ALGORITHMS FOR TRACKING OF FISH PATH USING IMAGE PROCESSING

S. Abdelaziz[1], A. Al-Nuaimi[2], P. Chugh[3], M. D. Bui[1], and P. Rutschmann[1]

[1]Institute of Hydraulic and Water Resources Engineering, Technische Universität München - Germany

[2]Institute for Media Technology, TU München - Germany

[3]Indian Institute of Technology, Kanpur - India

E-mail: shokry@bv.tum.de

Abstract

Understanding the fish behavior as a function of hydraulic conditions is the key for a good design of the fishway and other fish barriers. For that we need a method to extract the fish movement over time and relate it to changes in flow velocity, depth and other hydraulic conditions.

In this study, video (image) processing techniques are applied to track the fish movement in a controlled test environment. Here we specifically use the technique motion detection through background estimation and subtraction. We further improve the results. Furthermore we present a color-histogram based segmentation approach to track the identified fish over time and hence achieve robust fish tracking. Finally we apply signal processing techniques for post processing of the produced movement curves to filter out outliers.

The algorithms are applied on two test datasets, one of them captured in the Laboratory of Hydraulics in Obernach of the Technische Universität München (TUM). Initial results are very promising and a number of improvements are furthermore suggested.

Keywords: fish path tracking, image processing technique

Introduction

The study of fish behavior has always been a topic of interest amongst hydraulic engineers as well as biologists. Recent studies have emphasized on the in-depth assessment of fish behavior for efficient design of the hydraulic structures that concern them. Traditionally, several methods are used to capture a moving fish track. These included the mark and recapture technique (Blank 2008, Warren and Pardew 1998), casting nets for collecting and examining fish, human underwater observation and photography (Rouse 2007, Schlieper 1972), combined net casting and acoustic tracking (sonar) (Brehmera et. al, 2006) and, more recently, human hand-held video filming (Spampinato et. al,2008). Videler and Weihs (1982) used a high-speed camera (up to 200frames/s) fixed over the middle of a tank to measure the fish Burst speed. From these frames, the displacement of the head and the tip of the tail in a horizontal plane is traced by digitizing the position every 0.01s. Xingqiao et al. (2008) proposed a particular and real-time image processing application to detect pathological changes to fish. They acquired Images from a video source periodically using specialized control software. They then used the concept of area-square to find out the changes of pathological fish. On the other hand, Wu and Zeng (2007) introduced a video system for tracking a free-swimming fish two-dimensionally. They used two CCD (charge coupled device) cameras to obtain three-dimensional kinematic parameters of the tail and pectoral fin of the fish in forward, backward and turning swimming modes.

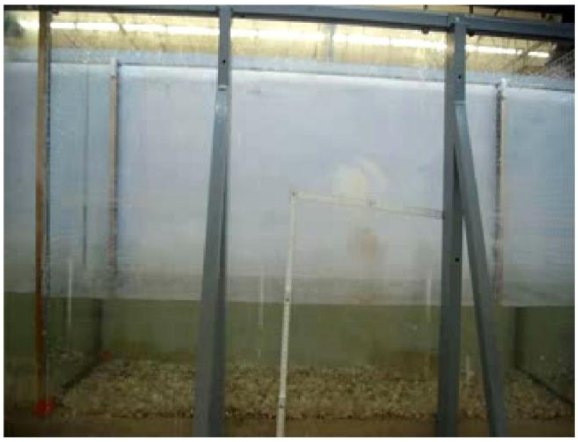
In this study, we investigate the conceptual applicability of image processing techniques for tracking fish path in a real dataset captured by off-the-shelf video cameras, emphasizing low acquisition cost

Datasets

Two sets of data are used in this paper. The first was captured in the Laboratory of Hydraulics in Obernach of the Technische Universität München (TUM) while the other was downloaded previously from internet http://www.youtube.com/watch?v=0_jbrSgTmb0&feature=related.

