SoftwareX 12 (2020) 100582

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

CIRN Quantitative Coastal Imaging Toolbox

Brittany L. Bruder *, Katherine L. Brodie

Coastal and Hydraulics Laboratory, U.S. Army Engineer Research and Development Center, 1261 Duck Road, Duck, NC 27949, United States

ARTICLE INFO

Article history:
Received 2 June 2020
Received in revised form 16 August 2020
Accepted 17 August 2020

Keywords:
MATLAB
Photogrammetry tutorial
Unmanned Aircraft Systems (UAS)
Coastal Imaging Research Network

ABSTRACT

The Coastal Imaging Research Network (*CIRN*)'s *Quantitative Coastal Imaging Toolbox* is a collection of MATLAB scripts to produce georectified images specifically tailored for quantitative analysis of coastal environments. The repository contains scripts that perform end-to-end georectification of oblique imagery from land-based multi/single-camera stations or stationary Unmanned Aircraft Systems (UAS). The toolbox produces georectified frames, ensemble statistical products, and subsampled pixel collections for optical wave, current and bathymetric inversion analyses. While experts have employed similar scripts operationally for years, this toolbox is for photogrammetry novices. Illustrative example scripts with demonstrative data teach fundamentals and can be easily transitioned to operational processing.

Code metadata

Current code version	*1.0*
Permanent link to code/repository used of this code version	https://github.com/ElsevierSoftwareX/SOFTX_2020_236
Code Ocean compute capsule	Submitted for Publication. Waiting for approval.
Legal Code License	*GNU General Public License, version 3*
Code versioning system used	*git*
Software code languages, tools, and services used	*MATLAB 2019b w/ Statistics and Machine Learning, Image Processing Toolbox*
Compilation requirements, operating environments & dependencies	*Cal-Tech Camera Calibration Toolbox preferred, not required (See User Guide for more Information).*
If available Link to developer documentation/manual	https://github.com/Coastal-Imaging-Research-Network/CIRN-Introduction-to-Quantitative-Coastal-Imaging-Toolbox/blob/master/user_manual.pdf
Support email for questions	Brittany.L.Bruder@erdc.dren.mil

1. Motivation and significance

Video monitoring is an accurate and cost-effective tool for scientists and engineers to gain qualitative and quantitative information about coastal environments [1]. Many nearshore processes such as wave propagation, wave breaking, and swash run up, provide optical and thermal signatures that can be exploited by color, grayscale, and infrared cameras to differentiate wave crests and troughs, water and foam, or land and water. Algorithms can utilize these signatures to estimate geophysical parameters including directional wave spectra, bathymetry, surface currents, and shoreline migration [2–19].

However, all of these algorithms require image georectification, where the effects of perspective are removed and an oblique (traditional) image is scaled to real world coordinates ('mapping an image'), for accurate extraction of quantitative geophysical information. Unfortunately for most coastal scientists and engineers, photogrammetry is not commonly included in oceanographic curricula and may be a roadblock for utilizing video monitoring quantitatively in their projects or research. As a result, despite its advantages and being employed for over 30 years, quantitative coastal imaging is not widespread outside of narrow academic research lineages.

* Corresponding author.
E-mail addresses: Brittany.L.Bruder@erdc.dren.mil (B.L. Bruder), Katherine.L.Brodie@erdc.dren.mil (K.L. Brodie).

