

RESEARCH

Open Access

Automated quantification of the schooling behaviour of sticklebacks

Reza Ardekani^{1*}, Anna K Greenwood², Catherine L Peichel² and Simon Tavaré^{1,3}

Abstract

Sticklebacks have long been used as model organisms in behavioural biology. An important anti-predator behaviour in sticklebacks is schooling. We plan to use quantitative trait locus mapping to identify the genetic basis for differences in schooling behaviour between marine and benthic sticklebacks. To do this, we need to quantify the schooling behaviour of thousands of fish. We have developed a robust high-throughput video analysis method that allows us to screen a few thousand individuals automatically. We propose a non-local background modelling approach that allows us to detect and track sticklebacks and obtain the schooling parameters efficiently.

Introduction

Threespine sticklebacks (*Gasterosteus aculeatus*) (Figure 1) have been a model organism in behavioural biology since the pioneering work of Niko Tinbergen over half a century ago [1]. Much is understood about stickleback behaviour in both the field and the laboratory [2,3]. More recently, sticklebacks have become a model system for understanding the genetic basis for divergence in phenotypic traits, including behaviour [4]. Differences in schooling behaviour between two populations of sticklebacks that inhabit dissimilar environments have been characterized [5]. Marine sticklebacks live in open water and school very strongly, whereas freshwater bottom-dwelling lake populations (benthics) exhibit reduced schooling [5]. We have developed an assay using an array of artificial stickleback models to elicit and quantify schooling behaviour [5]. Using this assay, we showed that marine sticklebacks spend significantly more time schooling.

Our goal is to dissect the genetic basis for the divergent schooling behaviour between marine and benthic sticklebacks. Quantitative trait locus (QTL) mapping has successfully identified the genetic basis for many variant traits in sticklebacks [4]. The plan is to use QTL mapping in benthic-marine hybrids to identify genetic loci that contribute to differences in schooling behaviour.

To assay the hundreds of fish necessary for this technique, a robust high-throughput video analysis system is essential. In this paper, we present a custom approach for analysis of videos from our assay. We propose a method for background modelling for videos that are (semi-)periodic; i.e. those in which some or all of the background in each frame is repeated in at least a few other frames in the video. We show the result of this simple yet effective method for processing videos from our experiments.

Target detection for video tracking

For any video tracking system, target detection is an essential ingredient. One approach is to detect an object of interest based on appearance features such as geometric shape, texture and colour [6]. In this approach, the visual features should be chosen so that the target can be easily distinguished from other objects in the scene. This approach has become more popular recently, partially due to the great progress in object detection [7]. Another approach to detect moving objects in the scene is background subtraction [8]. This approach is especially useful for surveillance systems, such as for parking lots, offices, and controlled experimental environments, in which cameras are fixed and directed to the area of interest. The main property of these systems is that background is to some extent static, and a model of background can be calculated for each frame [9]. For example, Wu et al. used this method for detection and tracking of a colony of Brazilian free-tailed bat in nature [10]. Different methods have been developed to robustly maintain the background

*Correspondence: dehestan@usc.edu

¹Program in Molecular and Computational Biology, University of Southern California, Los Angeles, CA 90089, USA

Full list of author information is available at the end of the article

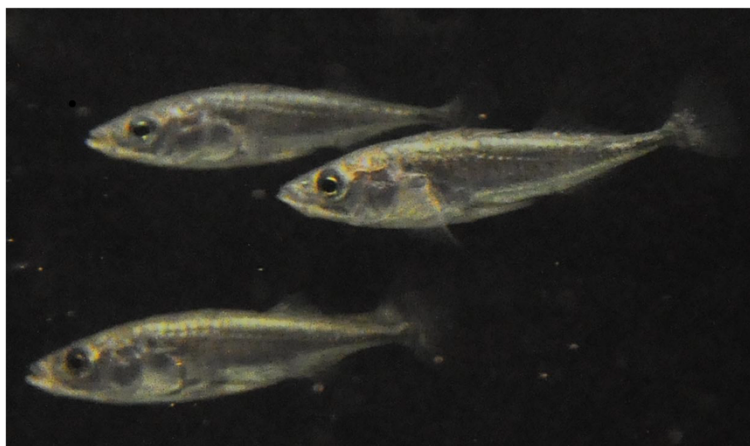


Figure 1 Threespine stickleback (*Gasterosteus aculeatus*) fish. It is an important model organism in behavioural biology.

model in scenes with possible changes in background such as gradual change in lighting and sudden changes in illumination due to light switches [8,9]. Moreover, there are studies that address background modelling in dynamic scenes with significant stochastic motion, such as water or waving trees [11,12]. Unfortunately, the aforementioned approaches are not applicable for our experiments due to our experimental set-up (see the ‘Challenges’ section). In this paper, we propose a non-local background modelling approach, which exploits the semi-periodic nature of the videos and overcomes the limitations of other approaches.

