Classification of Seabed Type from Underwater Video

Steven Tyner¹, James Wilson² and David Corrigan¹
¹ Department of Electronic and Electrical Engineering
² Department of Zoology
Trinity College Dublin
Dublin, Ireland.
{tyners, jwilson, corrigad}@tcd.ie

July 8, 2014

Abstract

This paper describes a method for the classification of seabed type from a video captured by a camera mounted on a towed vehicle that is dragged along the sea floor. Classification of seabed type is important for the mapping of marine habitats. Unlike other methods that are based on various sonar technologies, the proposed method is based purely on video frames. The aim is to tell from a single frame, what seabed type is present. A supervised learning approach is adopted, with a total of 5 different seabed types being represented. We developed a set of 6 image features to characterise the visual appearances of these seabed types. Both k-Nearest Neighbours (kNN) and Support Vector Machine (SVM) classifiers are implemented based on this feature set. Our analysis shows that is possible to achieve a cross-validation error of 10% for the 5-class problem.

Keywords: Marine Surveillance, Seabed Classification, Habitat Mapping, Supervised Learning

1 Introduction

Knowledge of seabed habitats is important for both environmental and commercial purposes [Robinson et al., 2009]. Environmental Impact Assessments (EIAs) are often required for the licensing of marine development or fishing operations. Furthermore, EU legislation, namely the Habitats Directive [European Community, 2007], requires member states to contribute to a European ecological network specifying special areas of conservation within their territories. To this end, each state must conduct seabed surveys to establish if areas of conservation exist within their territories [Sotheran et al., 1997].

At present the majority of seabed surveys are conducted using various sonar technologies such as backscatter and AGDS [Collier and Brown, 2005]. These acoustic methods, however, provide results which are hard to interpret and require ground-truthing (often by the acquisition of sample grabs or underwater photography) to establish the true seabed type [Blondel and Murton, 1997]. Video is less commonly used to acquire survey data as it requires the extra expense of mounting cameras on vehicles that are either towed or remotely operated [Robinson et al., 2009, Davie et al., 2008]. However, it is easier to visually identify seabed type from video and also allows the identification of marine flora and fauna that are indicative of habitat present.

The aim of our work is to develop a system than can automatically classify seabed type from an underwater video captured by a camera mounted on a towed vehicle dragged along the sea floor. This would reduce the amount of time spent on manual analysis of these surveys and hence accelerate the acquisition and analysis of new data. We treat the classification as an image texture analysis problem and we use 6 features to characterise the texture of each video frame. Unlike related work [Pican et al., 1998, Davie et al., 2008], our method adopts a
supervised learning approach. This allows the classifier to be explicitly trained to determine type according to accepted definitions (e.g., the EUNIS (eunis.eea.europa.eu) or JNCC (jncc.defra.gov.uk/page-1584) classification systems) and, given sufficient training data, would allow the classifier to work under different lighting conditions, geographical locations and with different vehicles. A second key difference is that classification is performed at frame resolution rather than identifying multiple seabed types within a frame. This is a reasonable approach as the field of view is typically much less than the recommended $5 \times 5$ m$^2$ resolution specified by the EUNIS mapping instructions.

The remainder of the paper is organised as follows. The next section describes the test data and how it is used to generate the ground truth for both the training and testing of the classifiers. This is followed by a description of the feature set. Section 3 provides details of the classifiers implemented and presents the results of our experiments. The paper concludes with a discussion of our results and outlines avenues of future work.

### 2 Methodology

A total of 111 images were extracted from an underwater video from the HABMAP dataset [Robinson et al., 2009] and were manually classified into five seabed types of Boulders, Cobbles, Pebbles, Sand and Shells. Figure 1 shows an example of a video frame containing each seabed type and the number of examples of each type is given in Table 1. Only a small number of the video frames were suitable due to the large amount of motion blur present and because the camera was periodically stationary. Since the salient image content is unchanging while the camera is stationary, features are only estimated for one frame of each stationary period. This frame was extracted manually. All of the images were first pre-processed to mitigate against the uneven lighting present. This was performed by filtering the image with a low-pass filter with a narrow bandwidth and subtracting it from the original image.

