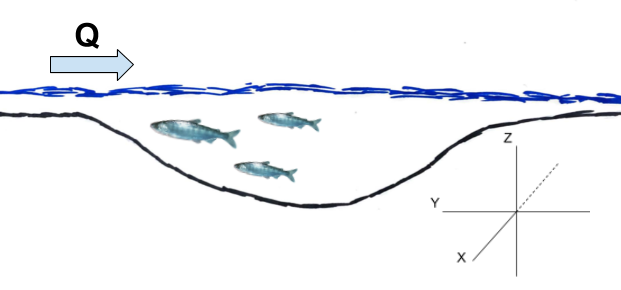
Videogrammetry Metrics for Assessing Fish Foraging Behavior: Implications for Behavioral Ecology and Management

Keane Flynn, Weston Slaughter, Gabriel Rossi, Mary Power, Stephanie Carlson

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).



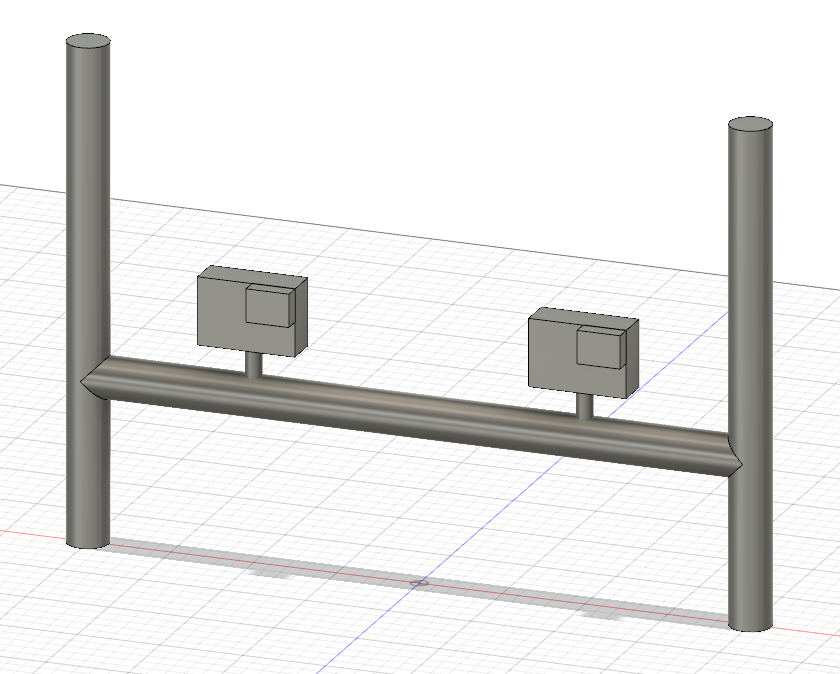
**Figure 1** Profile view of a pool-riffle sequence with X, Y, Z planes relative to a fish maintaining positive rheotaxis.

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.

