RivWidthCloud: An Automated Google Earth Engine Algorithm for River Width Extraction From Remotely Sensed Imagery

Xiao Yang, Tamlin M. Pavelsky, George H. Allen, and Gennadii Donchyts

***Abstract*— The wetted width of a river is one of the most important hydraulic parameters that can be readily measured using remote sensing. Remotely sensed river widths are used to estimate key attributes of river systems, including changes in their surface area, channel storage, and discharge. Although several published algorithms automate river network and width extraction from remote sensing images, they are limited by only being able to run on local computers and do not automatically manage cloudy images as input. Here we present RivWidthCloud, a river width software package developed on the Google Earth Engine cloud computing platform. RivWidthCloud automatically extracts river centerline and widths from optical satellite images with the ability to flag observations that are obstructed by features like clouds, cloud shadows, and snow based on existing quality band classification. Because RivWidthCloud is built on a popular cloud computing platform, it allows users to easily apply the algorithm to the platform's vast archive of remote sensing images, thereby reducing the users' overhead for computing hardware and data storage. By comparing RivWidthCloud-derived widths from Landsat images to *in situ* widths from the U.S. and Canada, we show that RivWidthCloud can estimate widths with high accuracy (root mean square error: 99 m; mean absolute error: 43 m; mean bias: −21 m). By making RivWidthCloud publicly available, we anticipate that it will be used to address both river science questions and operational applications of water resource management.**

***Index Terms*— Discharge, Google Earth Engine (GEE), remote sensing, river width.**

I. Introduction

VARIATIONS in centerline and width along river channels have been used to infer the morphodynamics of rivers and river deltas [1]–[3]. Moreover, repeated width measurements can be used to estimate river discharge based on classic hydraulic geometry relationships [4]. While *in situ* gauging stations measure river water level (height) to estimate discharge, the numbers of stations providing data to the public have declined since the 1980s [5], due to both decreased monitoring budgets and changes in international data sharing policies. Fortunately, recent advancements in theory and algorithms have enhanced our ability to gauge rivers from space. Among various satellite-based river discharge estimation methods that use river observations, many require river width as input. For example, point-scale and reach-scale [6] multitemporal river widths have been used to estimate river discharge using hydraulic geometry relationships established with *in situ* discharge measurements. Moreover, the recently discovered at-many-stations hydraulic geometry (AMHG) allows for estimating river discharge using widths alone [7]. In combination with the increasing number of earth-observing satellite missions, including the forthcoming Surface Water and Ocean Topography (SWOT) mission [8], satellite-derived widths will help us observe ungauged rivers and supplement *in situ* gauging networks, ensuring greater monitoring consistency globally.

Most existing algorithms to extract river widths are designed for optical images, and their application is heavily limited by the presence of clouds, as they require cloud-free images as input [1], [2], [9]–[14]. At any given time, clouds cover ∼67% of the earth's surface [15] and therefore a river width algorithm that can automatically run on images with cloud cover will vastly increase the amount of data available for river monitoring purposes.

Here, we present RivWidthCloud, the first Google Earth Engine (GEE)-based river width algorithm that automatically extracts river centerlines and widths from remotely sensed images (cloud-free or not) with minimal user input and minimal required computation hardware. By adapting the widely used RivWidth algorithm [13] to the GEE platform [16], this new tool contains several novel features that allow for rapid river width extraction from partially cloud-covered images. A suite of accompanying visualization and diagnostic tools and a detailed user manual are included in the software repository. In what follows, we describe the algorithm, validate its output, and discuss potential next steps for RivWidthCloud.

II. Method

Calculating river widths using RivWidthCloud incorporates three main steps: 1) extracting a river mask from a satellite image; 2) deriving a river centerline from the river mask; and 3) measuring river width along the centerline.

