

Image Segmentation for Automated Taxiing of Unmanned Aircraft

William Eaton

Department of Aeronautical and Automotive Engineering
Loughborough University
Leicestershire, UK
Email: w.h.eaton@lboro.ac.uk

Wen-Hua Chen

Department of Aeronautical and Automotive Engineering
Loughborough University
Leicestershire, UK
Email: w.chen@lboro.ac.uk

Abstract—This paper details a method of detecting collision risks for Unmanned Aircraft during taxiing. Using images captured from an on-board camera, semantic segmentation can be used to identify surface types and detect potential collisions. A review of classifier lead segmentation concludes that texture feature descriptors lack the pixel level accuracy required for collision avoidance. Instead, segmentation prior to classification is suggested as a better method for accurate region border extraction. This is achieved through an initial over-segmentation using the established SLIC superpixel technique with further untrained clustering using DBSCAN algorithm. Known classes are used to train a classifier through construction of a texon dictionary and models of texon content typical to each class. The paper demonstrates the application of said system to real world images, and shows good automated segment identification. Remaining issues are identified and contextual information is suggested as a method of resolving them going forward.

I. INTRODUCTION

A. Background and motivation

Bringing unmanned aircraft into civil airspace has been a key research area for many years. However, less focus has been given to how the aircraft will operate while on the ground. If the intention is for manned and unmanned aircraft to share the skies, it is likely they will also share ground facilities. Currently, there is no form of automated taxiing and pilots are responsible for all control while on the ground. In order for Unmanned Aircraft Systems (UAS) to operate alongside safety, this capability must be replicated. There are many requirements in order to produce an automated taxiing system, with some easier to implement than others. As all UAS require a complete avionics suite to fly, it is assumed that global positioning and computational hardware is already available. Therefore localisation of the aircraft and navigation around static obstacles can be easily achieved using accurate aerodrome maps. Of greater concern is detecting variable collision risks, such as other aircraft or ground vehicles.

As standard avionics do not include short range sensing capabilities, an additional sensor is required. Although active sensors (such as LIDAR) are being introduced for similar purposes in self-driving cars, it is unlikely that they will see widespread adoption for UAS, as allocating additional weight solely for ground operations will likely only be

possible on the largest of unmanned aircraft. Instead, this paper looks to achieve ground based sensing by using a single forward-facing monocular-camera; a piece of equipment already commonly found on unmanned aircraft.

B. Proposed System

Within an aerodrome environment, the size of moving obstacles can vary dramatically; ranging from small Foreign-Object-Debris (FOD) to large airliners. At the same time, these obstacles might be alone on an empty runway, or surrounded by other objects on a busy apron. Assuming that the UAS is limited to the same data currently available to human pilots, the size, type and location of all mobile risks start as unknowns. Therefore, it is impractical to attempt to detect risks on an individual basis (such as seeking any potential objects shaped specifically like an aircraft). Instead, the image captured by the forward facing camera must be interpreted in its entirety. The intention is to divide the image into visually distinct parts, through a process of image segmentation. As the smaller segments should ideally represent a single object or material type, the scene will be easier to interpret, with each region being labelled as the most likely class based on its contents. Where identification is not possible or confidence in identification is low, the segment is to be considered a collision risk by default. Once collision risks have been found within an image, they can be extrapolated from the image plane into real-world coordinates. As most UAS have fixed height landing gear, this is a straightforward exercise in trigonometry, using the height of the camera above the ground and its Field-of-View (FOV) to calculate range and bearing estimates. As a small variation in angle can result in a large error in range estimation, accurate segment edges are key.

II. IMAGE CLASSIFICATION

In order to produce a semantic breakdown of the image, elements will need to be classified through comparison to pre-defined models of expected object classes. These models are constructed prior to the classification attempt, using images manually selected as examples of each class. In order form a model, visual descriptors are required. The most basic of descriptors is the raw data that makes up the image, i.e. the colour data for each pixel. For regions, the information could be as simple as the average colour

data, or expressed in terms the colour distribution, using a histogram in either one or multiple colour channels. However, regardless of the colour space used, colour alone does not make a good descriptor. Multiple classes of object can share the same colour composition whilst still appearing visual different, and conversely the colour of a single object class can vary immensely under varying lighting conditions (outdoor environments are sensitive to both time of day and seasonal effects).

