Lifting Scheme for the Data on River Networks

Seoncheol Park and Hee-Seok Oh
Department of Statistics
Seoul National University
Seoul 08826, Korea

Draft: version of March 2, 2020

Abstract

This paper aims to suggest a new multiscale method for analyzing water pollutant data located in the river network. The main idea of this paper is adapting the conventional lifting scheme, which is one of the second-generation wavelets to the streamflow data while reflecting characteristics of the river network domain. It is challenging to apply the lifting scheme to streamflow data directly because of its complex data domain structure. To solve this, we propose a new lifting scheme algorithm for streamflow data that integrates flow-adaptive neighborhood selection, flow proportional weight generation, and flow-length adaptive removal point selection. Nondecimated version of the proposed lifting scheme is also provided. Based on the simulation study result, the proposed method successfully performs a multiscale analysis of streamflow data. We also provide a real data analysis of the Geum-River basin with a comparison of conventional smoothing methods.

Keywords: Lifting scheme; River network; Smoothing; Spatial adaptation; Spatial modeling; Streamflow data.

Status	Very good	Good	Slightly better	Normal	Poor	Bad	Very bad
TOC (mg/L)	≤ 2	≤ 3	≤ 4	≤ 5	≤ 6	≤ 8	> 8

Table 1: Environment standard provided by Water Environment Information System. Pollutants are total organic carbon (TOC).

1 Introduction

Environmental monitoring is the collection of observations and studies for the assessment of environmental data (Artiola *et al.*, 2004). Humans now know that the environment is crucially related to our health and survival. Therefore, one cannot emphasize too much environmental monitoring for humans. One of the main branches of environmental monitoring is water quality management. As human activities increase, more environmental costs are needed to rehabilitate water. Therefore, it is important to analyze the characteristics of water pollutants.

In this paper, we focus on an environmental pollutant called total organic carbon (TOC, mg/L). Recently, Korean Ministry of Environment announced that they changed the water pollution index for monitoring wastewater treatment performance of facilities from chemical oxygen demand (COD) to TOC. According to the ministry, cannot measure all organic matters in water. However, using TOC compensates for these shortcomings. Therefore, analyzing TOC data is meaningful for the society. The National Institute of Environmental Research (NIER), which is an institution of the Ministry of Environment, operates the Water Environment Information System for water quality monitoring. This system provides "Environment standard", which is a good guideline for the amount of water pollutant listed in Table 1.

On the other hand, one of the main characteristics of water pollutant data is that it is located on a river network. Therefore, the correlation between two points on the river network are closely related to the shape of the network. Since the stream network is different from the usual $\mathbb{R}^2$ domain, we need to consider a new method, which reflects the shape of the river network to analyze the river network data.

Given the scattered observations on the river network shown in Figure 1, our goal is to represent the underlying field of the water quality index on the river network domain. From

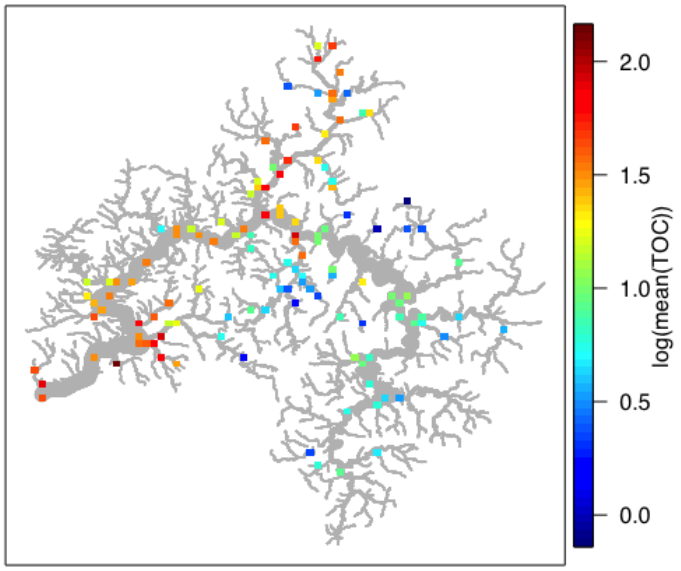

Figure 1: TOC data on Geum-River network. Gray lines mean streamflow segments with different weights represented by their widths, and colored points mean logarithm values of TOC means from December 2011 to November 2017 in 127 observation sites.

Figure 1, we observe some distinct characteristics of the water quality index: (i) The water quality index is located on the river network. It means that observed values are correlated across the river network, not the usual $\mathbb{R}^2$ domain. Most of the spatial statistical models have an interest in analyzing a spatial region that is a subset of $\mathbb{R}^2$ where Euclidean distance works well. On the other hand, for the streamflow data in Figure 1, Euclidean distance does not work well as a natural metric. (ii) As shown in Figure 1, the data have spatially inhomogeneous features with various dependent structures along with the river network. (iii) The observations are scattered; they are irregularly observed on the river network.

Thus, any such method of representing the above data should have the following features: (i) It is capable of effectively represent streamflow data on the river network. (ii) It provides a spatially adaptive framework that can estimate the inhomogeneous underlying function with reflecting inherent multiscale characteristics of data. (iii) It is applicable to scattered data. In this paper, we would like to propose a multiscale method that satisfies all the features mentioned above. Suppose that there are n observations in whole network. We denote x_i as the location of stations. Then assume that we observe a set of scattered data (x_i, y_i),

$i = 1, \ldots, n$ from the model,

$$y_i = g(x_i) + \varepsilon_i, \tag{1}$$

where x_i denote the locations of observations on the data domain, ε_i are the measurement errors, and g denotes an unknown underlying function of interest. Our goal is to estimate the underlying field $g(x)$ for every location x on the river network.

In literature, there exist some studies of stream data analysis. VerHoef *et al.* (2006) suggested the use of stream distance, which is defined as the shortest distance between two observation locations along with the stream network, as a good distance for data analysis on the river network. They showed that it is able to construct a large class of valid spatial autocovariance models using the stream distance. Also they further proposed a method to generate a class of covariance models for stream data by using kernel convolution.

O'Donnell *et al.* (2014) used nonparametric flexible regression models such as kernel methods and penalized splines to construct spatio-temporal models for river networks. They suggested a piecewise simple regression approach by diving the network into a large number of small pieces called "stream segments". They provided a regression-based stream data estimation approach by assuming that function values g are the same within the same stream segments.

On the other hand, because of the complexity of the data, it is not easy to fully understand the underlying structure of the data. Multiscale analysis is a possible solution to solve such problems, by considering multiple data resolutions. As existing multiscale methods, wavelets are the most popular choice. However, it does not properly work when the data is not observed on regular grids or the number of observations is not dyadic, i.e., $n = 2^J$, for some $J \in \mathbb{Z}$. To overcome these problems, Sweldens (1996) and Sweldens (1998) proposed a kind of second-generation wavelet called "lifting scheme". The lifting scheme has been extensively studied in signal processing and image analysis (Jansen and Oonincx , 2005).

However, all of previous works has a limitation that they cannot give a multiscale structure of the dataset. To the best of our knowledge, there is no direct literature addressing multiscale methods for streamflow data analysis. To achieve our goal, we propose a new lifting scheme method for streamflow data by coupling the conventional lifting scheme with novel modifications of neighborhood selection, prediction filter, and removal order that consider

the characteristics of streamflow data.

In this paper, we suggest a new multiscale method for river network data using the concept of lifting scheme. However, it is hard to directly apply the concept of lifting scheme into the river network because of the complexity of the network. Therefore, we suggest a streamflow lifting method with an appropriate modifications. The suggested method has two advantages. First, it gives a multiscale structure of streamflow dataset following the argument of lifting scheme. Second, under certain situations, the proposed method has an advantage compared to conventional smoothing methods for river netoworks from the perspective of signal denoising.

The rest of the paper is organized as follows. Section 2 reviews the conventional liftting schemes and smoothing method on the river network. In Section 3, the streamflow data used in this study are explained. Section 4 presents a new method, termed "streamflow lifting scheme". To evaluate the proposed method, simulation study and real data analysis are conducted in Sections 5 and 6. Finally, concluding remarks are provided in Section 7.

2 Backgrounds

2.1 Lifting scheme

In this section, we briefly summarize the concept of lifting scheme for self-contained material. Suppose that we observe a set of n irregular locations $\boldsymbol{x} = (x_1, \ldots, x_n)^T$, where the length of the data ($= n$) may not be dyadic. Assume that we have function values $y_1, \ldots, y_n$ at every location. We want to construct a multiresolution transform at the $j-1$th level, given the jth level data $\boldsymbol{x}_j$. The lifting scheme consists of following four steps:

1. **Split**: At each level $j-1$, divide observations of the data vector at j level $\boldsymbol{y}_j$ into two subsets, $\mathcal{P}_{j-1}$ and $\mathcal{U}_{j-1}$. We denote i for locations in set $\mathcal{P}_{j-1}$ and k for locations in set $\mathcal{U}_{j-1}$.