https://doi.org/10.1016/j.softx.2020.100582

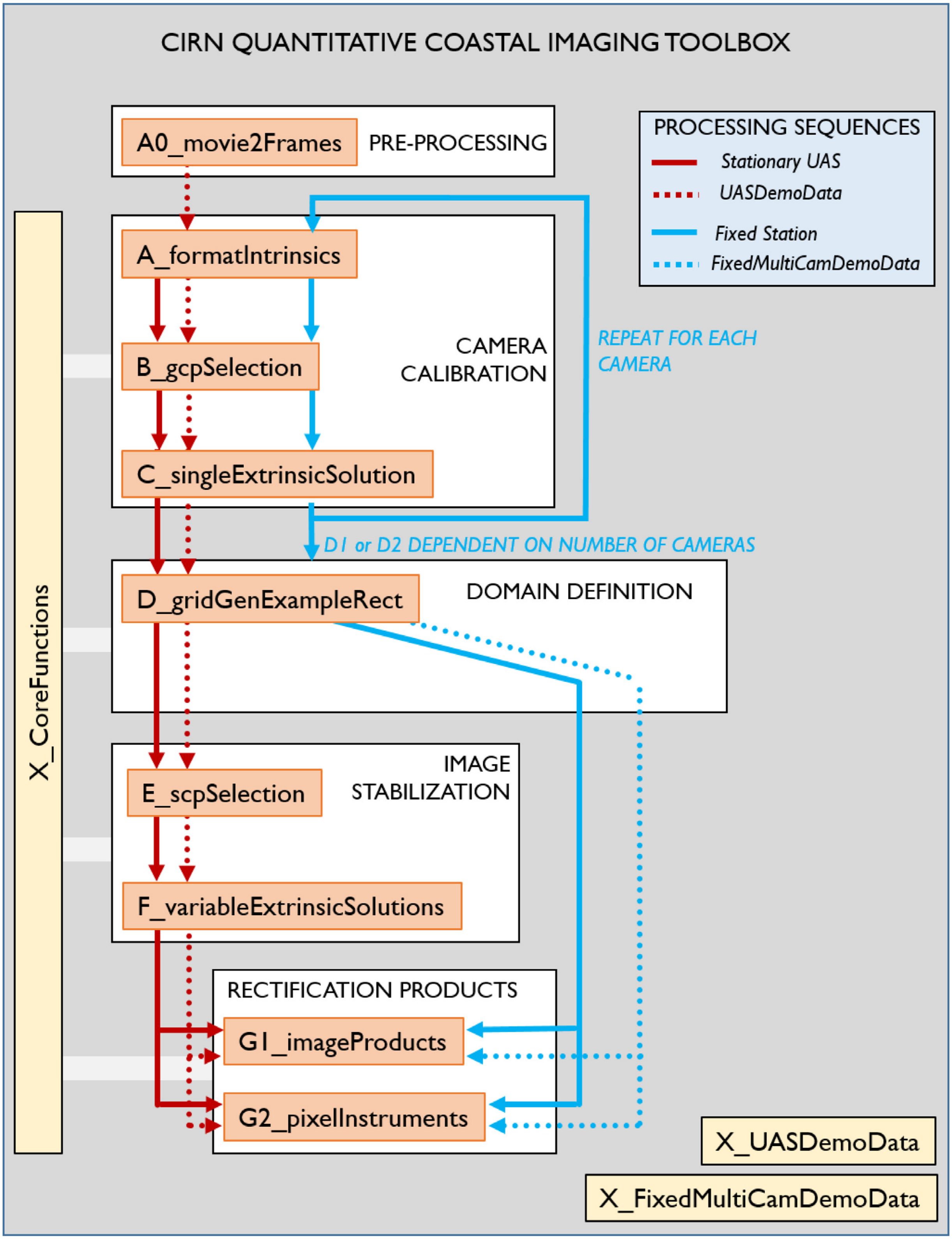

Fig. 1. CIRN Quantitative Coastal Imaging Toolbox repository and workflow. White boxes represents characterizations, yellow boxes represent folders, and orange boxes represent scripts.. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Recognizing this, subject matter experts (SMEs) formed the Coastal Imaging Research Network (CIRN) and developed 'bootcamps' beginning in 2015 designed to teach coastal photogrammetry fundamentals as well as more advanced analysis techniques. To accompany the bootcamp curriculum, a MATLAB code repository was developed to demonstrate end-to-end georectification of RGB video from a stationary unmanned aircraft system (UAS), based off of [20] and later developed into a Graphical User Interface (GUI) in [21]. Since 2015, the repository, known as the *UAV Toolbox*, has been cited numerous times in publications.

