# 3D processing TCRMP transects

L Olinger

2024-09-19

## Table of contents



## field methods

### materials

- camera + lights, memory card

- scale bars
- …

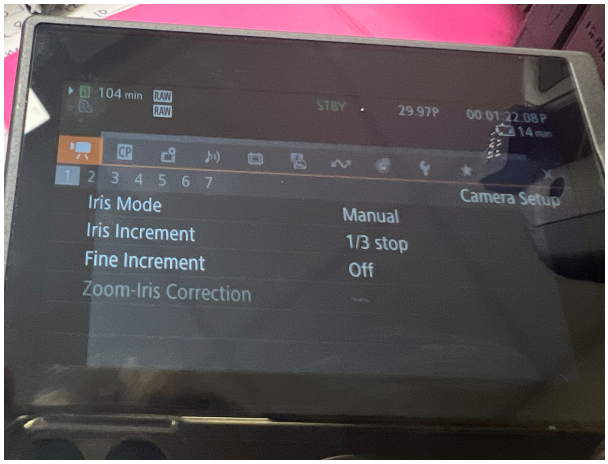
## procedure (4-pass method)

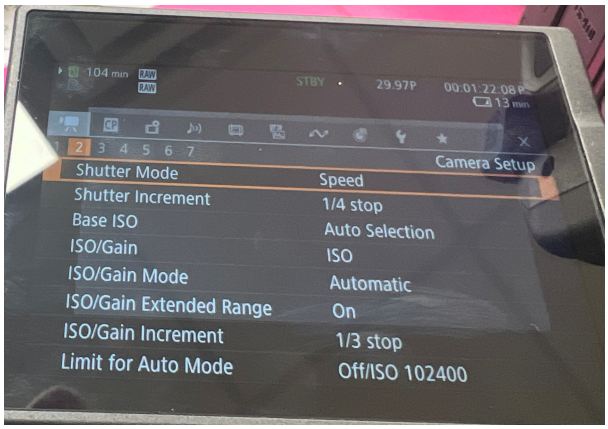
- general
  - everything with the pink ziptie belongs to the 3d camera
- camera maintenance:
  - camera cinema gear
  - settings
  - programmable buttons
- housing maintenance (every few weeks, or if any leaks detected)
  - o-rings greasing
- day before
  - check housing/o-rings
  - charge camera
  - charge external battery pack
  - charge strobe light batteries
  - memory card- initialize media
- morning of:
  - **closing / sealing cam**
    - install battery and memory card into camera
    - attach lens
    - check autofocus on (switch on lens)
    - remove camera lens cap
    - check for smudges on lens
    - pull switch on housing so that can insert camera with the cinema camera gear
    - seat camera in housing (pull out stage thing, screw camera on.

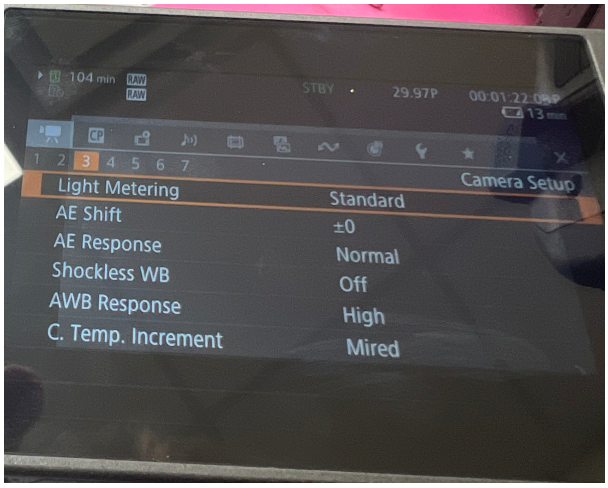
    - put camera into housing (careful to not scratch lens)
    - connect to external battery
    - add external battery to camera
    - once in housing:
        - turn alarm on
        - check for smudges on housing lens
        - check o-ring
        - close housing
        - use vacuum device until light turns green.