Experimental set-up

The model school is composed of eight plastic model sticklebacks that are arranged to mimic the formation of an actual school of sticklebacks [5]. The models are attached to wires and driven by a motor in a circular path within a circular tank. Trials are videotaped using a video camera mounted above the tank, as shown in Figure 2. For behavioural trials, fish are removed from their home tank and placed into individual isolation chambers for at least 1.5 h before the trial. Fish are then individually placed into the model school assay tank and given 5 min to acclimate. The motor controlling the artificial school is then turned on remotely, and the fish are given 5 min to interact with the models. The features we quantify in each video are the time taken for the fish to initially move within one body length of the model, the time of schooling with the model (i.e. swimming in the same direction as the model, within one body length), and the number of schooling bouts (i.e. the number of times that a fish starts schooling after it has stopped). These data can be obtained from the position and direction of the fish and the model in each frame. All research on live animals was approved by the Fred Hutchinson Cancer Research Center Institutional Animal Care and Use Committee (protocol 1575).

Challenges

There are two properties that make the task of tracking sticklebacks in our set-up challenging. First of all, the model fishes, as intended, look very similar to the real fish (see Figure 3). Therefore, no obvious visual feature can distinguish between the real fish and the model fish. So, even though it is possible to detect the real fish in the frames in which this fish is not close to the models using visual clues such as shape and intensity of fish contour, it is almost impossible to distinguish them in the frames where

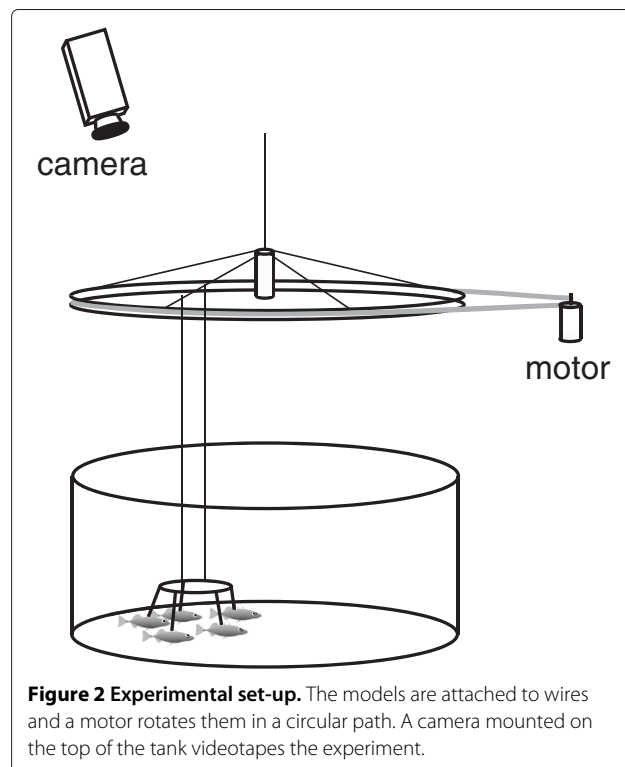


Figure 2 Experimental set-up. The models are attached to wires and a motor rotates them in a circular path. A camera mounted on the top of the tank videotapes the experiment.

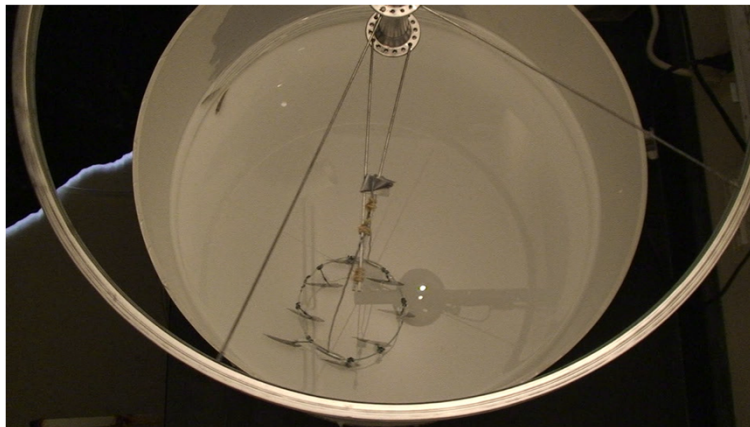


Figure 3 Sample frame. Sample frame from the video; the resolution of videos is 960×540 .

the real fish is schooling with the model fish. Problematically, these are the frames in which we are most interested because they represent the schooling behaviour.

