



FEB 29, 2024

OPEN  ACCESS

DOI:

[dx.doi.org/10.17504/protocols.io.
6qpvr4p5bgmk/v1](https://dx.doi.org/10.17504/protocols.io.6qpvr4p5bgmk/v1)

Protocol Citation: Loreto Paulino Jr., Colin J Anthony, Colin Lock, Justin T Berg, Anela E K Duenas, Bastian Bentlage 2024. Quantify coral paling with grayscale-normalized color intensity values. **protocols.io** <https://dx.doi.org/10.17504/protocols.io.6qpvr4p5bgmk/v1>

License: This is an open access protocol distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

Protocol status: Working
We use this protocol and it's working

Quantify coral paling with grayscale-normalized color intensity values

Loreto Paulino Jr.¹, Colin J Anthony¹, Colin Lock¹, Justin T Berg¹, Anela E K Duenas¹, Bastian Bentlage¹

¹Marine Laboratory, University of Guam

Colin J Anthony: Corresponding author

Bastian Bentlage: Corresponding author



Loreto Paulino Jr.

DISCLAIMER

This protocol was based on and modified from Winters et al. (2009).

Winters, J., Holzman, R., Blekhman, A., Beer, S., Loya, Y. (2009). Photographic assessment of coral chlorophyll contents: Implications for ecophysiological studies and coral monitoring. *Journal of Experimental Marine Biology and Ecology*, 380(1-2):25–35.

ABSTRACT

Color normalization is crucial for accurate assessment of coral health in ecological studies. Color intensity has conventionally been used as a rapid assessment of coral health. However, color differences often confound comparisons. Here we present a protocol for quantifying coral paling using photographic monitoring and grayscale normalization. RAW in situ images were converted to JPEG using PhotoscapeX. ImageJ is then used to collect grayscale and coral intensity values. A correction formula adjusts and normalizes coral values. Lastly, we provide an optional Excel macro code to help facilitate large data processing. This protocol ensures consistent color representation, vital for ecophysiological studies and coral monitoring, aligning with established methodologies and enhancing data reliability.

IMAGE ATTRIBUTION

Images courtesy of Colin J Anthony.

Created: Jan 17, 2023

Last Modified: Feb 29, 2024

PROTOCOL integer ID: 75382

Keywords: coral, bleaching, coral reefs, acclimation, photographic analysis, ecophysiology, color, photography

Funders Acknowledgement:

NSF Guam EPSCoR GECCO
Grant ID: OIA-1946352

MATERIALS

- Olympus Tough TG-6 Camera (Recommended but can be replaced with alternative)
 - Computer (>128 GB RAM support recommended)
 - ImageJ (imagej.net/software/imagej; v1.52)
 - PhotoscapeX (x.photoscape.org; v4.2)
 - Microsoft Excel (v2402)
- Optional:
- PVC pipe
 - 30% UV shade cloth
 - Bash and Python compatible environment (e.g. Ubuntu: ubuntu.com/download)

Designing a camera mount

- 1 In general, the goal is to build a rig to normalize the photographic conditions as variation in lighting across the colony and scale bar can skew the color intensity values generated. We built a PVC camera mount for an Olympus Tough TG-6 camera large enough to fit around coral transplants (13mm x 13mm x 34mm), and then wrapped the camera mount in 30% UV shade cloth (Figure 1A). It is important to shade the coral with 30% or greater. Blackout cloth, for example, may provide a more consistent light environment, although accompanied by transportation tradeoffs. Within the camera mount, include the Kodak grayscale (Figure 1B). The Kodak grayscale (Figure 1C) allows photographs to be analyzed across photographic conditions.