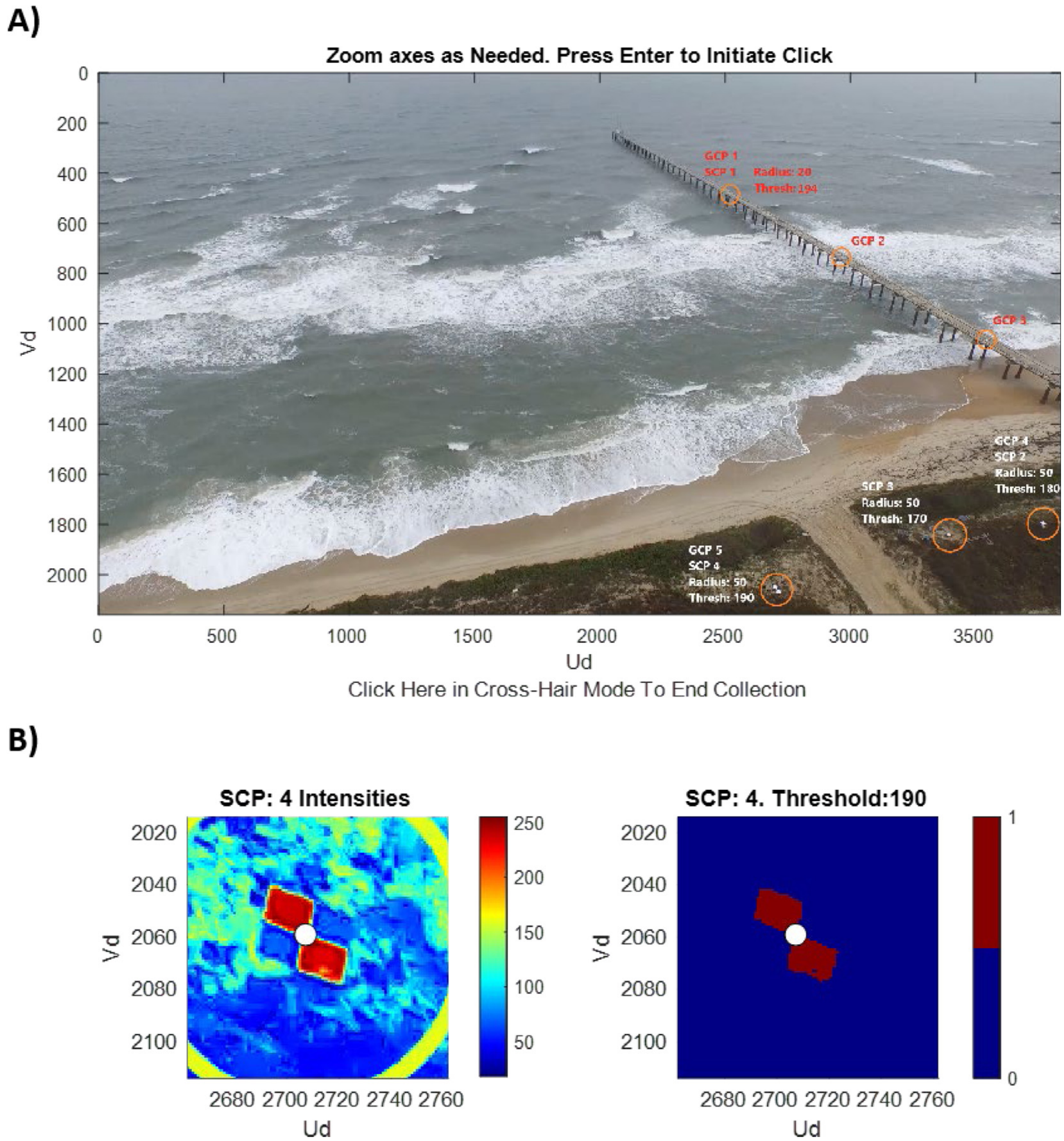

Fig. 2. Example GUI interface of B_gcpSelection (A) and E_scpSelection (A+B) using uasDemoData. (A) SCP and GCP clicking panel. (B) Panel displaying pixel intensity values within search radius of SCP 4 and resultant center of area calculations (white dot) when applying binary threshold.. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Despite its success, the *UAV Toolbox* has limitations and bootcamp participants have provided extensive feedback. Both novice and more experienced users find it difficult to implement for their own data, particularly outside the narrow UAS application. The original *UAV Toolbox* demo runs as a focused executable for UAS data and back-end sub-functions are highly decimated, requiring very specific data structures and metadata. This made it challenging for novices to extract key photogrammetric processing steps to develop their own code.

To address these issues, CIRN has developed the next generation bootcamp code: *The CIRN Quantitative Coastal Imaging Toolbox*, which is designed to be an instructive photogrammetry software development kit (SDK) applicable to a broad range of coastal imaging applications. The presented toolbox is a linear set of scripts to carry out end-to-end georectification processing of coastal imagery from both UAS and single (or multi) camera land-based stations. The toolbox demonstrates how to extract images from movie files, identify the pixel location of known ground control points (GCPs) in the field of view, solve for camera extrinsics (position and orientation), correct for movement between frames, georectify the imagery, calculate ensemble statistical rectification products, and extract rectified time-series for advanced processing. Scripts are segmented to save intermediate results and highlight key photogrammetry fundamentals. Sub-functions are prominently displayed in scripts and are aggregated to perform core functionalities. As a result, the presented toolbox acts as a turn-key application for novice users but also provides a repository of key sub-functions as users gain more experience to develop their own code. Ultimately, the goal of the presented toolbox is to bypass the photogrammetry impedance that exists for many coastal engineers, geoscientists, and oceanographers, and proliferate the use of video-monitoring in the wider coastal community.

2. Software description

The *CIRN Quantitative Coastal Imaging Toolbox* is series of MATLAB scripts that can be utilized to georectify sets of oblique imagery from stationary UAS and single (or multi) –camera land-based imagery for generation of classic coastal imaging data