In the first dataset, a primary investigation of the fish movement inside a Plexiglas channel that is 50cm wide, 80cm deep and 12m long with gravel bed material was done with two video cameras as shown in Figure 1. One of them was set up horizontally while the other was in the vertical direction. The fish was allowed to move only in 1.5m distance by two grids. the fish length was around 29.4 cm. Figure 2 depicts the camera locations with which the dataset was captured.

Figure 1: places of the two cameras



a) Vertical plane camera

b) Horizontal plane camera

Figure 2: Side and plan views taken from dataset 1

The second dataset shows the movement of 7 fish with different colors in a fish tank. This dataset was primarily used to validate the proposed methods and develop them since it posses a higher quality and clarity than that of dataset 1. Figure 3 shows an image from the video of the second dataset.

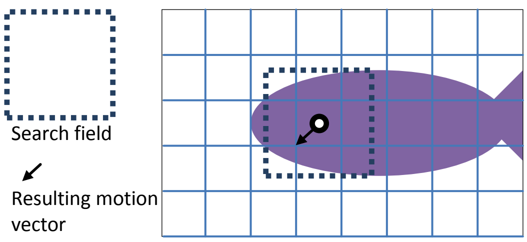
Figure 3: An Image from the video of dataset 2

Methodology

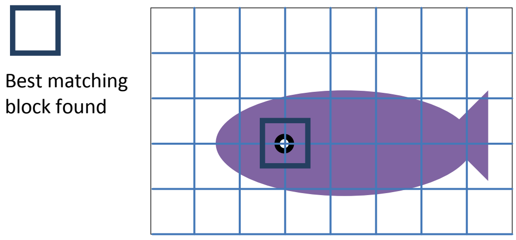
In this section we will explain the applied and tested methods. The concepts will be explained and some details of implementation will also be provided, since in image processing algorithms have to be adapted to the specific application.

Motion Detection using the Motion Vector Technique

Block segmentation of video frames combined with motion vectors make up the most fundamental components of almost all video compression standards (Jenq-Neng, 2009). Videos are made up of a series of images (called frames) taken at short time intervals such that the human eye cannot perceive the discrete transitions from frame to frame (typically 25fps or 30fps). Hence consecutive frames are temporally correlated over localized spatial domains. This temporal correlation is exploited in video compression: The idea is to segment each frame into a series of blocks (typically of size 8x8 or 16x16 pixels) as shown in Figure 4 (a). It is highly probable that a visually similar block can be found in the immediate (spatial) vicinity in the previous frame. Hence a similar image patch is looked for in a search field. Once such a similar block is found, the video codec simply stores the spatial offset values from the center of the block (the so-called *motion vector*) and no longer needs to store all pixel values. Such a motion vector shown in Figure 4(b) can effectively capture small-scale motion and encode them efficiently.



c) Block segmented frame ($t=t_n$) with motion vector shown for one of the blocks. The motion vector encodes the relative displacement of the most similar image patch found in the frame $t=t_{n-1}$



d) Most similar image patch found in the search area shown in Figure 2(a) to the block encompassing the fish eye.

Figure 4: Using motion vectors to estimate object movement

We used motion vectors to capture fish movement. The idea we exploited is again the high temporal correlation between consecutive frames. In essence, since a fish movement is constrained and changes within certain limits, we expected that a block search with motion vectors can sufficiently capture the fish movement if the background is static and the only moving objects are the fish.