2. **Predict**: Predict every sample $y_{j,i} \in \mathcal{P}_{j-1}$ from $y_{k,j} \in \mathcal{U}_{j-1}$ with a prediction filter $\mathbf{p}_{j-1,i}$ and store the prediction error $d_{j-1,i}$ $d_{j-1,i} = y_{i,j} - \hat{y}_{j,i} = y_{j,i} - \sum_{k \in \mathcal{N}_{j-1,i} \cap \mathcal{U}_{j-1}} p_{j-1,i,k} y_{j,k}$, where $\hat{y}_{i,j}$ represents the value of predicted values constructed from $\mathcal{U}_{j-1}$ neighbors of node i.

3. **Update**: Compute an update version of data at $j-1$ level $y_{j-1,k}$ in $\mathcal{U}_{j-1}$ with an appropriate update filter $\mathbf{u}_{j-1,k}$, $y_{j-1,k} = y_{j,k} + \sum_{i \in \mathcal{N}_{j-1,k} \cap \mathcal{P}_{j-1}} u_{j-1,k,i} d_{j-1,i}$.

4. **Repeat**: Repeat the above steps until the desired resolution level.

By iterating these steps, we generate coarse signals of data from updated subsamples. On the other hand, the inverse version of the lifting scheme algorithm can be easily obtained by undoing forward lifting scheme operations at each level $j-1$: (i) Undo update: $y_{j,k} = y_{j-1,k} - \sum_{i \in \mathcal{N}_{j-1,k} \cap \mathcal{P}_{j-1}} u_{j-1,k,i} d_{j-1,i}$. (ii) Undo predict: $d_{j,i} = d_{j-1,i} + \sum_{k \in \mathcal{N}_{j-1,i} \cap \mathcal{U}_{j-1}} p_{j-1,i,k} y_{j,k}$. (iii) Undo split. (iv) Repeat: repeat the above steps at the next level.

Finally, we remark that there are some crucial ingredients for construction of the lifting scheme.

- **The number of removing points at ones** ($|\mathcal{P}|$): The user should select how many points remain at the next (coarser) level.

- **Prediction filter**: It is essential to choose a prediction filter in the procedure. In lifting scheme literature, Haar (local constant), local linear, local polynomial or inverse distance weight are frequently used.

- **Removal order of points**: When removing the points, it is crucial to decide the order to remove the points. It is related to a question of which points are important or not important to represent the underlying field.

- **Neighborhood selection**: It is crucial to select several neighbors to construct a prediction filter. Too many neighbors make it hard to catch the local behavior of the data, while too few neighbors yield a bias to predict each node.

2.1.1 Lifting one coefficient at a time (LOCAAT)

We now review the lifting one coefficient at a time (LOCAAT) algorithm of Jansen *et al.* (2009). The LOCAAT algorithm constructs a removal order of data points and sequentially decomposes data with the order. Suppose we have the values $y_1, \ldots, y_n$, sampled at n irregularly spaced points $x_1, \ldots, x_n$ on the real line. Lifting scheme approximates the function g

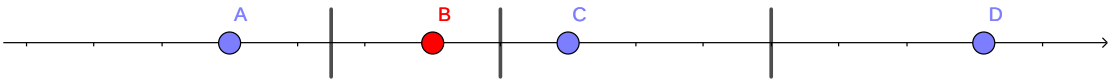

Figure 2: Illustration of removal point selection of lifting scheme in real line.

in (1) as

$$\tilde{g}(y) = \sum_{k=1}^{n} c_{n,k}\phi_{n,k}(x),$$

where $c_{n,i} := g(x_i)$ and $\phi_{n,k}(x_i) = \delta_{i,k}$ for $k, i \in \{1, \ldots, n\}$, where $\delta_{i,k}$ denotes the Kronecker delta.

In the initial stage, LOCAAT algorithm defines the index set of the scaling coefficients as $\mathcal{U}_n = \{1, \ldots, n\}$ and the index set of wavelet coefficients as $\mathcal{P}_n = \emptyset$. At the next step $n-1$, we choose a point to be lifted and denote its index by j_n, which is the point to be removed from the current set of scaling coefficients and to be converted into a detailed coefficient. The new set of indices corresponding to the scaling coefficients is $\mathcal{U}_{n-1} = \mathcal{U}_n \backslash \{j_n\}$, while $\mathcal{P}_{n-1} = \{j_n\}$ is the index set of the wavelet coefficient constructed at this stage.

To choose the point to be lifted, Jansen *et al.* (2009) used the minimum of the integral of scaling function $\phi_{n,k}$ concerning a suitable measure, denoted by I_{nk}. The point (x_{j_n}, c_{n,j_n}) is decided as the point with the smallest integral. To construct an update filter, Jansen *et al.* (2009) suggested a minimum norm solution based update weights at level r for reasons of numerical stability,

$$b_j^r = I_{ri_r} I_{r-1,j} / \sum_{k \in \mathcal{N}_r} I_{r-1,k}^2, \tag{2}$$

where i_r is an index of the remove candidate point. In this study, we consider the length or volume as a suitable measure. Figure 2 shows a toy example of choosing the point to be lifted by LOCAAT algorithm. A, B, C, D denote locations. One can define an area of each point by diving the real line into four blocks using mid-points, shown in vertical lines. LOCAAT algorithm selects a point that has the smallest area, which equals to the length of each block in one-dimensional data domain, among candidates. In this example, point B is chosen to be removed.

2.1.2 Other lifting schemes

Nunes *et al.* (2006) proposed a new lifting scheme method called "adaptive lifting". The

key ingredients of the adaptive lifting are the data-adaptive selection of the order of the regression and the neighborhood size in the lifting prediction step. Through these modifications, Nunes *et al.* (2006) provided flexibility to construct prediction filters under the one-dimensional signal denoising setting.

In the lifting scheme, finding an optimal removal order is not easy to decide because the optimal removal order does not exist from the perspective of minimizing mean-squared error. To enhance the performance of lifting scheme on a nonparametric regression setting, Knight *et al.* (2009) suggested a "nondecimated" concept into the lifting transform. It borrows the idea from a nondecimated wavelet transform, which improves the performance of the wavelet transform using over-determined basis functions. In Knight *et al.* (2009)'s approach, they produced many sequences of removal order called paths. We can generate Q different removal orders, called path, by permutation. Following notations of Knight *et al.* (2009), let $\hat{g}^{(q)}(x)$ is the estimate of the unknown function g at locations x, using q-th path. They showed that an averaged estimator

$$\hat{\bar{g}}(x_i) = \frac{1}{Q}\sum_{q=1}^{Q} \hat{g}^{(q)}(x_i), \qquad \forall i = 1, \ldots, n$$

could reduce the error between the actual signal and its estimator.

In a nondecimated lifting scheme, it is also important to select trajectories to get a better smoothing result. In Knight *et al.* (2009), they selected a few trajectories that have lower approximated average square errors $\widehat{\text{ASE}}$,

$$\widehat{\text{ASE}}(\hat{g}^{(q)}, g) = \frac{1}{n}\sum_{i=1}^{n} \{\hat{g}^{(q)}(x_i) - \hat{\bar{g}}(x_i)\}^2.$$

They generated other trajectories as variations of such "well-behaved" trajectories mentioned above. According to Knight *et al.* (2009), the use of genetic algorithms gives lower average mean square error.

2.2 Shrinkage in lifting scheme

Likewise wavelets, lifting scheme also applied to the nonparametric regression problem by incorporating a shrinkage approach. The main idea of wavelet shrinkage is based on the assumption that the information of the true signal is only contained in large values of

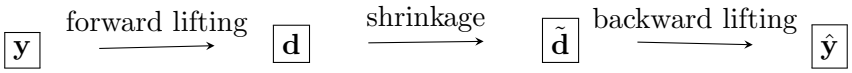

Figure 3: Illustration of lifting scheme with shrinkage.

the elements of $\mathbf{d}$. Therefore, by changing d coefficients, which are smaller than a certain threshold to zero, we can obtain a reconstruction result, which is more similar to the true signal.

In the proposed streamflow lifting scheme, we use the same shrinkage strategies used in Nunes *et al.* (2006) and Knight *et al.* (2009). There are several types of shrinkage approaches. In this paper, we focus on median thresholding and hard thresholding. They are implemented as `median` and `hard` in `adlift` and `nlt` package in R. To use the lifting scheme one must decide the number of scaling coefficients to be kept in the final representation of the initial signal. In addition, the user specifies `nkeep` in `adlift` and `nlt` package in R. In this paper, we always use the fully decomposed result (`nkeep=2`) in Knight *et al.* (2009), which produces $(n-2)$ detail coefficients from length-n dataset.