Moreover, since the model school is rotating, the associated poles and wires are also moving in the scene, but these are not the desired targets. Therefore, detecting real fish by background subtraction using a static model or using the most recent frames as the background model is not effective. We define a new 'background' model in which all objects (including moving ones) are a part of the background, and only the target, which is the real fish, is detected as foreground. It is possible to create such a background if objects in the video have a predictable motion model. Our main contribution is to exploit the periodicity of the videos and build a background model, which enables us to discount all moving parts of the set-up except the fish.

Proposed method

Model school detection

To detect the schooling behaviour of the fish, we need to detect the model school. As can be seen in Figure 3, the fish are suspended from a circular wire. An obvious choice for circle detection is the generalized Hough transform [13], and since the radius of the circle (aside from the negligible variation due to perspective effect) is constant, the model fish are effectively located. The process of model detection can be expedited using the previous frame information for each frame and searching for a circle in the neighbourhood of the region of interest (close to the last frame detected) instead of searching the whole image. By finding the centre of the circle at each frame, the movement direction of model fishes is extractable; this is needed to calculate the statistics we need from each experiment.

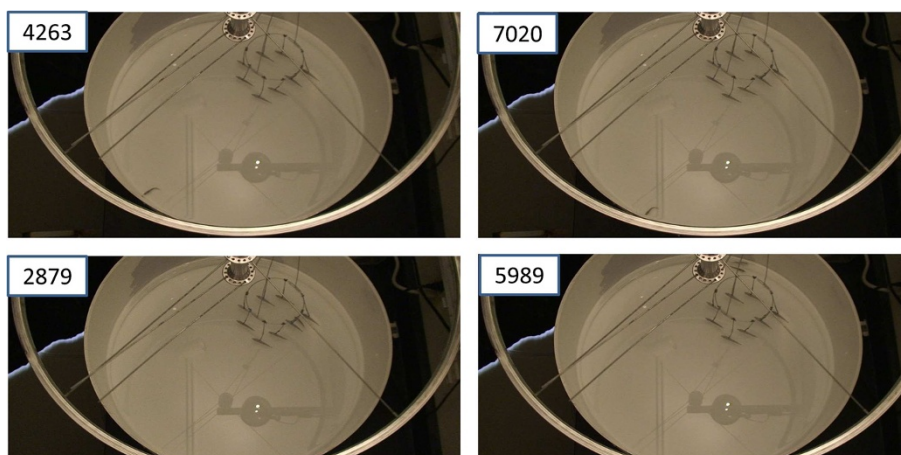
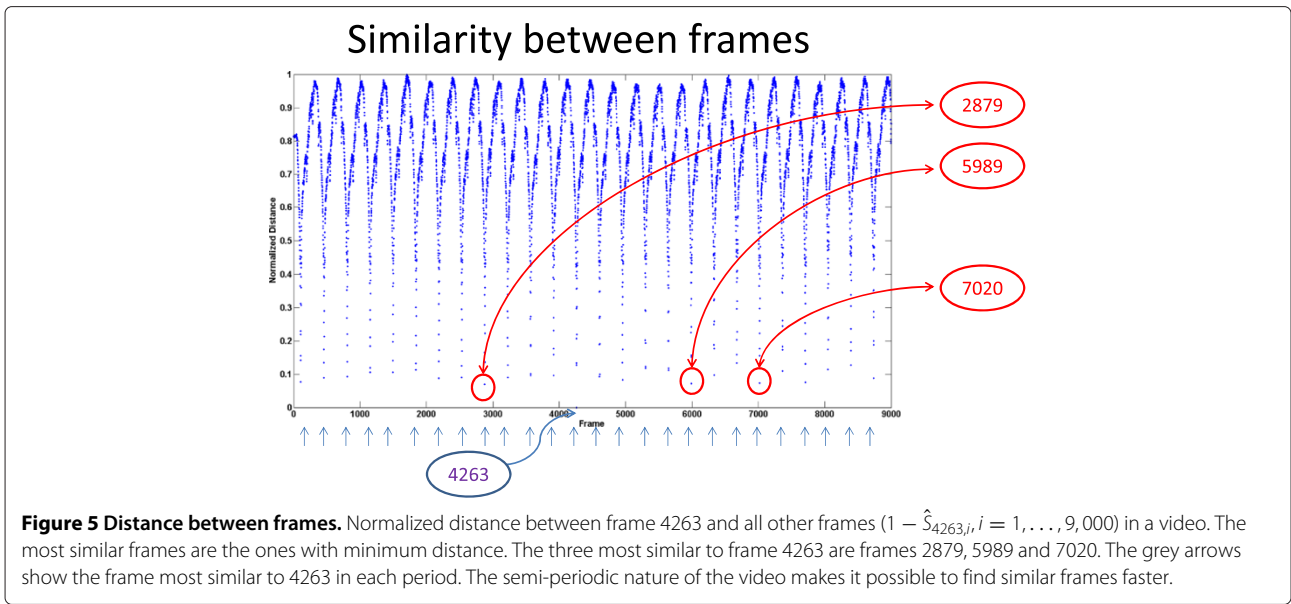