The motion vectors are typically calculated for each color channel separately (Jenq-Neng, 2009). For our application, however, it suffices to perform motion estimation on the gray-level intensity image. For that, the gray-level intensity at any location *(u, v)* at time *(t)* can be simply calculated from the three color intensity values I_R (for red), I_G (for green) and I_B (for blue) using following equation (Solomon & Breckon, 2011):

$$I(u,v,t) = 0.3I_R(u,v,t) + 0.59I_G(u,v,t) + 0.11I_B(u,v,t)$$

To calculate the motion vectors (and hence the amount of movement) for a particular block, the most similar patch of the same size in the previous frame has to be determined. The most similar patch in the previous frame is determined as being the one within the search field that has the smallest sum of squared differences (SSD). The SSD between any two patches of image intensity of size *NxN* is defined as:

$$SSD = \sum_{i=1}^{N}\sum_{j=1}^{N}(I_{curr}(i,j) - I_{cand}(i,j))^2 \qquad (1)$$

In this specific case I_{curr} denotes the block in the current frame for which a similar patch in the previous frame is being looked for and I_{cand} being the currently considered patch. In other words, the most similar patch is the one that minimizes the difference energy.

The motion vector technique has a fundamental limitation: it does not always perform *true motion estimation* since the focus is on finding the most similar patch only. Hence, sometimes the motion vectors do not really represent the true motions of the objects over time. Moreover, the search range is typically limited to a small rectangular area around the center point of the block whose match is to be found. This is necessary since the SSD is computationally expensive if performed using the *exhaustive search* (ES) technique (*O (N2)*). More efficient search algorithms exist (Manikandan, Vijayakumar, & Ramadass, 2006) nevertheless the search field would have to be adapted according to the speed of the fish and the video sampling rate adding complexity to the algorithm.

Motion Detection through Frame Difference Calculation

Due to the issues presented in the motion vector technique, we decided to use another method of motion detection based on calculating differences between consecutive frames. This is justified if the background is static which can be insured using a proper capturing arrangement (static cameras, appropriate lighting, etc.). In this case, as can be seen in Figure 5, only the area covered by the fish in two consecutive frames will have values different from *0* after the subtraction operation since the pixels outside that area will remain the same. Hence by identifying the locations where this is the case and looking for the center of mass of this area, the approximate location of the fish can be identified.

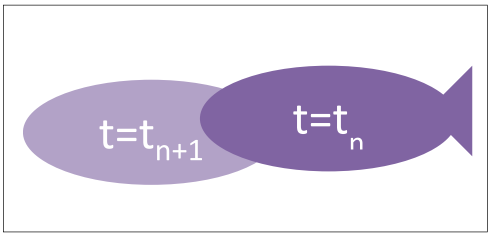


Figure 5: Fish location at time t=tn and t=tn+1. Only colored areas will have values different than 0 in the difference image.

The difference is calculated using the gray-level image representation as in the motion vector technique case. The calculation of the gray-scale image is followed by three processing steps:

a) Calculating the *absolute difference image* between frame n and frame n+1 as shown in Figure 6(a).
b) Converting the gray-scale image to a binary image by applying thresholding Figure 6(b). The threshold is calculated using the method described in (Otsu, 1979).
c) Filtering the binary image using an *opening* operation Figure 6(c). This type of *morphological filtering* is effective in filtering out small particles stemming from noise and smaller insignificant objects (Russ, 1995).

a) Absolute difference image

b) Threshold difference image

c) Filtering using image opening

Figure 6: processing steps for moving object detection.

The problem with this technique is that the retrieved location always deviates from the true one as it lies somewhere between the old and new fish locations (The centroid of the area where the absolute difference is bigger than zero). Furthermore, after implementing this technique, we noticed an oscillatory behavior of the tracking point. Hence we applied following refinements.

Background subtracted Motion Detection

We have seen that the location is always biased behind the fish when it is moving forward, and that a static or slow-moving fish cannot be tracked. To solve these two problems we decided to estimate the background first and use this as a reference frame out of which each video frame is subtracted. Insuring that the fish is the only moving object, this helps in identifying the true fish location. This method also works when the fish is suddenly static or moves slowly.