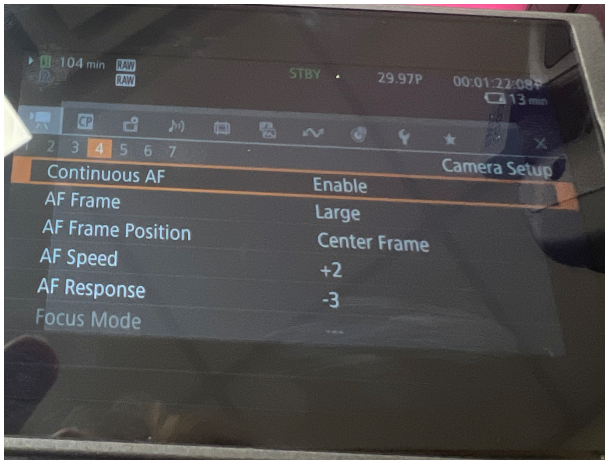
- **check have materials**:
    - camera + mem card
    - housing
    - field box with extra stuff/towels, o-ring grease, cleaning materials, dry towels, etc.
    - slate
    - scale bars x 2
    - handle (clips, rope)

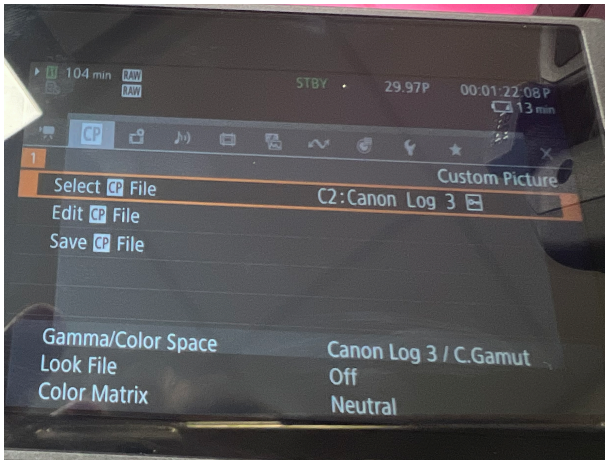
- **check camera settings**:
    - CP file : C2: Canon log 3 / C.Gamut Color matrix neutral.
    - Sensor mode: full frame
    - freq 59.94hz
    - Rec = RAW LT
    - Dest = CFexpress
    - Frame = 59.94 fps


104 min
RAW
STBY
29.97P
00:01:22:08
14 min
1 2 3 4 5 6 7
Camera Setup
Iris Mode
Manual
Iris Increment
1/3 stop
Fine Increment
Off
Zoom-Iris Correction

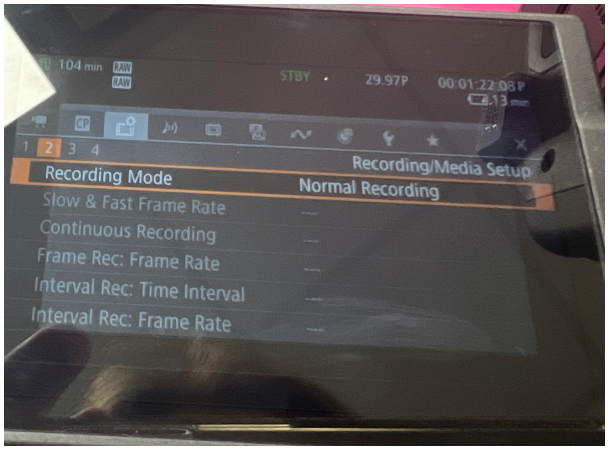

104 min
RAW
STBY
29.97P
00:01:22:08
13 min
1 2 3 4 5 6 7
Camera Setup
Shutter Mode
Speed
Shutter Increment
1/4 stop
Base ISO
Auto Selection
ISO/Gain
ISO
ISO/Gain Mode
Automatic
ISO/Gain Extended Range
On
ISO/Gain Increment
1/3 stop
Limit for Auto Mode
Off/ISO 102400


