Developing an Automated Detection Algorithm to quantify surface swarms of moon jellies, *Aurelia* Spp., and brown sea nettles, *Chrysoara fuscescens*, prey for Pacific leatherback turtles, *Dermochelys coriacea*

By

ERIN E. FROLLI

CAPSTONE PROJECT

Submitted in partial satisfaction of the requirements for the degree of

BACHELOR OF SCIENCE

in

MATHEMATICS

in the

COLLEGE OF SCIENCE, MEDIA ARTS, AND TECHNOLOGY

at

CALIFORNIA STATE UNIVERSITY,

MONTEREY BAY

Approved:

(Rachel Esselstein, Ph.D.)

Mathematics Capstone Committee

2012

Acknowledgments

Special thanks to Barry Hansen, Aspen Helicopters Inc., Bill Watson, Lisa Web, Luke Spence, Rachel Esselstein, Scott Benson, the NOAA Aerial Survey Team, and many others who made this project possible. Aerial surveys were conducted under National Marine Fisheries Service permit 1596-02, and National Marine Sanctuary permit MULTI-2008-003. Funding was provided by the Dr. Earl H. Myers and Ethel M. Myers Oceanographic and Marine Biology Trust and the National Science Foundation, Undergraduate Research Opportunities Center, CSUMB, Dr. William Head CSU-LSAMP Senior Alliance project, NSF grant no. HRD-0802628.

Abstract

The leatherback turtle (*Dermochelys coriacea*) has been listed as endangered since 1970 and critically endangered since 2006. Critical habitat has been established on nesting beaches in the US Virgin Islands. Recently, there has been a proposal to revise the designation of critical habitat to included foragging grounds in the US West Coast. Including foraging grounds for a species critical habitat has never been done for this species, so in order to include foraging grounds, the availability of the most common prey will be used to assess the need for the revision. Along the US West Coast the most common prey for the leatherback are large medusa that aggregate in naritic waters. The goal of this study is to develop an efficient method to quantify surface aggregations of large medusa (jellies) to detect significant annual changes in the two most common leatherback prey, the brown sea nettle (*Chrysoara fuscescens*) and moon jelly (*Aurelia* Spp.). An automated detection algorithm was developed to help monitor the abundance and availability of these two species from a novel photograph sampling method done over central coast waters off California. The output of this algorithm was used to help researchers understand the minimum extent of the medusa population off the US West Coast.

Contents

0.1 Introduction

The leatherback turtle (*Dermochelys coriacea*) is the largest living reptile and has been listed as endangered throughout its range by the U.S. Endangered Species Act in 1970 [5, 14]. Being listed as endangered, the leatherback population is under extreme risk of extinction in the future. In 2000, published estimates of the mature adult female population had been reduced in size by 70% of the global population [14]. Due to the major reduction in mature individuals over a ten year period, the World Union Conservation Red List increased the criteria and listed the leatherback turtle as critically endangered in 2006 [14].

With major decreases in the global population, conservation efforts have been made to secure the future of the Pacific leatherback by eliminating harvesting of eggs on nesting beaches. In 1979, critical habitat was established for leatherback nesting beaches on Saint Croy in the U.S. Virgin Islands. In 2009, revision to the critical habitat designation was proposed to include the U.S. West Coast as a foraging ground because of the dense aggregations of jellies [3, 5]. This proposal is the first of its kind, and includes Monterey Bay and surrounding waters [3, 5].

The justification for this new proposal is due to the Pacific leatherback's unique migration across the Pacific from nesting grounds in Indonesia to foraging grounds in neritic waters off California, Oregon, and Washington during late summer and fall [3]. Large scyphozoan jellies are the primary diet of the leatherback turtles, and are found in large swarms of the U.S. West Coast. In particular, two species that are known to be part of the leatherback diet are the brown sea nettle (*Chrysoara fuscescens*) and moon jelly (*Aurelia Spp.*). Leatherbacks only stay in the area to feed before moving south into warmer water [3, 5].