From each of these pre-processed images six features were extracted for use in the seabed classifier. The features were designed to exploit the various image texture properties of each of the seabed classes. Each feature, bar one, was scaled so as to have zero mean and unit variance. This step was performed as the scale of each feature differed greatly and so added an error to the classification results.

#### Number of Edges

The first feature calculated is the number of edge pixels in the example image. This is an obvious choice since, for example, the sand class would contain fewer edges than other classes. An edge map was estimated from the \texttt{edge()} function in Matlab using the Prewitt edge detector [Prewitt, 1970]. The threshold for all of the frames was chosen to be the mean of the threshold
values for all of the training examples obtained by setting the automatic threshold flag in the edge function. The number of edges feature is given by the total number of non-zero values in the edge map. An obvious drawback of this method is the falsely inflated number of edges introduced by the time and date stamp in the bottom left corner. For training and testing of the classifier with the same dataset this offset in the number of edges would remain relatively constant and so can be ignored. However, to make the feature useful in general, this could be accounted for by ignoring the parts of the frame where the timestamp is located.

Mean Colour

As colour is a prominent feature across the seabed types, with sand exhibiting a greater yellow hue compared to the remaining classes, a mean colour feature was calculated by taking the mean of each of the RGB channels and returning the average of these three values.

Discrete Wavelet Transform Coefficient Energies

The Discrete Wavelet Transform (DWT) is a useful tool for texture analysis [Smith and Chang, 1994] and breaks down the texture into various bands according to texture orientation and frequency content. We developed three separate features based on the energy of the horizontal ($C_H$), vertical ($C_V$) and diagonal ($C_D$) bands at the third level of the wavelet transform [Arivazhagan et al., 2005, Kociolek et al., 2001]. For this paper the ‘bior2.2’ wavelet, a symmetric biorthogonal wavelet, was selected for the DWT. Each of the features, $E_x$, are calculated from the DWT coefficients, $C_x$, by

$$E_x = \frac{1}{N} \sum C_x^2$$

where $x = H, V$ or D and N is the number of coefficients.

Co-occurrence Matrix Correlation

A Grey-Level Co-occurrence Matrix (GLCM) is a statistical method used to examine the texture of an image through the spatial relationship between the pixels [Nanthagopal and Sukanesh, 2012]. The GLCM is created by calculating the number of times pixels with intensity $i$ and $j$, separated by a distance $d$ in a specified direction, occur in an input image [Haralick et al., 1973, Pican et al., 1998]. The GLCM is constructed with the $(i,j)$ location of the matrix representing the calculated number of occurrences of the $i$ and $j$ pixels.

The GLCM is, in fact, calculated from a quantised version of the input image. The default quantisation is used for this feature which scales all of the grey level intensities to integers between 1 and 8. This decreases the size of the GLCM to an 8x8 matrix and so lowers memory overhead and computation time and allows the matrix to be more densely populated. However, the distance and direction of the pixel pairs, known as the offset, was altered from the default of one pixel distance in the horizontal plane. For this feature the distances were chosen as 1, 2, 3, 8, 16 and 32 pixels so as to give a range of GLCMs based on increasing distance [Ghazali et al., 2007]. In addition, GLCMs were estimated for both the horizontal and vertical orientations, giving a total of 12 GLCMs.

Haralick proposed a number of features suitable for texture analysis that can be extracted from a GLCM [Haralick et al., 1973]. In this work the GLCM correlation was chosen as a feature and is calculated on each of the GLCMs after having been normalised to sum to one. The correlation of a GLCM is calculated as

$$Correlation = \frac{\sum_i \sum_j (ij)p(i, j) - \mu_x \mu_y}{\sigma_x \sigma_y}$$

where $p(i, j)$ is the normalised GLCM and where $\mu_x$, $\mu_y$, $\sigma_x$ and $\sigma_y$ are the means and standard deviations respectively of the sums of the rows and columns.
<table>
<thead>
<tr>
<th>Kernel Type</th>
<th>Boulders</th>
<th>Cobble</th>
<th>Pebbles</th>
<th>Sand</th>
<th>Shells</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>29%</td>
<td>19%</td>
<td>8%</td>
<td>5%</td>
<td>5%</td>
</tr>
<tr>
<td>RBF ($\sigma = 1$)</td>
<td>10%</td>
<td>18%</td>
<td>6%</td>
<td>5%</td>
<td>3%</td>
</tr>
<tr>
<td>Polynomial ($3^{rd}$ order)</td>
<td>5%</td>
<td>15%</td>
<td>9%</td>
<td>4%</td>
<td>5%</td>
</tr>
</tbody>
</table>

This correlation provides an indication of how correlated a pixel is to its neighbour over the whole image. This value ranges between -1 and +1, corresponding to a perfectly positively or negatively correlated image. As the correlation of constant textures is non-determined, feature scaling was not performed on this feature. The correlation feature is the mean of the coefficients for the 12 GLCMs.