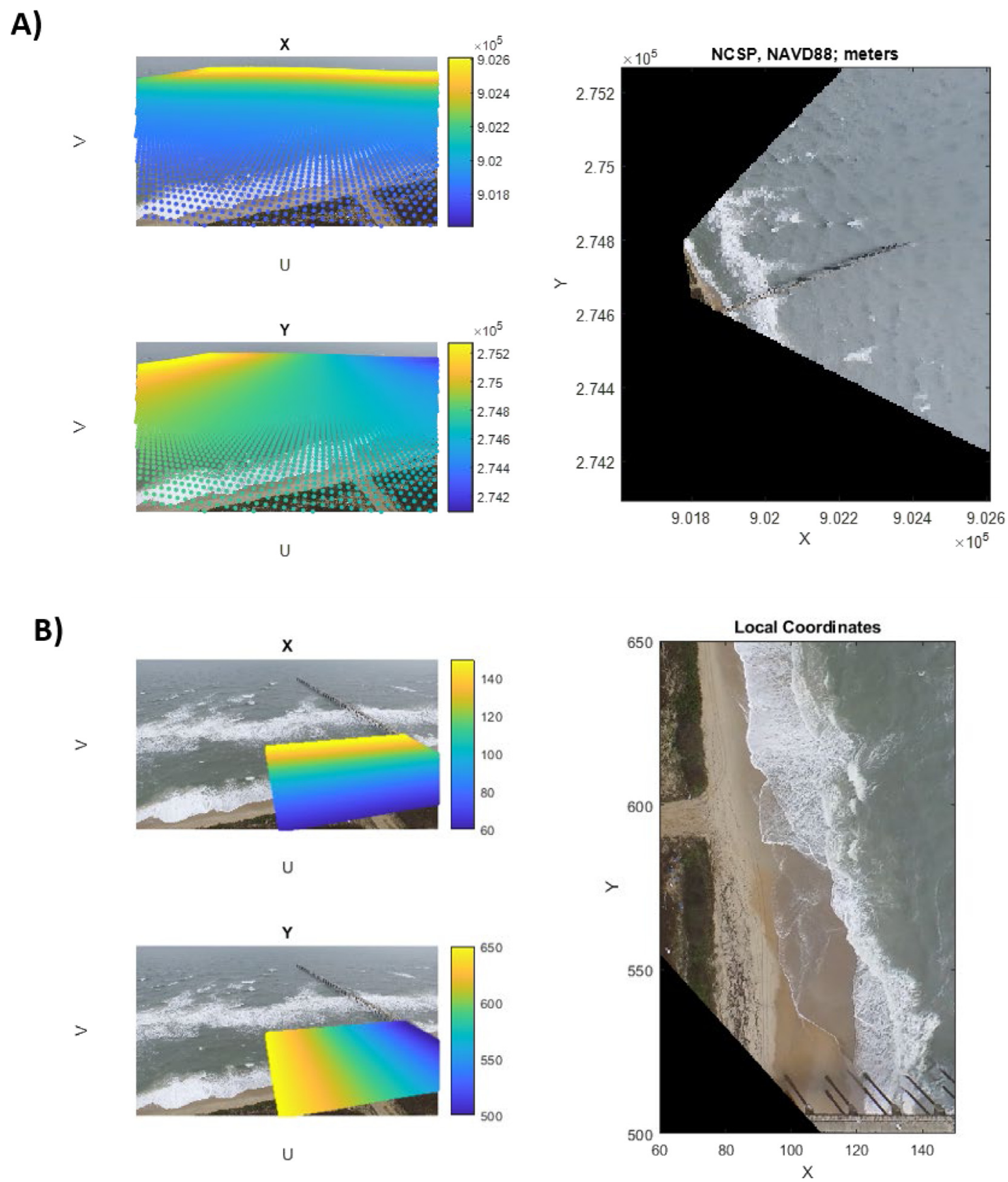

Fig. 3. Example output of D_gridGenExampleRect using uasDemoData. (A) Example geo-rectified image in geographic coordinates with low 5-m resolution (right panel). Left panels are reprojections of rectification grids in oblique imagery. Colors are X (top) and Y (bottom) coordinates. (B) Example with higher 10 cm resolution, smaller extent and displayed in local rotated coordinate system.. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

products [1]. The scripts prominently call and highlight a series of sub-functions that implement fundamental photogrammetry calculations. The scripts are prepopulated with demo UAS and multi-camera data and can be run as downloaded. To adapt to other data sets, the user is required to provide: image data with suitable ground control points (GCPs), GCP geographic coordinates, and camera intrinsic calibration parameters. More details on collection requirements for georectified imagery can be found in [22] and camera intrinsic calibrations at [23].

2.1. Software architecture

The application is a series of linear MATLAB m-scripts that demonstrate a series of functional steps: (1) pre-processing (image extraction from a video file); (2) camera calibration (solving camera extrinsics); (3) projection domain definition; (4) image stabilization; and (5) generation of rectified products (Fig. 1). These scripts can be run in different processing sequences depending on the application. Script modularity reduces replicative code and highlights commonalities between image collection techniques (UAS vs Ground). The user-manual provides detailed dependencies, input, and output of scripts and sub-functions.

Scripts are run sequentially as inferred by the alphanumeric prefixes in the filenames and Fig. 1. Each script represents a key photogrammetric processing step or decision point, (e.g. what GCPs to use) and saves intermediate products (e.g. extrinsics, etc.) for input into the next script. The intermediate products allow the user flexibility in processing and an ease of exploring processing decisions (e.g. changing coordinate systems or grid resolutions) without having to restart the entire end-to-end processing. Metadata such as GCP projection errors are saved in the output mat-files as well.

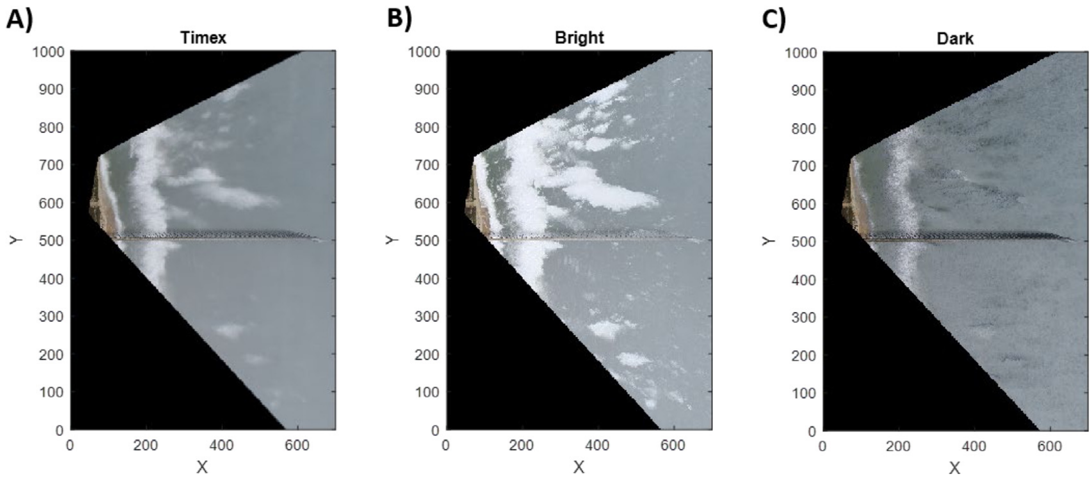

Fig. 4. Example ensemble output of G1_imageProducts for uasDemoData in a local rotated coordinate system.

In addition to the demonstrative scripts in the main repository directory, there is a X_CoreFunctions folder that contains subfunctions that execute fundamental photogrammetry processes (e.g. solving camera extrinsics and calculating a camera projection matrix) . These functions can also be used independently of the demonstrative scripts making them easily adaptable to a user's needs.

The remaining folders in the main directory contain demonstration data for UAS (X_UASDemoData) and multi-camera land-based stations (X_FixedMultiCamDemoData). The processing sequences for each demo are highlighted in Fig. 1. The demo scripts prefixed (A0-G2) are prepopulated to process the UASDemoData end-to-end. The FixedMultiCamDemoData path starts at D_gridGenExampleRect since geometry solutions are already provided, and requires users to uncomment Section 4 of the code to execute the correct input parameters. Scripts prefixed G1-G2 are also prepopulated for FixedMultiCamDemoData and both require Section 5 to be uncommented. Both demos are demonstrated in Section 3.

