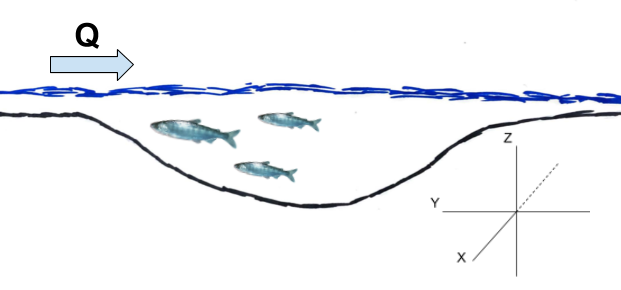
Videogrammetry Metrics for Assessing Fish Foraging Behavior: Implications for Behavioral Ecology and Management

Keane Flynn

4/30/2019

# **Introduction**

Fish foraging behavior has long been a topic of natural observation, wonder, and recreational curiosity (Walton 1655). In recent years, this area of observation has extended beyond the realm as an enigmatic phenomenon and taken its place as a series of qualitative metrics in fisheries ecology. But with recent advances in computer technologies and programming languages, we are capable of extending these qualitatives behavioral metrics into quantitative values which can be further mathematically and statistically analyzed to show distinct differences in behavior. These differences in foraging behaviors have been shown to be linked to changes in physical habitat characteristics that might be indicative of decreasing habitat quality for foraging fish (Rossi et al. 2019 in prep; Steingrímmson 2012; Fausch et al. 1997).   
Using the three dimensional videogrammetric program VidSync, quantifying previously generalized qualitative/quantitative values has become a relatively easy process that has the potential to offer results with more statistical significance than previous behavioral observation methods (Neuswanger 2014). This method makes use of two underwater cameras mounted in a fixed location relative to one another recording simultaneous video of fish behavior. Doing so allows for an almost entirely non-invasive way to record fish behavior, whereas many current methods rely on snorkel observations. Once the videos have been processed, an X, Y, Z coordinate, a timecode, and a behavioral observation type are assigned to each point created for the video sample in the program. These coordinates, timecodes, and observations can then be exported and processed into code-based programs to create statistical and graphical representations of the data collected.  
This program offers a number of advantages to traditional fish foraging observations, the largest of which being ease of quantifying spatial habitat use metrics within sub-millimeter accuracy, which will be expanded upon in the methods section. Additionally, this method of observation allows to expand forage observations beyond focal animal sampling, which has been the most common method for recording fish behavior in natural experiments, to other methods of behavior sampling methods such as sampling all occurrences of behaviors, sequence sampling, and scan sampling (Altmann 1974). While previous field observation methods could only account for one of these observation methods based on physical restrictions, replayable video allows for a higher confidence in behavioral observation assignment. Being able to track multiple individuals throughout a video sample allows for a complete assessment of behavioural observations for all individuals in the sample, hence creating a larger sample size for improved statistical analysis.  
In previous research, these estimations of distances travelled, area of habitat, size of subjects, specific foraging patches, etc. conducted during snorkel surveys were capable of assigning numeric values to these metrics, but with a higher degree of uncertainty (Nakano et al. 1999; Nielsen 1992; Steingrímsson & Grant 2008). While some of these methods have proved useful for changes in habitat usage, these methods are inherently based on location estimations and lack critical pieces of information for properly classifying habitat use. The issue with these previously used methods is that the metrics for habitat use is generally two-dimensional, which disregards many phenomenon such as physical habitat characteristics and location of prey input that might affect location of fish along the Z-axis (Figure 1).



**Figure 1** *Profile view of a pool-riffle sequence with X, Y, Z planes relative to a fish maintaining positive rheotaxis.*

We hope to address some of the potential uses of these methods, but also realize that there are situations in which it might not be in the best interest to use these methods: they involve more post-capture editing and analyzation than that of snorkel observations, plotting movement movement points and territories can be more complex than doing so using the animal movement extension in ArcView (Hooge & Eichenlaub 2000). Given the resources to use this method are available, we find that it has many attributes that make up for the shortcomings of previously used methods. In this paper, we will elaborate upon the metrics that can be accurately derived using VidSync and the ecological implications of each.

## Load Libraries

# Importing VidSync CSV File

VidSync\_Test\_Data <- readr::read\_csv(file = "Porter\_BACI\_RoachRun\_5July2018\_Part1.csv",   
 skip = 2,  
 col\_names = c("objects", "event", "timecode", "time\_sec", "X", "Y", "Z", "pld\_error", "projection\_error", "nearest\_camera\_distance", "screen\_coordinates"),  
 col\_types = "cccdddddddd") %>%   
 mutate(subsample = as.numeric(str\_extract(objects, "\\d"))) %>%  
 mutate(fish\_ID = as.numeric(str\_extract(objects, "\\h\\d{1,2}"))) %>%  
 mutate(species = str\_extract(objects, "Omykiss|Okisutch")) %>% #Note that the use of regular expressions in this line are to be specified to whatever species are recorded in your experiment  
 select(subsample, fish\_ID, species, time\_sec, X, Y, Z, objects, event) #All other data vectors were excluded for the sake of neatness and what is useful in the metrics below