3 Results

The k-Nearest Neighbours (kNN) and Support Vector Machine (SVM) classifiers were both developed to classify the seabed images. The kNN classifier is referred to as a lazy classifier and implementation of a multi-class classifier is straightforward. However, it is computationally inefficient at classifying unknown examples when the amount of training data is large. The key parameter in the kNN classifier is the value of k, which implicitly defines the complexity of the decision boundary between the classes. On the other hand, SVMs are much more efficient at classifying unknown examples once training has been completed. However, optimising an SVM for more than 2 classes is not straightforward. The notion of a kernel is central to SVMs as it allows complex decision boundaries between the classes. In our experiments, we tested both the value of k in the kNN and the Polynomial and Radial Basis Function (RBF) kernels of the SVM to minimise the classification error on the training data.

As SVMs are designed for 2-class problems, we trained a one-v-the-rest SVM classifier for each of the 5 classes. By performing a K-fold cross-validation with 10 folds, the average cross-validation error can be estimated on the training set and this value is used to optimise either the polynomial degree or sigma parameter for the respective kernels. From our experiments it was determined that the optimal degree of the polynomial kernel was 3 and achieved an average cross-validation error of 7.6% for the 5 one-v-the-rest classifiers. The optimal value of the sigma parameter for the RBF was 1, resulting in a slightly higher than average cross-validation error of 8.4%. For comparison we estimated the average cross validation error for the default linear kernel as 13.2%. A summary of the classification errors for each classifier is given in Table 2.

We used a kNN to implement a full 5-class classifier and use K-fold cross-validation to optimise the value of k. The optimum result was achieved with a value of k of 1 (ie. A nearest Neighbour Classifier) and the cross-validation error achieved was 10%.

As expected with these results the linear SVM kernel causes the greatest classification error due to the lack of linearly separable data. The more complex, non-linear polynomial and Gaussian kernels had a lower classification error with the polynomial kernel outperforming the Gaussian with the optimal order of three. It is also expected that this classification error is less than the kNN classifier due to the the fact that it is classifying between 5 rather than 2 classes.

Throughout the testing of both classifiers it was found that there was a constant misclassification of several of the sand images. Analysis of these specific images revealed they depicted coarse sand, as shown in Figure 2, not the smooth, silty sand that would typically be associated with a sandy seabed. As such the features detected this increased level of coarseness and likely viewed these images as resembling the features of the pebble images. The Joint Nature Conservation Committee (JNCC) classification framework defines the various classes based predominantly on the physical size of the individual stones or grains of sand. Hence, there
is scope for miss-classifications where the sizes are near the specified boundaries between the classes.

4 Conclusion

This paper has proposed a method for the automatic detection of seabed type from a video taken from a towed or remotely operated vehicle. The key idea is to use a supervised learning approach that allows a classifier to be trained to accepted definitions of seabed type and given sufficient training data should be robust to variations in lighting conditions and geographical location. Our experiments show that on a small dataset it is possible to get a classification error of 10% for the 5 class problem.

Acquiring more labelled videos of seabeds is key to improving the performance of the classifications. It will allow both improvements in the design of the classifiers and on the design of features as well as to obtain a more reliable estimate of the accuracy of the classifier. For example, including data from more than one survey will establish the accuracy of the classifier on newly acquired marine surveys and will ensure that the features used are not overfit to the training data. We would also like to explore multi-class implementations of the SVM as well as Neural Networks as frameworks for the multi-class classification problem.

Acknowledgments

This work was funded in part by a grant from Science Foundation Ireland (Project Number 08/IN.1/I2112). The authors would like to thank Dr. Bryn Jones from Natural Resources Wales for providing some of the training data for this study.

References