With artificial objects, such as cars or aircraft, the colour of an object is often unrelated to what it is. However, the texture is usually indicative of the surface material, making identification easier. Materials with similar colours can often be distinguished by their texture, and unlike colour, the texture of a surface varies less with changes in illumination.

A. Texture based Classification

Paper [1] provides a comparison of common texture descriptors for three-dimensional object classification. It concludes that different descriptors outperform each other on different texture types, and at the time of writing there is no single descriptor that is universally better than others. As texture is essentially a pattern recognition task, certain texture descriptors are limited to work only when the texture is orientated correctly. As the UAS moves about an aerodrome, it can come across objects from any angle and as such the required descriptors must be invariant to scale and rotation as well as lighting conditions. As finding an optimal descriptor for outdoor environment is not part of this work, a method which has already been applied to outdoor semantic recognition was selected instead, namely Textons.

Textons, originally proposed in [2] and explained in great detail in [3], are visual descriptors specifically designed to describe the appearance of three-dimensional surfaces. To accomplish this, they work with image elements that are mostly invariant to differing lighting conditions and camera orientations, while also benefiting from simple implementation. The implementation of a texton-based classifier is dealt with later in the methodology section.

A drawback of textons is that they are computationally intensive. However since their inception computational power has increased enormously and more literature has started to use them in image segmentation, such as [4]. Of further benefit, [4] suggests that as textons rely on dense features, both highly textured and untextured objects receive the same level of distinction. This is highly useful when dealing with objects with gradual texture changes, such as occur over distance (e.g. a runway tending to the horizon).

B. Colour based classification

Although texture data alone should be sufficient to separate most expected classes (as can be easily demonstrated through human interpretation of greyscale images) in an aerodrome environment, features of different classes may have very similar textures. For example, taxiway paint markings have different colours but the same texture.



Fig. 1. Example of Superpixel representation of outdoor scene.

In these cases, a texture-only interpretation method may struggle; therefore a colour classifier can be used in conjunction with a texture classifier as a secondary step to differentiate the two classes.

III. IMAGE SEGMENTATION

Semantic segmentation is a process consisting of two parts; segmenting the image into distinct regions and classifying each region as a known type. Depending on the method employed, classification and segmentation can be done together, or separately. For this work, the main examination was into whether to use classification based segmentation, or to segment the image first and then classify afterwards. The majority of modern methods attempt to perform both steps simultaneously, segmenting the image directly into the expected classes. Methods, such as [4] and [5], use the pre-defined classes as prototypes and the contents of the image are partitioned according to which class they are closest to, if any. Provided that the class models are well constructed and are representative of the image contents, segmentation is achieved quickly. Additional efficiency can be introduced by reducing the number of clusters used to fit the data. Contextual information, such as where the image was taken, can be used to achieve this. As this work seeks to detect unknown collision risks, this is difficult to implement as collision risks could take any form. Alternatively [6]

examines the image as a whole to create an image level prior, which lists the most likely classes present within the image, based on dominant features. Further classification is then performed with the classifier biased towards these expected classes.

As classification led segmentation relies on pre-determined data, each segment in the end result is either labelled as one of several known classes, or as an unknown. This can have the effect that two visually distinct regions which belong to the same class are only represented as a single region in the final result (e.g. two aircraft where one obscures part of the other become a single region). However, as the collision risk would still be identified, this is not considered to pose too great a problem.

The major limiting factor when using simultaneous segmentation and classification is the size of the visual descriptor. Results from [6] and [7] show that segmentation based on texture feature descriptors often shifts regional boundaries from their original position. Where regions are small enough, this can result in them being entirely misclassified, usually assigning the same class as a larger adjacent region.

This is due to texture based descriptors not having the resolution to work down to the pixel level. Despite being data rich, images are comparatively information devoid, with a great number of pixels required to provide any significant feature useful for recognition. Accordingly, a texture based feature descriptor must be large enough to differentiate textures. Texture based feature descriptors often rely on large sample areas to produce results; for example, the filter bank suggested in [8] encompasses several thousand pixels in every filter used. When the filters are applied at the centre of a region in the original image, the increased sample size provides a more detailed output which can aid during classification. However, when the filters are applied near region boundaries, they capture data from multiple regions at the same time, making classification more difficult. If the feature descriptor overlaps multiple classes simultaneously, the data produced is less likely to match a known class. If contextual biasing of the classifier has also been introduced (such as in [7] and [9]) dominant classes will often be categorised more favourable. This causes class boundaries to shift from their position in the original image, and can even completely conceal smaller regions.