Aerial surveys off the coast of California indicate leatherback turtles and jellies are primarily found in retention zones [3, 8, 9]. For the assessment of designating critical habitat for foraging grounds, the availability of jellies as prey plays an essential role. There has to be a large enough stock of jellies to sustain the population now and in the future [2]. In the past, aerial observers have used a quantitative scoring index to track the density of surface aggregations of jellies during leatherback surveys. However, no quantitative estimates of jelly abundances are available in central California. Thus, there is a need for quantitative abundances for surface swarms of jellies to confirm qualitative estimates given by aerial observers.

Several methods have been used to obtain jelly abundance including camera sleds, net trawls, side-scan sonar and bottom facing acoustic echosounders [13, 10, 12, 7]. However, these methods are not reliable in the upper few meters of the water due to orientation of the equipment [13, 10, 11]. This leaves us blind to the surface aggregations of jellies where leatherbacks concentrate their dive time [3].

The goal of this study is to develop an inexpensive, efficient method to quantify surface aggregations of large jelles to detect significant annual changes in their abundance to help define critical habitat for leatherback foraging grounds in the U.S. West Coast. With the development of a sampling and interpolation method to quantify surface aggregations of jellies from aerial transects, the abundance among years is precluded until such a method is developed. This is essential for refining critical habitat of leatherback turtles, as well as documenting potential increases in scyphozoan jellies which could drastically alter the central coast ecosystem.

0.2 Body

Aerial line-transect surveys were conducted for leatherback turtles in the fall of 2010 and 2011 off central California using a twin engine fixed wing aircraft, The Partenavia, P 68-OBS ”Observer”. The aircraft was flown at a height of about 620-690 ft (above the surface of the water) with a speed of 90-100 knots following methods described by Forney et al. 1991 and Benson et al. 2007 [6].

In addition to existing protocols, photographs of jellies were taken with a 10.1 megapixel digital still camera, Canon Rebel XTi, DS126151. A custom mount was used to secure the camera just forward of the belly observer with the lens extending through a small hole in the pexiglass of the observing window to prevent distortion (Fig 1). Photographs were taken opportunistically when jelly aggregations were sited and each frame was taken so that there was no overlap of individual jellies. The GPS location and elevation, from the radar altimeter for the aircraft, was recorded for photographs whenever possible by the data recorder.

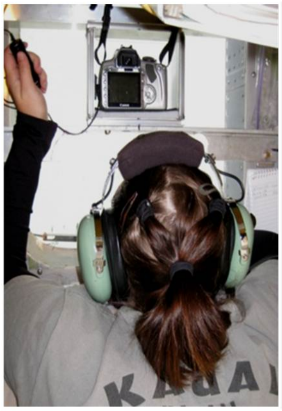

Figure 1 is an aerial observer taking photos of jellies during a transect

Photographs were categorized by the observer qualitative jelly density categories (low, moderate, high density of jellies in each aggregation). Photographs were eliminated from the data set by factors such as glare, no known jellies, and fog. Finally, to minimize autocorrelation, 30 of the remaining photographs were chosen from different days and times of observation. Fifteen photographs for each species of jelly were chosen for analysis. Each photograph was analyzed using two programs, ArcGIS and MATLAB. An automated detection algorithm in MATLAB was used to create an automated count. We consider the ArcGIS count to be the proxy for the actual total number of jellies observed at the time of the photograph. This number will be used to ground-truth the automated count in MATLAB.

Each photograph (n=30) was imported into ArcGIS and given the projected coordinate system NAD 1983 UTM Zone 10N. A geodatabase was created for each photograph and projected to the same coordinate system. Because distinct individuals can be identified by shape and color, jellies were marked with a point feature to obtain a proxy of the total number of jellies in the photograph. All photographs were analyzed by the same person to minimize human error. After each photograph is thoroughly checked for jellies the total number for each was recorded. The software program, R was used to find 100 randomly selected jellies.