To estimate the background we again use morphological filtering. More specifically we filter each pixel along the *t* dimension with a structuring element of width 41 pixels (the same pixel in the previous 20 frames and the next 20 frames). The median value is taken as background value at that particular frame. This is exemplary shown in Figure 7: the fish body is temporarily seen on the location *(u, v)* and changes the intensity of the pixel over a short period of time. If the structuring element length is chosen appropriately, this temporary change in the intensity will be filtered out and the prevailing intensity value (that of the actual background) will be chosen as the background intensity value.

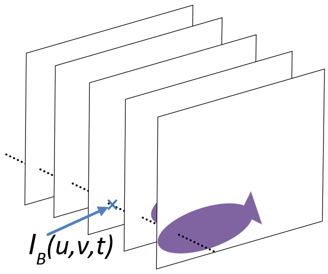


Figure 7: median filtering for background estimation.

In Figure 8, the estimated background at time t=10 frames is shown for the actual test sequence captured in our lab.

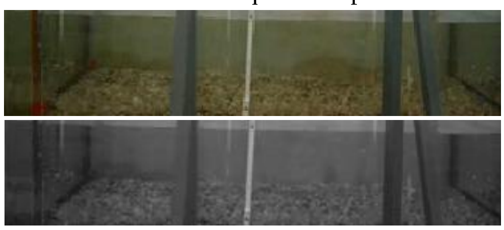


Figure 8: Frame #10 from test video (above) and estimated background using pixel-wise median filtering (below).

Using Color Filtering for Robust Tracking and Multiple Fish Tracking

The background subtraction motion detection technique delivered the required improvement. Nevertheless we needed a method to filter outliers (sudden jumps in tracking point). Also we needed a method to associate tracking points to multiple fish tracked over time. The problem can perhaps be seen in Figure 9 more clearly: While individual fish can be detected and tracked (red dots representing centroids of detected moving regions), these dots cannot be assigned to the different fish over time.

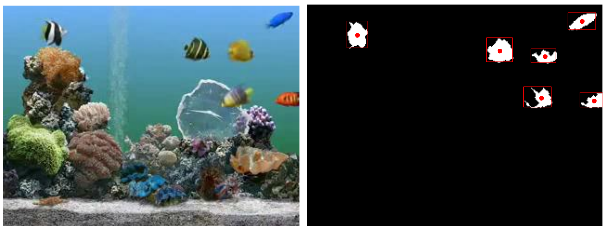


Figure 9: detected moving regions (with centroids and bounding boxes).

Simply assigning the found locations to the fish based on the area size does not work well. Figure 9, however, shows that color might be a good information source upon which to identify which location belongs to which fish, Matching colors can be achieved using matching of color histograms. Such histograms are shown for two of the fish in Figure 10. It can be seen that the histogram of the blue channel for the blue fish has most energy concentrated in the higher values range. A similar thing can be obsereved for the orange fish where the histogram of the red channel has a big part of its energy in the high values range.

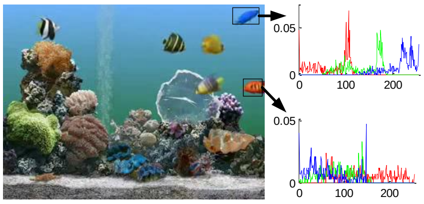


Figure 10: fish color histograms

The idea is to extract the color histogram of each fish to match them to color histograms extracted from the bounding box regions out of each frame. This way, each detected region can be uniquely assigned to one fish; namely that with the most similar color histogram.

The only thing that remains to be explained is how color histograms can be compared. There exists a number of metrics to compare color histograms. Two notable metrics are the Kullback-Leibler divergence and the Bhattacharyya distance. We have chosen to use the Bhattacharyya distance where the similarity between two discreet histograms expressed as vectors of length M can be calculated using:

$$D_B(\boldsymbol{h_1}, \boldsymbol{h_2}) = -\ln\left(\sum_{i=1}^{M} \sqrt{\boldsymbol{h_1}(i)\boldsymbol{h_2}(i)}\right)$$

h1 and ***h2*** being color histograms expressed as vectors having M bins (in our case we use M=256 bins since intensity values for each channel range from 0 to 255 since our videos have 8 bit color depth). of Since there are three channels per image patch and consecuently three histograms, there needed to be a way to combine them together. The most simple is to perform distance calculation between patches using each channel histogram seperately and then calculate one distance score by linearly combining the individual distances using a weighting equal to that found in Equation 1.

