Underwater Event Classification

Mathias Kölsch, kolsch@nps.edu

Abstract

Classifying events in underwater video is currently a laborious task that technicians have to spend hours on, that delay the availability of survey results, and that drastically limit the feasibility of large-scale or long-term open ocean wildlife surveys. Automatic recognition of some events (animals) would open doors for many avenues of new research and help improve our understanding of the deep-sea environment.

This report describes the outcome of a class project in a computer vision course taught at the Naval Postgraduate School. The problem setting to the students was to classify images from sighting events involving four different animals: Flatfish, Sea cucumber, Sea star, and Rockfish. The students were provided with annotated training data and had to choose appropriate image features and classification methods. The results varied widely (across students and animals :-) but were overall very promising, with classification accuracies reaching almost 90%.

CS 4330: Introduction to Computer Vision, Fall Quarter 2006

This course is taught annually at the Computer Science department at the Naval Postgraduate School (NPS). The Fall 2006 (FY07-1) course was the second incarnation since the course's inception in 2005. Its goal is to provide an overview of *computer vision*, which is the field of study of enabling computers to understand visual input. The course introduces the major concepts and methods and gives the student a sample of problem settings and techniques for solving them. It has strong ties between theory, examples of its application, and practical exercises in order to promote fast and persistent learning. The course is intended for the student who wants to get an understanding of how computer vision works and how it can perform such diverse tasks as object recognition, robot navigation, and increasing situational awareness. No prior experience or knowledge in computer vision or image processing is required.

More information can be found on the course web page [1].

The final project in this class was to classify said objects in underwater images. This was a group project (1 or 2 people) with the goal of correctly classifying a previously detected animal into one of four classes. To produce the best results, students had to balance their effort between selecting/integrating a good feature descriptor and a good classification method. The quality of the classifier was to be determined based on the confusion matrix. The students were free to choose their implementation platform and image processing library. They teamed up in 6 teams which will be referred to as Team A through F.

Training and Evaluation Data

Instrumental to the success of this class project was the access to large numbers of labeled training data that were graciously provided by Dr. Duane Edgington at the Monterey Bay Aquarium Research Institute (MBARI). Those data were grouped into "events," sightings of animals in a sequence of frames in underwater video. These events had previously been extracted semi-automatically (Edgington et al. [2]). One event consists of one or many cropped images of one individual animal from various viewpoints and possibly engaged in some form of crawling, swimming, or another motion. The following table gives an overview of the entire dataset and which parts were used for training and classification.

animal picture and name (images courtesy MBARI)		events	total num. images	training images	test images
	Flatfish	26	577	16	19
	Sea cucumber	37	2202	31	38
	Sea star	102	2642	44	82
	Rockfish	7	642	15	17

Training as described in the Results section was performed on distinct, labeled data sets. Each training set contained between 15 and 44 images from selected events, with up to two images from the same event, and up to three images from some Rockfish events. Classification was tested with different subsets of images (those that were not used for training) from independently selected events. Each test set contained between 17 and 82 images from selected events, with up to two images from the same event, and up to four images from some Rockfish events. There was some overlap between training and test data in terms of the *events* that the images were taken from, but no *image* was in both the training and test datasets. This overlap was necessitated by the limited number of events and the need for sufficient training data. Another solution would have been to train on all images of some events and to evaluate on all images of the remaining events. That was not deemed to be beneficial, primarily because of only seven Rockfish events and the requirements on data diversity for successful classifier training.

Students were given said training data set. If desired, they could split it into training and test sets for their own evaluations. This split was not required nor documented in general. Some teams used leave-one-out round-robin methods to determine the performance of their classification methods, others used more ad-hoc methods and possibly even the same training and test sets. The results reported below are with respect to the "complete" training and test sets.

Image Features and Classifiers

At the time the students were given the image classification problem, they had learned about a number of image features in detail, including grey-level gradients, edges, line histograms, and Hough transforms. In brief student presentations to the class, they were also introduced to the following appearance-based features, their properties, advantages, and limitations, as well as freely available implementations, if any:

- Eigenfaces (PCA on image data)
- Gabor wavelets
- Viola-Jones
- Active Shape Models
- Active Appearance Models
- Good Features to Track
- Laplacian of Gaussian (LoG)
- Edge orientation histograms
- Shape context
- SIFT
- SURF
- Gradient PCA

They were to choose appropriate features or a combination of features for their task. Given the short time period (2-3 weeks), it could not be expected that they would create their own implementation of a well-known feature, much less that they would invent an entirely novel feature type. Accordingly, the features that students preferred were those with an existing or otherwise well-supported implementation in the software library of their choice. The following table lists what the individual teams chose as their image features.