A vector N_{Photo} is made:

$$N_{Photo} = [1, 2, 3, ...n]$$

where n is the total number of jellies counted in the photograph N_{Photo} can be then randomly sampled from to obtain 100 random jelly values.

For the photographs containing the moon jelly counts the *Select Attribute* function will be used to locate each of the 100 randomly selected jellies, then measure individual bell diameters in pixel width (assuming horizontal orientation) and record in the attribute table. The bell diameters are used to obtain the average surface area, in pixels, for each species.

$$\mathit{Surface\ Area}_{Asp.} = \pi r^2$$

where $r = d/2$ and d is the mean bell diamete,r in pixels, of all moon jellies in each photograph for that year.

For the photographs containing the brown sea nettle counts the *Select Attribute* function will be used to locate each of the 100 randomly selected jellies, and then the height and width of individual jellies were measured in pixel width and recorded in the attribute table. Brown sea nettles take a rectangular shape in photographs due to their orientation being mostly on their side with oral arms extended so the surface area for this species is

$$Surface\ Area_{cf} = WL$$

where W is the average width and L is the average length, in pixels, of moon jellies in each photograph for that year.

0.2.1 Detection Algorithm

In order to obtain data from multiple photographs in an efficient amount of time, an automated detection algorithm was designed (Appendix A). The algorithm consists of a custom code written to automate the summation of pixels within the specified color range. This color range was calibrated for water color, glare and other oceanographic conditions.

First the photograph is loaded into MATLAB as a large matrix, I, that consists of red, green, and blue (RGB) values for each pixel in the photograph

$$I = \begin{bmatrix} rgb_{(1,1)} & \cdots & rgb_{(1,m)} \\ \vdots & \ddots & \vdots \\ rgb_{(n,1)} & \cdots & rgb_{(n,m)} \end{bmatrix}$$

To find the specific color range, five areas of interest for a specific species color based on depth, water type, and glare were extracted from I such that

$$RGB_{Values} = \{C_1, C_2, C_3, C_4, C_5\}$$

where

$$C_i \subset I$$

and where

$$C_i = \begin{bmatrix} rgb_{(1,1)} & \cdots & rgb_{(1,6)} \\ \vdots & \ddots & \vdots \\ rgb_{(6,1)} & \cdots & rgb_{(6,6)} \end{bmatrix}$$

such that

$$RGB_{Values} = \begin{bmatrix} r_{(1,1)} & g_{(1,2)} & b_{(1,3)} \\ \vdots & \ddots & \vdots \\ r_{(24,1)} & g_{(24,2)} & b_{(24,3)} \end{bmatrix}$$

. Vectors were then made for each color (RGB)

$$r = [r_{(1,1)} \cdots r_{(24,1)}]$$

$$g = [g_{(1,2)} \cdots g_{(24,2)}]$$

$$b = [b_{(1,3)} \cdots b_{(24,3)}]$$

The maximum and minimum value for each vector was then found using the maximum and minimum function. So X is the set of all values in *I*, and for all $x \in X$ where there exists a maximum value $x^{\circ} \in X$ if and only if $x \leq x^{\circ}$. Similarly there exists a minimum value $x_{\circ} \in X$ if and only if $x \geq x_{\circ}$. Once the maximum and minimum values were found for all three vectors a FOR END loop was created to search the entire photograph matrix for values. Each individual value was evaluated using the following requirements.

$$RGB_{(m,n)} = (x_1, x_2, x_3)$$

where $x_1, x_2,$ and x_3 are the red, green, and blue values for that term (pixel) in matrix *I*. So if $RGB_{(m,n)}$

$$x_1 \geq r_{\circ} \text{ and } x_1 \leq r^{\circ} \text{ and...}$$

$$x_2 \geq g_{\circ} \text{ and } x_2 \leq g^{\circ} \text{ and...}$$

$$x_3 \geq b_{\circ} \text{ and } x_3 \leq b^{\circ} \text{ and...}$$

where $r_{\circ}, g_{\circ},$ and $b_{\circ}$ is the red, green, and blue minimum values and $r^{\circ}, g^{\circ},$ and b° is the red, green, and blue maxamum values. Once the pixel meets the above requirments the red value, x_1, was changed to 225 (the brightest red value) and then tallied for the total count. When finished, MATLAB then displays the image with the translucent bright red pixels for verification that the program is selecting appropriate cells in the matrix that contains "jelly" color. The total number of pixels was used to obtain the total number of jellies registered by the program