Manuscript received March 15, 2019; revised May 15, 2019 and May 22, 2019; accepted May 28, 2019. This work was supported by the SWOT Project Office, NASA/Caltech Jet Propulsion Laboratory. *(Corresponding author: Xiao Yang.)*

X. Yang and T. M. Pavelsky are with the Department of Geological Sciences, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599 USA (e-mail: yangxiao@live.unc.edu; pavelsky@email.unc.edu).

G. H. Allen is with the Department of Geography, Texas A&M University, College Station, TX 77843 USA (e-mail: geoallen@tamu.edu).

G. Donchyts is with Deltares, 2600 Delft, The Netherlands (e-mail: gennadii.donchyts@deltares.nl).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

Color versions of one or more of the figures in this letter are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LGRS.2019.2920225

RivWidthCloud in principle works with any remote sensing product capable of providing classifications of water and visual obstructions (cloud, snow/ice, etc.). The methods for classifying water and the accuracy of such classifications vary among remote sensing products. Thus, the software is structured so that new classification modules associated with different remote sensing products could be easily inserted. Because of its accurate cloud classification algorithm [17] and its long mission history, we describe and validate RivWidthCloud using data from the Landsat program, specifically imagery from the Thematic Mapper (TM), Enhanced TM Plus, and Operational Land Imager instruments.

A. Developing the River Mask

To classify surface water [Fig. 1(b)], RivWidthCloud uses a simple, spectral-based water classification formula with a reported accuracy of ∼97% [18]. The water classification makes use of multiple spectral indexes like modified normalized difference water index [19], normalized difference vegetation index [20], and enhanced vegetation index [21] with thresholds determined based on training data across the contiguous U.S. Then, RivWidthCloud relies on the Global River Widths from Landsat (GRWL) dataset [22] to distinguish river pixels from nonriver water pixels. GRWL contains width values associated with each centerline location, derived from Landsat images using RivWidth, for rivers wider than 30 m globally. RivWidthCloud calculates a channel mask [Fig. 1(c)] as all water pixels [Fig. 1(b)] that are connected to the GRWL river centerlines. By default, the algorithm checks connectivity to the GRWL centerline within a 4-km radius, a modifiable value that allows measurement of rivers with the maximum width ≤8 km. RivWidthCloud then removes all islands with a surface area smaller than a default of ∼0.3 km^2 (333 Landsat pixels, also a modifiable value) in the channel mask to reduce the topological complexity of the resulting centerline. After removing the islands, the image is then referred to as a river mask [Fig. 1(d)].

We note that the accuracy of the channel/river mask and the widths derived from them is highly dependent on the accuracy of the inundation map, and hence on the water classification method that is used to generate the inundation map. While RivWidthCloud uses a default spectral-based water classification method, an alternative, the Dynamic Surface Water Extent (DSWE) classification method [23], is also available in RivWidthCloud, and can be easily selected by the user.

B. Delineating the River Centerline

There are two major steps in generating a 1-pixel-wide river centerline: 1) deriving a raw centerline by applying three numeric operations consecutively on the river mask: distance transform, gradient, and skeletonization and 2) removing the spurious branches from the raw centerline. The details of these two steps are provided in the following paragraphs.

The distance transform assigns each river pixel the value of its distance to the closest nonriver pixel [Fig. 1(e)]. After the distance transform, pixels close to the center of the river have local maximum values, as they are furthest