3. Illustrative examples

The repository has two sets of demonstration data and is prepopulated to process both. In this section, both processing sequences (UAS and Fixed Station, Fig. 1) are demonstrated. The UAS path is demonstrated first since it runs the entire processing workflow and requires no user modification. Then, the products from the multi-camera demonstration data set are presented to highlight the multi-camera capability. Imagery of intermediate output is not shown but can be found in the user manual.

3.1. UASDemoData

To run the UASDemoData set, the user first runs the A0_movie2Frames script. This script converts a video file to a collection of sequential oblique images at a user-specified framerate. The example file is 4 K video taken by a DJI Phantom Pro 4 at Duck, NC USA at 60 m elevation. The video has been modified with annotations and reduced framerate (2 Hz) for demonstrative purposes.

The user then runs A_formatIntrinsics. This script loads a user-specified calb_result.mat file from the Cal-Tech Camera Calibration Toolbox, a third-party MATLAB repository to perform camera intrinsic calculations [23], and outputs correctly formatted camera intrinsic coefficients. More advanced users can skip this step, entering camera intrinsics manually if calculated themselves. The user-manual provides more information on the coefficient definitions and distortion model utilized by the toolbox.

B_gcpSelection loads the first frame of the UAS video (or any oblique image as specified by the user) and presents a GUI for the user to click GCPs (Fig. 2A), which are then identified numerically in the command window. Fig. 2A shows the raw first frame which has been modified to show GCP locations and numbering suggestions for the user. Once selection is completed, the function outputs a mat-file with the image (UV) coordinates of the GCPs.

The next function, C_singleExtrinsicSolution, solves for the extrinsics of the first frame utilizing mat-files output by B_gcp Selection (GCP UV coordinates) and A_formatIntrinsics (intrinsic information) in addition to a comma delimited text file specifying the XYZ coordinates of the clicked GCPs, linked by the GCP number. The script uses a nonlinear solver (nlinfit, in MATLAB Statistical Toolbox) to solve for the combined camera extrinsic/intrinsic (EOIO) solution to reproject XYZ GCP points and minimize the error between their UV counterparts. Users can specify which combination of GCPs to utilize and must input an initial guess. The XYZ values can be in any geographic reference frame. The function saves a mat-file of the camera EOIO along with calculated extrinsic uncertainties and GCP reprojection errors (a horizontal RMSE of 0.14 m is calculated in this example). The script also displays a figure of the first frame with clicked and reprojected GCP values to qualitatively evaluate the solution.

Next, the user defines the desired rectification grid and a rotated coordinate system aligned with the coastline in D_gridGenExampleRect by entering grid limits/resolution and local rotation angle/origin. Run as is, the script will load the output from C_singleExtrinsicSolution (EOIO solution) and produce corresponding georectified images of the first frame in both geographic (Fig. 3A) and rotated local references frames. Rotated coordinate systems are frequently used to easily delineate cross and alongshore processes and are generally required for many optical coastal analysis. The script provides outputs for both reference frames including .pngs of the example rectifications and a mat file with the rectification grids and rotation parameters. Note, the demo grids assume a flat user-entered elevation, typically the tide elevation during the flight. However for more accurate rectification (ortho-rectification) the user can modify the script to incorporate varying relief. The script is an instructive and exploratory tool to determine effects of grid resolution, rotation, extent and Z-elevation. Fig. 3B demonstrates rectifications at a user-specified higher resolution in a local coordinate system. In addition to the geo-rectified image, the script produces figures where the rectification grids are reprojected onto the oblique