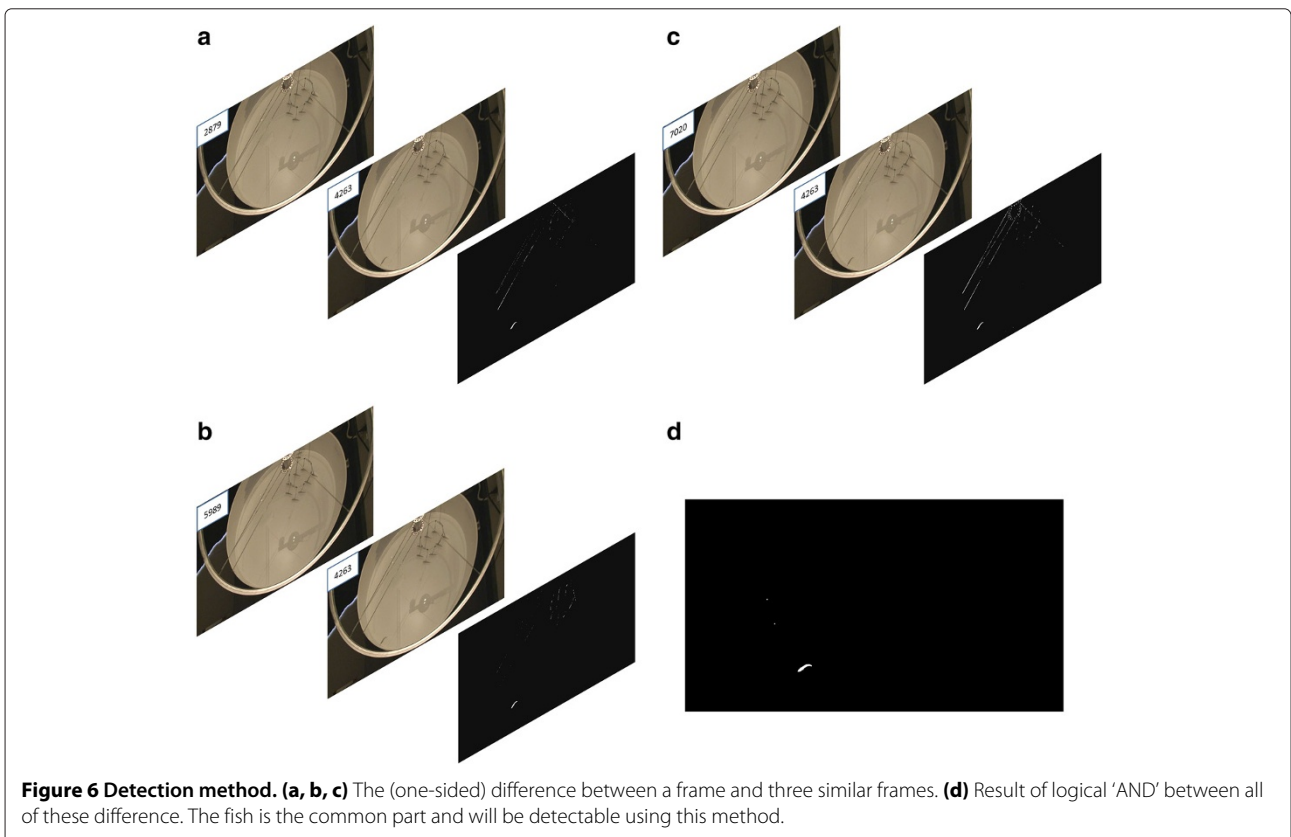
Figure 4 Similar frames. Four frames that have the minimum distance with each other. As can be seen, the position of the school model, wires and poles is almost the same in these frames, whereas the position of the real fish is different between frames.



Real fish detection

We want to build a background model for each frame such that the only 'foreground' would be the real fish. This means we want to have the model school, poles and wires as background.