2.3 Smoothing method on the river network

In this subsection, we briefly summarize the work of O'Donnell *et al.* (2014). One of the main ideas of O'Donnell *et al.* (2014) is to simplify the information of the given network by using the concept of streamflow segments. They also suggest a penalize spline-based method with spatial, seasonal, temporal, and interaction basis. In this paper, we focus on the analysis of the spatial behavior of pollutants, considering the structure of river networks like (1). Therefore, we can consider a straightforward spatial additive model like

$$y_i = \mu + m_x(x_i) + \varepsilon_i = g(x_i) + \varepsilon_i, \tag{3}$$

where the function m_x describes spatial trends. The main idea of the spline method is to use a set of basis functions to estimate g in Equation (1). Suppose that we use p basis functions, then the estimator is $\hat{g}(x) = \sum_{j=1}^{p} \beta_j \phi_j(x)$. In O'Donnell *et al.* (2014), they use B-spline basis, which uses polynomial pieces. Under the above setting, a B-spline model can be formulated as $\mathbf{y} = B\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, where

$$B = \left(\mathbf{1}\ B_s\right),$$

where B_s is a design matrix of spatial components, and $n \times p$ vector $\boldsymbol{\beta}$ is a response vector. In this paper, we set $p = 1+$ the number of streamflow units. The P-spline model in O'Donnell *et al.* (2014) used a penalized version of B-spline model. It minimizes the following penalized sum of squares

$$(\mathbf{y} - B\boldsymbol{\beta})^T(\mathbf{y} - B\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}^T D^T D\boldsymbol{\beta}, \tag{4}$$

with respect to β. In short, (4) becomes

$$\|\mathbf{y} - B\boldsymbol{\beta}\|^2 + \boldsymbol{\beta}^T P\boldsymbol{\beta}.$$

In Equation (4), D means the penalty matrix makes the differences of β values within nearby stream units. According to O'Donnell *et al.* (2014), the solution of Equation (4) is $\hat{\boldsymbol{\beta}} = (B^T B + \lambda D^T D)^{-1} B^T \mathbf{y}$, where λ is a smoothing parameter. To select the optimal value of λ, O'Donnell *et al.* (2014) considers λ minimizing $\log(\hat{\sigma}^2) + 1 + \frac{2+2\text{dof}}{n-\text{dof}-2}$, where dof is degree of freedom.. For more information, O'Donnell *et al.* (2014) gives a detail of smoothing methods in the stream network.

3 Geum-River TOC data

The data used in this paper are observed in the Geum-River basin, which is located in the central part of South Korea. See Figure 5 (a). According to the Water Environment Information System, operated by the Ministry of Environment in Korea, the Geum-River basin is divided by 14 sub-regions called catchments. All 14 catchments are also divided into several sub-catchments, which are plotted as dotted lines in Figure 5 (a) and (b). Among them, Miho-Cheon catchment marked by orange in Figure 5 (b) is one of the sub-regions. It contains many observational stations compared to other catchments, and there are several cities and factories around it. We believe that taking a closer look at this area is meaningful. We also applied this river network to construct the network model of the simulation study in Section 5.

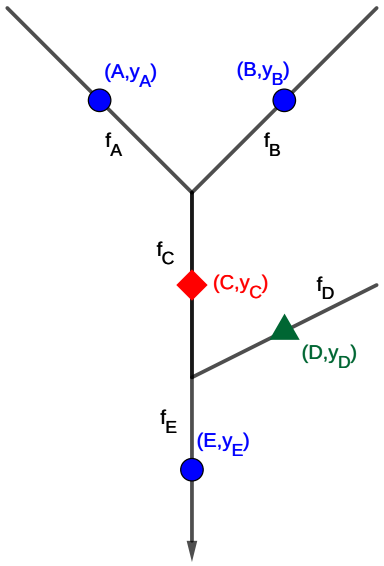

Figure 4: A simple streamflow data. Five solid black lines represent streme segments, indexed by A, B, C, D, and E. Each line segment has its flow volume, called $f_A, f_B, \ldots, f_E$. We denote $y_A, \ldots y_E$ to represent water quality values.

The orange lines in the Miho-Cheon catchment of Figure 5 (b) denote streamflow segments which are defined as lines between junctions in a stream network (VerHoef *et al.*, 2006, 2010). We note that the Miho-Cheon catchment has 113 streamflow segments and 28 observation stations. In total, the Geum-River network has 942 streamflow segments and 127 observation points.

Figure 5 (c) shows populations of cities, counties, and districts located in the Geum-River basin. Note that these administrative area does not fully match to the streamflow segments. From Figure 5 (c), we can check that most of the populations are concentrated on the Northern and Central parts of the Geum-River basin. Figure 5 (d) shows the locations of industrial areas in Geum-River basins. Note that general, national, and urban industrial sites are gathered in Miho-Cheon and its nearby area. Therefore, we can conjecture that many water pollutants will be produced in Miho-Cheon and its nearby river basin.

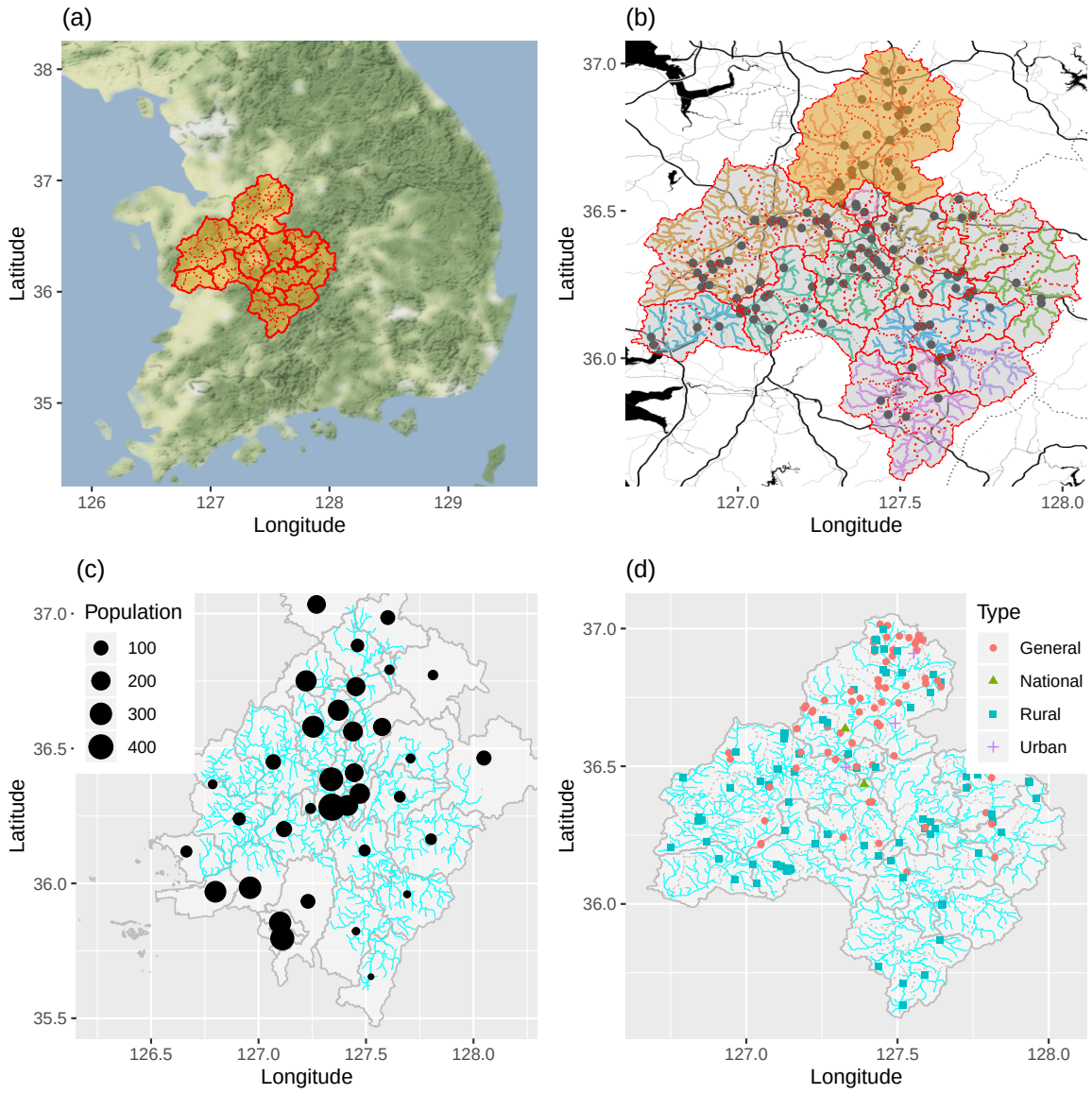

Figure 5: (a) Map of Geum-River basin in South Korea. Geum-River catchments are represented in red lines. (b) Enlarged figure of (a). Colored lines denote Geum-River stream networks and black dots are 127 observation points. (c) Populations (2017/12/31, thousands) in Geum-River basin. Gray lines mean cities. (d) Four types of industrial areas in Geum-river basins in 2018.