Whether these effects are significant depends on the intended use of the segmented image. Currently, a major use of image segmentation is in automatic content recognition for image categorization. In this case, the goal is often to produce a simplified semantic description that a human would appreciate, for example *“a dog on a grass background, with trees”*. Here, failure to precisely adhere to a boundary is unimportant as the dominant elements remain intact. As including all minor elements would clutter the semantic output, the loss of minor elements within the image can actually help improve the simplification of the scene and likely has little detrimental effect on categorization. By contrast, collision avoidance relies on accurate range and bearing estimates to potential risks. As these estimates are extrapolated from the position of borders within the image, any significant shift in their position is undesirable.

A suitable set of classes for the aerodrome environment will

include obvious risks (such as aircraft, signage and ground vehicles) and surface types that are both safe and unsafe (asphalt, grass, etc.). However, unknown threats (such as FOD) also pose a large risk but are implicitly difficult to classify. As classification lead segmentation tends to bias results towards the known classes, a small object of an unknown class could be missed entirely until it is close enough to represent a significant portion of the image. Biasing the classifier using context (as in [9]) is successful when the scenario conforms to expected outcomes, but could conceal collision risks when they are most important.

This problem could be countered by increasing the resolution of the images until the size of the filter descriptor has negligible effect on the accuracy of the boundary lines. However, as the resolution increases, the number of pixels representing a texture element also increase, making it more difficult for the descriptor to differentiate accordingly. In addition, increasing image resolution requires similar increases in computational capabilities, which may prohibit the real time application of this method on UAS.

As the imprecise borders generated by these methods have been noticed by others, methods of overcoming the resolution limitations of texture lead approaches have been suggested. Paper [4] makes use of traditional edge detection to reduce the tendency for neighbouring nodes to share the same class, if an edge is detected between the regions. However, this now relies on accurate edge detection, which is a field of study in itself. As edges become less sharp at distance, a weak edge detection method is required, which may in turn produce many useless edges in the rough asphalt surface close to the UAS.

Despite the widespread use of classification lead segmentation, there are alternatives which may be more appropriate for this task. In order to maintain precise region boundaries, the regions must be defined down to pixel resolution. The best descriptor used at pixel level is colour, and there are many segmentation methods which have been shown to work well at this level. Obviously, this requires segmentation be performed first, with classification then used to identify the defined segments.

1) Segmentation without Classification: As texture based feature descriptors need to be fairly large, attempting to segment an image using texture will result in the extracted regions having imprecise borders. However, as textures are invariant to scale, rotation and lighting changes, they remain a viable method to classify outdoor scenes. A solution is to segment the image using a separate feature descriptor and identify each region afterwards.

Untrained segmentation methods that produce pixel level accuracy are well established for image processing applications. Region growing algorithms based on pixel similarity are some of the most common. In this process, pixels are grouped based on their colour and spatial distance from each other, creating clusters both uniform in colour and of similar size. Using starting pixels distributed throughout the image, neighbouring pixels are inspected. When a pixel is too different or too distant from the starting pixel, a borderline is created between them. This produces an over segmentation of the original image into many small pixel clusters known as ‘superpixels’.

The main benefit of superpixels is data reduction. As each superpixel is similar to all of the pixels it contains, the mean data of the superpixel can be used to represent many pixels. In addition, as superpixels will not group dissimilar pixels, the edges in the original image are captured by the borders between superpixels. Therefore an image with millions of pixels can be reduced to a meaningful representation of only a few hundred superpixels, dramatically increasing computational speed for many applications.

However, superpixels also have drawbacks. As each superpixel is fairly small, the technique will over-segment large regions and introduce many borders which are not present in the original image. In addition, the number of pixels within each superpixel may be too few to provide adequate texture information, making it difficult to perform classification at the superpixel level.

The problem of over-segmentation is discussed in [7], where semantic segmentation is achieved using superpixels as the first stage. In [7], the classifier is augmented with relative location data (e.g. pixels above trees are likely to be sky, and pixels below trees are likely to be grass) which proves to be an effective method of increasing accuracy. However, the final result still suffers from significant border drift.