# Methods

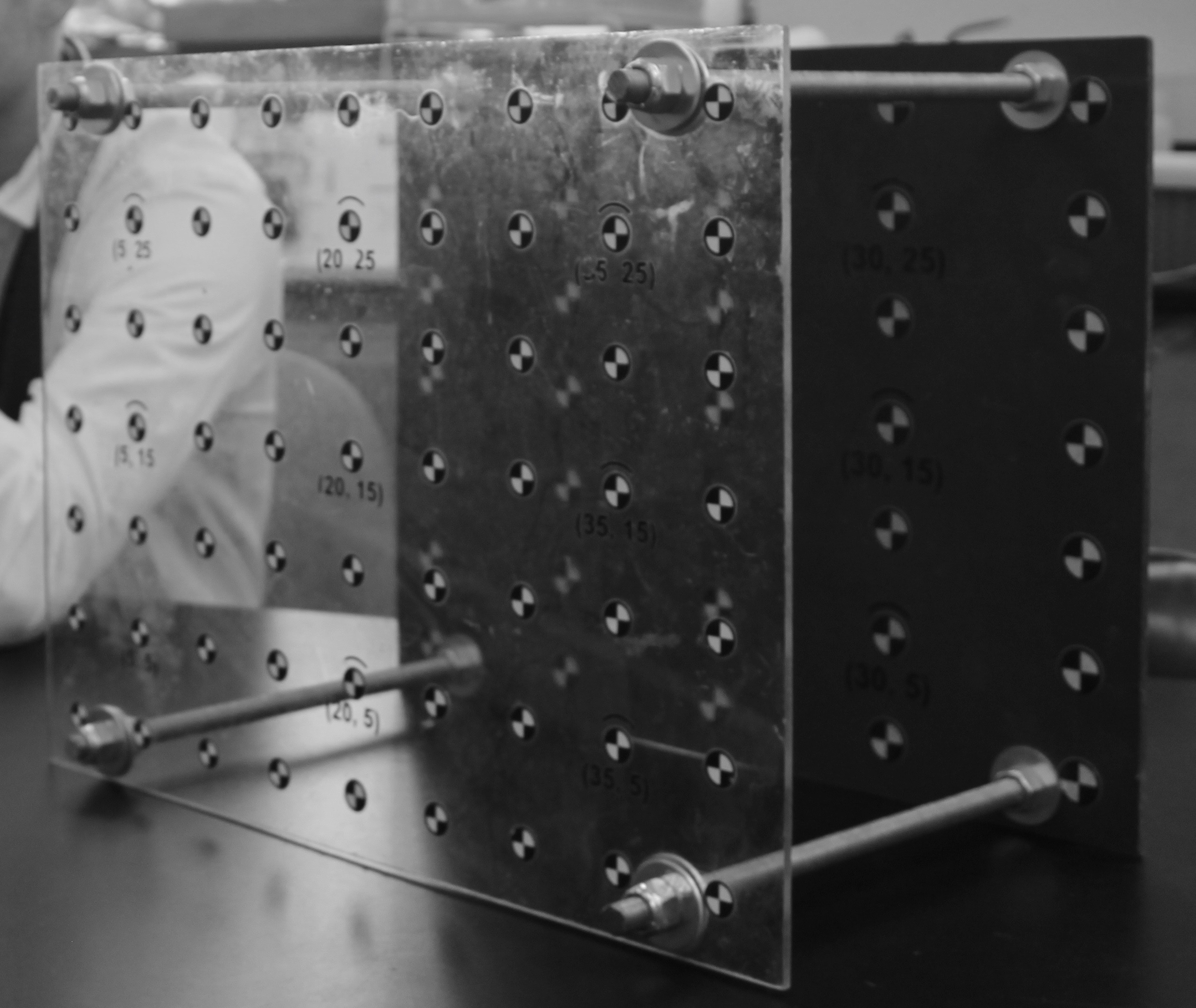
### *Streamside set up and use*

For underwater videography, we found that using GoPro cameras allowed for a relatively low-cost method of video capture as well as wide angle lens to capture all fish in the study area continuously. To properly mount these cameras for multiple samples at the same site, we found that having multiple mount points to the streambed via rebar hammered into the alluvium ensures that the same habitat area is recorded each time. To fix the cameras to the mount, a simple device made from PVC pipe and bicycle mounts covers the widest amount of habitat while minimizing area where the camera angles do not overlap (Figure 2).



**Figure 2** Fusion 360 schematic for how to mount GoPro cameras for VidSync applications

When recording behavior, especially in low-flow situations, it is important to enter the river to mount the camera from downstream to avoid increasing turbidity in the study area. That being said, this method should not be employed in areas of high turbidity as behavior observations will be abysmal. While it depends on the specific study, we recommend roughly 25-30 minutes of recording to ensure usable behavioral observations are recorded. Immediately following the recording period, someone must enter the creek with a calibration frame (from downstream to avoid increasing turbidity) and place it in front of the cameras where all points on the frame are visible (Figure 3).



**Figure 3** Calibration frame used in the VidSync program to calibrate the two videos to calculate positions in relativve space

### *Test Measurements*

Once the videos have been recorded, loaded into VidSync, and calibrated, this calibrated video must be tested to ensure that the measurements are valid (Neuswanger 2014; Neuswanger 2016). Using the known distances between points on the calibration frame, test measurements can be made along the X (front frame to rear frame), Y, and Z axes using the length function in the program to ensure that what you are measuring in VidSync matches empirical measurement data (Figure 1).

### *Video Period Selection*

The first 5 minutes of the video will be discarded to allow for an adjustment period to allow fish to resume normal foraging behavior (Nakano et al. 1999; Fausch et al. 1997). The video period being chosen will be randomly selected but first we must, by a quick scan sample, find the most fish-dense 10 minute period of the recording. In order, each 30 second interval of the video is then assigned an integer (1-20) and using a random number generator, 6 random integers (1-20) representing a unique 30 second period are selected to make up a total of 3 minutes of data supported behavioral observations (Neuswanger 2014; Rossi et al. 2019 in prep). Each of these 6 chosen integers will represent a subsample within the sample video to allow for a period of total observation similar to what has been previously used with this method, however it allows for a higher sample size while removing any potential bias in sampling period.