104 min
RAW
STBY
29.97P
13 min
1 2 3 4 5 6 7
Camera Setup
Light Metering
Standard
AE Shift
±0
AE Response
Normal
Shockless WB
Off
AWB Response
High
C. Temp. Increment
Mired


104 min
RAW
RAW
STBY
29.97P
00:01:22:08
13 min
2 3 4 5 6 7
Camera Setup
Continuous AF
Enable
AF Frame
Large
AF Frame Position
Center Frame
AF Speed
+2
AF Response
-3
Focus Mode


104 min
RAW
RAW
STBY
29.97P
00:01:22:08
13 min
1
Custom Picture
Select CP File
C2:Canon Log 3
Edit CP File
Save CP File
Gamma/Color Space
Canon Log 3 / C.Gamut
Look File
Off
Color Matrix
Neutral


104 min
RAW
RAW
STBY
29.97P
13 min
3 4
Recording/Media Setup
Initialize Media
Sensor Mode
Full Frame
System Frequency
59.94 Hz
Main Rec Format
RAW LT
Main Recording Destination
CFexpress
Main Resolution/Bit Rate
8192x4320 1290Mbps
Frame Rate
59.94P
Main Audio Format (MP4)


104 min
RAW
ARW
STBY
29.97P
00:01:22:03
1 2 3 4
Recording/Media Setup
Recording Mode
Normal Recording
Slow & Fast Frame Rate
Continuous Recording
Frame Rec: Frame Rate
Interval Rec: Time Interval
Interval Rec: Frame Rate

**in water :**

- **Start of dive:**
  - **B**uttons: press all buttons to prime them
  - **Power:** turn on camera and lights (hold in/out buttons 1s, press middle button). put lights to sleep (hold center 2s)
  - **L**eaks: green light stays green, if turn red, return to boat.
- **Transect and Camera Setup**
  - **S**cale bars: Place at each end of transect, one perpendicular, one parallel to transect. make sure targets visible in some footage and scale bars do not move at any time during filming.
  - **T**ime code: reset (Mode button).
  - **A**rms: extend to position lights as far apart as possible
  - **L**ights: turn on (hold Center Button 2 sec).
  - **W**hite balance: press Button 13, hold camera over white part of one of the scale bars.
  - **E**xposure: open WFM (Button 6) and false color (Button 9), Use F-stop dial to slightly overexpose, just below 100% on WFM.
  - **A**ltitude: position camera so viewfinder covers length of scale bar, note height (should be ~70cm). Maintain this altitude throughout filming.
  - **R**ecord: Record button, show transect number, autofocus
- **Filming (4 passes, 10 m each, ~1 min, consistent altitude):**
  - Pass 1: Start at one end, camera facing straight down, transect line visible in left quarter of viewfinder.
  - Pass 2: Turn around, camera facing straight down, slightly away from transect line such that viewfinder sees 1m distance from transect while keeping ~0.5 m overlap with Pass 1 (~arm's length from transect)
  - Pass 3 & 4: Move ~20 cm from pass 1/2 position, tilt camera 45°, capture angled view of transect from either side.
- **After Filming:** press center button on each light for 2s to put light to sleep.

## set up (this template)

download and unzip this template

rename the folder to some descriptive name of the footage going to process - eg `TCRMP_2024_postBL`

open the `.Rproj` in Rstudio to knit and follow along.

good idea to save this template to somewhre like dropbox where it can be constantly backed up.

open and populate `00_list.csv` with the transects to be reconstructed with this template. use this to keep track of the steps, and it is used in step 3.py below.