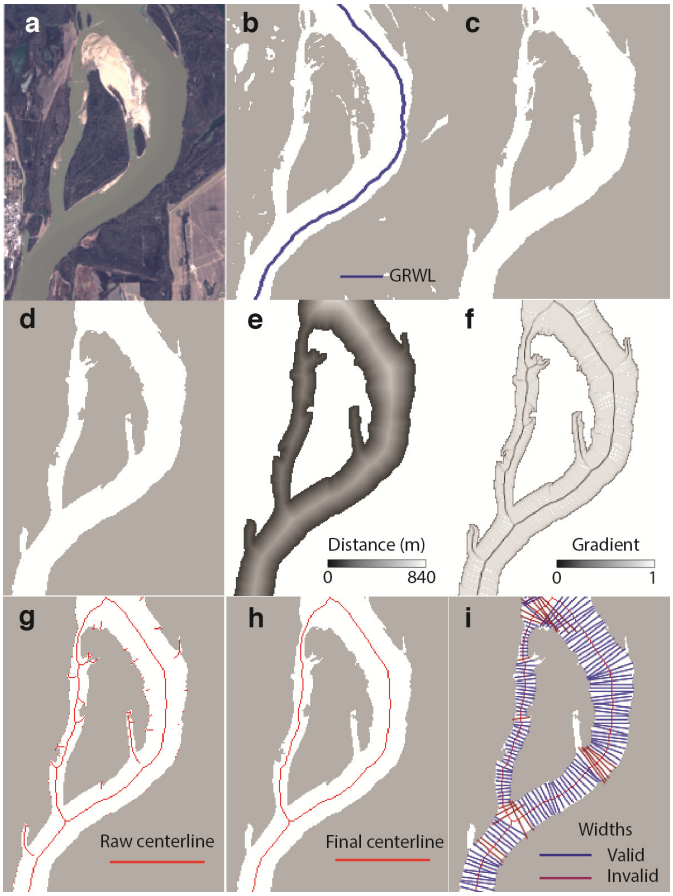

Fig. 1. Steps to derive river centerlines. (a) Landsat RGB image. (b) Water mask. (c) Channel mask. (d) River mask. (e) Distance to closest nonwater pixel. (f) Gradient of the distance map. (g) Raw 1-pixel-wide centerline. (h) Final river centerline after removing spurious branches. (i) Cross-sectional lines with length scaled by the widths calculated using RivWidthCloud (only every third cross-sectional line plotted).

from the banks. Next, RivWidthCloud convolves the distance map with a pair of 3 × 3-pixel kernels (1–2) to generate a gradient map [Fig. 1(f)]. While the maximum distance scales with river size, the gradient close to the centerline is nearly always a local minimum and close to zero [Fig. 1(f)]. Thus, the algorithm generates an initial centerline by classifying gradient-map pixels with values less than 0.9 as belonging to the centerline. This threshold was chosen from the range used by RivWidth [14] to balance algorithm speed—achieved with a lower threshold, and improved centerline connectivity from using a higher threshold. To ensure that the centerline is only one pixel wide, three iterations of skeletonization, or thinning, were applied to the initial centerline mask [Fig. 1(g) shows the resulting 1-pixel-wide centerline]. The implementation of skeletonization in GEE is modified from [24].

$$\begin{bmatrix} \frac{1}{8} & 0 & -\frac{1}{8} \\ \frac{2}{8} & 0 & -\frac{2}{8} \\ \frac{1}{8} & 0 & -\frac{1}{8} \end{bmatrix} \tag{1}$$

$$\begin{bmatrix} -\frac{1}{8} & -\frac{2}{8} & -\frac{1}{8} \\ 0 & 0 & 0 \\ \frac{1}{8} & \frac{2}{8} & \frac{1}{8} \end{bmatrix}. \quad (2)$$

The resulting 1-pixel-wide centerlines often exhibit spurious branches—artifacts mainly produced by irregular bank shapes. RivWidthCloud trims these branches by starting from the endpoint of each centerline and removing either 500 pixels (∼1500 m for Landsat) or the total length of the segment to the next confluence, whichever is shorter. The resulting final centerline [Fig. 1(h)] preserves the river's channel structure such that each channel retains its own centerline, and as a result, each channel's width is measured separately. This approach differs substantially from that of the RivWidth algorithm [13], which outputs a single centerline with total flow width summed across all channels.

C. Calculating River Width

Once the 1-pixel-wide centerline is computed, RivWidthCloud uses two steps to calculate river width. First, for each centerline pixel, it calculates the direction orthogonal to the local centerline. Second, it computes the river width along these orthogonal directions.