Method Validation

Dataset 1 is used to verify the Alogrithm without the color filtering technique, since only one fish is moving. The two videos included in this datasets show that the fish rested in its place for some time before starting to move quickly with high speed. The recorded video has a frame rate of 25fps and spans a period of 10.5s. . Figure 11 shows three frames of the resulting tracking. The successful fish tracking outcome can better be seen in the video we provide at (www.wb.bv.tum.de) under Research Video Section (Dataset_1.avi). Figure 12 shows the smoothed fish path as captured in the vertically aligned camera

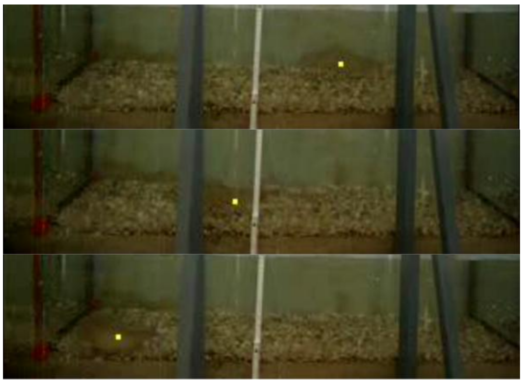


Figure 11: tracking the fish in dataset1.

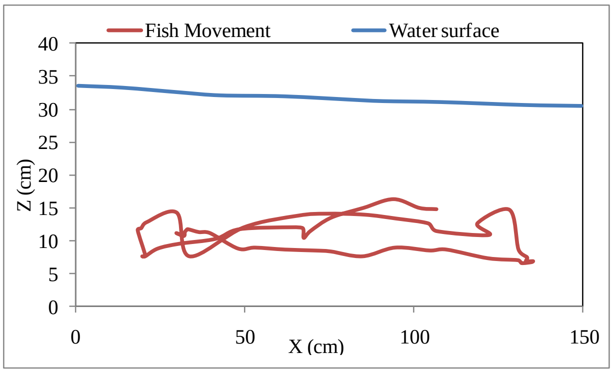


Figure 12: The resultant vertical fish movement for dataset 1.

While Figure 12 shows the fish track, it can hardly be used to determine the accuracy of the used method. To verify that the results we got are accurate, the true fish location (the ground truth) was manually determined and recorded by human observation. For that we programmed a Matlab tool that displays the frames and captures the location which the user clicks on. With this tool we manually recorded the true fish location at every 5th frame (corresponding to 0.2s time space). The determined accuracy is best seen in an error histogram showing the distribution of absolute location error in the estimated location compared to the ground truth location. The relative error histogram was calculated relative to the fish length as shown in Figure 13. The mean error in X and Y directions are 0.1490L ± 0.1471L and 0.0635L ± 0.0537L respectively (where the tolerance values indicated by the ± sign represent the standard deviation). The main reason for this small difference is the presence of some supporting beams (see Figure 2(a)) which partially occlude the scene.

Unfortunately, the tracking method could not work with other camera (the one looking at the fish from above) due to the randomly changing lighting reflections resulting from the water surface turbulence. Hence, we suggest to capture a new dataset where the camera capturing the horizontal plane is placed below the fish tank to avoid this problem. We are confident however, that the method would work just as well if a proper dataset is captured for the horizontal plane and thus providing us with the 3D location of swimming fish over time.

