Introduction

The primary currency of image analysis for abundance consists of count data. Targets are counted on a frame basis. Or in the case of video data, total counts may span a time interval. An often used metric of abundance in marine systems is MaxN, where this number represents the maximum number of targets observed at any one time during a sampling period. This is a conservative value, and represents a relative index that is thought to be proportional to true abundance. Volumetric density represents true absolute density, and therefore is a quantity that allows for direct comparisons between different systems and sampling methodologies. The StereoCamVolume package provides tools to estimate volumetric density for targets and provide methods to easily convert from frame-level counts to density. This is done by computing the effective sampling volume for a particular target species and set of viewing conditions, which once established can be used for a wide set of count data.

Volume estimation

Using a stereo-camera, targets can be ranged and their 3d position known, but only if they are visible in both cameras at the same time. To compute the volume of this joint view space, a numerical method is employed using the calibration parameters estimated using a standard calibration routine. This package requires a yml file that lists the parameters, and provides a utility script (accessory\_functions/convert\_calibration\_to\_yaml.R) to construct this yml file from three sources. These include the MATLAB camera calibration toolbox from Jean-Yves Bouguet (<https://robots.stanford.edu/cs223b04/JeanYvesCalib/>), the OpenCV library, specifically the Python implementation (pyOpenCv), and the SeaGIS stereo camera analysis suite (<https://www.seagis.com.au/>). There’s also an included helper function for MATLAB to translate the output from the built in MATLAB stereo camera calibrator app (<https://www.mathworks.com/help/vision/ug/using-the-stereo-camera-calibrator-app.html>) to the Bouguet toolbox, which can then be turned into the yml file via the utility script.

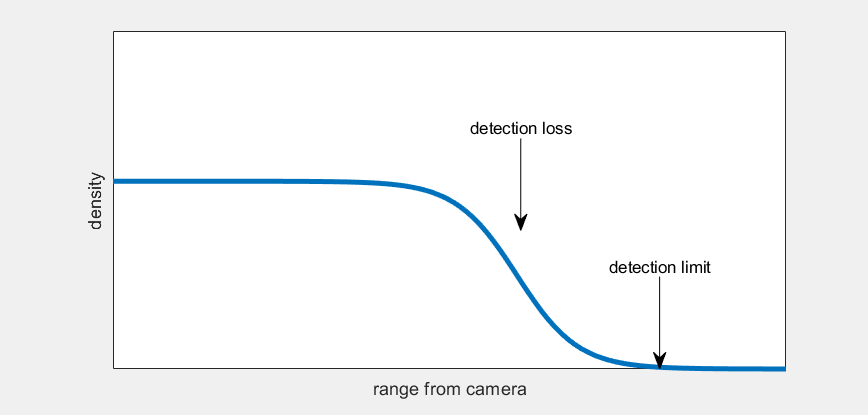
Once a calibration file is available, it needs to be read in using the yaml package in R and the command yaml.load(read\_yaml(“filename”)). The volume is estimated the get\_volume function.

get\_volume also requires a maximum field extent (max\_extent), and the grid\_size input, which specifies the grid density: smaller grid size values result in high grid density, which in turn can deliver higher precision at the cost of being more computationally intensive.

The process of estimating volume is detailed in Willliams et al., 2018. In brief, a grid of 3d points is created using the parameters above and encompassing the space occupied by the view fields of both cameras out to the specified maximum extent. These points are then projected onto the first camera’s virtual; image using the camera calibration coefficients. These coefficients contain both intrinsic parameters (e.g. those specifying the optical properties of each camera) based on the pinhole model, and extrinsic parameters to do with the location of the camera in the grid 3D space. The series of equations used for projecting 3D points into 2D image pixel coordinates are given in detail at <https://docs.opencv.org/4.x/d9/d0c/group__calib3d.html>. Grid points whose projected values (pg) satisfy 0>pgx<image width and 0>pgy<image height are considered “viewable” by the first camera. These viewable 3D grid points are then projected onto the second camera using its calibration coefficients, and the ones viewable by the second camera define the set that is viewable by both cameras. To find parameters that define how this joint volume changes with range, the number of grid points in a given 1 m range interval is counted and scaled by the volume that is represented by each point, defined as grid\_size^3. For a parallel camera setup, where both cameras are facing in the same direction and the optical axes can be said to be close to parallel, the change in volume is an exponentially expanding curve that follows a second order polynomial function. Other systems where cameras are turner in toward each other may require a different volume function, but a polynomial form should be able to represent most expanding situations.