4 Streamflow lifting scheme

In this section, we present our procedure to construct a new lifting scheme for streamflow data by modification of LOCAAT algorithm of Jansen *et al.* (2009) that reflects some features of streamflow data. Our main idea is to develop a multiscale concept in the streamflow data by incorporating the idea of Nunes *et al.* (2006) into O'Donnell *et al.* (2014). The idea of the streamflow lifting scheme are (i) performing a network-adaptive neighborhood selection, (ii) constructing a prediction filter with flow-adaptive weighted averages, and (iii) determining a removal order by defining a proper contribution measure of each observation point for the stream network.

For a better presentation, we consider a toy example network plotted in Figure 4. Suppose that there are five observation points (A, B, C, D, E) located at different stream segments of a river network. Assume that each segment has a flow volume of f. Let $f_A, f_B, \ldots, f_E$ denote flow volume of station $A, B, \ldots, E$. We further denote $y_A, \ldots, y_E$ as water quality observation values at station $A, \ldots, E$.

4.1 Neighborhood selection

The concept of "flow-connected" introduced in VerHoef *et al.* (2006) is useful to construct a neighborhood set of a point in stream network. VerHoef *et al.* (2006) defined that two locations are connected when the intersection of upstreams of two stations is a non-empty set. In our example, segments A, C and B, C are "flow-connected" because the water in A and B can go to location C. On the other hand, C and D are not flow-connected since the water in C cannot go to station D, or vice versa.

We now use this concept of "flow-connected" to decide whether two segments are neighbors or not. In this study, when two points are flow-connected, we consider that they are neighbors of each other. In Figure 4, suppose that we are interested in removing point C at a specific resolution level. By following the concept of flow-connected, we define A, B, and E (blue circles) as its neighborhood, while D (green triangle) is excluded from the neighborhood of C.

One of the distinct characteristics of the proposed neighborhood selection is that it considers both upstream and downstream neighborhoods. By doing so, it can reduce the number

of boundary points. At first glance, including downstream points into neighborhood seems awkward. However, by combining an appropriate prediction filter construction explained in Section 4.2, it is able to generate reasonable prediction filters.

4.2 Construction of prediction filter

In this subsection, we consider a problem of prediction filter construction. The simplest prediction filter is constructed using an equally weighted valued vectors. However, every stream network has its mainstream and substreams. It is plausible that observation values on the mainstream usually has more powerful effect to the nearby observation values. Therefore, the impact of each stream onto the given segment should be different. To consider the influence, it is reasonable to consider the flow volumes. The main idea of the proposed prediction filter construction is to consider the amount of flow volumes compared to others, where O'Donnell *et al.* (2014) called it as "relative flow volumes".

Suppose that we have neighbors of a specific point in the stream network. The simplest way to assign weights is to give an equal weight for all neighbors, which may not be desirable. For example, in the case that f_A is much bigger than f_B, y_A has a more massive impact on y_C, compared to y_B. Also, if f_D is bigger than f_C, then y_E is much more different from y_C. Thus, we would like to construct flow-adaptive weights that reflect the above consideration.

In the previous toy example of Figure 4, suppose that we predict the response value of point C with its neighbors. Since $f_C = f_A + f_B$, flow-adaptive weights for point C can be defined as ratios of flows,

$$w_A = \frac{f_A}{f_C}, \quad w_B = \frac{f_B}{f_C}, \quad \text{and} \quad w_E = \frac{f_C}{f_E}. \tag{5}$$

Then we obtain a predicted value of y_C as $\hat{y}_C = \tilde{w}_A y_A + \tilde{w}_B y_B + \tilde{w}_E y_E$, where

$$\begin{aligned} \tilde{w}_A &= \frac{f_A/f_C}{f_A/f_C + f_B/f_C + f_C/f_E}, \\ \tilde{w}_B &= \frac{f_B/f_C}{f_A/f_C + f_B/f_C + f_C/f_E}, \text{ and} \\ \tilde{w}_E &= \frac{f_C/f_E}{f_A/f_C + f_B/f_C + f_C/f_E}. \end{aligned} \tag{6}$$

Therefore, the predicted value of the segment C, $\hat{y}_C$ is

$$\hat{y}_C = \tilde{w}_A y_A + \tilde{w}_B y_B + \tilde{w}_E y_E. \tag{7}$$

Note that $\tilde{w}_A, \tilde{w}_B$, and $\tilde{w}_E$ denote normalized flow-adaptive weights to make the sum of weights to be 1, i.e., $\tilde{w}_A + \tilde{w}_B + \tilde{w}_E = 1$. Hence, it is able to construct a lifting scheme for streamflow data by combining flow-adaptive weights of (5) and (6) into the conventional lifting scheme.

In practice, it is uncommon that one knows all f values throughout the entire streamlines. Hence, it is necessary to estimate flow values. For example, in VerHoef *et al.* (2006), they used an simple equal weight for each split.In this study, we assume that flow volume f of the most upstream segments is proportional to their Shreve stream order and segment length. Stream order is a positive whole number frequently used in hydrology to define stream-based distance in a stream network system. There are various kinds of stream order. Among them, the Shreve stream order is one of the most straightforward stream orders (Cressie *et al.*, 2006; VerHoef *et al.*, 2010). Cressie *et al.* (2006) defined the stream order as merely a count of the number of sources in the upstream portion of a stream network. Shreve stream order starts from setting all upper segments to 1. Magnitudes increase at all junctions in the stream network system. For example, if a stream has magnitude 1 and it combines with a new stream having magnitude 2, that has magnitude 3. By doing so, it is able to construct all magnitudes of the given network.

In this paper, we approximate f values by following the definition of Shreve stream order above and additionally letting the flow of the upper-most segments be proportional to their lengths to prevent multiple tie values in flow volumes. After defining flow volumes of upper-most segments, one can define a flow volume of the next upstream segments as a sum of their upstream segments. By repeating this approach, we obtain all f values in the stream network. In this paper, we assume that we know flow volume related weights, which generates by $\log(\sqrt{f})$ values. By following the O'Donnell *et al.* (2014), we normalize all $\log(\sqrt{f})$ values are lying between 0.2 to 1.5.

4.3 Removal point selection

For the streamflow lifting scheme, it is necessary to decide the removal order. In the case that the data lie in the real line such as Figure 2, it is easy to adapt a conventional approach such as Nunes *et al.* (2006). They used the length of a point in real line for the

computation of an integral. It can be extended in the two-dimensional data, suggested by Jansen *et al.* (2009). The basic idea of Jansen *et al.* (2009) to decide the removal point is setting an appropriate integral of a scaling function to find the densest observation in the Euclidean domain. Jansen *et al.* (2009) suggested Voronoi-polygon based area measurement as a candidate of a proper integral and selected a point that has the smallest integral as a removal point in LOCAAT algorithm.

However, these methods cannot be directly applicable to the streamflow data since the network is not easily projected onto one or two-dimensional data. Here, we suggest a simple approach to distinguish which points are located in the densest region of the river network by using the measure of the contribution of each segment in data. We define an integral as a contribution of each observation point to the network. To define a contribution of each point in streamflow data, we use flow-adaptive weights defined in (6). Consider the simple example described in Figure 4. Suppose that at the jth level, we want to remove point C with neighborhood points A, B, and E. Let I_A^j denote an integral of point A at the jth level, which is defined by the volume of the segment where A is located, say V_A. We define the volume V as a product of flow f and length of the segment ℓ,

$$\begin{aligned} I_A^j &= V_A = f_A \times \ell_A, \\ I_B^j &= V_B = f_B \times \ell_B, \text{ and} \\ I_E^j &= V_E = f_E \times \ell_E. \end{aligned} \tag{8}$$

At the next level, $j-1$ after point C is removed, we need to update the integral of neighborhood points. For this purpose, we use a weighted volume of the point C according to the weights of neighbors in (6). Thus, I_A^j, I_B^j, and I_E^j are updated to

$$\begin{aligned} I_A^{j-1} &= I_A^j + \tilde{w}_A \times V_C, \\ I_B^{j-1} &= I_B^j + \tilde{w}_B \times V_C, \text{ and} \\ I_E^{j-1} &= I_E^j + \tilde{w}_E \times V_C. \end{aligned} \tag{9}$$

Note that since $\tilde{w}_A \times V_C + \tilde{w}_B \times V_C + \tilde{w}_E \times V_C = I_C^j$, the sum of integrals is not changed. For the point to be removed at the $j-1$th level, we select a point that has the minimum value of I^{j-1}. For the update filter, we use the minimum norm solution based filter in (2).