To calculate the orthogonal direction, the algorithm convolves the river centerline image with a 9 × 9-pixel kernel (see (S1) in the Supplemental Materials) filled with values of zero except for the pixels along the rim. The value of each rim pixel is assigned in the following manner: Starting from the center-right position moving anticlockwise, each pixel is assigned the value of the angle (in degrees) between the horizontal line and the line connecting the pixel itself and the kernel center. For example, the rim pixel at top-right corner is assigned a value of 45°, and the rim pixel at top-center is equal to 90°. The values of the centerline pixels from the convolved image, when divided by two, approximate their orthogonal directions. However, the resulting orthogonal direction is only accurate when exactly two pixels intersect the rim of the kernel, one from each end of the overlapping centerline segment (Fig. 2). For pixels located near the end of river centerline that intersect with the kernel once, we estimated their orthogonal direction by adding 90° to the convolved value. We then discard those orthogonal directions if more than two centerline pixels intersect the kernel—a condition that often occurs close to confluences.

After obtaining the orthogonal directions, the river width is calculated for each centerline pixel as follows. First, RivWidthCloud generates a line segment w_m along the orthogonal direction with length equals three times the distance to the closest nonwater pixel (w_b; Fig. 2); then, the algorithm calculates the river width (w) by multiplying w_m by the mean value of the channel mask [Fig. 1(c)] along w_m.

$$w = w_m \times u \quad (3)$$

$$u = \frac{w}{w_m} = \frac{\sum_{i=1}^{N} p_i}{N} \quad (4)$$

where N is the total number of pixels intersecting the segment w_m, and p_i is the value of the pixel i on the channel mask. The mean value u equals to the ratio between the wetted length of w_m and w_m itself. This particular method to calculate the final widths is chosen for its easy implementation in GEE.

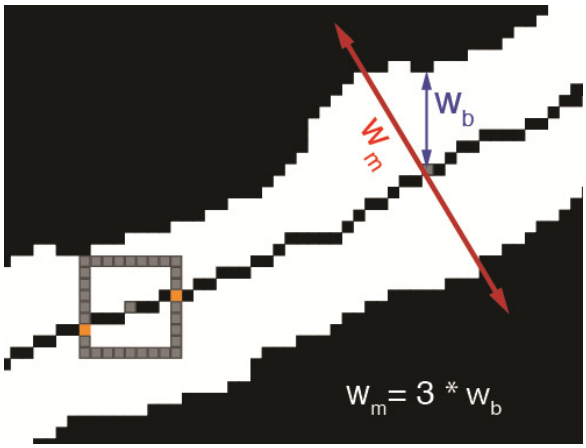

Fig. 2. Calculating the orthogonal direction and river width. The box is a 9 × 9-pixel kernel that is used to calculate the orthogonal direction for its center point (gray). The red line along the orthogonal direction of the centerline is used to calculate river width.

D. Assigning Quality Flags

The accuracy of RivWidthCloud's width measurements depends on the accuracy of the water classification, which can be impacted by clouds, cloud shadows, topographic shadows, and the presence of snow/ice. Recent work has pioneered the ability to distinguish these features using a machine learning approach [25], [26], and we expand on this ability by allowing such processing automatically in GEE, which allows easy global application. Specifically, with each width measurement, RivWidthCloud provides several quality flags indicating the potential influence of each of these features [see Fig. 1(i) for an example of RivWidthCloud derived centerline, orthogonal direction, and widths; widths flagged are shown as invalid].

Cloud, Cloud Shadow, Snow/Ice: When using Landsat imagery as input, RivWidthCloud uses the Fmask classification algorithm [17] to flag whether a width measurement is affected by clouds, cloud shadows, or snow/ice. Each of these conditions has its own flag, which was assigned the mean value of its corresponding image along the cross-sectional line w_m. Thus, a flag with a nonzero value suggests the width measurement may be impacted by the flagged condition.