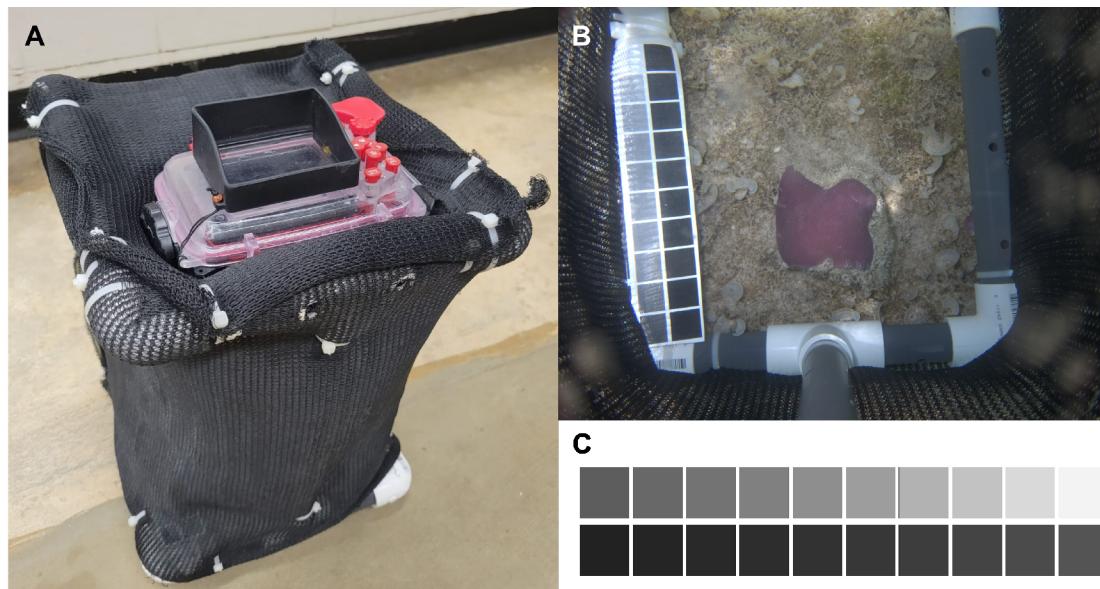


Figure 1: **A)** Example camera mount for Olympus Tough TG-6. **B)** Example photograph with Kodak grayscale in view. **C)** Kodak grayscale used for presented protocol.

Grayscale Files for Download

GrayScale_Kodak.svg 54KB

GrayScale.png 32KB

Taking photographs

- 2** Photographs were taken with an Olympus Tough TG-6 underwater camera (Figure 1). RAW images are best as they preserve all color information leading to a more accurate conversion to grayscale.

Camera Settings

- ISO AUTO
- WB AUTO
- AF
- RAW
- 4000x3000

Converting Raw Images into .JPG Files

- 3** Label photographs (.ORF) (Figure 1A) based on location, species, time-point, site, or any other metadata required for post-processing.

Note

For our largest experiment, we wrote a simple code that automatically enumerated and renamed files based on the order of photographs taken, given our repeated sampling design.

Sample Naming Code for Download

 PhotoRenameEnumerate.py 0B

 GardenPhotoRename07062021.py 21KB

4 Converting ORF Files into JPEG using PhotoscapeX

1. Open PhotoScapeX.
2. Select batch and the folder with the respective time point.
3. Drag photos to "Drag photos here" box.
4. Checkmark Color on the right-hand side and click on color then select "Grayscale" (Figure 2B).
5. Press save and select format output to be .JPEG.
6. Send .JPEG files to the same folder for easy access.

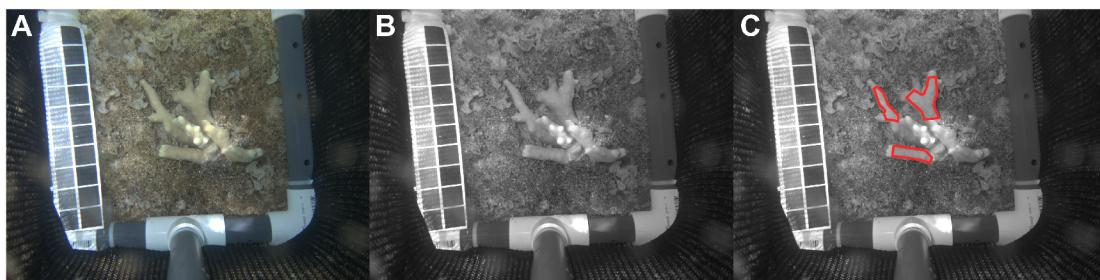


Figure 2 **A)** Example RAW photograph (.ORF) from field. **B)** Photograph after grayscale conversion. **C)** Grayscale converted photograph with red polygons representing tissue fit for color sampling.

Using ImageJ Analysis Version 1.52(100) for Collecting GrayScale Data

5 Basic Commands:

Open a photo in the ImageJ program:

Press Ctrl + o or go to the File menu and select "Open".

Locate and select the desired photo from your computer.

Set measurements for data collection:

Go to the "Analyze" menu and select "Set Measurements". Select "Standard deviation" and "Mean". Press "Ok" to confirm the measurements.

6 First Measurement: Grayscale Intensity Values

The first step is to define a grayscale calibration curve that we can reference our coral colony to.

1. Open photograph as described in Step 6.
2. Select "Square Tool" in ImageJ.
3. Define a square that is the size of your individual grayscale boxes.
4. Drag a Square to one of the square scales on the grayscale by pressing Ctrl + M on the keyboard.
5. Measure the first grayscale (lightest) to the 19th interval (second darkest) in odd intervals. 1, 3, 5, 7, 9, 11, 13, 15, 17, 19 (10 Intensity Values).
6. Copy and paste measurement to Microsoft Excel using the transpose key (Ctrl + x) [See Step 9].