## video encoding (premiere pro)

- upload contents of memory card into computer (connect using special mem card reader compatible with CF Express, careful of magnets on the mem card reader)
- **starting with .CRM, export to MP4 without applying LUTs**
- export preset : `step0_premierepro_uhd_8k_23sept2024.eps` located in `00_scripts`
- use the following naming convention :
  - eg `TCRMP20240311_3D_BID_T1_3.MP4`
    - `TCRMP20240311` = TCRMP plus date (YYYYMMDD)
    - `3D` = demographic
    - `BID` = sitecode
    - `T1` = transect number
    - `3` = pass number (if separated vids by pass)
- export to folder `01_vids` (start backup to dropbox)
- format/wipe/initialize memory card after the videos have been uploaded

## extract frames (R)

requires `ffmpeg` - much faster than using av package in R.

```r
library(av)

npics <- 100 # number of pics to extract

# functions
get_video_duration <- function(video) {
  info <- av_media_info(video)
  return(info$duration)
}

extract_frames <- function(video, fps, dest_folder) {
  video_name <- tools::file_path_sans_ext(basename(video))
  output_pattern <- file.path(dest_folder, paste0(video_name, "_%04d.tiff"))

  cmd <- sprintf(
    '"%s" -i "%s" -vf fps=%f -c:v tiff -pix_fmt rgb24 "%s"',
    ffmpeg_path,
    video,
    fps,
    output_pattern
  )

  system(cmd)
}

get_transect_id <- function(file) {
  sub("\\.mp4", "", sub(".mp4$", "", basename(file)))
}

# script
directory <- paste(getwd(), "01_vids", sep = "/")

files <- list.files(directory, pattern = "\\.mp4$", full.names = TRUE)

ffmpeg_path <- "/opt/homebrew/bin/ffmpeg"  # Update this path based on your installation

transects <- unique(sapply(files, get_transect_id))
```

```r
base_output_dir <- "02_pics"

# iterate over transects, apply the functions to extract frames
for (i in 1:length(transects)) {
  transect <- transects[i]
  transect_files <- files[grep(paste0(transect, ".mp4$"), files)]
  total_duration <- sum(sapply(transect_files, get_video_duration))
  frames_per_part <- sapply(transect_files, function(x) {
    duration <- get_video_duration(x)
    return(round(npics * (duration / total_duration)))
  })
  Folder1 <- file.path(base_output_dir, transect)
  dir.create(Folder1, showWarnings = FALSE, recursive = TRUE)
  for (j in seq_along(transect_files)) {
    video_file <- transect_files[j]
    duration <- get_video_duration(video_file)
    num_frames <- frames_per_part[j]
    fps <- num_frames / duration
    extract_frames(video_file, fps, Folder1)
  }
}
```

## editing photos (lightroom classic)

In adobe lightroom, batch edit photos

apply `step0_lightroom_hdrphoto_r5c.xmp` located in `00_scripts`

summary of the settings:

- HDR Editing Enabled: Allows for greater dynamic range adjustments with a maximum HDR value of +8.00.
- Auto Tone Enabled: Automatically adjusts tonal settings for optimal exposure.
- Lens Corrections Applied:
  - Uses the Adobe lens profile for the Canon EF 24-70mm f/2.8 L USM lens.
  - Vignetting correction is intensified with a scale set to 200.
- Most other settings remain at their default values, making this preset suitable for general use without over-processing the image.

Export from lightroom as uncompressed tif with srgb colorspace and HDR , save in `03_editedpics/{SITE_TRANSECT}`

> **Note**
>
> in future may color correct algorithmically (eg Sea Thru)

## 3D processing (metashape pro)

### file setup

- open new metashape file
- add photos to it corresponding to each transect. can do 15-20 transects per psx, click and drag folder in edited pics for each transect into the metashape workspace where chunks are labeled.
- save as psx in `04_psx` with title as all the site names (eg BIX_BID….psx)

### step 1.py

Run `00_scripts/step1.py` : in metashape, select tools, run script, and point to the `00_scripts/step1.py`.

Briefly, This script automates the initial stages of photogrammetric processing using Metashape's Python API. It processes image alignment, point cloud generation, and camera optimization for each chunk in a Metashape project. Key tasks include:

- **Photo alignment**: Matches and aligns photos, creating a sparse point cloud.
- **Camera optimization**: Filters tie points based on reconstruction uncertainty, reprojection error, and projection accuracy, followed by camera optimization. For an explanation of the parameters, see @olinger2019
- **Region reset and transformation**: Resets the bounding region and rotates the coordinate system to the bounding box.
- **Optional 3D model construction steps**: Depth map generation, 3D model building, smoothing, and UV/texture building (commented out).