4.4 Nondecimated lifting scheme for streamflow data

As mentioned in Section 2.1.2, sometimes we can reduce the mean squared error of the lifting scheme in a nonparametric regression setting by considering a nondecimated version of the lifting scheme. In this paper, we applied a similar approach to the streamflow data. In this paper, we assume that the current stream distance-based removal is one of the well-behaved trajectories from minimizing the root mean squared error perspective. Then we generate multiple trajectories produced from permutations. Permutations are only performed within the same sub-catchment regions, which are dotted lines in Figure 5. In this paper, we fixed the number of permutations `Q=10`. To generate such well-behaved trajectories, we define clusters first and do permutation to observations located in the same cluster.

In this paper, we consider two kinds of tuning parameters: (i) the number of trajectories (`Q=10`) and (ii) the number of permutation within single trajectory (`v=5`) Note that we can have the same denoising results when we use `Q=10` and `v=0`.

5 Simulation study

In this section, we conduct numerical experiments for the evaluation of our approach for spatial streamflow data analysis. Suppose that the data are observed from the regression model in Equation 1. We mainly focus on the situation that the underlying mean-field of the data is piecewise constant. Therefore, there are several discontinuous function values in a stream network, which sometimes fails to have an appropriate estimation using conventional smoothing-based methods. For comparison, we consider the flexible smoothing approach of O'Donnell *et al.* (2014). For the proposed approach, we consider three kinds of different methods: ordinary streamflow lifting scheme with median thresholding (`S-Lifting (M)`), ordinary streamflow lifting scheme with hard thresholding (`S-Lifting (H)`), and nondecimated streamflow lifting scheme with median thresholding (`S-Lifting (N)`).

For simulation setup, we consider two types of stream networks: one is the Miho-Cheon streamflow segments (Figure 6 (a) and (b)), and the other is a simulated river network used in Gallacher *et al.* (2017) (Figure 6 (c) and (d)). Each network consists of 113 and 80 stream segments. For each river network, we classify whole stream segments into two groups,

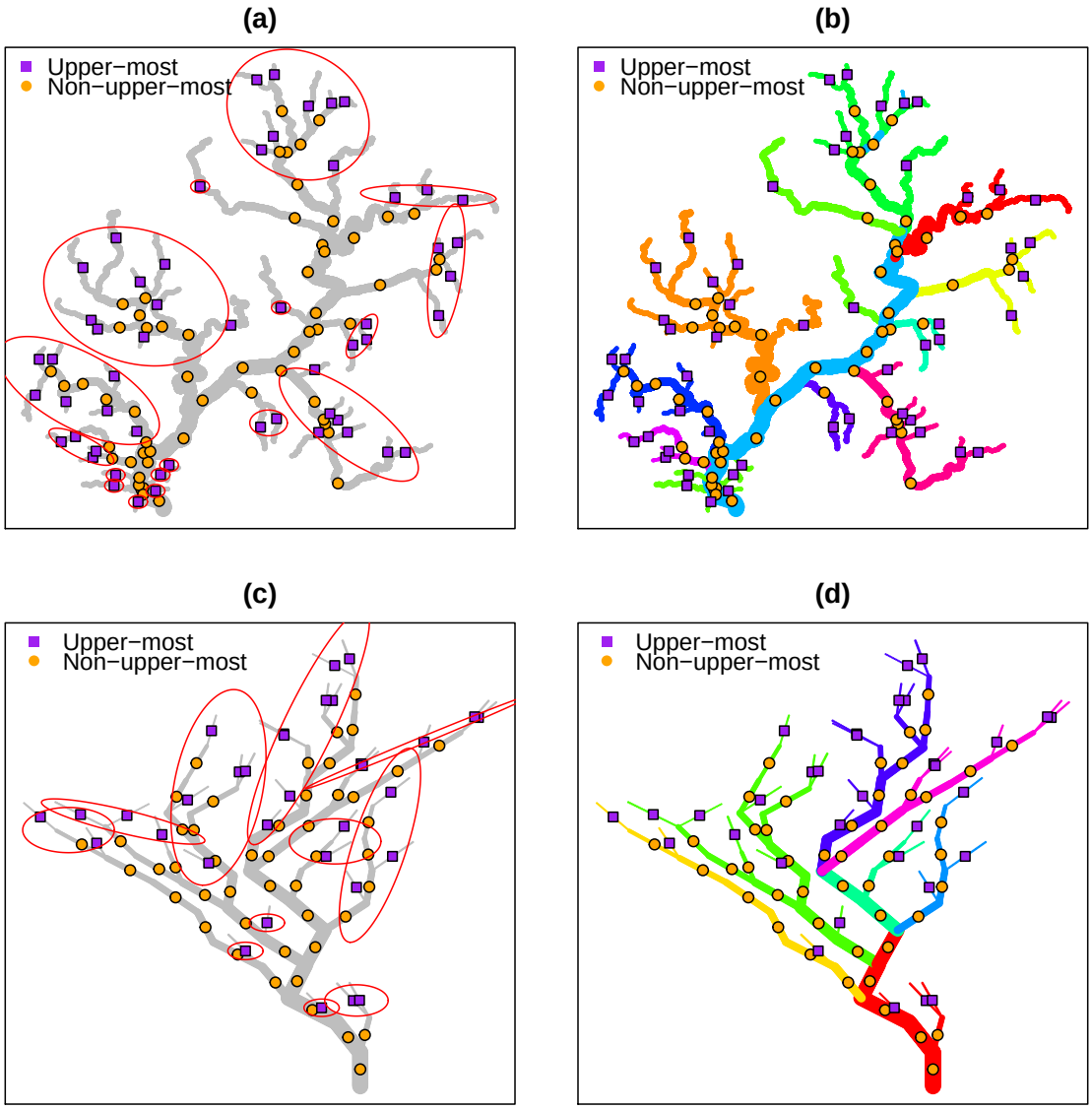

Figure 6: (a) Cluster construction (red circles) on the Miho-Cheon stream network. We classify streamflow segments into two, upper-most segments (purple squares) and non-upper-most segments (orange circles). (b) Colors represent substreams for the sampling procedure. The sampling probability is proportional to the number of streams of each substream.

upper-most segments and non-upper most segments, shown in Figure 6 (a). We assume that there are no intrinsic sources to change the simulated signal values. Then the signal values of non-upper-most segments are generated from a weighted average of nearby upstream signal values of upper-most segments. It implies that it is enough for the simulation only to generate signal values of upper-most segments.

Moreover, we divide upper-most segments into a few clusters, shown in Figure 6, to generate inhomogeneous stream network data. We assume that within a cluster, all segments have the same signal value. For each simulated data set, $g(x_i)$ values of upper-most segments are generated as follows: (i) all $g(x_i)$ values of upper-most segments are set to be 9, (ii) a cluster is randomly selected from the clusters in Figure 6, and (iii) $g(x_i)$ values in the selected cluster are replaced with a values that is randomly chosen from 12, 15, 18. This procedure is repeated until at least 30 upper-most segments have a value bigger than 9. An example of the simulated data generation is plotted in Figure 7 (a).

We further consider three different spatial designs for the simulated data in the stream network. (i) For a sparse design, 40 observation stations located on 40 different segments among the total 113 Miho-Cheon stream segments. A realization is shown in Figure 6 (b). (ii) Eighty stations, which is almost two-thirds of the number of Miho-Cheon streams. (iii) As for a dense case, 113 stations are considered. Along the same lines, we analyze the simulated network of Gallacher *et al.* (2017) in two situations, (i) suppose that we only have observations in 40 stations, and (ii) we have one observation at each segment.