7 Second Measurement: Coral Intensity Values

Now, that we have a calibration curve, we can measure the coral colony's grayscale intensity value.

1. Open photograph as described in Step 6.
2. Select "Polygon Tool" in ImageJ.
3. Set Legacy to "10".
4. Set Mode to "Legacy" then select "Ok".
5. Select an area on coral you want to measure (e.g. Figure 2C).
6. Press Ctrl + M on the keyboard to record the grayscale intensity.

Note

Often the coral does not have a uniform tint due to colony morphology or lighting. We try to reduce this as much as possible with shade cloth, but it is important to choose an area of the coral colony that is a fair representative of the overall colony. A larger area measurement will give you a more robust estimate, but you must find a balance between a large area and a representative area.

8 Correcting Coral Intensity Values

Normalization relies on the fact that under perfect conditions each gray within this scale, the intensity of each of the three RGB color channels is equal to one another ($I_{(Rj)} = I_{(Gj)} = I_{(Bj)}$, where $I_{(Rj)}$, $I_{(Gj)}$, $I_{(Bj)}$ are the intensities of the each color (R,G,B) for pixels located within the j^{th} gray). Since the decrease in the intensity of the grays on this scale is linear, the intensity of the red channel (for example) as a function of gray number can be described as equation 1.1.

$$(1.1) I_{(Rj)} = a_{(R)}xj + b_{(R)}$$

When taking images in realistic underwater scenarios, colors change due to differential absorption of by water across the spectrum, as well variations in illumination conditions. Consequently, $a^*(R) \neq a^*(G) \neq a^*(B) \neq -(255/20)$ and $b^*(R) \neq b^*(G) \neq b^*(B) \neq 255 - a$ [$*$ indicates coefficients in the underwater images]. However, the intensity within each channel is still linearly correlated with gray number following the Eq. (1.1). Between different underwater images, $a^*(R)$ and $b^*(R)$ are expected to vary due to varying illumination, reflectance and absorption conditions, producing the variation in color observed in unprocessed images. Since the original values of a and b are known, it is possible to calculate the corrected intensity of each channel for each pixel through linear transformation so that:

$$(1.2) I_{\text{corrected}(Rxy)} = a_{(R)}/a^*(R) (I_{\text{measured}(Rxy)} - b^*(R)) + b_{(R)}$$

Variable Definitions

$I_{\text{corrected}(Rxy)}$: Corrected color intensity to be used for down-stream analysis

$I_{\text{measured}(Rxy)}$: Measured color on coral value

$a_{(R)}$ and $b_{(R)}$: These are the known measured values (Outside of the ocean)

$a^*(R)$ and $b^*(R)$: These are the observed values (Taken From photos)

Excel Macro Code Set Up for Mass Data Collection

9 Set Up Excel to Have Use the Data Collected in Previous Steps.

Excel Macro Code Set-Up and Use

1. Open a blank Excel Sheet.
2. Go to the "Tools" menu.
3. Select "Record New Macro".
4. Name the macro, "Macro2" (Preference)
5. Select the "Command Option" button and input the letter "x" (Preference).
6. Go to the "Tools" menu and select "Macros".

7. Select "Visual Basic Editor".
8. You will be prompted to a new window and there will be two drop-down tabs: "General" and "Macro2".
9. Beneath these two tabs, copy and paste the provided macro code (below).
10. Select "Save" and save the file as a macro-enabled excel sheet.

Using the Code

Open the excel sheet, paste the data with the color mean values under Column "C".

Press "Command + x" and the data should appear in rows starting on Column "F".

Example Datasheet



Color_Data.xlsx 159KB

Note

This code can collect up to 400 grayscale values and is easily modified.