An alternative is to use a second application of clustering, to group the superpixels into larger, visually similar regions. This reduces the risk of misclassification as each region has more information for the classifier to use. In addition, by merging superpixels into larger regions the actual region boundaries should be preserved. Work achieved in [10] proposes a double clustering effort to achieve this, where the superpixels are themselves clustered based on their mean colour data. Rather than similarity to an initial superpixel, [10] proposes using a density reachable approach. This allows for regions with gradual colour changes to be created, despite significant colour distance at either end. This is especially useful in this work, as atmospheric scattering causes all object to gradually change colour over distance. i.e. asphalt near the horizon will differ dramatically in colour from that close by.

IV. METHODOLOGY

This work aims to combine the speed and precision of untrained superpixel creation, with the reliable texture identification methods commonly used in trained segmentation methods. In the chosen approach, segmentation and classification are handled independently; an initial colour based over-segmentation and subsequent untrained clustering is paired with a primarily texture based classification approach.

Similar approaches have recently been developed for automotive use. For example, [?] proposes using superpixel based semantic segmentation for terrain classification to allow for automatic suspension adjustment based on driving conditions. However, [?] only attempts to identify the terrain class and does not attempt collision detection. This work contributes a method of collision risk detection without depth information relying solely on monocular images.

A. Image Segmentation

Despite superpixels being a relatively new concept, there are already many different algorithms available to generate them. Based on the method described in detail in [11], this work uses the well known Simple Linear Iterative Clustering (SLIC) approach to generate the initial segmentation. Compared to other superpixel generation techniques, SLIC is easier to implement, with the only tunable variable being the number of desired superpixels. This is accomplished using a single distance metric, with groups pixels based on both their XY distance, and their colour difference in the CIELAB colourspace. This simplicity makes the algorithm very computationally efficient, with the work done in [12] demonstrating that a standard desktop computer provides sufficient computational power for SLIC to process high resolution images in real time, easily matching the typical 30Hz refresh rate of cameras. Finally, SLIC produces superpixels of consistent size and of minimal colour range, while adhering well to region boundaries. Although difficult to quantify, it is subjectively considered to produce a better end result. An example of SLIC superpixel representation is shown in Figure 1.

During testing images were captured at 1680 by 1050 pixels, with approximately 4500 superpixels generated for each image using SLIC. As this is an obvious over-segmentation, a further clustering effort is required. The method outlined in [10] suggests combining superpixels using the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm. DBSCAN is a method of clustering that uses density reachability. This is particularly useful for outdoor images, as atmospheric scattering creates a gradual change in colour with distance. Therefore, a surface that is actually uniform in colour (such as a runway) will appear to vary in colour as it approaches the horizon. Provided that the superpixels between the closest and furthest point captured the change in colour gradually, the two extremes will be considered density reachable and therefore will be combined into a single region.

The distance metric used is colour distance within the Lab colourspace. To improve computational speed, only the colour distance between adjacent superpixels is considered, ensuring a high speed clustering is achieved. As the original superpixel boundaries formed by SLIC are preserved during DBSCAN clustering, the final result retains the sharp resolution required for 3D estimation.

B. Segment Classification

Classification is divided into two stages; a training stage where models of each class are created and a classification stage where the segmented image is compared to these models. To be able to draw a comparison, the same underlying feature extraction technique must be used throughout. In this work, textons have been selected as the descriptor and the classification methodology closely follows that outlined in [2].

1) *Classifier Training*: For each class that is to be identified, image data is required for use in training. To ensure the best match between model and segment, the chosen images must depict each class under various illumination