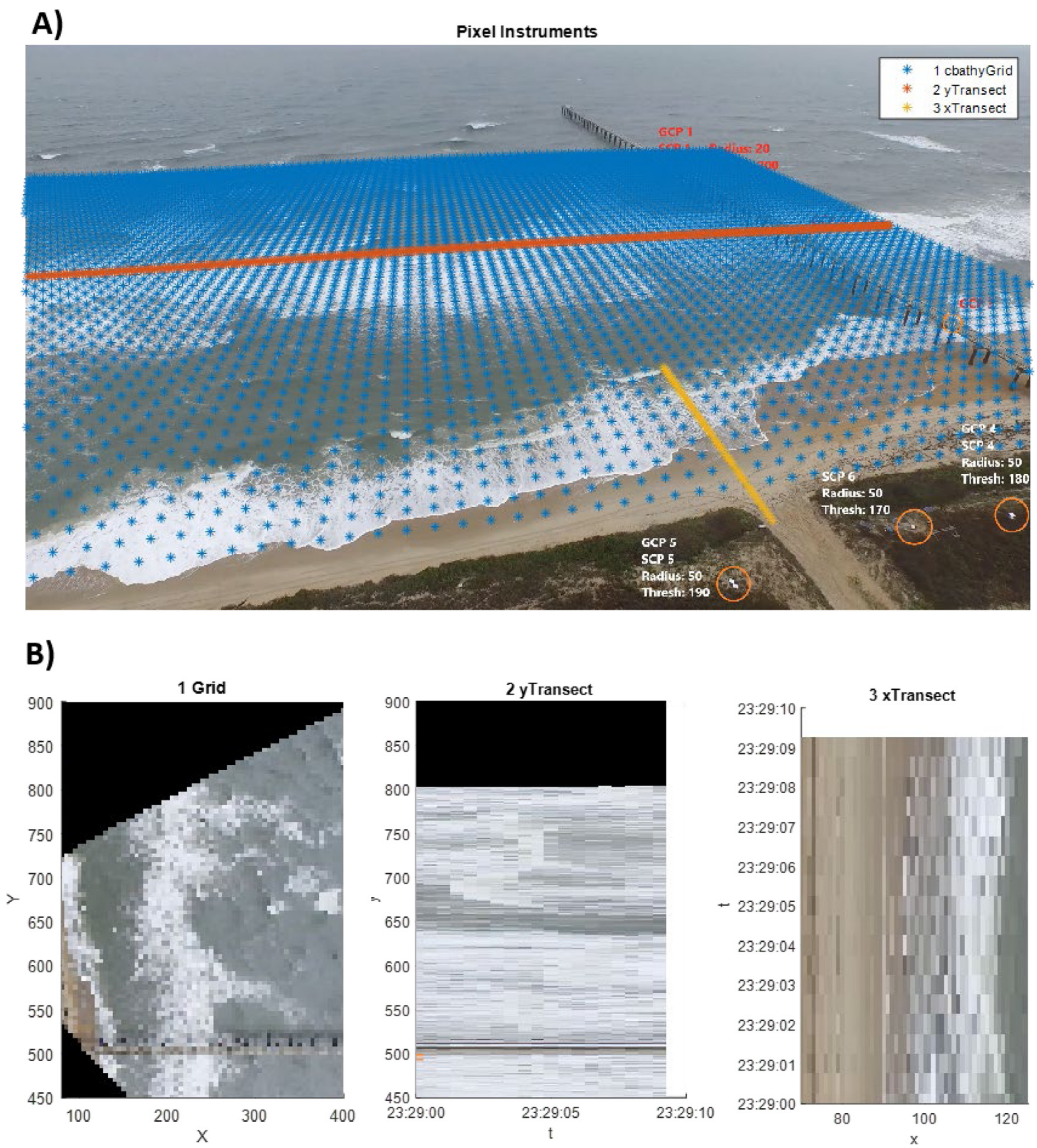

Fig. 5. Example output of G2_pixelInstruments for uasDemoData. (A) Location of pixel transects reprojected onto oblique image. (B) Example pixel instrument data from left to right. 5 m × 5 m grid (first frame), Alongshore transect timestack at X = 225 m and 0.2 m resolution, and cross-shore timestack at Y = 600 m and 0.2 m resolution.

image (Fig. 3). This is a useful tool for evaluating pixel density and explaining the rectification process.

The next function is E_scpSelection, where the user selects Stabilization Control Points (SCPs) that are used to adjust for slight movements of the UAS during collection. The script runs similarly to B_gcpSelection, allowing the user to define SCP locations in the image, but also prompting the user to identify a search radius and pixel intensity threshold (Fig. 2B). The aim is to pick a threshold where only a bright or dark stationary object is identified (dark red color), so that its center of area (white dot) can be consistently and automatically calculated within the defined search radius in pixels. Like in B_gcpSelection, the first frame of uasDemoData has suggestions for the users for SCP, radius, and threshold values. The output of this function is a mat file with SCP UV coordinates as well as thresholds and radii.

The script F_variableExtrinsicSolution utilizes the initial IOEO from C_singleExtrinsicSolution and SCP information from E_scpSelection to calculate extrinsic solutions for a series of sequential oblique images. Similarly to C_singleExtrinsicSolution, the script uses a nonlinear solver to solve each frame's extrinsics using the automatically identified SCPs. The script plots calculated SCP centers for each new frame in a figure and the function outputs a mat file with EOIO solutions for each frame as well as a figure plotting the camera extrinsics through time.

G1_imageProducts rectifies each individual frame given the corresponding EOIO frame solved for in F_variableExtrinsic Solution to the user-specified rectification grid produced by D_gridGenExampleRect. The script displays a rectified example frame and outputs sequential frames as png files in either reference frame as well as pngs of georectified statistical image products of the collection ensemble (Fig. 4). More information on statistical products and utilization can be found in [1].

