DETECTING FISH IN UNDERWATER VIDEO USING THE EM ALGORITHM

Fiona H. Evans

The School of Mathematics and Statistics
The School of Electrical, Electronic and Computer Engineering
University of Western Australia

ABSTRACT

We consider the problem of detecting fish in underwater video. We adopt a modeling framework, where the shape of each fish is assumed to be multivariate Gaussian. Mixture modeling is used to classify noise and varying numbers of fish. The mixture parameters are estimated using an EM algorithm that incorporates an Akaike information criterion to simultaneously estimate the number of components in the mixture. In addition, the algorithm does not require careful initialization.

1. INTRODUCTION

The traditional methods for monitoring marine species involve either visual assessment by scuba divers or their removal from the water. In the first case, accurate estimates are difficult to obtain because of the optical distortion of the relationship between size and distance underwater. In the second case, stress and possible death can occur. An impersonal and nondestructive system of measurement is required.

Underwater imaging systems provide a potential solution. Stereo-imaging systems (where two cameras are used to obtain depth measurements) were examined for use underwater as early as 1964 [3]. Since that time, there has been a move by the marine research community towards the use of both still and stereo systems [1;8;9]. Concurrently, research has been undertaken to quantify improvements in accuracy obtained by the use of imaging systems over divers [7].

Very few studies; however, have used automated techniques for the analysis of underwater images. Measurements are usually made by human interpretation of the images. In other words, the human user is required to examine the image data to make subsequent measurements.

We examine the use of an EM algorithm for automatically identifying fish in images. We approach the problem from a modeling perspective, by assuming that individual fish can be modeled using a simple parametric statistical distribution. We then model the image as a mixture model. An EM algorithm is then used to estimate the mixture parameters, including the number of components in the mixture.

2. THE DATA

Southern bluefin tuna (SBT) are a migratory ocean fish that are increasingly subject to over-exploitation. In Australia, the SBT fishery operates by holding wild-caught fish for fattening after capture. Fish are held temporarily in towing cages and then released into moored "grow-out" cages where they are fed for several months on a diet of baitfish and then harvested between three and eight months later.

Given their high value, farmers are reluctant to cause the SBT stress by removing them from the water. This makes it difficult to monitor any catch. Currently, monitoring is performed by removing a small sample of fish from each tow cage and taking length and weight measurements. An underwater video attached to the side of the gate is used manually count the SBT as they are transferred from the tow cage to the grow-out cage. The total biomass per tow-cage is estimated by multiplying the count by sample mean. We aim to automate the process and improve its accuracy by developing an algorithm that will use the video to first detect, and then count the fish.

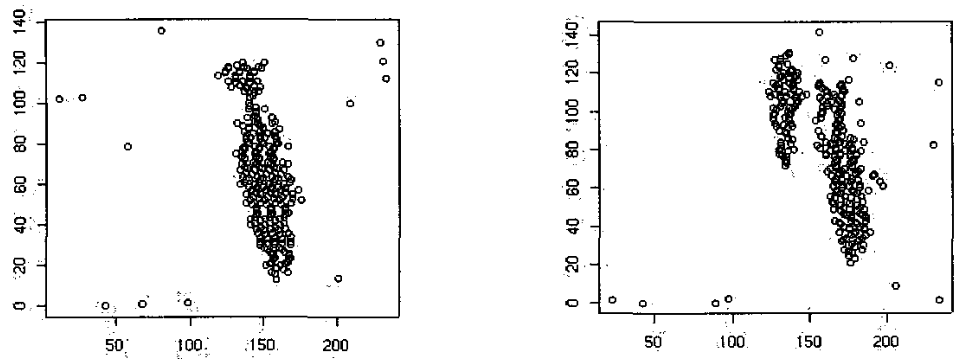
Image data for the project are being provided by Dr. Euan Harvey (Marine Biology Group, Department of Botany, University of Western Australia).

2. STATISTICAL MODELS FOR FISH

Figure 1, overleaf, shows typical frames collected from the underwater video camera. The quality of the data is poor. Spatial noise caused by suspended particles and matter in the water can be clearly seen in the thresholded data, shown in Figure 2. The SBT are roughly elliptical in shape and have higher intensities towards their centers. For this reason, we choose to model individual SBT using a 2-dimensional Gaussian distribution, using the X and Y

(a) Frame showing a single SBT. (b) Frame showing two SBT

Figure 1. Typical frames with intensity increasing from black to white.



(a) Frame showing a single SBT. (b) Frame showing two SBT

Figure 2. X and Y pixel locations extracted from the thresholded frames.

pixel locations from the thresholded images as our input data. By modeling the image data as a mixture, we aim to fit Gaussian distributions to each component in the image; that is one component for each SBT and one for the spatial noise. The noise components are easily identified by having the lowest fitted mixture weights.