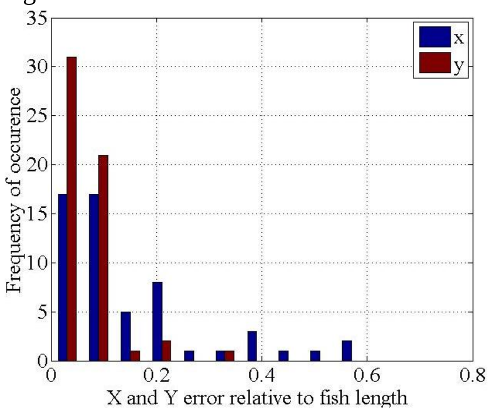


Figure 13: Relative error Histogram

To also verify that the suggested improvement of using color filtering (histogram matching) does bring the expected gains, we ran the the refined method on dataset 2. More specifically, we provide the color histogram of the blue fish in the fish tank as template and with that successfully track and distinguish the blue fish. Figure 14 shows two frames of the resulting tracking. The resulting tracking can better be seen in the video on (www.wb.bv.tum.de) under Research Video Section (Dataset_2.avi)

Figure 14: tracking the blue fish using the color histogram aided fish tracking

Conclusion

In this paper the methods of motion vector-based motion estimation and frame difference-based motion estimation are investigated towards their usability to track swimming fish for the purpose of automatic path capturing. The motion vector technique is found to be of limited performance. The frame difference technique is improved by performing background estimation and subtraction using morphological filtering, resulting in accurate fish tracking in a realistic dataset captured at our lab. We furthermore add color filtering for fish identification in a multiple fish tracking scenario and demonstrate the principal capability to track and identify different fish.

Although we achieve good results in tracking fish in realistic scenarios, we consider this paper as a foundation for more elaborate methods that could result in more accurate and robust tracking by incorporating suitable post-processing techniques (outlier removal and adaptive smoothing).

References

Blank, M.,(2008). Advanced studies of fish passage through culverts: 1-d and 3-d Hydraulic modeling of velocity, fish energy expenditure, and a New barrier assessment method. PhD thesis, Montana State University.

Brehmera, P., Do Chib, T., Mouillotb D., (2006) Amphidromous fish school migration revealed by combining fixed sonar monitoring (horizontal beaming) with fishing data. Journal of Experimental Marine Biology and Ecology. Vol. 334, Issue 1, pp. 139-150.

Jenq-Neng, H., (2009). Mutliemdia Networking: from Theory to Practice. Cambridge: Cambridge University Press.

Manikandan, M., Vijayakumar, P., & Ramadass, N. (2006). Motion Estimation Method for Video Compression - an Overview. International Conference on Wireless and Optical Communications Networks. Bangalore: IEEE.

Otsu, N. (1979). A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man and Cybernetics , 62-66.

Rouse W. (2007). Marine Biology Research Experiment: Population Dynamics of Barnacles in the Intertidal Zone. May 27, 2007.

Russ, J. C. (1995). The Image Processing Handbook. CRC Press.

Schlieper, C. (1972). Research methods in marine biology. University of Washington Press, Seattle, 1972.

Solomon, C., & Breckon, T., (2011). Fundamentals of Digital Image Processing. John Wiley & Sons.

Spampinato, C., Chen-Burger, H., Nadarajan, G., Fisher, R., (2008): Detecting Tracking and Counting Fish in Low Quality Unconstrained Underwater Videos. VISAPP (2) 2008: 514-519

Videler, J., and Weihs, D., (1982). Energetic advantages of burst-and-coast swimming of fish at high speeds. J. exp. Biol

Warren, M.L., and Pardew, M.G., (1998). Road crossings as barriers to small-stream fish movement. Transactions of the American Fisheries Society 127: 637-644.

Wu, G., and Zeng, L., (2007). Video tracking method for three-dimensional measurement of a free-swimming fish. Science in China Series G: Physics Mechanics and Astronomy Volume 50, Number 6, 779-786

Xingqiao,L., Jiao, G., Feng, J., and Dean, Z., (2008). Using MATLAB Image Processing to Monitor the Health of Fish in Aquiculture. Proceedings of the 27th Chinese Control Conference