team	feature(s)
A	RGB mean, RGB stddev, edge orientation histogram using Scharr kernel
B	equalized hue and saturation histograms on color-segmented foreground
C	PCA on grey-scale image data
D	PCA on resized, Gaussian-smoothed image
E	Sobel edges, thresholded, contours, 10-bin histogram on foreground object distribution across the image
F	brightness normalization, edge orientation histogram, hue histogram

After extracting the respective features from the images, a multi-class classification had to be performed in feature space. Due to the small size of the data set, most teams chose k-nearest neighbor (k-NN) classification, arguably the best choice for poorly defined decision boundaries. Multiple teams varied the parameter k and determined that a 1-NN classification produced the best results. Other teams experimented with unsupervised k-means clustering (with k=4) to segment the feature space, and then used the cluster mean to test proximity. They eventually abandoned this approach in favor of classification based on the full training data.

team	**classifier and distance function**
A	Bhattacharyya distance on histograms, best match (1-NN)
B	Bhattacharyya and Chi-squared distance on histograms, 5-NN
C	sum of squared differences in 2-dimensional PCA space, 2-NN
D	closest distance (1-NN)
E	closest distance (1-NN)
F	closest distance (1-NN), weighted 66% edge, 33% hue histogram

Results

Team F chose Matlab as their implementation platform and the Image Processing Toolbox for library functions, all other teams implemented their feature detectors and classification methods in C/C++ with the help of the OpenCV library. The table below shows how well each team's classifier performed on the test data. The Rockfish caused the most difficulty due to its rather varied appearance without highly distinguishable texture or shape. The Sea cucumber, a worm-like animal, can also appear very differently from image to image. It is generally easier distinguished from the background, however, and edge detection methods produce better results. Images of the Flatfish were surprisingly easy to classify despite the animal's nondescript surface and its excellent camouflage before the sea floor. Image brightness normalization and other pre-processing proved valuable to the successful classification of this animal. The Sea star is relatively easy to pick out, particularly with edge-based methods that register a high number of strong edges in all orientations. Hence the good overall results for this animal.

team	**Flatfish**	**Sea cucumber**	**Sea star**	**Rockfish**	**average**
A	89.5	50.0	96.3	29.4	**66.3**
B	88.9	89.2	88.9	18.8	**71.5**
C	88.9	62.2	85.2	18.8	**63.8**
D	88.9	78.4	90.1	87.5	**86.2**
E	52.6	50.0	73.2	53.0	**57.2**
F	63.2	55.3	96.3	41.2	**64.0**
average	78.67	64.18	88.33	41.45	**68.17**
max	**89.5**	**89.2**	**96.3**	**87.5**	

No one feature detector and classifier seems to perform equally well on all four classes. The overall best classifier achieved a very respectable recognition rate of 86.2%. Details of team D's results are shown below in a confusion matrix. Note that this team's program unintentionally ignored one training image and one test image of every class due to a bug in the software.

found / actual	Flatfish	Sea cucumber	Sea star	Rockfish	% correct
Flatfish	**16**	**1**	**1**	**0**	*88.9*
Sea cucumber	**6**	**29**	**1**	**1**	*78.4*
Sea star	**5**	**0**	**73**	**3**	*90.1*
Rockfish	**2**	**0**	**0**	**14**	*87.5*

Summary and Conclusions

Automatic classification of animals in underwater video is a challenging task. This CS4330 class project yielded overall surprisingly good results, despite tight time constraints and this course being the students' first exposure to computer vision. The best classifier reached an average of 86% correct classification. However, additional tests on unseen events have to be performed to determine the robustness of the results.

A system with over 86% correct classification already is a tremendous help to a human annotator who traditionally has to manually inspect hours of video and hand-label events. Considering that most events produce many images of the same animal, a temporal integration of the results from individual image frames is likely to further improve the results.

Acknowledgments

This project would not have been possible without Dr. Duane Edgington's support and his preparation of the data. A huge thank-you! goes to the students in this CS4330 class who performed such great work on such difficult data: Paul Avellino, Eric Forsyth, Ron Hemmelgarn, Jeffrey Hyink, John Kelly, Donnie Mayo, Michael McVicker, Ed Murphy, Marcin Pohl, and Eric Sjoberg. Last but not least I would like to thank Ishbel Kerkez and Judith L. Connor at MBARI for their help with obtaining access to the image data.

References

[1] http://www.movesinstitute.org/~kolsch/courses/CS4330/CS4330.html

[2] Edgington, D.R., Cline, D.E., Davis, D., Kerkez, I., and Mariette, J., Detecting, Tracking and Classifying Animals in Underwater Video (060331-207). MTS/IEEE Oceans 2006 Conference Proceedings, Boston, MA September 2006. IEEE Press.