3. GAUSSIAN MIXTURE MODELING VIA THE EM ALGORITHM

We have an observed data set $D_X = \{x_i\}_{i=1}^{n}$ and consider a k-component mixture model where each component has a multivariate Gaussian distribution. We apply the EM algorithm [5] with update equations:

$$w_m^{(p+1)} = \frac{1}{N}\sum_{i=1}^{n} p\left(y_m \mid x_i, \theta^{(p)}\right)$$

$$\mu_m^{(p+1)} = \frac{\sum_{i=1}^{n} x_i\, p\left(y_m \mid x_i, \theta^{(p)}\right)}{\sum_{i=1}^{N} p\left(y_j \mid x_i, \theta^{(p)}\right)}$$

$$\Sigma_m^{(p+1)} = \frac{\sum_{i=1}^{n} p\left(y_m \mid x_i, \theta^{(p)}\right)\left(x_i - \mu_m^{(p+1)}\right)\left(x_i - \mu_m^{(p+1)}\right)^T}{\sum_{i=1}^{n} p\left(y_m \mid x_i, \theta^{(p)}\right)}$$

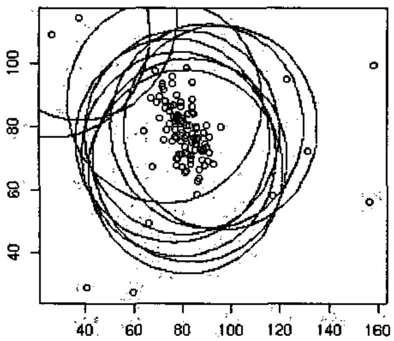
(see [2] for the detailed derivation).

4. INCORPORATING AN AKAIKE INFORMATION CRITERION

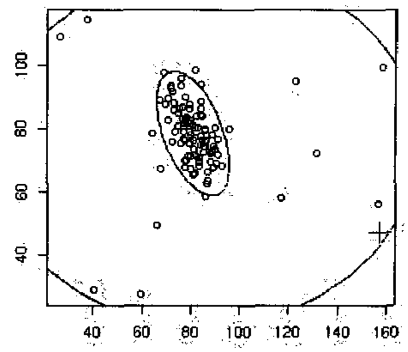
We use Bozdogan's consistent Akaike information criterion [4] for a k-component mixture:

$$AIC = -2\log L(\theta \mid D_x) + 2c$$

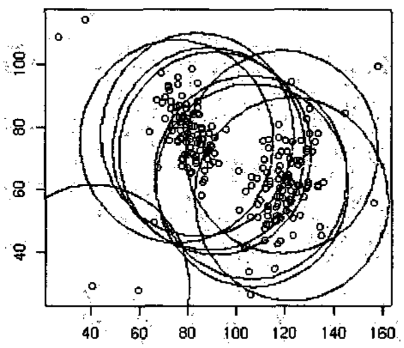
where $c = Nk + k$ is the number of parameters in the mixture density, N is the number of parameters in each component density and $L(\theta \mid D_x)$ is the likelihood function. We incorporate this into the EM algorithm by minimizing the expected AIC over all values of θ and k, rather than maximizing the expected log-likelihood. Since we require that $w_j \geq 0$ for all j, we simply truncate negative weights to zero.



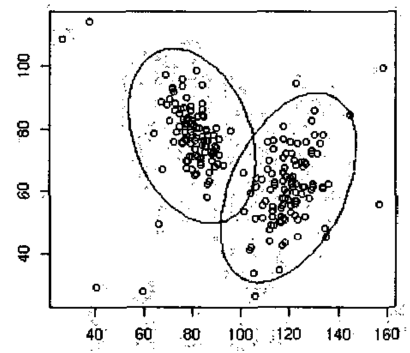
(a) Example 1: Initialization k=10



(b) Example 1: 15 iterations k=2



(c) Example 2: Initialization k=10



(d) Example 2: 15 iterations k=3

Figure 3. Fitting a Gaussian mixture model to generated data.

In the case of Gaussian mixture modeling, the update equation for the mixture weights is then:

$$w_m^{(p+1)} = \frac{\max\left\{0, \sum_{i=1}^{N} p\left(y_m \mid x_i, \theta^{(p)}\right) - N\right\}}{\sum_{j=1}^{k} \max\left\{0, \sum_{i=1}^{N} p\left(y_j \mid x_i, \theta^{(p)}\right) - N\right\}}$$

where the denominator ensures that the weights sum to one.