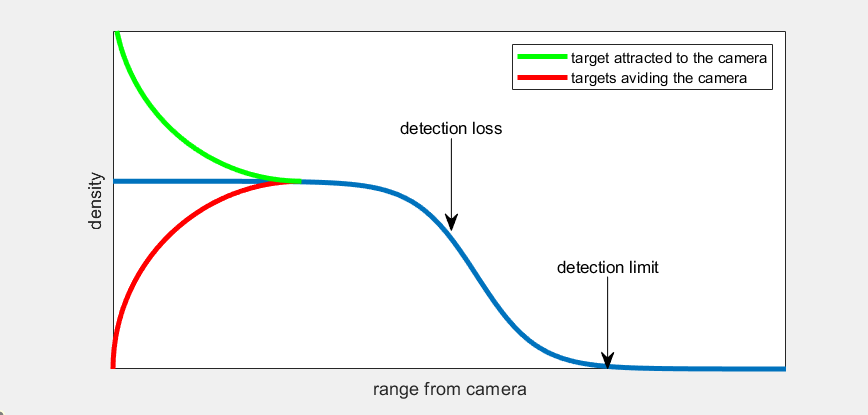
Theoretical volumetric density estimation

Cameras view targets of interest such as fish or birds in a 3D space and while the images contain information on where targets are located in the image pixel coordinates, a second camera is needed to stereo-triangulate the 3D positions of these targets. Once 3D positions are known one can estimate the volumetric target density, or how many animals occupy a particular volume. Because the ability to resolve and identify targets is range dependent, there is a necessary for field limit to target detection. This might appear as a reduction in density with range, but is actually a property of the sampling system, and is dependent on many factors, including the optical setup, visibility, and identifiability of specific target classes. One can imagine a theoretical model where there exists some steady-state density in the observable space, and this is starts being reduced at ranges where the ability to detect targets is likewise reduced. This function is unlikely to be hard edged, and may reflect a process dependent on target aspect.



In this scenario, the ultimate goal is to determine the steady-state local density to ultimately evaluate real abundance. At a certain range we start to underestimate density, and ultimately we reach a limit at which no targets can be detected.

As sampling devices often elicit behavioral responses from animal targets, such as fish, one can imagine these processes impacting density in the near field of the camera as well, which could be positive (attraction) or negative (avoidance) reactions. These add complexity to the relationship between range and density, as shown in this example:



In the attraction case, it may be difficult to correctly estimate true local density. However, in the avoidance case, if the avoidance reaction does not extend to range at which detection loss starts to occur, we can still make some inferences about local abundance.

Detection probability function

The StereoCamVolume provides the tools to estimate a detection probability function, which can include both the detection loss function and behavioral components such as avoidance. This detection function is combined with the volumetric function described in the previous chapter to derive an effective volume, which can then be used to convert frame-level counts to frame-level density. This process requires several steps, which are outlined below.

The detection function is assumed to be proportional to the change in observed density as a function of range from the camera, as shown in the illustrations above. To derive the pattern of range-dependent change in density, the number of targets in a given range interval is divided by the volume in that interval. To allow for higher data amounts, the target ranges are combined from a large set of frames, with the assumption that the relationship between density and range is not dependent on any given frame. The interval size is determined by the user. A function is then fit to the density per range values. Several function options can be used. The base is a simple logistic function, referred to as “single logistic”, which closely resembles the blue line in the theoretical examples. This function was chosen as it approximates the data patterns well and not because the underlying data are binary. To capture additional complexity of avoidance behavior, a double logistic model can be fit where one function captures the ascending arm of the change in density closer to the camera, and a second logistic captures the detection decline in the far field same as with the single logistic detection function. Finally, if the density per range data appear strongly symmetrical, a normal curve can be used, although this is not assuming the underlying data are consistent with a normal distribution but is only useful for modeling the shape of the data. All functions are fit using an additional parameter for scale, which allows the functions to follow the arbitrary density values derived from combining observations across a set of individual frames. The functions can then be used to approximate a probability of detection by setting the scale parameter equal to 1.