully processed and classified video can be found on our YouTube channel [1]

There a number of extensions that could be performed on this work. The texture classifier could provide certainty in its classification from pseudo probabilities derived from the distance metrics. This could be used by entering this as soft evidence, as less information would be lost this would increase performance.

Acknowledgment

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] A. M. Lekkas, "Guidance and path-planning systems for autonomous vehicles," 2014.

[3] J. Yao, S. Fidler, and R. Urtasun, "Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, June 2012, pp. 702–709.

[4] W. Eaton. and W.-H. Chen, "Image segmentation for automated taxiing of unmanned aircraft," in *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*, June 2015, pp. 1–8.

[5] W.-H. C. Matthew Coombes, William Eaton, "Colour based semantic image segmentation and classification for unmanned ground operations," in *Submitted to International Conference on Unmanned Aircraft Systems 2016*, 2016.

[6] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context," *Int. Journal of Computer Vision (IJCV)*, January 2009.

[7] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Ssstrunk, "SLIC Superpixels," EPFL, Tech. Rep., 2010.

[8] C. Y. Ren and I. Reid, "gslic: a real-time implementation of slic superpixel segmentation," *University of Oxford, Department of Engineering, Technical Report*, 2011.

[9] P. D. Kovesi, "MATLAB and Octave functions for computer vision and image processing," Centre for Exploration Targeting, School of Earth and Environment, The University of Western Australia, available from: <http://www.csse.uwa.edu.au/∼pk/research/matlabfns/>.

[10] M. Varma and A. Zisserman, "Texture classification: Are filter banks necessary?" in *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

[11] B. Julesz, "Textons, the elements of texture perception, and their interactions," *Nature*, vol. 290, no. 5802, pp. 91–97, 1981.

[12] G. Madzarov and D. Gjorgjevikj, "Multi-class classification using support vector machines in decision tree architecture," in *EUROCON 2009, EUROCON '09. IEEE*, May 2009, pp. 288–295.

[13] J. Luo, A. E. Savakis, and A. Singhal, "A bayesian network-based framework for semantic image understanding," *Pattern Recognition*, vol. 38, no. 6, pp. 919 – 934, 2005, image Understanding for Photographs.

[14] M. Boutell and J. Luo, "Bayesian fusion of camera metadata cues in semantic scene classification," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2, June 2004, pp. II–623–II–630 Vol.2.

[15] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, 1988.

[16] R. G. Cowell, P. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter, *Probabilistic networks and expert systems: Exact computational methods for Bayesian networks*. Springer, 2007.

[17] N. Sebe, I. Cohen, T. Huang, and T. Gevers, "Skin detection: a bayesian network approach," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 2, Aug 2004, pp. 903–906 Vol.2.

[18] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[19] H.-Z. Yuan, X.-Q. Zhang, and Z.-L. Feng, "Horizon detection in foggy aerial image," in *Image Analysis and Signal Processing (IASP), 2010 International Conference on*, April 2010, pp. 191–194.

[1] https://www.youtube.com/channel/UCd$_{U}m7LY4xSoIc53j_{7}Uy1A$