$$T_{jellies} = \frac{N_{Pixels}}{\mu_{sa}}$$

where N_{Pixels} is the total number of pixels that met the color requirements and μ_{sa} is the average surface area per species of jelly. Mean ± standard deviation was used to create the maximum and minimum number of jellies that can be detected by the algorithm.

The total number of jellies using the manual ArcGIS count method was plotted against total number of jellies derived from the automated MATLAB program. A regression is performed to determine the relationship between the two methods and a t-test will determine if the regression slope was significantly different from 1.0.

0.2.2 Analyzing the Aerial Observer Counts

The area of the still photographs is calculated to determine the density of jellies (Fig. 2). Assuming that the height of the plane is perpendicular to the transect line and the angle is determined by the zoom of the camera, the actual distance width of the photograph can be determined.

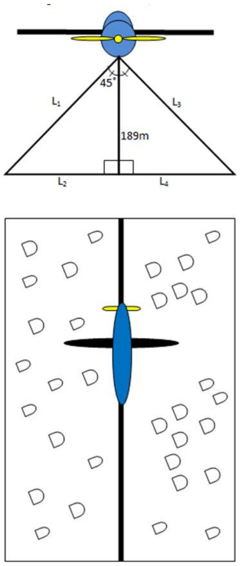

Figure 2 is the plane flying along the transect line. The camera is downward facing from the floor window of the plane so the camera angel is depicted at the top of this figure.

The aerial survey was ground-truthed (compared with a known measurement) with a photograph taken at 30in above known measurements of length and width (17.25in x 12in). All photographs were taken at a focal length of 50mm and the camera was set to the highest resolution of 3540 x 2336 pixels per photograph. The angle of view for length and width is needed to calculate the photograph surface area during the aerial survey. This was done by taking the dimensions of a photograph with known measurements. Let θ_w be the angle of view for the photograph width, so let

$$\theta_w = \mathrm{ArcTan}(w/a)$$

where w is the width of the photograph (known measurement) and a is the altitude of the camera. The same can be done for θ_l or the angle of view for the length of the photograph, so let

$$\theta_l = \text{ArcTan}(l/a)$$

where l is the width of the photograph (known measurement) and a is the altitude of the camera.

For example, let the photograph be taken over a known area of 17.25in x 12in with the camera being at an altitude of 30in (Fig. 3).

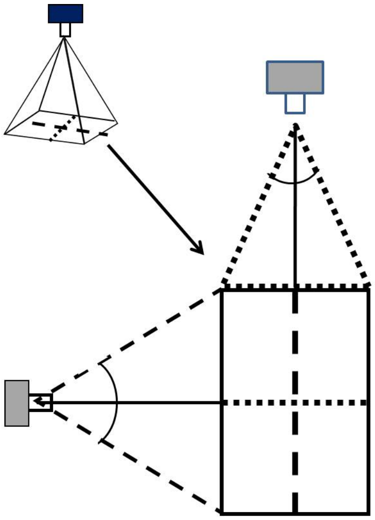

Figure 3 is a cartoon of the of the camera angle. The dotted triangle is used to find the width of the photograph and the dashed triangle is used to find the length of the photograph.