Command

Excel Macro Code (Windows)

```
Sub Macro2()
'
' Macro2 Macro
'
' Keyboard Shortcut: Ctrl+x
'
    Range("C1:C10").Select
    Selection.Copy
    Range("F1").Select
    Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=_
        False, Transpose:=True
    Range("C11:C20").Select
    Application.CutCopyMode = False
    Selection.Copy
    Range("F2").Select
    Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=_
        False, Transpose:=True
    Range("C21:C30").Select
    Selection.Copy
    Range("F3").Select
    Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=_
        False, Transpose:=True
    Range("C31:C40").Select
    Application.CutCopyMode = False
    Selection.Copy
    Range("F4").Select
    Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=_
        False, Transpose:=True
    Range("C41:C50").Select
    Selection.Copy
    Range("F5").Select
    Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=_
        False, Transpose:=True
    Range("C51:C60").Select
    Application.CutCopyMode = False
```

```
Selection.Copy
Range("F6").Select
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=_
    False, Transpose:=True
    Range("C61:C70").Select
Selection.Copy
Range("F7").Select
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=_
    False, Transpose:=True
Range("C71:C80").Select
Application.CutCopyMode = False
Selection.Copy
Range("F8").Select
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=_
    False, Transpose:=True
    Range("C81:C90").Select
Selection.Copy
Range("F9").Select
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=_
    False, Transpose:=True
Range("C91:C100").Select
Application.CutCopyMode = False
Selection.Copy
Range("F10").Select
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=_
    False, Transpose:=True
    Range("C101:C110").Select
Selection.Copy
Range("F11").Select
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=_
    False, Transpose:=True
Range("C111:C120").Select
Application.CutCopyMode = False
Selection.Copy
Range("F12").Select
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=_
    False, Transpose:=True
    Range("C121:C130").Select
Selection.Copy
Range("F13").Select
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=
```

```
    raise, transpose:=true
Range("C131:C140").Select
Application.CutCopyMode = False
Selection.Copy
Range("F14").Select
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=_
    False, Transpose:=True
Range("C141:C150").Select
Selection.Copy
Range("F15").Select
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=_
    False, Transpose:=True
Range("C151:C160").Select
Application.CutCopyMode = False
Selection.Copy
Range("F16").Select
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=_
    False, Transpose:=True
Range("C161:C170").Select
Selection.Copy
Range("F17").Select
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=_
    False, Transpose:=True
Range("C171:C180").Select
Application.CutCopyMode = False
Selection.Copy
Range("F18").Select
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=_
    False, Transpose:=True
Range("C181:C190").Select
Selection.Copy
Range("F19").Select
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=_
    False, Transpose:=True
Range("C191:C200").Select
Application.CutCopyMode = False
Selection.Copy
Range("F20").Select
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=_
    False, Transpose:=True
Range("C201:C210").Select
Application.CutCopyMode = False
```

```
Selection.Copy
Range("F21").Select
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=_
    False, Transpose:=True
Range("C211:C220").Select
Selection.Copy
Range("F22").Select
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=_
    False, Transpose:=True
Range("C221:C230").Select
Application.CutCopyMode = False
Selection.Copy
Range("F23").Select
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=_
    False, Transpose:=True
Range("C231:C240").Select
Selection.Copy
Range("F24").Select
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=_
    False, Transpose:=True
Range("C241:C250").Select
Application.CutCopyMode = False
Selection.Copy
Range("F25").Select
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=_
    False, Transpose:=True
Range("C251:C260").Select
Selection.Copy
Range("F26").Select
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=_
    False, Transpose:=True
Range("C261:C270").Select
Application.CutCopyMode = False
Selection.Copy
Range("F27").Select
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=_
    False, Transpose:=True
Range("C271:C280").Select
Selection.Copy
Range("F28").Select
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=
```

```
    False, Transpose:=True
Range("C281:C290").Select
Application.CutCopyMode = False
Selection.Copy
Range("F29").Select
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=_
    False, Transpose:=True
Range("C291:C300").Select
Selection.Copy
Range("F30").Select
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=_
    False, Transpose:=True
Range("C301:C310").Select
Application.CutCopyMode = False
Selection.Copy
Range("F31").Select
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=_
    False, Transpose:=True
Range("C311:C320").Select
Application.CutCopyMode = False
Selection.Copy
Range("F32").Select
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=_
    False, Transpose:=True
Range("C321:C330").Select
Selection.Copy
Range("F33").Select
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=_
    False, Transpose:=True
Range("C331:C340").Select
Application.CutCopyMode = False
Selection.Copy
Range("F34").Select
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=_
    False, Transpose:=True
Range("C341:C350").Select
Selection.Copy
Range("F35").Select
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=_
    False, Transpose:=True
Range("C351:C360").Select
```

```
Application.CutCopyMode = False
Selection.Copy
Range("F36").Select
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=_
    False, Transpose:=True
    Range("C361:C370").Select
Selection.Copy
Range("F37").Select
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=_
    False, Transpose:=True
Range("C371:C380").Select
Application.CutCopyMode = False
Selection.Copy
Range("F38").Select
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=_
    False, Transpose:=True
Range("C381:C390").Select
Selection.Copy
Range("F39").Select
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=_
    False, Transpose:=True
Range("C391:C400").Select
Application.CutCopyMode = False
Selection.Copy
Range("F40").Select
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=_
    False, Transpose:=True
End Sub
```