Topographic Shadow: RivWidthCloud estimates the extent of topographic shadows using the *ee.Terrain.hillShadow* function from GEE. The Multi-Error-Removed Improved-Terrain DEM (MERIT DEM) [23] and the image specific sun elevation and azimuth are used to estimate the topographic shadow. For each width measurement, the flag for the topographical shadow is assigned the mean shadow value along the cross-sectional line w_m.

III. Results and Discussion

To test the performance of RivWidthCloud, we validated the widths it produces against *in situ* width measurements from the United States Geological Survey (USGS) and

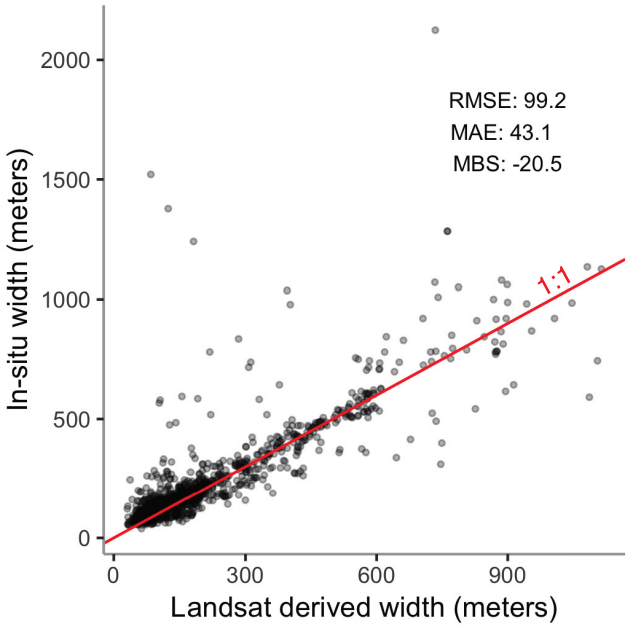

Fig. 3. Validation of Landsat-derived river width with in situ width measurements from the USGS and Water Survey of Canada.

the Water Survey of Canada. For the period 1984–2018, we obtained 1514 valid (flag-free) widths from Landsat images that had same-day *in situ* measurements at gauging stations, across the U.S. (492 stations with 1476 widths) and Canada (27 stations with 38 widths). We found that the widths computed using RivWidthCloud on Landsat imagery (w_L) closely matched the *in situ* measurements (w_G) (root mean square error = 99.2 m; mean absolute error = 43.1 m; mean bias = −20 m) (see also Fig. 3). Large inconsistencies between widths obtained *in situ* and remotely were mainly from stations where: 1) nearby river widths changed rapidly or 2) the *in situ* width was taken a certain distance upstream or downstream from the gaging station. In both situations, the *in situ* widths did not serve as good validation data (validation can be assessed for each individual station using the RivWidthCloud GEE app: https://eeproject.users.earthengine.app/view/rivwidthcloud-validation). For the majority of the width validation pairs, the accuracy of Landsat-derived widths depends on the accuracy of the image classification—the classification for water determines the extent of the river while the classification of cloud/shadow/snow/ice determines the flagging accuracy.

The centerlines from RivWidthCloud by default do not guarantee connectivity at confluences. This lack of connectivity is a necessary side-effect of our default method of calculating gradient to accelerate the centerline computation. In cases where connectivity is needed, it can be enforced by directly calculating the centerline from the river mask via skeletonization only, skipping the steps of distance transform and gradient. However, solely relying on skeletonization to calculate the centerline is slow and could potentially exceed the computational resources available to a GEE user when applied over a large area.

IV. Conclusion

We present RivWidthCloud, an automated river width extraction algorithm that uses the cloud computing services available on GEE. By validating river widths calculated using RivWidthCloud with same-day *in situ* width measurements, we show that the algorithm can accurately extract river width from Landsat images. It is likely that the error assessment included here is conservative because it does not factor in errors in *in situ* width measurements or mismatches between *in situ* and satellite measurement locations in areas where width changes rapidly (see Fig. S1).