So if we look at the width of the photograph the width of the dotted triangle (Fig. 3) is 6in and so

$$\theta_w = \text{ArcTan}(6/30)$$

$$\theta_w = 11.31^\circ$$

Similarly for length if we looked at the dashed triangle (Fig. 3), the width is 8.625in and so

$$\theta_l = \text{ArcTan}(8.625/30)$$

$$\theta_l = 16.03^\circ$$

We can now use the altitude of the plane and the viewing angle (for both the width and length) to determine the surface area of the photograph during flight. So let W_{photo} be the width of the photograph and let L_{photo} be the length of the photograph. To calculate surface area of the photograph at the given altitude

$$Surface\ Area_{Photo} = W_{photo} * L_{photo}$$

where

$$W_{photo} = \text{Tan}(\theta_w) * a$$

and

$$L_{photo} = \text{Tan}(\theta_l) * a$$

where a is the altitude of the plane at the time the photograph was taken. This can be done for the range of airplane altitudes (620-690 ft) for all photographs. For example, if we use the camera angles from our first example we can determine width and length of the camera for an altitude of 650ft

$$W_{photo} = \text{Tan}(11.31^\circ) * 650$$

$$W_{photo} = 260\text{ft}$$

and

$$L_{photo} = \mathrm{Tan}(16.03°) * 650$$

$$L_{photo} = 374\mathrm{ft}$$

so

$$\textit{Surface Area}_{Photo} = 260ft * 374\mathrm{ft}$$

$$\textit{Surface Area}_{Photo} = 92,175\mathrm{ft}$$

The surface area is used to determine density of jellies per photograph. So let

$$D = \frac{Surface\ Area_{photo}}{N}$$

were N is the number of jellies observed in each photograph. We will calculate the mean density for each jelly abundance category by

$$\mu_D = \frac{\sum D}{N_{photo}}$$

where N_{photo} is the number of photographs used to calculate density for each abundance category.

The GPS data taken during aerial line transects will be put into ArcGIS and filtered for start of transect, points where jelly abundance categories changed, and the end of transect. The length of each category segment will be measured using spatial analysis tools in ArcGIS. The strip transect lengths of each category will be summed to obtain the total strip transect length for each category. The strip transect length will then be multiplied by the strip width to determine the total area for each jelly abundance category on the transect line. The average density for each jelly category will be multiplied by the area of that category and summed to obtain the total estimated abundance of jellies per transect line. An interpolation method will be used to estimate the total abundance of jellies from the water's surface to 1m in depth for the entire California survey area (Fig. 4).

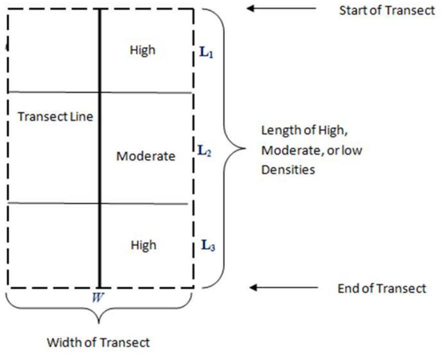

Figure 4 is the resulting strip transect after observer GPS points of the start transect, jelly densities, and end transect have been mapped and an interpellation method has been used to fill in the line.

0.3 Conclusion

With the development of the detection algorithm, the amount of time spent on processing the photographs was decreased significantly. Processing individual photographs went from taking 3-4hrs/photo to 12-15min/photo. With this reduction in time, the researcher is able to process more photographs in a shorter amount of time.

The efficacy of the detection algorithm to detect the total number of jellies was assessed using a regression comparing the relationship of the automated count versus the manual count. The results showed that the automated count underestimates for the total number of jellies for both species. This is especially true when aggregations are categorized as high (>1,000 individuals). This is most likely due to the nature of the aggregations of both species. Moon jellies tend to stack on top of each other in high aggregations, so this hinders the algorithm's ability to distinguish individuals based on surface area. For the brown sea nettles, the shape of the animal affects the algorithms ability to distinguish individuals based on surface area. This is because brown sea nettles most often are seen horizontal in the water column. Both bell and oral arms are seen and in high density aggregations the oral arms over lap and/or blend into the water color.

So even though the detection algorithm underestimates, the regression test provided us with a correction equation that will now enable us to correct the number of jellies detected by the algorithm to match with the proxy number of jellies found in the photograph.