One useful property of the videos from our system is that the model school is turning around almost periodically; thus, for each frame, there are some other 'similar' frames in the video in which the position of the model school, as well as poles, wires and even shadows are almost



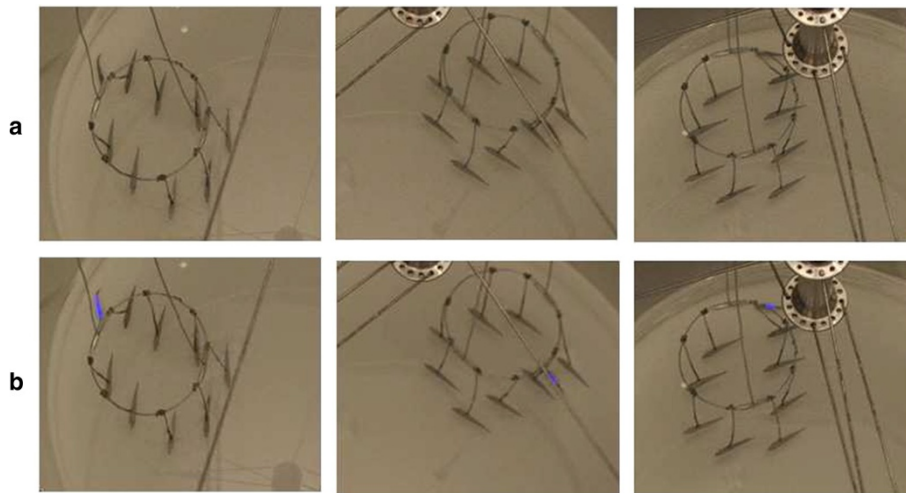


Figure 7 Detection results. (a) Original frames. (b) Processed frames in which detected fish are coloured blue. The fish is very close to the model, and yet the proposed method can detect it.

the same. Figure 4 shows this property; as one can see in the illustrative frames, the position of the model school is almost the same. We exploit this specific feature of these videos to build a background model for each frame using the similar frames that exist in the whole video. So, instead of using the neighbouring frames (neighbour in terms of time), we search the whole video to find the frames that are similar to the current frames. Our proposed approach for background modelling for videos has some similarities with the NL-means algorithm described in [14]. In [14], for denoising a pixel, instead of just using the neighbours of the pixel or local pixels, all other pixels in the entire image that are similar to the current pixel are used. The measure of similarity is based on the intensity value of a square neighbourhood of fixed size.

Our similarity measure is based on the absolute distance between frames. More precisely, S_{f_1, f_2} , the similarity score between frame f_1 and f_2 is defined as

$$S_{f_1, f_2} = 1 - C \times \sum_{i=0}^w \sum_{j=0}^h |I_{f_1}(i, j) - I_{f_2}(i, j)|$$

in which h and w are the height and width of the region of interest, respectively, C is a normalization factor and $I_f(i, j)$ is the intensity value of the pixel (i, j) which is between $0 - 255$ at frame f . To keep S_{f_1, f_2} between 0 and 1, we choose C to be $(255 \times w \times h)^{-1}$.

Since the area of the real fish is only about 0.1% of the whole image, the position of the fish does not make that much contribution to the value of the similarity score. This means that frames that are similar to each other have the same or very similar background (see Figure 4). To speed up the process of calculating the similarity score between frames, each frame is summarized as a vector

of Haar-like features [15,16] that can be computed very efficiently using an integral image [17]. In this case, the similarity distance is

$$\hat{S}_{f_1, f_2} = 1 - \hat{C} \times \sum_{k=0}^L |V_{f_1}(k) - V_{f_2}(k)|$$

in which V_f is a vector containing L rectangular Haar-like features and \hat{C} is a normalizing constant $((L \times 255)^{-1})$. Using feature differencing is faster for two reasons. First, for calculating distance between frames using feature vectors, we need to perform L subtractions, whereas using the difference of the frames themselves, we need $w \times h$ subtraction operations. Second, reading from a compressed AVI file is slow if the frames that are grabbed are not consecutive. By having the feature vector, we make a short signature for each frame with which we can compare frames quickly. Since we are doing the comparison operation around 500 times for each frame, the efficiency of this step is important (see the 'Implementation and results'

Table 1 Detection performance in five video segments, with 1,000 frames each

Segment number	Number of MD	Number of FD	Number of CD	Precision(%)	Recall(%)
1	32	0	968	96.8	100
2	54	4	942	94.5	99.5
3	131	1	868	86.9	99.8
4	33	0	967	96.7	100
5	25	0	975	97.5	100
Average	55	1	944	94.5	99.9

Number of missed detections (MD), false detections (FD) and correct detections (CD) as well as precision and recall rates are shown.

section). For our application, it is sufficient to use a small Haar-like feature space, i.e. the first-order feature, which is the average value of a rectangular region; We used rectangles with a size of 20×20 pixels in the region of interest which is inside the tank (of size of 500×500); thus, $L = 625$. Figure 5 shows the normalized distance ($1 - \hat{S}_{f_1, f_2}$) between frame 4263 and the rest of the frames in a sample video. As indicated, the three closest frames are 2879, 5989 and 7020 which are shown in Figure 4. For each frame, after ranking the similarity scores, we pick the N frames that have the highest scores; we used $N = 3$. The background for the current frame is then calculated using these frames. For calculating each change mask, we subtract frame 4263 from other frames and keep

only positive values. Since the fish is dark, the real fish in frame 4263 is detected while the real fish in other frames is ignored. Doing a logical 'AND' between change masks removes the water waves and other non-periodic changes in the image. Finally, we filter the components in the change mask based on their size and remove those components that are much smaller or larger than the real fish. Figure 6 shows this process and the output result for frame number 4263.