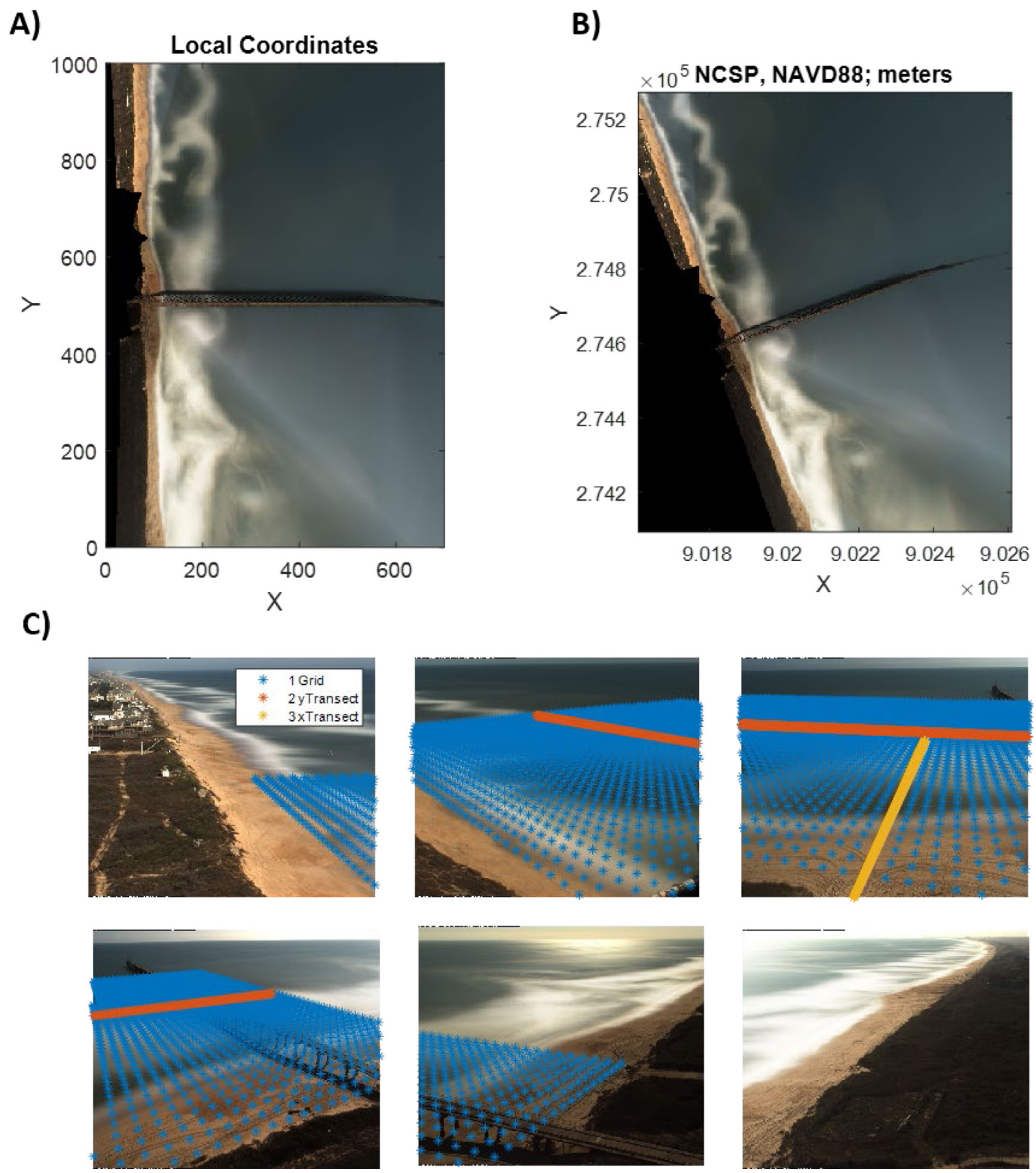

Fig. 6. Example output of D_gridGenExampleRect (A–B) and G2_pixelInstruments (C) for fixedMultiCamDemoData. Example georectified image in (A) local coordinates. And (B) geographic coordinates (C) Pixel instruments projected across image space in multiple cameras.

G2_pixelInstruments produces mat files of pixelInstruments (pixel arrays) used by optical-based geophysical processing algorithms. The script runs similarly to G1_imageProducts, requiring the same EOIO and grid input from F_variableExtrinsicSolution and D_gridGenExampleRect respectively, but instead of georectifying and saving full imagery, it rectifies single geo-spatial transects or subsampled grids. These data points are rectified for each sequential oblique image and saved sequentially in a mat file. The script is prepopulated to create an example pixel array, alongshore transect, and cross-shore transect and plots an example of each (Fig. 5B) as well as their location reprojected onto the first frame imagery (Fig. 5A).

3.2. FixedMultiCamDemoData

To process the multi-camera dataset, the user needs to uncomment sections in D_gridGenExampleRect, G1_imageProducts, and G2_pixelInstruments to direct the functions to the correct image directories, grid files, and EOIO solutions. FixedMultiCamDemoData data is from six fixed cameras on top a 43-m tower at Duck, NC; each oblique image is a 10-min time average taken every 30 min on October 8, 2015.

The fixed station path starts with function D_gridGenExample Rect, assumes no camera movement, and uses pre-populated IOEO solutions from the multiple cameras (FixedMultiCamDemoData; Fig. 6A–B). For user's own data, scripts prefixed A-C would have to be run for each camera to determine the IOEO solution. . Since at the same location as the uasDemoData, the grid parameters are the same for both demos. For multi-camera data, D_gridGenExampleRect is also a tool to explore georectification accuracy between camera seams (blended using a feathering method [1]).

G1_imageProducts and G2_pixelInstruments functions output the same products and figures for FixedMultiCamDemoData as uasDemoData. Both functions allow the user to enter/load a varying tidal elevation vector to account for the change in tidal and therefore rectification elevation over extended periods of time; this vector overrides the specified elevation in D_gridGenExampleRect. Fig. 6C highlights pixel instruments reprojected across multiple cameras. Since each image is taken at 30 min interval, these data products are daily averages and timestacks as compared to the 10 min uasDemoData collect.

Functions E_scpSelection and F_variableExtrinsicSolution do not need to be run for the FixedMultiCamDemoData since there is no camera movement. However, they could be applied to fixed camera imagery if daily thermal expansion causes movement of the cameras and there are fixed points that could serve as SCPs. Subsequent functions do have the option of inputting a time varying extrinsic solutions for fixed stations.

4. Impact

The primary impact of the *CIRN Quantitative Coastal Imaging Toolbox* is to eliminate the photogrammetry roadblock that many coastal engineers, geoscientists, and oceanographers face when implementing quantitative video analysis in their own work. Quantitative video techniques are transformative technology for monitoring many engineering and research projects. Imagery of the coast can be exploited qualitatively and quantitatively to provide information on coastal processes, beach & dune topography, nearshore water depths, sandbar positions, wave and runup elevations, as well as the condition of coastal infrastructure or navigability of harbor or inlet entrances.

With lower operational costs, in both finance and time, engineers and scientists can respond and monitor more nearshore processes in a more timely fashion with video monitoring. The recent proliferation of low-cost video technology (smartphones, UAS, go-Pros), makes the hardware for this technology more accessible than ever. The presented toolbox aims to provide an equally accessible software solution and the *impact will be an increase* in quantitative coastal data in a multitude of locations, environmental conditions, and spatial scales which could ultimately help reduce coastal hazards worldwide.