The ability to run numerous photographs in a short amount of time, will enable the researchers to have a better foundation to understand the minimum surface density of each aggregation photographed. This will be used to find the conservative qualitative number of jellies seen during the leatherback flight for that year. Having a conservative estimate of jellies in each density category will also help researchers understand the magnitude of the population based on what the observers see during an average flight.

The future goal is to use this new method to understand the number of individuals seen in past years. If we assume that the jelly populations are the same in any given year we can assume that what we saw in 2010 and 2011 will be an estimate for years prior to these years. With the average number of jellies seen for each jelly density category we can use past data to calculate the conservative total number of jellies seen during transects of those years as well. This will be helpful for researchers who are trying to determine the sustainability of the leatherback's U.S. West coast foraging grounds.

All of this data can be used to make density maps for each year, where leatherback sighting data and weather conditions can also be included. These maps will be helpful in showing researchers were leatherbacks were in association to the jelly aggregations. We can then use thess maps to predict where leatherbacks will be in future years, based on their patterns in the past.

To improve this method, better technology is needed. Because the budget for this project was so small, the camera for this project was a personal camera dontated for the duration of the project. The increase in resolution will help the detection algorithm to distinguish individuals at higher aggregation densities. Also being able to attach an altimeter and GPS to the camera system (like in photogrammetry) the ability to count the total number seen over the whole transect will be a more accurate estimate. For the purposes of this project the algorithms ability to detect individuals is efficient and repeatable for future years at a moderately low cost.

Bibliography

[1] Sea notes. Downloaded on 08 Nov 2011.

[2] Scott R. Benson. personal communication, 2010.

[3] S.R. Benson, K.A. Forney, J.T. Harvey, J. V. Carretta, and P.H. Dutton. Abundance, distribution, and habitat of leatherback turtles (dermochelys coriacea) off california, 1990–2003. *Fish. Bull.*, 105:337–347, 2007.

[4] C.J. Calton and J.W. Burnett. Sea nettle *Nematocysts:anantomy, toxicology, and chemestry. Washington, DC: MTS*, pages 337–347, 1973.

[5] NOAA Fisheries. Leatherback turtles. *Office of protected resources*, Downloaded on 15 Mar 2010.

[6] K.A. Forney, D.A. Hanan, and J. Barlow. Detecting trends in harbor porpoise abundance from aerial surveys using analysis of convariance. *Fish. Bull.*, 89:367–377, 1991.

[7] T.R. Graham. Scyphozoan jellies as prey for leatherback sea turtles off central california. *Unpublished master's thesis. San Jose State University, San Jose, CA*, 2009.

[8] W.M. Graham, J.G. Field, and D.C. Potts. Persistent ”upwelling shadows” and their influence on zooplankton distributions. *Mar. Bio.*, 114:561–570, 1992.

[9] W.M. Graham and J.L. Largier. Upwelling shadows as nearshore retention sites: the example of northern monterey bay. *Continental Shelf Research*, 17:509–532, 1997.

[10] W.M. Graham, D.L. Martin, and J.C. Martin. In situ quantification and analysis of large jellies using a novel profiler. *Mar. Ecol. Prog. Ser.*, 254:129–140, 2003.

[11] W.M. Graham, F. Pages, and W.M. Hamner. A physical context for gelatinous zooplankton aggregations: a review. *Hydrobiologia*, 451:199–129, 2001.

[12] J.D. Houghton, T.K. Doyle, J. Davenport, and G.C. Hays. Developing a simple, rapid method for identifying and monitoring jellies aggregations from the air. *Mar. Ecol. Prog. Ser.*, 314:159–170, 2006.

[13] J.E. Purcell, E.D. Brown, K.D.E. Stokesbury, L.H. Haldorson, and T.C. Shirley. Aggregations of the jellies aurelial abiata: abundance, distribution, association with age-0 walleye pollock, and behaviors promoting aggregation in prince william sound, alaska, usa. *Mar. Ecol. Prog. Ser.*, 195:145–158, 2000.

[14] A.L. Sarti Martinez. Dermochelys coriacea. *In: IUCN 2010*, 2010(1), 2000.