The noise terms are generated from $\varepsilon_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2)$ with (i) $\sigma = 0.5$, (ii) $\sigma = 1$, and (iii) $\sigma = 1.5$. As for evaluation measure, we consider the root mean square error (RMSE) as

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N_{tot}} (g(x_i) - \hat{g}(x_i))^2}{N_{tot}}}, \tag{10}$$

where $\hat{g}(x_i)$ is an estimated smoothed value at segment i, and N_{tot} means the total number of stream segments in the river network. For each combination of three spatial designs and two σ's, we compute RMSE values according to our method and the approach of O'Donnell *et al.* (2014) over 100 simulated data sets. Table 2 lists the RMSE values. As listed, the proposed method outperforms the approach of O'Donnell *et al.* (2014) for all the combinations. For visual inspection, we look at fitting results of the spatial design with 60 stations and $\sigma = 1$

RMSE	Obs=40			Obs=80			Obs=113		
(Std. error)	$\sigma = 1$	$\sigma = 1.5$	$\sigma = 2$	$\sigma = 1$	$\sigma = 1.5$	$\sigma = 2$	$\sigma = 1$	$\sigma = 1.5$	$\sigma = 2$
O'Donnell	2.0988	2.2101	**2.2882**	1.5270	1.6336	1.7542	1.3252	1.4458	1.5798
	(0.1957)	(0.2052)	**(0.2317)**	(0.1561)	(0.1513)	(0.1654)	(0.1364)	(0.1320)	(0.1609)
Proposed	1.7089	2.1410	2.3588	1.0662	1.3488	1.6377	0.8362	1.1656	1.4141
(Median)	(0.4181)	(0.5202)	(0.5134)	(0.1877)	(0.2343)	(0.2686)	(0.1187)	(0.1929)	(0.2119)
Proposed	**1.6287**	2.1281	2.3647	**1.0382**	1.3440	1.6400	0.8386	1.2142	1.5172
(Hard)	**(0.3882)**	(0.4913)	(0.4704)	**(0.1528)**	(0.1903)	(0.2524)	(0.1074)	(0.1569)	(0.1985)
Proposed	1.7210	**2.1143**	2.3292	1.0428	**1.3212**	**1.6076**	**0.8000**	**1.1225**	**1.3705**
(Median, nlt)	(0.3907)	**(0.4517)**	(0.4460)	(0.1849)	**(0.2241)**	**(0.2588)**	**(0.1158)**	**(0.1760)**	**(0.2081)**

Table 2: RMSE of simulation result (and their standard error). In each simulation, the number of iterations is 100.

shown in Figure 7. The proposed method performs well by reflecting the inhomogeneous features of the underlying field.

Table 2 and 3 are the RMSE summary of the simulation study. We find that the proposed methods work well under the given simulation settings, especially when σ is small. O'Donnell *et al.* (2014)'s method generally gives more robust RMSE compared to proposed methods. When the number of observations is small and the proposed methods usually give a lower RMSE. The proposed methods work well under situations that nearby segments have similar values while there are big differences among different streams. On the other hand, the performance of the proposed methods is also dependent on the number of sampling observations. Therefore, when the number of observations is small compared to the number of streamflow segments, it is better to consider both the proposed methods and O'Donnell *et al.* (2014)'s approach. For the nondecimated version of the proposed lifting scheme, we find that it works well under the simulation setting, which assume that we have knowledge about the behavior of the data.

6 Real data analysis

In this section, we apply the proposed lifting scheme to our real dataset in Section 3. We consider TOC water pollutant, observed from 2012 to 2017. Water pollutants typically have some extreme values, which yield skewed empirical distributions. Therefore, we consider log-

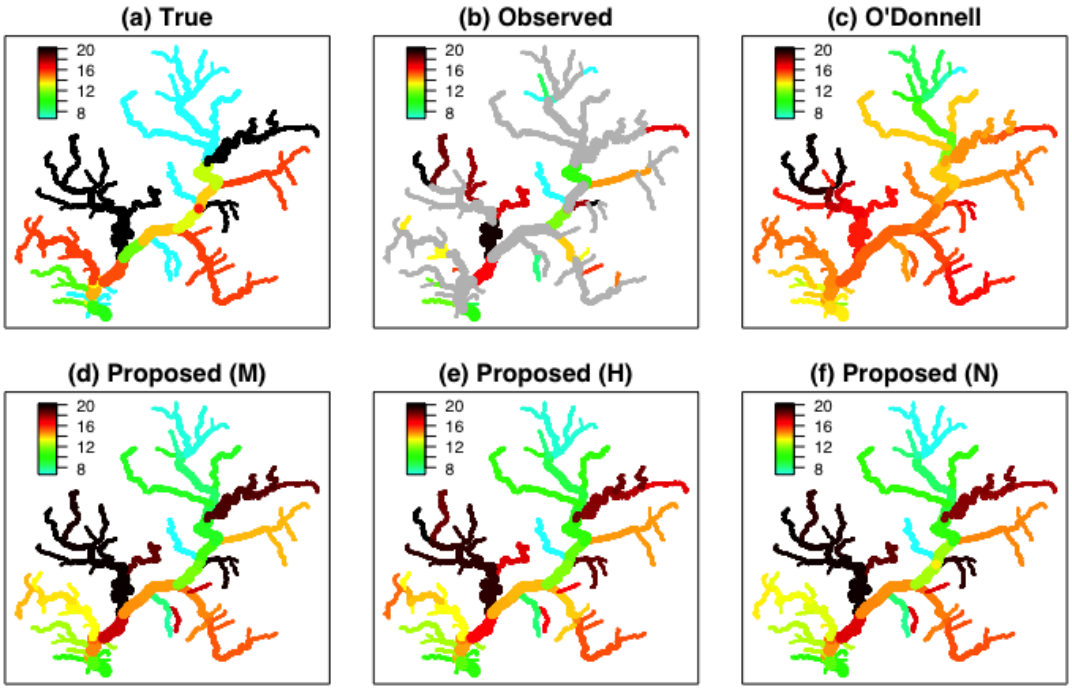

Figure 7: (a) True signals, (b) noisy observed values when Obs=40, (c) estimation results of O'Donnell *et al.* (2014), (d) the proposed method with median thresholding, (e) with hard thresholding and (f) the proposed nondecimated method with median thresholding.

RMSE	Obs=40			Obs=80		
(Std. error)	$\sigma = 1$	$\sigma = 1.5$	$\sigma = 2$	$\sigma = 1$	$\sigma = 1.5$	$\sigma = 2$
O'Donnell	1.7266	1.8695	1.9698	1.2698	1.3815	1.5421
	(0.2638)	(0.2835)	(0.2839)	(0.1251)	(0.1727)	(0.2017)
Proposed	1.1716	1.5464	1.8121	0.7265	1.0249	1.2818
(Median)	(0.2530)	(0.3611)	(0.3539)	(0.1212)	(0.1510)	(0.2599)
Proposed	**1.1417**	1.5769	1.9040	0.7396	1.0666	1.3705
(Hard)	**(0.2244)**	(0.3050)	(0.3286)	(0.1317)	(0.1678)	(0.2495)
Proposed	1.1443	**1.4988**	**1.7329**	**0.7162**	**1.0106**	**1.2816**
(Median, nlt)	(0.2494)	**(0.3267)**	**(0.3385)**	**(0.1219)**	**(0.1678)**	**(0.2712)**

Table 3: RMSE of simulation result of riverflow network data used in Gallacher *et al.* (2017) (and their standard error). Note that in each simulation, the number of iterations is 100.

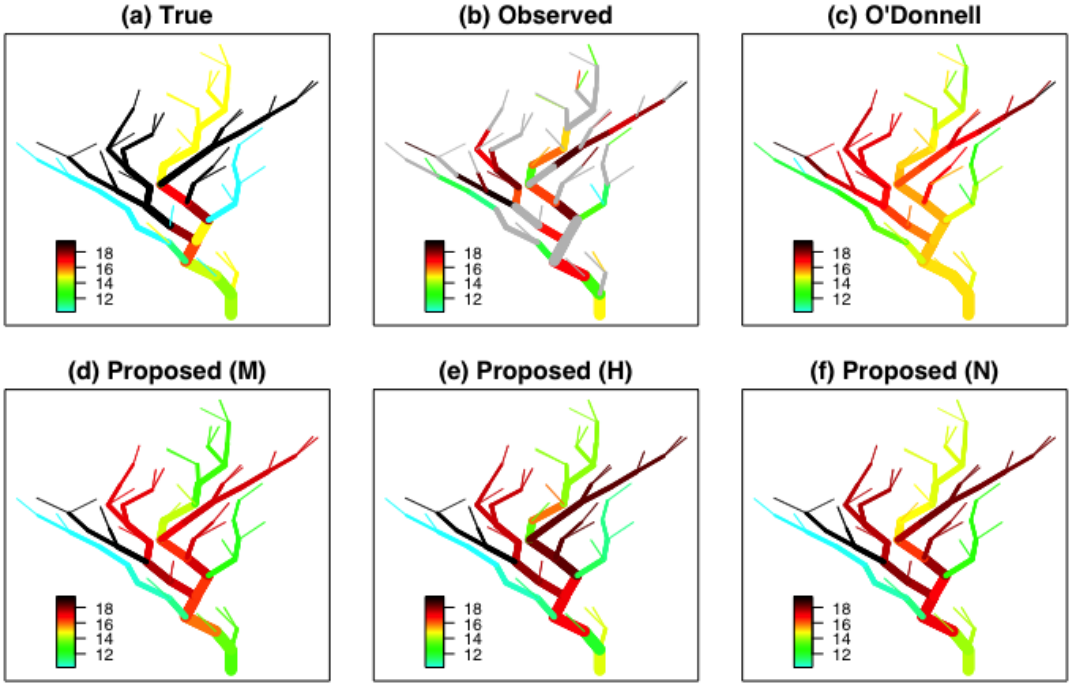

Figure 8: (a) True signals, (b) noisy observed values when Obs=60, (c) estimation results of O'Donnell *et al.* (2014), (d) the proposed method with median thresholding, (e) with hard thresholding and (f) the proposed nondecimated method with median thresholding.