5. Conclusions

The *CIRN Quantitative Coastal Imaging Toolbox* is a collection of MATLAB scripts to produce geo-rectified images from UAS and land-based camera imagery specifically tailored for quantitative analysis of coastal environments. While experts have employed similar scripts operationally for years, this toolbox is for photogrammetry novices and part of the CIRN curriculum. Demos teach fundamentals but also can be easily transitioned to applications for operational data processing. The presented toolbox acts as a turn-key application for novice users but also a repository of key sub-functions as users gain more experience to develop their own code.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Funding was provided by the Coastal Ocean Data Systems (CODS) program under the US Army Corps of Engineers (USACE). The authors would like to thank the CIRN community for evaluating the code and providing valuable feedback. In addition, great gratitude is extended to the original Argus code developers, Dr. Rob Holman and John Stanley, for their extraordinary guidance and repository foundation.

References

[1] Holman RA, Stanley J. The history and technical capabilities of Argus. Coastal Eng 2007;54(6):477–91. http://dx.doi.org/10.1016/j.coastaleng.2007.01.003.
[2] Holman R, Plant N, Holland T. cBathy: A robust algorithm for estimating nearshore bathymetry. J Geophys Res Ocean 2013;118(5):2595–609.
[3] Cohen AB, Aarninkhof SGJ, Chickadel CC. Video-derived observations of longshore currents. In: Coastal engineering 2004: (in 4 Volumes). World Scientific; 2005, p. 1468–79.
[4] Bergsma EWJ, Almar R. Video-based depth inversion techniques, a method comparison with synthetic cases. Coastal Eng 2018;138:199–209.
[5] Plant NG, Holland KT, Haller MC. Ocean wavenumber estimation from wave-resolving time series imagery. IEEE Trans Geosci Remote Sens 2008;46(9):2644–58. http://dx.doi.org/10.1109/TGRS.2008.919821.
[6] Alexander PS, Holman RA. Quantification of nearshore morphology based on video imaging. Mar Geol 2004;208(1):101–11. http://dx.doi.org/10.1016/j.margeo.2004.04.017.
[7] Baldock TE, Moura T, Power HE. Video-based remote sensing of surf zone conditions. IEEE Potentials 2017;36(2):35–41.
[8] Harley MD, Turner IL, Short AD, Ranasinghe R. Assessment and integration of conventional, RTK-GPS and image-derived beach survey methods for daily to decadal coastal monitoring. Coastal Eng 2011;58(2):194–205.
[9] Pape L. BLIM toolbox manual. IMAU rep. R08-02, dep. phys. geogr. utr. univ., 2008, available from author upon Req.
[10] Ruessink BG, Pape L, Turner IL. Daily to interannual cross-shore sandbar migration: Observations from a multiple sandbar system. Cont Shelf Res 2009;29(14):1663–77.
[11] Power HE, Holman RA, Baldock TE. Swash zone boundary conditions derived from optical remote sensing of swash zone flow patterns. J Geophys Res Ocean 2011;116(C6).
[12] van Dongeren A, Plant N, Cohen A, Roelvink D, Haller MC, Catalán P. Beach Wizard: Nearshore bathymetry estimation through assimilation of model computations and remote observations. Coastal Eng 2008;55(12):1016–27. http://dx.doi.org/10.1016/j.coastaleng.2008.04.011.
[13] Holland KT, Puleo JA, Kooney TN. Quantification of swash flows using video-based particle image velocimetry. Coastal Eng 2001;44(2):65–77.
[14] Almar R, Blenkinsopp C, Almeida LP, Cienfuegos R, Catalán PA. Wave runup video motion detection using the radon transform. Coastal Eng 2017;130:46–51. http://dx.doi.org/10.1016/j.coastaleng.2017.09.015.
[15] Valentini N, Saponieri A, Molfetta MG, Damiani L. New algorithms for shoreline monitoring from coastal video systems. Earth Sci Inf 2017;10(4):495–506.
[16] Almar R, et al. A new breaking wave height direct estimator from video imagery. Coastal Eng 2012;61:42–8.
[17] Almar R, Bonneton P, Senechal N, Roelvink D. Wave celerity from video imaging: a new method. In: Coastal engineering 2008: (in 5 Volumes). World Scientific; 2009, p. 661–73.
[18] Plant NG, Aarninkhof SGJ, Turner IL, Kingston KS. The performance of shoreline detection models applied to video imagery. J Coast Res 2007;658–70.
[19] Aarninkhof SGJ, Ruessink BG, Roelvink JA. Nearshore subtidal bathymetry from time-exposure video images. J Geophys Res Ocean 2005;110(C6).
[20] Holman RA, Brodie KL, Spore N. Surf zone characterization using a small quadcopter: Technical issues and procedures. IEEE Trans Geosci Remote Sens 2017;55(4):2017–27. http://dx.doi.org/10.1109/TGRS.2016.2635120.
[21] Vos K. Remote sensing of the nearshore zone using a rotary-wing UAS. University of New South Wales; 2017.
[22] Holland KT, Holman RA, Lippmann TC, Stanley J, Plant N. Practical use of video imagery in nearshore oceanographic field studies. IEEE J Ocean Eng 1997;22(1):81–91. http://dx.doi.org/10.1109/48.557542.
[23] Bouguet J-Y. Camera calibration toolbox for Matlab, vol. 1080. 2008, URL http://www.vision.caltech.edu/bouguetj/calib_doc.