Implementation and results

We implemented our method in C++ and using OpenCV library. We have a pre-processing block in which the Haar-like features as well as the position of the model

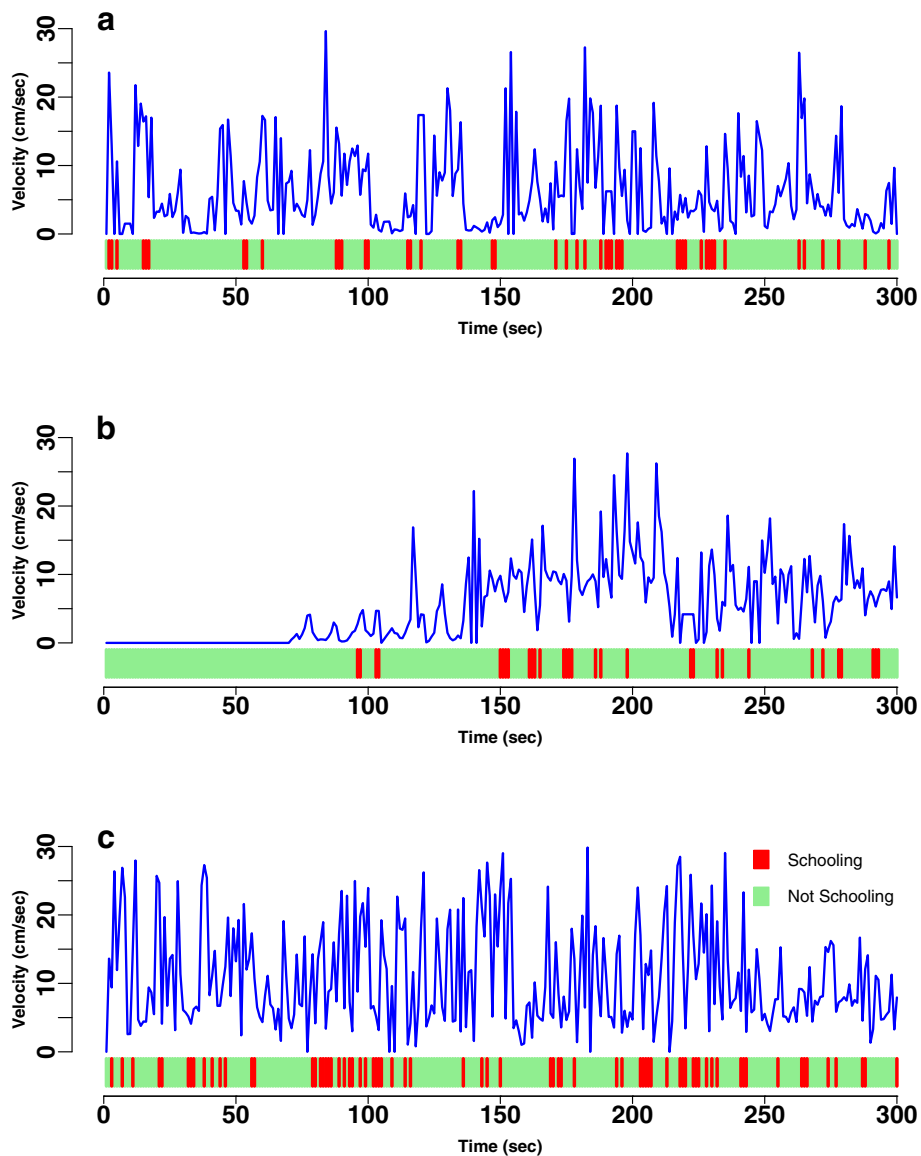


Figure 8 Schooling annotation results. Speed of movement and schooling behaviour for three sample videos (a, b, c). Red bars indicate inferred periods of schooling.

fish are extracted at each frame. In the processing step, we use extracted features to identify the similar frames for each frame and detect the fish as described in the methods section. Since the model school is moving semi-periodically, we can limit our search space to find similar frames and search in a limited number of frames instead of searching in all frames. In our set-up, the model school turns almost 25 times during the 5-min video (approximately 9,000 frames). As mentioned, the period of turning is not constant and differs between and within videos. By assuming a constant period of 350 frames per turn, we find frames in other periods that should be the most similar to the current frame; we then add the 10 frames before and after to the search space. Thus, instead of searching all 9,000 frames, we find the most similar frame by looking at around 500 frames. This expedites the processing of the videos. Finally, in the post-processing block (implemented in the R language), we look at the extracted trajectory of the fish from the model school and annotate each frame using the distance of the fish and model school as well as the speed of the fish.