	ODonnell	S-Lifting (M)	S-Lifting (N)
$\widetilde{\text{RMSE}}$	0.1240	0.0818	0.1856

Table 4: $\widetilde{\text{RMSE}}$ results of interpolation of Geum-River dataset.

transformed mean values of TOC data at each station from 2012 to 2017, shown in Figure 1.

To construct result figures, we consider a simple interpolation method. Equation (6) provides an appropriate tool for interpolation. For example, consider the simple river network in Figure 4. Suppose that there are no observations in segment C. That is, we assume that the actual value of y_c is unknown. Instead, we estimate the value of y_c using observations y_A, y_B, and y_E. The interpolated value is $\hat{y}_C = w_A y_A + w_B y_B + w_E y_E$, where $w_A + w_B + w_E = 1$. In this section, we plot different interpolation results. First of all, we consider the interpolation result with the raw dataset (`Raw`). For comparison, we compute the interpolation of the smoothing approach of O'Donnell *et al.* (2014).

For the nondecimated version of the streamflow lifting scheme, we have to set clusters to get a stable and enhanced smoothing result. In this section, we only consider a permutation of observations in the same stream segments, under the assumption that the original removal path defined through Section 4 is such a well-behaved removal order. Among 127 observation stations, 12 stations are located in the segments which have two or more observations.

Suppose that the underlying model is following Equation (1). Although we do not know the actual function g, we can compare the similarity of interpolation with the approximated root mean square error ($\widetilde{\text{RMSE}}$) as

$$\widetilde{\text{RMSE}} = \sqrt{\frac{\sum_{i=1}^{N_{tot}} (\tilde{g}(x_i) - \hat{g}(x_i))^2}{N_{tot}}},$$

where N_{tot} is the number of total stream segments in the network. In this case, $N_{tot} = 942$. $\widetilde{\text{RMSE}}$ measures the similarity between raw data interpolation result ($\tilde{g}(x_i)$) and estimated data interpolation results ($\hat{g}(x_i)$).

Figure 9 shows the streamflow lifting scheme results for the Miho-Cheon TOC dataset. The interpolation result using the raw dataset is shown in (a). For comparison, we also consider the approach of O'Donnell *et al.* (2014), whose results are in (b) of Figure 9. The

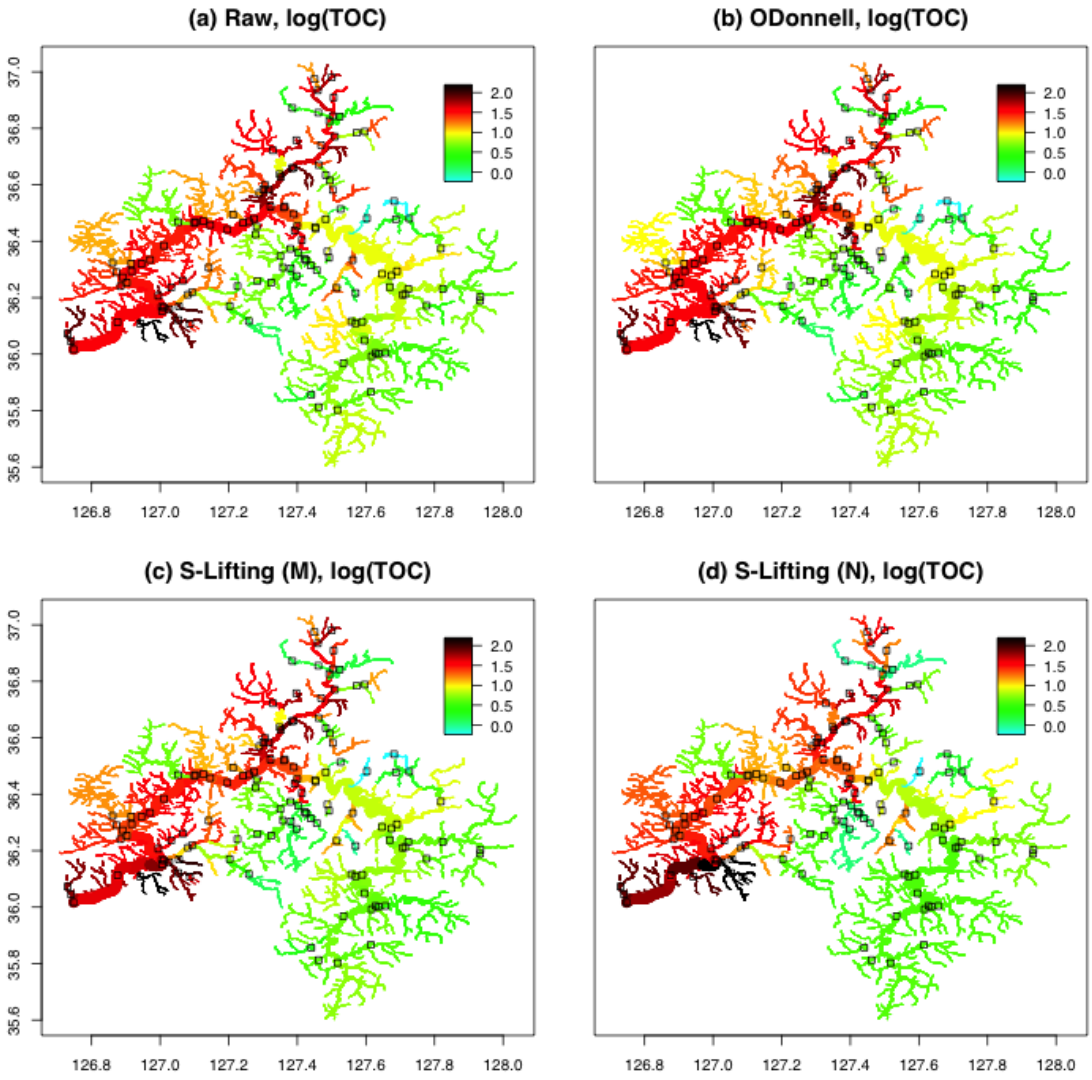

Figure 9: Real data analysis result of TOC data. Colored lines represent log(TOC) value prediction result using (a) row dataset, (b) O'Donnell's method, (c) proposed streamflow lifting scheme with median thresholding, and (d) proposed nondecimated streamflow lifting scheme with median thresholding. Colored bar represents interpolation values.

interpolation results of the proposed methods are in (c) and (d) of Figure 9. Since non-decimated lifting scheme uses a bunch of random trajectories, the result could be different at each execution. According to Figure 9, we can easily find that the high TOC values of Geum-River downstream are affected by the water qualities of Miho-Cheon. In conclusion, pollutants from Miho-Cheon dominate the water pollutant pollution. Concerning Figure 5, we can find that many industrial factories are located near the Miho-Cheon catchment area. It is a plausible conclusion that industrial factories may affect the amount of TOC in the river network.

From Figure 9, all methods presents more smoothed results compared to stream network representation using the raw dataset, plotted in (a). Among them, the proposed streamflow lifting scheme method (c) and (d) make more smoothed estimation results. For example, if we compare the (b) and (c) of Figure 9, we can find some orange colors on the East side of the network when we use O'Donnell's approach. However, these colors are changed to green colors in the streamflow lifting scheme. On the other hand, the proposed lifting scheme decides some green values of the left side on the network are noises. Therefore, the method also changes it into orange or red.

For the nondecimated version, we can find that several values on the East side of the network become higher. Compared to the simulation dataset, the real data has such a mixed pattern, not like piecewise. Therefore, we conjecture that it is crucial to construct appropriate clusters when it comes to the nondecimated streamflow lifting scheme. When we would like to apply a nondecimated version of the proposed approach to the real-world dataset, we should be careful to make a cluster. On the other hand, the original streamflow lifting scheme and O'Donnell *et al.* (2014)'s approach can give more robust results.