### manual checks

go through each chunk and check

- each transect is roughly visible when you use 0 to reset view
- there are no mirror images, or huge gaps in photos (< 90% photos aligned should be a red flag )
- scale bars are visible

disable chunks that are not good quality (eg < 90% of photos aligned)

### step 2.py

> **Warning**
>
> this is the most time consuming step, and can take a few hours per chunk.

Run `00_scripts/step2.py` : in metashape, select tools, run script, and point to the `00_scripts/step2.py`.

This script automates the creation of dense clouds, 3D models, and textures in Metashape using Python. It also generates and exports a processing report for each chunk. Key tasks include:

- Dense cloud creation: Generates depth maps (with no filtering).
- Model creation and smoothing: Builds a 3D model based on depth maps, smooths the model, and optionally decimates the face count (commented out).
- UV and texture generation: Generates UV mapping and builds a texture with GPU acceleration and various quality optimizations.
- Report export: Saves a detailed processing report for each chunk in the designated reports directory.

Key Parameters:

- **buildDepthMaps**: Ultra high quality and no depth filtering.
- **buildModel**: Uses depth maps data, arbitrary surface type, high face count, no volumetric masking, enabled interpolation, and vertex colors.
- **smoothModel**: Strength of 3, does not apply to selection, fixes borders, does not preserve edges.

- **buildUV**: Generic mapping mode, texture size of 8192, and one texture page.
- **buildTexture**: Texture size of 8192, diffuse map type, source images data, mosaic blending, fills holes, uses ghosting filter, enables GPU, and relaxed precision.

### step 3.py

run `00_scripts/step3.py` : in metashape, select tools, run script, and point to the `00_scripts/step3.py`.

This script automates the process of copying chunks between Metashape projects based on a CSV file. It reads the CSV, appends chunks from source projects to destination projects, and saves the results. Key steps include:

- read the CSV file location:
- Project handling: For each unique project name in the CSV, the script opens or creates a destination .psx project and appends chunks from specified source projects.
- Chunk copying: It searches for the specified chunk in the source project and appends it to the destination project.
- Saving results: After processing, the destination project is saved, ensuring all chunks are properly appended.

### manual straightening and scaling

Open each project and do the following for each chunk:

**straightening**

- load textured model
- change to rotate model view and rotate the transect so the transect line lines up horizontally and is top of view.
- model- region- rotate region to view
- resize region to "crop" to transect area (do this for top xy and side views)
- use rectangular crop tool to crop to transect area bound by region. may automate this in step 4.py in the future.

**scaling (only if scaling manually)**

- place markers on scales - set up at least 2 scale bars
- set distance in reference pane

- press refresh button -make sure error less than .01

**save project**

### step 4.py

run `00_scripts/step4.py` : in metashape, select tools, run script, and point to the `00_scripts/step4.py`.

This script automates several key processes in Metashape, including decimation and Sketchfab uploads. The script processes multiple .psx projects based on values in a 00_list.csv file and performs the following steps for each project:

- **Model Scaling and Alignment**: If the model isn't scaled, the script detects circular 20-bit markers and creates scale bars between specific marker pairs.
- **Gradual Selection**: Applies a point cloud filtering step using connected components, removing points based on a threshold.
- **Orthomosaic Generation**: Builds an orthomosaic from the 3D model and exports it in TIFF format.
- **Textured Model Export**: Exports the textured model in OBJ format with JPEG textures.
- **Chunk Duplication and Decimation:** Duplicates the original chunk, decimates the duplicate to a specified number of vertices (nverts), and re-textures the model if needed.
- **Sketchfab Upload**: Uploads the decimated model to Sketchfab with customizable titles, descriptions, and tags.
- **Temporary Chunk Deletion**: Deletes the duplicated temporary chunk after uploading to Sketchfab.