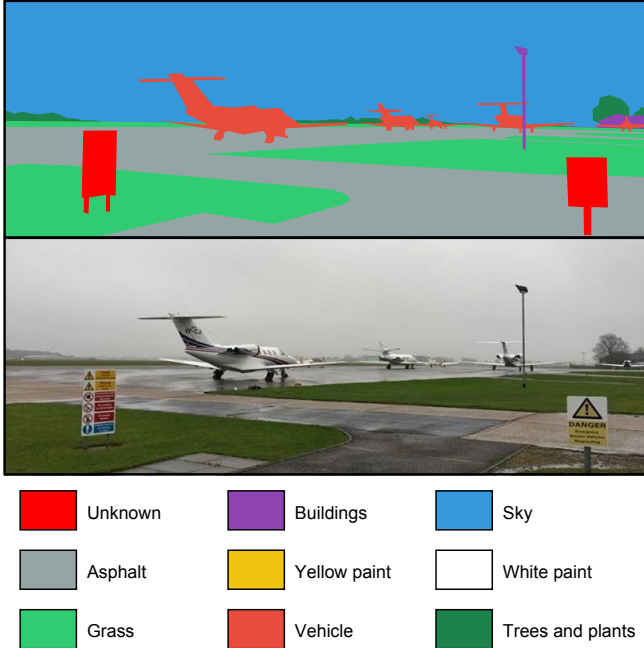


Fig. 2. Example of Training Image and Manually Segmented Representation

conditions, scales and rotations, consistent with what might be encountered at an aerodrome. The eight selected classes are listed in Figure 2. For each class, 20 training images are manually selected. Each image is then convolved using the same filter bank. Unlike [2] which only dealt with greyscale images, the images are captured in the RGB colourspace. To avoid the data loss in conversion to an intensity image, each of the colour channels are independently convolved with each of the filters. The two dimensional responses are then concatenated into layers, such that each pixel in the original image is now represented by an 'appearance vector' of filter responses, as it was originally represented by the colour channels. To allow for comparison, both the filters and the order in which they are arranged must be kept consistent throughout. In this case, the filter bank used is provided by Oxford Robotics group, and is demonstrated in [8].

As Gaussian filter responses are pixel orientated, each response vector is independent of its neighbours. This allows the vectors from all training images of a single class to be amalgamated into a single list. With each class now be represented by at least several hundred thousand filter vectors, K-means clustering is then applied to determine 20 centres within the data. These 20 centres are considered the typical appearance vectors and are labelled textons. The textons from each class are then collected to form a universal texton dictionary; with 8 classes, this results in 160 textons. In order to form a model of each class, the distribution of each texton within the images is required. Typical examples of each class are again convolved using the filter bank. Using nearest neighbour approximation, each pixel is then labelled as its closest texton counterpart. For the class as a whole, a normalised histogram can be established as the model of expected texton content.

For those classes with very similar texture models, colour data must be incorporated. Colour is modelled using a single RGB histogram, representing the pixel colour data within the entire sampled class. Although the colour histogram could be applied in any colourspace, (e.g. Lab, HSV) the RGB colourspace was found to provide the best results. This was due to it being the only colourspace which encodes colour information in all three channels, therefore requiring the most detailed histogram. The resulting colour histogram is also normalised.

2) *Classifier Application:* Applying the classifier uses a similar process to the training stage. Prior to the segmentation effort, the entire image is first convolved using each filter in the filter bank, in order to produce the required feature vectors. Using nearest neighbour approximation, each feature vector is then replaced by the most similar texton in the texton dictionary.

After the segmentation effort is complete, the image is broken down in a series of large regions. As the textons have now replaced pixels, the contents of each region can be simply extracted using a mask. The histogram of textons within each segment is then compared to the known models for each class using Bhattacharyya coefficient measure. Based on the resulting similarity measure, the system can provide a confidence as to how well the region fits each model, in addition to the most probable class type.

As expected, the colour classifier alone produces poor results. Instead, the colour classifier is used as a decider when the texture similarity measure indicates that a region is similar to more than one known class.

V. EXPERIMENTAL RESULTS

In order to test the proposed system, images taken from a viewpoint typical of a UAS camera system were required. As UAS cannot yet operate from civil aerodromes, a manned aircraft was used instead, with a Go-Pro camera mounted under-wing on a Cessna 152 during a routine round-trip flight from Coventry Airport, UK. Using images from the taxiing portion of outbound flight to train the classifier, a selection of frames from the taxiing segment of the return journey were put through the classification procedure. As the purpose of the experiment was to test the classifier, not perform depth extraction, the lens distortion was not corrected. However, the images were cropped to remove the airframe so as to provide a clear view of the aerodrome.

As the data was post-processed, runtime performance was not optimised. However, SLIC, DBSCAN and texton classification algorithms have all previously been shown to run quickly enough for real-time application, which is a future goal of this work.

The chosen scenario provides a visually challenging 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. Therefore multiple different texture types are required in order to produce a model for

TABLE I. PERCENTAGE BREAKDOWN OF TEXTURE-ONLY CLASSIFICATION OF TEN IMAGES.