Two distinct advantages of RivWidthCloud over previous river width measuring algorithms [1], [2], [9]–[14] are its ease of use based on a popular cloud computing environment and its flagging capability to automatically reduce adverse impact from features like cloud and shadows. These advantages allow users to quickly extract typical width statistics and time series without the overhead of downloading, storing, and processing remote sensing data locally. It also allows for the generation of spatially continuous multitemporal width data that can be used to provide *a priori* data set of variations of river location and width for future satellite missions like SWOT, or to estimate river discharge either via AMHG alone or via in conjunction with other remote sensing data products.

RivWidthCloud is currently implemented with remote sensing products from Landsat program, given its long mission history, wide application, and sophisticated cloud algorithm. However, the range of remote sensing data for RivWidthCloud can be readily expanded to include other remote sensing data products (e.g., from Sentinel 2 or Moderate Resolution Imaging Spectroradiometer).

Acknowledgment

Source code for RivWidthCloud is hosted on GitHub at (https://github.com/seanyx/RivWidthCloudPaper) and can also be accessed directly in GEE at (https://code.earthengine.google.com/?accept_repo=users/eeProject/RivWidthCloudPaper).

References

[1] J. Schwenk, A. Khandelwal, M. Fratkin, V. Kumar, and E. Foufoula-Georgiou, "High spatiotemporal resolution of river planform dynamics from Landsat: The RivMAP toolbox and results from the Ucayali river," *Earth Space Sci.*, vol. 4, no. 2, pp. 46–75, 2017.

[2] F. Monegaglia, G. Zolezzi, I. Güneralp, A. J. Henshaw, and M. Tubino, "Automated extraction of meandering river morphodynamics from multitemporal remotely sensed data," *Environ. Model. Softw.*, vol. 105, pp. 171–186, Jul. 2018.

[3] J.-M. Vesakoski *et al.*, "Arctic Mackenzie Delta channel planform evolution during 1983–2013 utilising Landsat data and hydrological time series," *Hydrolog. Processes*, vol. 31, no. 22, pp. 3979–3995, 2017.

[4] L. B. Leopold and T. J. Maddock, "The hydraulic geometry of stream channels and some physiographic implications," USGS, Washington, DC, USA, Geol. Surv. Prof. Paper 252, 1953.

[5] T. M. Pavelsky *et al.*, "Assessing the potential global extent of SWOT river discharge observations," *J. Hydrol.*, vol. 519, pp. 1516–1525, Nov. 2014.

[6] T. M. Pavelsky, "Using width-based rating curves from spatially discontinuous satellite imagery to monitor river discharge," *Hydrol. Processes*, vol. 28, no. 6, pp. 3035–3040, Feb. 2014.

[7] C. J. Gleason and L. C. Smith, "Toward global mapping of river discharge using satellite images and at-many-stations hydraulic geometry," *Proc. Nat. Acad. Sci. USA*, vol. 111, no. 13, pp. 4788–4791, 2014.

[8] S. Biancamaria, D. P. Lettenmaier, and T. M. Pavelsky, "The SWOT mission and its capabilities for land hydrology," *Surv. Geophys.*, vol. 37, no. 2, pp. 307–337, 2016.

[9] A. Golly and J. M. Turowski, "Deriving principal channel metrics from bank and long-profile geometry with the R package CMGO," *Earth Surf. Dyn.*, vol. 5, no. 3, pp. 557–570, Sep. 2017.

[10] K. Yang, M. Li, Y. Liu, L. Cheng, Q. Huang, and Y. Chen, "River detection in remotely sensed imagery using Gabor filtering and path opening," *Remote Sens.*, vol. 7, no. 7, pp. 8779–8802, 2015.