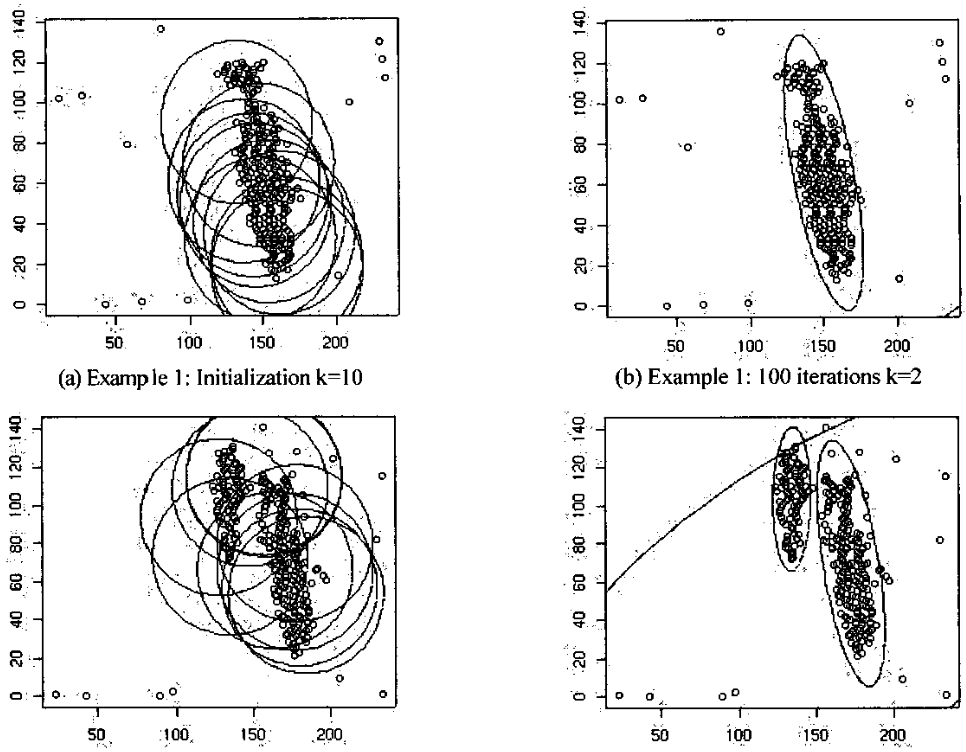
When a component's weight is set to zero, it is eliminated from the mixture. To speed convergence, the covariance matrices of the remaining components are re-initialized whenever this occurs. The EM algorithm is thus performed by randomly initializing the mixture using a large number of components, and eliminating those that are unnecessary. The algorithm is similar to that of [6] which uses a minimum message length (MML) criterion [10], but tends to converge at a faster rate.

5. EXPERIMENTS

In the first experiment we used 100 samples from a 2-dimensional Gaussian distribution and 10 samples from a uniform distribution (to represent noise). The EM algorithm was initialized using 10 components and converges successfully to 2 components. In our second experiment, we added 100 samples from an additional Gaussian distribution. The initializations and final estimates are shown in Figure 3 – the ellipses are drawn at 2 standard deviations from the component means. Note that in Figure 3(d), the ellipse for the third component is drawn outside of the plotting area.

The results of applying the EM algorithm to the image data are shown in Figure 4, overleaf. In the first example, the image contains a single fish and the algorithm converges to two components, separating the background noise from the fish. In the second example there are two fish in the image, and the algorithm converges to three components, separating the two fish and the background noise.

Further work is required to determine the robustness of the algorithm, particularly in cases where the SBT are occluded by each other.



(a) Example 1: Initialization k=10

(b) Example 1: 100 iterations k=2

(c) Example 2: Initialization k=10

(d) Example 2: 100 iterations k=3

Figure 4. Fitting a Gaussian mixture model to image data.

7. BIBLIOGRAPHY

[1] Beddow, T. A., Ross L. G., and Marchant J. A., "Predicting salmon biomass remotely using a digital stereo-imaging technique", *Aquaculture*, 146, 189-203, 1996.

[2] Bilmes, J. A., "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models", Technical Report TR-97-021, Computer Science Division, Department of Electrical Engineering and Computer Science , U. C. Berkeley, 1998.

[3] Boyce, R. E., "Simple scale determination on underwater stereo pairs", *Deep Sea Research*, 11, 89-91, 1964.

[4] Bozdogan, H., "Model selection and Akaike's information criterion (AIC): The general theory and its analytical extensions", *Psychometrika*, 52, 3, 345-370, 1987.

[5] Dempster, A. P., Laird, N. M., and Rubin, D. A., "Maximum likelihood from incomplete data via the EM algorithm", *Journal of the Royal Statistical Society, Series B*, 39, 1-38, 1977.

[6] Figueiredo, M. A. T and Jain, A. K., "Unsupervised learning of finite mixture models", *IEEE Transactions on Pattern Analysis and Machine Learning*, 24, 3, 381-396, 2002.

[7] Harvey, E., Fletcher D., and Shortis M., "A comparison of the precision and accuracy of estimates of reef-fish lengths determined visually by divers with estimates produced from a stereo-video system", *Fishery Bulletin*, 99, 1, 63-71, 2000.

[8] Petrell, R. J, Shi X, Ward R. K., Naiberg. A., and Savage C. R., "Determining fish size and swimming speed in cages and tanks using simple video techniques", *Aquacultural Engineering*, 16, 63-84, 1997.

[9] Ruff, B. P., Marchant J. A., and Frost A. R., "Fish sizing and monitoring using a stereo image analysis system applied to fish farming", *Aquacultural Engineering*, 14, 155-173, 1995.

[10] Wallace, C. and Freeman, P., "Estimation and inference via compact coding", *The Computer Journal*, 42, 4, 241-252, 1987.