Manual Segment Class	Automatic Segmentation Output as Percentage of Manually Segmented Classes								
	Unknown	Asphalt	Grass	Sky	White paint	Trees/plants	Buildings	Yellow paint	Vehicle
Asphalt	0.02	94.83	0.62	0.23	0.65	0.42	1.74	0.2	1.28
Grass	0.02	24.41	68.07	0.01	0.09	2.79	0.8	0.9	2.91
Sky	0.11	0.11	0.01	97.87	0.15	0.16	1.32	0.18	0.1
White paint	0.04	18.16	0.89	0.4	66.16	0.37	8.35	3.1	2.52
Trees/plants	0.18	5.49	6	0.86	0.37	61.91	16.12	0.12	8.96
Buildings	2.51	2.26	5.97	3.04	3.47	12.12	55.52	0.08	15.02
Yellow paint	0.13	13.8	5.54	0.06	3.46	0.13	0.72	74.93	1.24
Vehicle	3.58	2.03	2.32	2.08	6.32	19.19	9.4	2.82	52.27

a single class.

Figure 3 shows six frames captured at different points across the aerodrome. In order to evaluate the automated output, each image was manually segmented into the same classes to allow for comparison. Assuming the manual segmentation to be optimal (as human level capability is the current goal) the per-pixel results were compared to examine how successful the method has been. These results have been compiled in Table I. Each row in Table I represents the percentage breakdown of the original manually segmented class, in terms of the automated segmentation results. I.e. 0.23% of the pixels manually identified as asphalt were classified as Sky pixels by the automated classifier.

It must be recognised that whilst these results are indicative of success rate, they are based on real world images and therefore the regions used are of differing sizes. Accordingly, larger regions will show a smaller percentage error for the same per-pixel misclassification. This is partially responsible for why the region with the greatest percentage match is the sky, as it makes up the second largest area in each image. In addition, sky regions are usually only bordered along their bottom edge, resulting in a less border regions that can be misclassified. However, the result does confirm that textures are appropriate for classifying surfaces with sparse texture data and that the classifier works well on both overcast and clear skies.

A better indication of classifier performance is the asphalt class. Although asphalt is the largest region in all images, the consistency of asphalt varies at different points around the aerodrome, as well as with distance from the camera. From Table I it can be seen that the classifier performed well in matching asphalt despite the class requiring a more complex model than what is required for sky.

The errors in asphalt classification came from two sources; the first being the *patchiness* of asphalt, caused by different levels of wear and repair. As the clustering process works on superpixel colour distance, small regions of repaired asphalt must be sufficiently distant from their neighbours that they cannot be joined using DBSCAN. As such, a much smaller area is passed to the classifier, which often contains texture data which is not typical of the surface type. This results in the somewhat 'random' classification of smaller segments, as is especially apparent in the forth

image. Despite the classifier producing a very high success rate, false positive collision risks need only to occur very rarely to pose a problem to the system. As the majority of the training data was of the taxiway and runway surfaces which are less worn, it is likely that more example data is required that shows damaged asphalt, in order to produce a better overall class definition. Alternatively, a separate class for damaged asphalt could be introduced. This would not only be beneficial for classification, but could also be useful for path planning, as the UAS could attempt to avoid potholes which are not technically collision risks.

The second large source of misclassification for asphalt was caused by the inclusion of a concrete section in the taxiway, in the sixth image. Despite being manually labelled as asphalt, the surface was automatically labelled as part of a building. It can be concluded that despite both representing taxiway surfaces, asphalt and concrete will most likely require separate classes, with contextual information required to differentiate surface concrete from that used to make buildings. As buildings are static collision risks, this information could be sourced from an aerodrome map.

Yellow paint was well classified in most scenes, with the main misclassification being asphalt. As there is a relatively low amount of yellow paint in the images and yellow paint is almost always bordered by asphalt, this slight discrepancy is most likely caused by a difference in perceived border position between manual and automatic segmentation. A more clear result is that the addition of the colour classifier helped to ensure separation of white and yellow lines, with very little misclassification of either as the other. White paint was also fairly well matched, but was most commonly miss-classified as asphalt. Where white paint has been highly eroded it takes on the same texture as the asphalt underneath. In addition, there is very little colour difference between white paint and grey asphalt, making a colour classifier ineffective at distinguishing the two, especially if the classifier must remain illumination invariant. As human vision relies on the relative luminance of the white paint compared to the surfaces around it, a classifier that uses the surrounding regions as context is required.