The most important part of the problem is detecting the fish. Figure 7 shows the result of real fish detection in three difficult situations. The detected area is indicated in blue in Figure 7b. This shows that our method is able to find the foreground or real fish, even in situations with partial occlusion (see Additional file 1).

To quantify the performance of our algorithm, the detected object was indicated in an output video (as is shown in the sample video we have provided), and videos were watched frame by frame to see if the fish was detected correctly. We did this process of verifying on five segments of video of length 1,000 frames. Table 1 shows the performance of the proposed method in terms of the number of missed/false detections. On average, the precision of detecting the fish is 94.5% and the recall rate is very close to 100%. This shows that our detection algorithm works effectively. The method is based on the assumption that there are frames in the whole video in which the position of the model school and poles etc. are very close to the current frame, and by finding them, we can detect the fish in the current frame. However, if there are no frames similar enough to the current one, due to an unusual position of the model fish in the current frame, detecting the fish in that frame will fail. This situation can happen if the whole set-up shakes due to an external force or motor glitch. That is what has happened in segment 3 in Table 1.

We present the result of processing three sample videos with the proposed method. Videos are recorded in a controlled environment with fixed lighting conditions. The assay tank was illuminated with indirect lighting from a 60-W incandescent lamp. The resolution of videos is 960×540 , and all are recorded at 30 fps. For each frame,

the distance between the model and the fish and the speed of the fish are obtained. If the distance between the fish and the model is less than a predefined threshold (5 cm) and the speed of the fish is more than a threshold (2 cm/s), we identify that frame as schooling. There are frames in which the fish is occluded. However, handling occlusion in our case is fairly easy since we only have one target. We can estimate the position of the fish in occluded frames by linear interpolation between two known frames. Since occlusion usually does not last more than a few frames, this gives us a reasonable trajectory of the fish. Figure 8 shows the result of quantifying speed and schooling behaviour. As can be seen, the patterns of schooling and activity differ between individuals. To compare the result of our method with human annotation, we manually annotated ten different experiments, and in each video, the total amount of schooling time was recorded. The comparison between manual and automated annotations is shown in Table 2.

For each video, what we are ultimately interested in is the proportion of time in which the fish schools. Each video lasts 300 seconds, and for each second, we determine if the fish is schooling. This results in two vectors of 0 and 1 (0 for not schooling and 1 for schooling), one for manual and one for automated annotation. To assess the concordance between the manual and the automated annotation, we used the Kappa statistic [18]. Values of Kappa can be at most 1, with larger values corresponding to better agreement between human and machine; observed values are given in Table 2. To determine the significance of the Kappa statistic for each experiment, we produced 1,000 permutations of the automated annotation and computed the observed value of the Kappa

Table 2 Comparing automated and manual schooling time (in seconds) for 10 experiments, each of which lasts 5 min

Trial number	Automated schooling time	Manual schooling time	κ
1	62	47	0.82
2	108	109	0.49
3	0	0	-
4	154	154	0.62
5	194	214	0.49
6	245	260	0.43
7	112	115	0.38
8	90	69	0.46
9	124	166	0.53
10	218	219	0.75

The statistic Kappa (κ) is used to assess the concordance between manual and automated annotation (there was no schooling behaviour observed in trial 3 by either manual and automated scoring; Kappa is undefinable since its denominator is zero.) See the text for further details.

statistic for the comparison between the human annotation and the permuted one. The observed value of Kappa was compared to the values obtained under the permutation procedure. In all experiments, the observed value was larger than the largest simulated statistic; this corresponds to a nominal p value of 0.001, confirming the agreement between the manual and automated annotation.

Conclusions

We have proposed a method to automate the quantitative analysis of stickleback schooling behaviour. We exploit the semi-periodic nature of the videos to build an accurate background model for each model. Since we are processing recorded videos, our background modelling algorithm does not need to be causal; however, it can be extended for causal systems, e.g. real-time applications. The proposed method enables us to detect the fish in difficult situations, for example, when the fish is very close to the model and/or is partially occluded. Most modern online tracking methods rely on the visual features and/or motion model of the targets [6,7]. These approaches would fail in the frames in which the actual fish is swimming close to the models since they are similar in appearance and movement pattern. If a switching between the real fish and one model fish happens, this might lead to tracking the model throughout the rest of the video, thereby giving a much higher schooling score to the real fish. This leads to another advantage of the proposed method: since the detection in each frame is independent of the neighbouring frames, detection errors will not propagate to the other frames. Using our approach, we can find the important parameters of schooling behaviour. This enables us to screen many individuals with different genotypes efficiently and conduct association studies between genotype and schooling behaviour. Moreover, the new definition of background can be used in situations where the moving part of the background is predictable or periodic, for example, in detecting an object in assembly lines that use robotic arms with repetitive moves.