0.4 Appendix A

0.4.1 Detection Algorithum

Jelly_Counting_code.m

Written by Erin Frolli

```
% This code was a product of Erin Frolli's 2010-2012 Capstone project @ CSU
% Monterey Bay.

% The code was designed to automate counting of individual
% jellies from Aerial photographs taken from the belly window of a
% twin-engine fix wing aircraft. All transects were conducted using three
% person observer team in a twin engine fix wing following methods
% described in Forney et al. 1991 and Benson et al. 2007. The height of
% the plane during transects was 198m with a speed of 90-100 knots as
% previously described in Forney et al. 1991 and Benson et al. 2007.

% For this code to work the photograph must be in a format that allows
% Matlab to surch through the digital RGB values of each pixel. Before this
% program is run you must also have the average surface area in pixels of
% each species for the last section of the code.

% Please note that this code has to be run one photograph at a time.
```

Uploading the Photo

```
% NOTE: make sure that the photos that you would like to use are in a
% folder in the Matlab directory. Make sure that the folder is opened so
% that Matlab is pulling the photos directly from this folder when the
% imread function is activated.

% The imread function is used to load your image into Matlab as a large
% matrix of both position and RGB values for each individual pixel of the
% photograph. F

% NOTE: For this function you must have the whole name of the photograph
% with file extension in (' ').

I=imread('IMG_0007.JPG'); %Loads the photo for use as a matrix. Note that
% named the photo I for future used and for simplicity.

% The imshow function will open a figure window that shows you the photo
% matrix as the actual photo. We bring up the photo first to make sure
% that we have the correct photo and second to collect X,Y coordinates of 5
% jellies around the photo in order to build our pixel database.

imshow(I) %brings the original photo up in a separate window

% Once the photo is up in the second window look at the top tabs.
% Tools > Data Courser
% This tool will turn the courser into a t-shape. This function will now
% allow you to find X,Y coordinates of 5 areas of interest. It would be best
% to zoom completely in on individual jellies so that you can select the
% appropriate pixel.  For the imcrop function that you will be using in the
% next section of this code you will want the top-left most corner of the
% jelly.  Only need this coordinate for the code.
```

```
% Its best if you see the photograph having four quadrates.  Then choosing
% a jelly that best represents the color that is seen in that quadrate
% (i.e. so that we can capture all attributes of this photo graph including
% glare, depth, watercolor, and fog).  For the fifth jelly pick a one that
% is in the center of the photo.
```

```
% Once this is done you can run the remaining code.  AFTER you imput the
% 5 coordinates and average jelly size.
```

Collecting Pixel Color Data

```
% This section crops the photo around the five individual jellies that you
% have chosen as your representatives of the photo.
```

```
% The imcrop function useses the original photo (in this case I) with a
% small matrix.  The first two values of the matrix are your X,Y coordinates
% of the jelly.  The second two numbers is the size of the area that you
% would like to crop.  For example, (I,[120 825 6 6]) says that I would like
% to crop from photo I into a 6X6 matrix starting at the top left corner
% at (120,825).  If you want to make sure that the program has cropped the
% correct area of interest please use the imshow code that has been made
% into a note by putting a % in front of the code.
```

```
C1=imcrop(I,[1046 8886 6 6]); % Crops first area of interest
C2=imcrop(I,[3642 230 6 6]); % Crops second area of interest
C3=imcrop(I,[715 2278 6 6]); % Crops third area of interest
C4=imcrop(I,[2926 1839 6 6]);% Crops fourth area of interest
C5=imcrop(I,[1665 1309 6 6]); % Crops fifth area of interest

% Use imshow to shows your cropped areas of interest
%figure
%imshow(C1)
%figure
%imshow(C2)
%figure
%imshow(C3)
%figure
%imshow(C4)
%figure
%imshow(C5)
```