The remaining classes each had inferior performance. One of the main reasons for this was a combination of distance and weather conditions. As is typical for aerodromes, neither trees nor buildings were located near to the runway.

As such, these classes constitute only a small fraction of each test images, with any error representing a large percentage of the region. In addition, atmospheric scattering of light causes 'blur' over distance, concealing the texture data and making regions difficult to identify at range. As grass regions are shown in both the foreground and at distance, a comparison can be observed. Distance grass in images C and F suffered and was misclassified as asphalt, while closer grass, such as in images E and F, was correctly classified. As these effects only occur at distance, and objects at distance are not immediate collision risks, this is not considered a large problem.

Instead, the failure to classify objects closer to the camera is more dangerous. In the images each individual collision risk was detected at least in part. However, often elements of the object are misclassified as the wrong class. In image A, the hangers were mostly classified correctly as buildings, but some elements were incorrectly labelled as vehicle, due to the similarity in material. As texture based methods are essentially surface material indicators, it is difficult to overcome these issues using image segmentation alone. For static obstacles (i.e. buildings, grass and trees) map data can be used to contextualise the image, helping differentiate regions of certain materials into their correct class. Even so, as these static obstacles are already shown on the map, the UAS should avoid them based on good path planning alone. Moving collision risks that are not shown on a map are of more concern.

In this test, the vehicle class has the lowest percentage match with the manually segmented image. A possible cause of this is the limited data available for the training phase, with this aerodrome having very few aircraft to use for sample data. However, more training data might not help overcome an underlying issue. As different elements of the aircraft will vary immensely in terms of texture and colour, building a single class to encompass all elements has been shown to not produce good results. Instead, separate classes for elements of aircraft, such as 'painted metal' may be more useful. Additionally, pilots already rely on contextual information, such as provided over the radio, to make judgements. By switching to a material based analysis, detecting a metal obstacle on a runway where there is no known collision risk should be all the contextual information required to indicate the likely presence of an aircraft. Therefore, further work is required into introducing a contextual step to the classifier.

VI. CONCLUSION

In this paper we have presented a method for segmenting images and semantically labelling the resulting regions. Using a double clustering approach of SLIC and DBSCAN, images are initially segmented using colour data alone. Classification is then achieved by matching texture features to known models.

From the results, it can be seen that the system can successfully identify elements within image that correspond to expected classes. The current system has difficulty in classifying regions that are not typical examples of that class

and is limited by the few classes that were created. Despite this, the automated segmentation was very similar to the manually segmented image in all cases.

The intention is to use a contextual approach to improve the results in the future. This will make use of adjacency within the image, in addition to using other data sources, such as an aerodrome map.

ACKNOWLEDGEMENT

The authors would like to thank BAE Systems for their continued support throughout this project.

REFERENCES

- [1] W. R. Schwartz, F. Roberti de Siqueira, and H. Pedrini, "Evaluation of feature descriptors for texture classification," *Journal of Electronic Imaging*, vol. 21, no. 2, pp. 023 016–1–023 016–17, 2012. [Online]. Available: <http://dx.doi.org/10.1117/1.JEI.21.2.023016>
- [2] T. Leung and J. Malik, "Representing and recognizing the visual appearance of materials using three-dimensional textons," *International Journal of Computer Vision*, vol. 43, no. 1, pp. 29–44, 2001.
- [3] S.-C. Zhu, C.-e. Guo, Y. Wang, and Z. Xu, "What are textons?" *International Journal of Computer Vision*, vol. 62, no. 1-2, pp. 121–143, 2005.
- [4] 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. [Online]. Available: <http://research.microsoft.com/apps/pubs/default.aspx?id=117885>
- [5] G. Csurka and F. Perronnin, "An efficient approach to semantic segmentation," *International Journal of Computer Vision*, vol. 95, no. 2, pp. 198–212, 2011. [Online]. Available: <http://dx.doi.org/10.1007/s11263-010-0344-8>
- [6] J. Shotton, M. Johnson, and R. Cipolla, "Semantic texton forests for image categorization and segmentation," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, June 2008, pp. 1–8.
- [7] S. Gould, J. Rodgers, D. Cohen, G. Elidan, and D. Koller, "Multi-class segmentation with relative location prior," *Int. J. Comput. Vision*, vol. 80, no. 3, pp. 300–316, Dec. 2008. [Online]. Available: <http://dx.doi.org/10.1007/s11263-008-0140-x>
- [8] M. Varma and A. Zisserman, "Classifying images of materials: Achieving viewpoint and illumination independence," in *Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark*, vol. 3. Springer-Verlag, May 2002, pp. 255–271. [Online]. Available: <http://www.robots.ox.ac.uk/vgg>
- [9] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie, "Objects in context," in *ICCV*, Rio de Janeiro, 2007.
- [10] 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/>.
- [11] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Ssstrunk, "SLIC Superpixels," EPFL, Tech. Rep., 2010.
- [12] C. Y. Ren and I. Reid, "gslc: a real-time implementation of slic superpixel segmentation," *University of Oxford, Department of Engineering, Technical Report*, 2011.