One of the advantages of the proposed streamflow lifting scheme method is that it gives a multiscale representation of the given streamflow data. Let the estimated function is $\hat{g}$, then we can represent the true function into the sum of two components,

$$g(x_i) = \hat{g}(x_i) + e(x_i).$$

Furthermore, we generate plots of $e(x_i)$ by applying the same interpolation strategy used in Figure 9 (c). Figure 10 shows the streamflow lifting scheme with multiple scales, using the original version of our proposed streamflow lifting scheme. Note that there is no thresholding

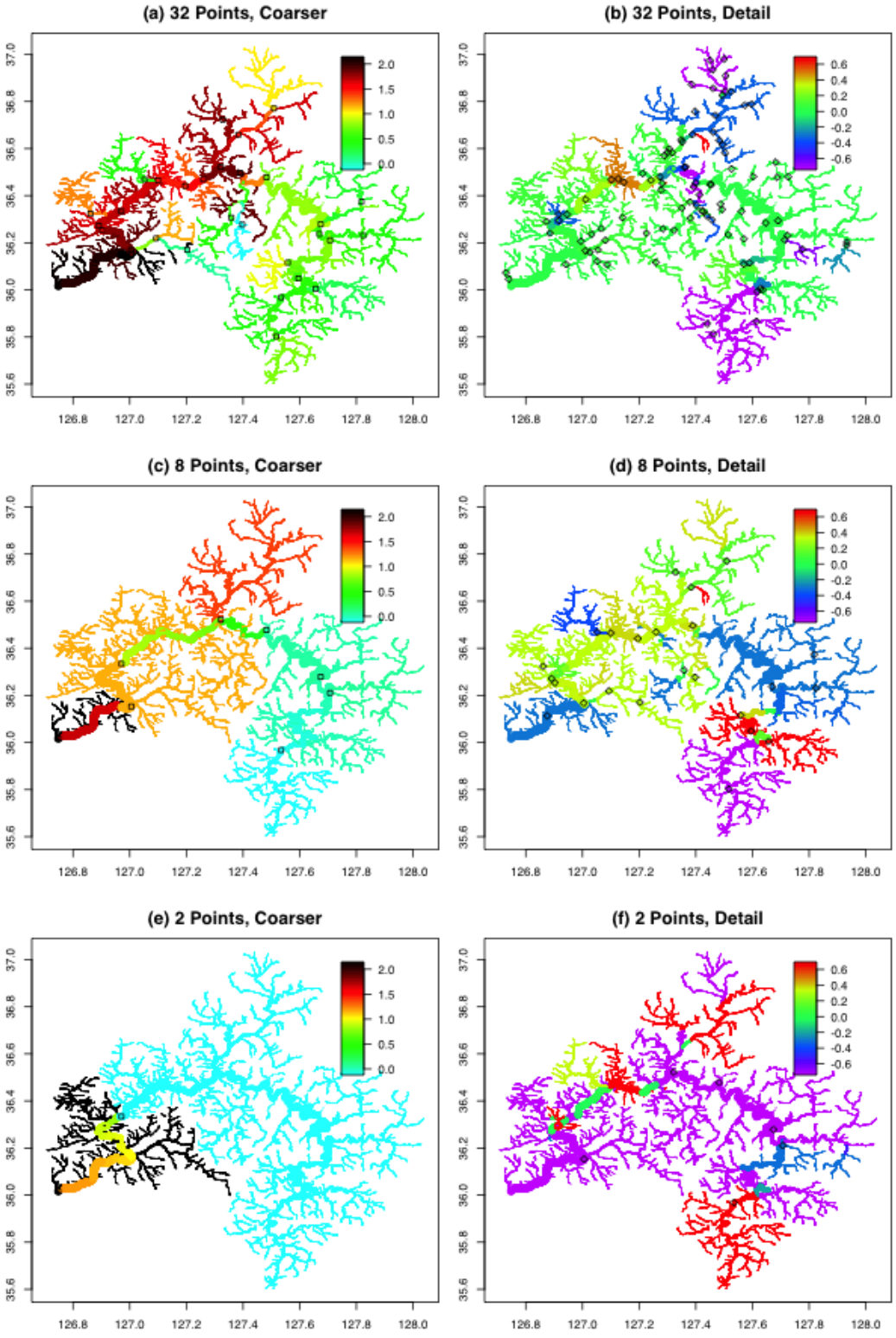

Figure 10: Result TOC details. (a), (c), and (e) show coarser-level of streamflow field representations and (b), (d), and (f) are corresponding details. The number of points (`nkeep`) are `32, 8, 2,` respectively.

procedure in Figure 10. From Figure 10 (a), if we reconstruct the river network field using 32 points, we can have relatively similar results, compared to Figure 9. When we reduce the number of data points for the reconstruction, the coarser data fields are different from their original interpolation result. Instead, detail fields have more information about the network.

7 Summary and further works

In this article, we suggest a new lifting scheme for streamflow data. The proposed methods enable lifting scheme to streamflow data setting by (i) adopting a stream network adaptive neighborhood selection, (ii) constructing a prediction filter with flow-adaptive weighted averages, and (iii) setting a removal order by defining neighborhood flows of each observation point. By using the proposed neighborhood selection method, we can reduce the number of boundary points and predict the values of upstream streamflow points. Besides, we also suggest a nondecimated version of the proposed streamflow lifting scheme. A simulation study shows that the proposed method works well compared to the smoothing approach for streamflow data under certain situations, especially when data has some discontinuities.

However, the proposed approach has some limitations. First, it assumes that the volume of the water flow is proportional to the length of the segments and Shreve order. However, in reality, the volume of the water flow is different from the length of the segments and Shreve order. The volume of the water is also different from the season. For example, in Korea, most of the precipitation is focused on the Summer season. Second, in the simulation study, we gathered segments in the given river network into several artificial groups to enhance the performance of the proposed lifting scheme. However, it is not easy to define optimal groups in real data analysis. Therefore, one of the future research would be suggesting an appropriate measure to find optimal groups. Third, the removal order of the proposed method is deterministically decided, not dependent on the value of the streamflow dataset. If possible, finding a data-adaptive removal order selection algorithm is also useful for data analysis. In this paper, by following the argument of Knight *et al.* (2009), we make a nondecimated version of the proposed streamflow lifting scheme, by assuming that there is no optimal removal order selection under LOCAAT algorithm setting.

On the other hand, the proposed approach does not provide spatio-temporal data anal-

ysis. Since the TOC data are irregularly observed in both space and time domain, we need additional methods to do spatio-temporal streamflow analysis. Lindström *et al.* (2014) and O'Donnell *et al.* (2014) solve the problem by computing biweekly or a monthly average of data at each station. Then by O'Donnell *et al.* (2014)'s method, one can build a space-time basis with a tensor product. However, if we find a way to construct a multiscale spatio-temporal basis without data merging, it will be more useful to capture the multiscale spatio-temporal behavior of the data. This is left for future research.

Acknowledgement

This work was supported in part by ∼.

References

Artiola, J, Pepper, I. L. and Brusseau, M. L. (2004). *Environmental Monitoring and Characterization*. Elsevier Science and Technology Books.

Cressie, N., Frey, J., Harch, B. and Smith, M. (2006). Spatial Prediction on a River Network. *J. Agric. Biol. Environ. Stat.*, **11**, 127–150.

Gallacher, K., Miller, C., Scott, E. M., Willows, R., Pope, L. and Douglass, J. (2017). Flow-directed PCA for monitoring networks. *Environmetrics*, **28**, e2434.

Jansen, M. H. and Oonincx, P. (2005). *Second Generation Wavelets and Applications*. Springer Science and Business Media.

Jansen, M., Nason, G. P. and Silverman, B. W. (2009). Multiscale methods for data on graphs and irregular multidimensional situations. *J. Roy. Stat. Soc. B.*, **71**, 97–125.

Knight, M. I. and Nason, G. P. (2009). A 'nondecimated' lifting transform. *Stat Comput.*, **19**, 1–16.

Lindström, J., Szpiro, A. A., Sampson, P. D., Oron, A. P., Richards, M., Larson, T. V. and Sheppard, L. (2014). A Flexible Spatio-Temporal Model for Air Pollution with Spatial and Spatio-Temporal Covariates. *Environ. Ecol. Stat.*, **21**, 411–433.

Nason, G. P. (1993). *Wavelet Methods in Statistics with R*. Springer Science and Business Media, New York.

Nunes, M. A., Knight, M. I. and Nason, G. P. (2006). Adaptive lifting for nonparametric regression. *Stat. Comput.*, **16**, 143–159.

O'Donnell, D., Rushworth, A., Bowman, A. W. and Scott, E. M. (2014). Flexible regression models over river networks. *J. Roy. Stat. Soc. C.*, **63**, 47–63.

Sweldens, W. (1996). The lifting scheme: A custom-design construction of biorthogonal wavelets. *Appl. Comput. Harmon. Anal.*, **3**, 186–200.

Sweldens, W. (1998). The lifting scheme: A construction of second generation wavelets. *SIAM J. Math. Anal.*, **29**, 511–546.

Ver Hoef, J. M. and Peterson, E. E. and Theobald, D. (2006). Spatial statistical models that use flow and stream distance. *Environ. Ecol. Stat.*, **13**, 449–464.

Ver Hoef, J. M. and Peterson, E. E. (2010). A moving average approach for spatial statistical models of stream networks. *J. Am. Stat. Assoc.*, **105**, 6–18.