[11] F. Isikdogan, A. Bovik, and P. Passalacqua, "RivaMap: An automated river analysis and mapping engine," *Remote Sens. Environ.*, vol. 202, pp. 88–97, Dec. 2017.

[12] J. C. Rowland *et al.*, "A morphology independent methodology for quantifying planview river change and characteristics from remotely sensed imagery," *Remote Sens. Environ.*, vol. 184, pp. 212–228, Oct. 2016.

[13] T. M. Pavelsky and L. C. Smith, "RivWidth: A software tool for the calculation of river widths from remotely sensed imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 5, no. 1, pp. 70–73, Jan. 2008.

[14] D. Yamazaki, F. O'Loughlin, M. A. Trigg, Z. F. Miller, T. M. Pavelsky, and P. D. Bates, "Development of the global width database for large rivers," *Water Resour. Res.*, vol. 50, no. 4, pp. 3467–3480, Apr. 2014.

[15] M. D. King, S. Platnick, W. P. Menzel, S. A. Ackerman, and P. A. Hubanks, "Spatial and temporal distribution of clouds observed by MODIS onboard the Terra and Aqua satellites," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 7, pp. 3826–3852, Jul. 2013.

[16] N. Gorelick, M. Hancher, M. Dixon, S. Ilyushchenko, D. Thau, and R. Moore, "Google earth engine: Planetary-scale geospatial analysis for everyone," *Remote Sens. Environ.*, vol. 202, pp. 18–27, Dec. 2017.

[17] Z. Zhu, S. Wang, and C. E. Woodcock, "Improvement and expansion of the Fmask algorithm: Cloud, cloud shadow, and snow detection for Landsats 4–7, 8, and sentinel 2 images," *Remote Sens. Environ.*, vol. 159, pp. 269–277, Mar. 2015.

[18] Z. Zou *et al.*, "Divergent trends of open-surface water body area in the contiguous United States from 1984 to 2016," *Proc. Nat. Acad. Sci. USA*, vol. 115, no. 15, pp. 3810–3815, Apr. 2018.

[19] H. Xu, "Modification of normalised difference water index (NDWI) to enhance open water features in remotely sensed imagery," *Int. J. Remote Sens.*, vol. 27, no. 14, pp. 3025–3033, 2006.

[20] J. R. G. Townshend and C. O. Justice, "Analysis of the dynamics of African vegetation using the normalized difference vegetation index," *Int. J. Remote Sens.*, vol. 7, no. 11, pp. 1435–1445, Nov. 1986.

[21] A. R. Huete, K. Didan, T. Miura, E. P. Rodriguez, X. Gao, and L. G. Ferreira, "Overview of the radiometric and biophysical performance of the MODIS vegetation indices," *Remote Sens. Environ.*, vol. 83, nos. 1–2, pp. 195–213, Nov. 2002.

[22] G. H. Allen and T. M. Pavelsky, "Global extent of rivers and streams," *Science*, vol. 361, no. 6402, pp. 585–588, 2018.

[23] J. W. Jones, "Improved automated detection of subpixel-scale inundation—Revised dynamic surface water extent (DSWE) partial surface water tests," *Remote Sens.*, vol. 11, no. 4, p. 374, Feb. 2019.

[24] G. Donchyts, J. Schellekens, H. Winsemius, E. Eisemann, and N. van de Giesen, "A 30 m resolution surface water mask including estimation of positional and thematic differences using Landsat 8, SRTM and OpenStreetMap: A case study in the Murray-Darling basin, Australia," *Remote Sens.*, vol. 8, no. 5, p. 386, May 2016.

[25] F. Isikdogan, A. C. Bovik, and P. Passalacqua, "Surface water mapping by deep learning," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 11, pp. 4909–4918, Nov. 2017.

[26] F. Isikdogan, A. Bovik, and P. Passalacqua, "Learning a river network extractor using an adaptive loss function," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 6, pp. 813–817, Jun. 2018.