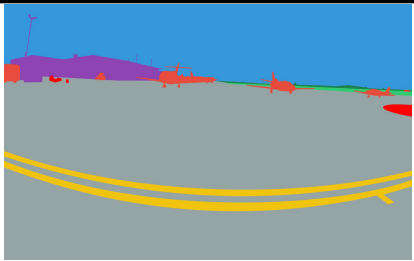






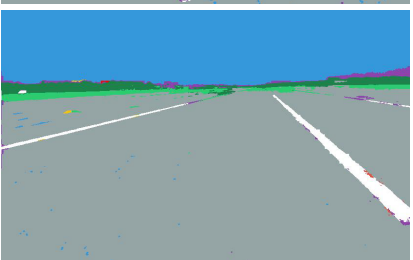

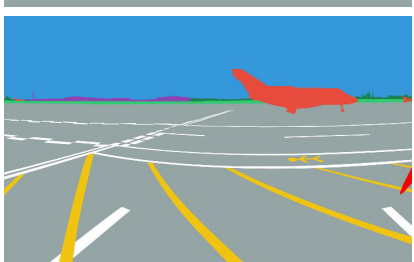
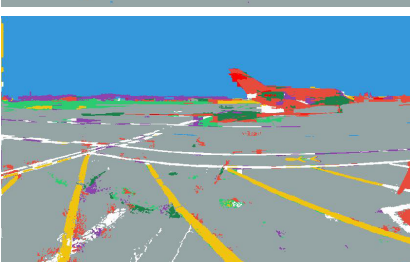

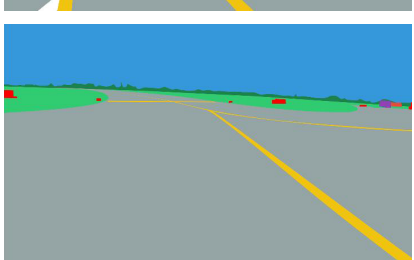
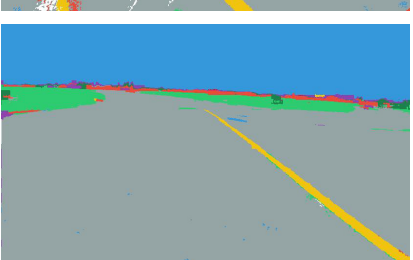

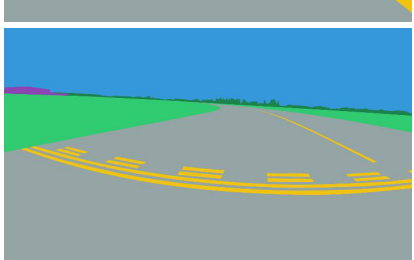
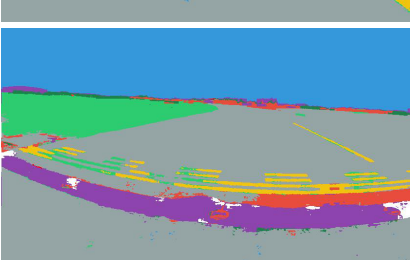
	Original Image	Manual Segmentation	Automatic Segmentation
A			
B			
C			
D			
E			
F			

Fig. 3. Comparison of Original Captured Image, Manually Segmented Image and Automatically Segmented Result