Additional file

Additional file 1: SticklebackTracking.avi - sample video. This video shows one typical experiment that has been processed. Detected fish is indicated in blue and a red circle shows the position of model school at each frame.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

AKG and CLP designed the schooling assay, and AKG performed the experiments. RA and ST designed the video analysis method and RA implemented the method. RA and AKG wrote the paper. All authors read and approved the final manuscript.

Acknowledgements

Research reported in this publication was supported by the National Human Genome Research Institute of the National Institutes of Health under awards number P50HG002790 (RA, ST) and P50HG002568 (AKG, CLP), and National Science Foundation grant IOS 1145866 (AKG, CLP). The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health or the National Science Foundation.

Author details

¹Program in Molecular and Computational Biology, University of Southern California, Los Angeles, CA 90089, USA. ²Division of Human Biology, Fred Hutchinson Cancer Research Center, Seattle, WA 98109, USA. ³DAMTP, University of Cambridge, Cambridge CB3 0WA, UK.

Received: 31 January 2013 Accepted: 26 September 2013

Published: 9 November 2013

References

1. N Tinbergen, The curious behavior of the stickleback. *Sci. Am.* **187**, 22–26 (1952)
2. MA Bell, SA Foster, *The Evolutionary Biology of the Threespine Stickleback*. (Oxford University Press, Oxford, 1994)
3. RJ Wootton, *The Biology of the Sticklebacks*. (Academic Press, London, 1976)
4. DM Kingsley, CL Peichel, in *Biology of the Three-Spined Stickleback*, ed. by S Ostlund-Nilsson, I Mayer, and F Huntingford. The molecular genetics of evolutionary change in sticklebacks (CRC Press Boca, Raton, 2007)
5. AR Wark, AK Greenwood, EM Taylor, K Yoshida, CL Peichel, Heritable differences in schooling behavior among threespine stickleback populations revealed by a novel assay. *PLoS ONE*. **6**, e18316 (2011)
6. A Yilmaz, O Javed, M Shah, Object tracking: a survey. *ACM Comput. Surv.* (2006). doi: 10.1145/1177352.1177355
7. S Hare, A Saffari, PH Torr, Struck: structured output tracking with kernels. *IEEE International Conference on Computer Vision, Barcelona 6–13 Nov* (2011)
8. M Piccardi, Background subtraction techniques: a review. *IEEE Int. Conf. Syst. Man Cybern.* **4**, 3099–3104 (2004)
9. K Toyama, J Krumm, B Brumitt, B Meyers, Wallflower: principles and practice of background maintenance. *ICCV*. **1**, 255–261 (1999)
10. Z Wu, TH Kunz, M Betke, Efficient track linking methods for track graphs using network-flow and set-cover techniques. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Colorado Springs, 20–25, June 2011
11. Y Sheikh, M Shah, Bayesian modeling of dynamic scenes for object detection. *PAMI*. **27**, 1778–1792 (2005)
12. AB Chan, V Mahadevan, N Vasconcelos, Generalized Stauffer-Grimson background subtraction for dynamic scenes. *Mach. Vision Appl.* **22**, 751–766 (2011)
13. RO Duda, PE Hart, Use of the Hough transformation to detect lines and curves in pictures. *Commun. ACM*. **15**, 11–15 (1972)
14. A Buades, B Coll, JM Morel, A non-local algorithm for image denoising. *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*. **2**, 60–65 (2005)
15. CP Papageorgiou, M Oren, T Poggio, A general framework for object detection. *Sixth International Conference on Computer Vision (ICCV 98)*, Bombay, 4–7 Jan 1998
16. P Viola, M Jones, Robust real-time face detection. *Int. J. Comput. Vis.* **57**, 137–154 (2004)
17. FC Crow, Summed-area tables for texture mapping. *Proc. SIGGRAPH*. **18**, 207–212 (1984)
18. J Cohen, A coefficient of agreement for nominal scales. *Educ. Psychol. Meas.* **20**, 37–46 (1960)

doi:10.1186/1687-5281-2013-61

Cite this article as: Ardekani et al.: Automated quantification of the schooling behaviour of sticklebacks. *EURASIP Journal on Image and Video Processing* 2013 **2013**:61.