Export RGB Values

```
% If you would like to import the pixel data into excel you would use the
% impixel function to collect the RGB values for you to export.  The FULL
% range of coordinates has to be put into the two vectors in order to
% obtain the full all of the RGB values.
% Note this sections is noted out because we do not need to export the data
% in order to run the code.
% Xs=[120:126,1989:1996,673:679,3421:3428,1988:1995];% x coordinates range
% Ys=[825:831,683:690,2229:2235,2398:2405,682:689];% y coordinates range
% RGB_Values=impixel(I,Xs,Ys); % Stores all of the RGB values from all three
% sets into a single matrix
```

Finding the Max and Min RGB Values

```
% In this section of the code we are finding the maximum and minimum RGB
% values in order to define the strict range of color we would like MatLab
% to find.  This will use the cropped data from before.

% First we must combined all of the cropped values together in once matrix.
% Since we would like to stack the values we write the vector as such.

RGB_Values = [C1 % Vector of RGB values
    C2
    C3
    C4
    C5];

M_=max(RGB_Values);
MAX = max(M_);% Finds the OVERALL Max values of the RGB Values
M__=min(RGB_Values);
MIN = min(M__);% Finds the OVERALL Min values of the RGB Values
```

```
MAXr=MAX(1,1,1); % names the Max red value
MAXg=MAX(1,1,2); % names the Max green value
MAXb=MAX(1,1,3); % names the Max blue value
MINr=MIN(1,1,1); % names the Min red value
MINg=MIN(1,1,2); % names the Min green value
MINb=MIN(1,1,3); % names the Min blue value
```

Pixel Finder Algorithm

```
% In this section we will be finding the overall number of pixels contained
% in the photo that are in our specific maximum and minimum RGB values.

% We first need to find the size of the photo.  Which we have named [Y X
% color].
[Y X color]=size(I);

% We then need to define an empty vector to hold our overall number of
% pixels found with our algorithm.
count=[0]; % Starts the pixel count at 0

% We are going to use a double For-End lope that has an IF statement for
% each RGB color. This tells MatLab to start at the top-left corner of the
% photo matrix and move from pixel to pixel looking for the defined RGB
% requirements.

% NOTE: That the last column of the photograph is data storage of the
% photographs properties not color data.

% We are using the maximum and minimum values to define the IF statements
% for each RGB values
% 1 = r = red
% 2 = g = green
% 3 = b = blue

% Once a pixel is found it is tagged using the brightest red value
% (225 = Bright Red). This is done so that the technician can double check
% that the code was finding the correct values when searching through the
% photograph.

% Once the pixel is tagged red it is added to the count. This number
% always replaces the number before it in the count vector.



for y = 1:Y-1; % tells the code to start at the first pixel in the top-left
%  and go through the whole photo until the last row

for x = 1:X-1; % tells the code to start at the first pixel and go
% through the whole photo

if(I(y,x,1)>= MINr)&&(I(y,x,1)< MAXr)...
&&(I(y,x,2)>= MINg)&&(I(y,x,2)< MAXg)...
```

```
&&(I(y,x,3)>=MINb)&&(I(y,x,3)< MAXb);

I(y,x,1)=225; % highlight all pixels that match the correct criteria

count=count+1; % includes this pixel to the count
end% ends first loop
end% ends second loop
end% ends third loop

% Once the lope is finished MatLab brings the original photo up in a
% separate window with new highlighted pixels shown.
imshow(I)
```

Species Counter

```
% Final Number of jellies estimated by the algorithm comes from the number
% of pixels tagged in the photograph (count vector number) divided by the
% average surface area (in pixels) of the species of interest.

% NOTE: Replace the Average_Jelly with the correct average surface area
% (in pixels) of the species of interest. Do so below
% for Average_Jelly_Surface_Area

Average_Jelly_Surface_Area = 137.32;

% The format long function is used to expand your number of interest out to
% 15 decimal places for a more accurate number.


format long
No_Jellies = count./Average_Jelly_Surface_Area

% FINAL NOTE: Make sure that you Clear all and Close all when you start a
% new image so that you do not overload MatLab with several different
% large matrixes.
```