## Warning in rbind(names(probs), probs\_f): number of columns of result is not  
## a multiple of vector length (arg 1)

## Warning: 475 parsing failures.  
## row # A tibble: 5 x 5 col row col expected actual file expected <int> <chr> <chr> <chr> <chr> actual 1 1 <NA> 11 columns 10 columns 'Porter\_BACI\_RoachRun\_5July2018\_Part1… file 2 2 <NA> 11 columns 10 columns 'Porter\_BACI\_RoachRun\_5July2018\_Part1… row 3 3 <NA> 11 columns 10 columns 'Porter\_BACI\_RoachRun\_5July2018\_Part1… col 4 4 <NA> 11 columns 10 columns 'Porter\_BACI\_RoachRun\_5July2018\_Part1… expected 5 5 <NA> 11 columns 10 columns 'Porter\_BACI\_RoachRun\_5July2018\_Part1…  
## ... ................. ... .......................................................................... ........ .......................................................................... ...... .......................................................................... .... .......................................................................... ... .......................................................................... ... .......................................................................... ........ ..........................................................................  
## See problems(...) for more details.

# Nearest Neighbor Distance (NND)

NND is calculated by finding the nearest neighbor to each unique fish over the course of a subsample. The final NND column will represent the closest nearest neighbor for each fish and exclude any larger distances (in cm) so as to not skew the data away from any values that are the absolute nearest neighbor point over the course of the subsample. It is important to note that subsamples with only one observed fish will produce an NND that is infinite and should be dealt with accordingly when graphing/calculating statistics on these values.

NND\_Data <- VidSync\_Test\_Data %>%   
 filter(grepl("^Subsample.\*", objects)) %>%  
 filter(!grepl("^Length.\*", event)) %>%  
 filter(!grepl("^Attack.\*", event)) %>%  
 filter(!grepl("^Nip.\*", event)) %>%  
 na.omit() %>% #To remove any additional rows from the data set not associated with fish (i.e. surface shots)  
 arrange(time\_sec) %>%  
 group\_by(time\_sec) %>%  
 mutate(distance\_between\_X = X - lag(X, default = first(X))) %>%  
 mutate(distance\_between\_Y = Y - lag(Y, default = first(Y))) %>%  
 mutate(distance\_between\_Z = Z - lag(Z, default = first(Z))) %>%  
 mutate(nnd\_cm = sqrt((distance\_between\_X)^2 +   
 (distance\_between\_Y)^2 +  
 (distance\_between\_Z)^2)) %>%  
 group\_by(time\_sec) %>%  
 filter(!nnd\_cm == 0) %>%  
 filter(nnd\_cm == min(nnd\_cm)) %>%  
 ungroup() %>%   
 distinct(fish\_ID, .keep\_all = TRUE) %>%   
 arrange(subsample, fish\_ID, time\_sec) %>%  
 select(subsample, fish\_ID, time\_sec, nnd\_cm)

# Distance per Time

DistancePerTime\_Data <- VidSync\_Test\_Data %>%   
 filter(!grepl("Surface\_Shots.\*", objects)) %>%  
 filter(!grepl("^Length.\*", event)) %>%  
 group\_by(fish\_ID) %>%   
 mutate(distance\_travelled\_X\_cm = X - lag(X, default = first(X))) %>%  
 mutate(distance\_travelled\_Y\_cm = Y - lag(Y, default = first(Y))) %>%  
 mutate(distance\_travelled\_Z\_cm = Z - lag(Z, default = first(Z))) %>%  
 mutate(fish\_distance\_travelled\_cm = sqrt((distance\_travelled\_X\_cm)^2  
 + (distance\_travelled\_Y\_cm)^2  
 + (distance\_travelled\_Z\_cm)^2)) %>%  
 ungroup() %>%   
 group\_by(fish\_ID) %>%  
 mutate(distance\_cm\_per\_sec = fish\_distance\_travelled\_cm /(time\_sec - lag(time\_sec, default = first(time\_sec)))) %>%  
 filter(!distance\_cm\_per\_sec == Inf) %>%   
 arrange(subsample, fish\_ID, time\_sec) %>%  
 select(subsample, fish\_ID, time\_sec, distance\_cm\_per\_sec)

# Proportion of Forage Behaviors

ForageBehaviors\_Data <- VidSync\_Test\_Data %>%   
 filter(!grepl("Surface\_Shots.\*", objects)) %>%  
 filter(!grepl("^Length.\*", event)) %>%  
 mutate(Behaviors = if\_else(grepl("^Drift\_Forage", event), "Drift Forage",   
 if\_else(grepl("^Search\_Forage", event), "Search Forage",   
 if\_else(grepl("^Search\_Forage", event), "Search Forage",   
 if\_else(grepl("^Benthic\_Forage", event), "Benthic Forage",   
 if\_else(grepl("Nip|Attack", event), "Attack",   
 if\_else(grepl("^Surface\_Strike", event), "Surface Strike",   
 if\_else(grepl("^Movement", event), "Movement", "NA")))))))) %>% #Note that "nip" has since been changed to the definition of "attack" and attack to feint  
 arrange(subsample, fish\_ID, time\_sec) %>%   
 select(subsample, fish\_ID, time\_sec, Behaviors)

# Centroid to Forage Attempt Point

# Distance from Surface

# Length Calculations