### *VidSync Use and Behavior Classification*

Once the desired subsample intervals are selected, the video has been calibrated and corrected for lens distortion, object and event types can then be created to classify subsamples and behavioral observations to be associated with recorded coordinate points (Neuswanger 2016). The object types can be used to represent respresent unique subsamples and will be exported into a vector named, "Object(s)" which will include the title of the object as well as the "index" which is used to represent an individual fish idenfication number. Additionally, the option to include a name with each object is given which will also be included in the "Object(s)" vector which should be used to identify species or specific individual fish if unique identification is possible. These object types should be saved as follows: Subsample\_1, Subsample\_2, Subsample\_3, Subsample\_4, Subsample\_5, Subsample\_6.   
The event types will be represented by the behavioral classifications assigned to each 3 second sample increment in each subsample, this length of time allows for classification of unique behaviors as well as tracking movement accurately with small temporal increments (Rossi et al. 2019 in prep). The behavior classifications, as well as additional points to be recorded, and their definitions have been used or adopted from prior behavioral studies with some modifications (Kalleberg 1958; Nielsen 1992; Nakano et al. 1999). They are as follows  
  
Benthic Forage: A distinct forage event targeted at the stream benthos.  
Drift Forage: A fish maintaining positive rheotaxis to target prey carried through the water column in a sit-and-wait fashion.   
Search Forage: A fish covering large foraging patches in search of non-benthic prey.  
Surface Strike: A fish making contact with the water surface to obtain prey.  
Aspirating: A fish visibly flaring their gills while minimally moving and making no forage attempts.   
Attack: A fish making contact with another fish in an act of territoriality.  
Feint: Adopted from a fencing term, a fish will charge another in an act of territoriality, but make no contact and quickly resume foraging thereafter.  
Movement: Any movment throughout the pool not focused on foraging.  
Length: A fork length measurement taken by creating a point at the nose and the fork of the caudal fin and using the three-dimensional distance formula to calculate length.  
Surface Shots: Points taken on the surface (usually of an easily identifiable floating object) to record a Z-plane value for use in determining the distance fish are holding from the surface.

# R-code and Examplary Data for Quantitative and Qualitative Behavioral Metrics

## Loading Necessary Libraries

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(readr)  
library(tidyverse)

## ── Attaching packages ───────────────────────────────────────────────────────────────────────────────────── tidyverse 1.2.1 ──

## ✔ ggplot2 3.0.0 ✔ purrr 0.2.5  
## ✔ tibble 1.4.2 ✔ stringr 1.3.1  
## ✔ tidyr 0.8.1 ✔ forcats 0.3.0

## ── Conflicts ──────────────────────────────────────────────────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(ggplot2)  
library(geometry)

## Loading required package: magic

## Loading required package: abind

library(rgl)  
library(hypervolume)

## Loading required package: Rcpp

library(zoo)

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

library(stringr)  
library(data.table)

##   
## Attaching package: 'data.table'

## The following object is masked from 'package:magic':  
##   
## shift

## The following object is masked from 'package:purrr':  
##   
## transpose

## The following objects are masked from 'package:dplyr':  
##   
## between, first, last

# Importing VidSync CSV File

VidSync\_Test\_Data <- readr::read\_csv(file = "VidSync-TestData.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

# Nearest Neighbor Distance

Nearest neighbor distance (NND) has historically been used as a metric for understanding inter/intraspecies interactions as well as making observations about general trends regarding animal density in a study region (Clark 1954). NND is calculated by finding the nearest neighbor to each unique fish over the course of a subsample. The final NND column will return a single NND value for each fish in a subsample and 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 of these values. The exemplary data is as follows:

NND\_Data <- VidSync\_Test\_Data %>%   
 filter(grepl("^Subsample.\*", objects)) %>%  
 filter(!grepl("^Length.\*", event)) %>%  
 filter(!grepl("^Attack.\*", event)) %>%  
 filter(!grepl("^Feint.\*", 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, nnd\_cm)

# Distance per Time

The exemplary data is as follows:

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

The exemplary data is as follows:

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("^Feint", event), "Feint",   
 if\_else(grepl("^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

The exemplary data is as follows:

ControidToForage\_Data <- VidSync\_Test\_Data %>%   
 filter(!grepl("Surface\_Shots.\*", objects)) %>%  
 filter(!grepl("^Length.\*", event)) %>%   
 group\_by(fish\_ID) %>%   
 mutate(median\_X = median(X),  
 median\_Y = median(Y),  
 median\_Z = median(Z)) %>%   
 mutate(distance\_to\_forage\_point\_cm = sqrt((X - median\_X)^2 +   
 (Y - median\_Y)^2 +  
 (Z - median\_Z)^2)) %>%   
 select(subsample, fish\_ID, time\_sec, distance\_to\_forage\_point\_cm)

# Distance from Surface

# Length Calculations

# Occupied Volumes

# Overlapping Volumes