Rainfall Grid Interpolation from Rain Gauge and Satellite Data

Petchakrit Pinyopawasutthi
Pongsorn Keadtipod
Tanut Aranchayanont
Jitkomut Songsiri
Piyatida Hoisungwan

Department of Electrical Engineering
Chulalongkorn University

December 9, 2015

Abstract

Rain fall is conventionally collected by rain gauge on stations which is accurate but sparse. In order to improve an interpolation of rainfall between stations,reainfall data predicted from satellite is introduced. Both data set are merged by a linear estimator to interpolate a rainfall map.

1 Introduction

1.1 Overview and assumptions

By geodesic latitude and longitude coordinate, the rainfall map is equally divided into a flat rectangular, longitudinally n fragments, latitudinally m fragments, called *grid*. Provided indices (i, j) originated at top-left of the map, mathematically the grid (i, j) belongs to a Cartesian product of all possible grid:

$$I = \Big\{(i, j) \mid (i, j) \in \{1, 2, 3 \dots m\} \times \{1, 2, 3 \dots n\}\Big\}.$$

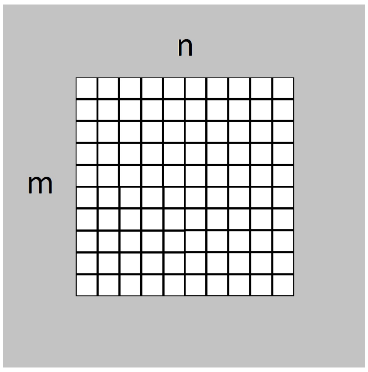

Figure 1: Example of $m \times n$ grid

The following would propose the way to predict $\hat{y}_{ij}$, a predicted rainfall corresponding to each (i, j) grid. There are two sources of data used to predict $\hat{y}_{ij}$. First from rain gauge x_{ij} in millimeter collected from stations every time interval of δt. A number of rain gauge station N is less than mn. Second source of data z_{ij}, predicted from the geodesic satellite, is available for all grid. Such data is predicted from humidity and atmospheric data.

Two assumptions are proposed:

1. Predicted data $\hat{y}_{ij}$ depends only on data from surrounding grid including itself. Mathematically, surrounding grid of the (i,j) grid is indexed by:

$$
\begin{aligned}
I_{ij} &= \Big\{(i-1), i, (i+1)\Big\} \times \Big\{(j-1), j, (j+1)\Big\} \\
&= \Big\{(i-1,j-1), (i-1,j), (i-1,j+1), (i,j-1), (i,j), (i,j+1), \\
&\qquad (i+1,j-1), (i+1,j), (i+1,j+1)\Big\}
\end{aligned}
$$

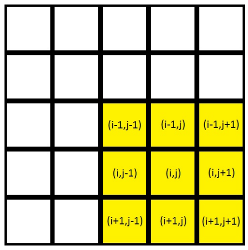

Figure 2: Surrounding grids are displayed in yellow colors where the center is the (i,j) grid, the grid of interest.

2. $\hat{y}_{ij}$ is a linear estimator. From 1, $\hat{y}_{ij}$ is a linear combination of x_{ij} and z_{ij} from surrounding grid. Let I_g be the index set of grids having ground stations. In other words, $\hat{y}_{ij}$ can be expressed as:

$$
\hat{y}_{ij} = \sum_{(k,l)\in I_g \cap I_{ij}} a_{ij,n} x_{kl} + b \sum_{(k,l)\in I_{ij}} z_{kl}, \tag{1}
$$

   where $a_{ij,n}$ and b are scalar parameters and n runs from 1 to M, where M is the cardinality of $I_g \cap I_{ij}$. In other words, $a_{ij,n}$ does not only depend on the grid location but also depends on $\hat{y}$ location intended to predict.

To illustrate the expression of $\hat{y}_{ij}$, consider Figure 8, where the yellow grids contain rain guage stations. As an example, suppose $(2,2)$ is the grid of our interest. Therefore,

$$
\hat{y}_{22} = a_{22,1}x_{13} + a_{22,2}x_{32} + a_{22,3}x_{33} + b(z_{11} + z_{12} + z_{13} + z_{21} + z_{22} + z_{23} + z_{31} + z_{32} + z_{33})
$$

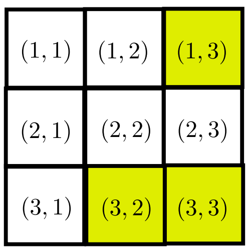

Figure 3: Example of grid system used to describe the interpolation of rainfall, $\hat{y}$.

1.2 Example of rainfall data along the year

As shown in Figure 4, the averaged values of satellite data in rainy season is higher than others. Figure 5 shows the difference of averaged value of rainfalls from ground station and from satellite measured at the location where ground stations are available. The green color represents the magnitude of 0, where we do not make a comparison since no stations available on those locations. We tend to see that in the rainy season, the satellite measurement is significantly overestimated.

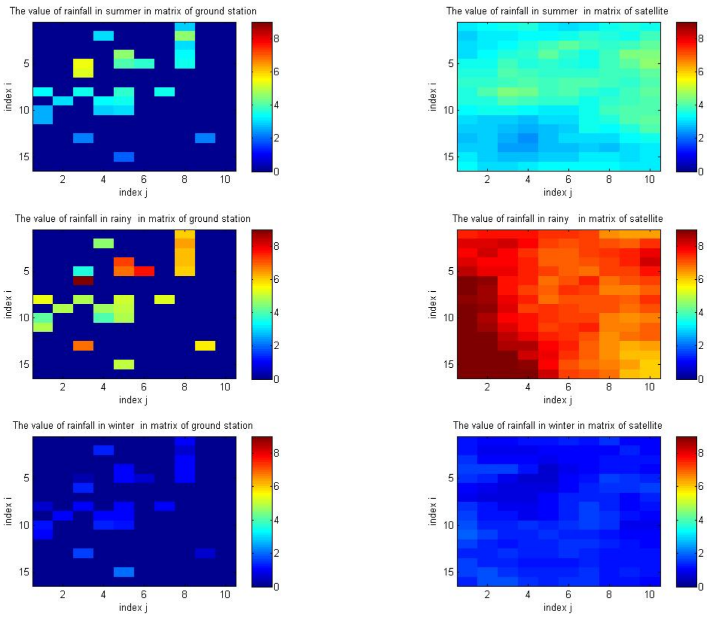

Figure 4: The averaged value (over days in a season) of ground station rainfall and satellite rain fall in each season. *Left.* Ground station data. *Right.* Satellite data. The plot in each row represents summer, rainy, and winter seasons, from top to bottom, respectively. The color maps of each plot are adjusted to be in the same scale.

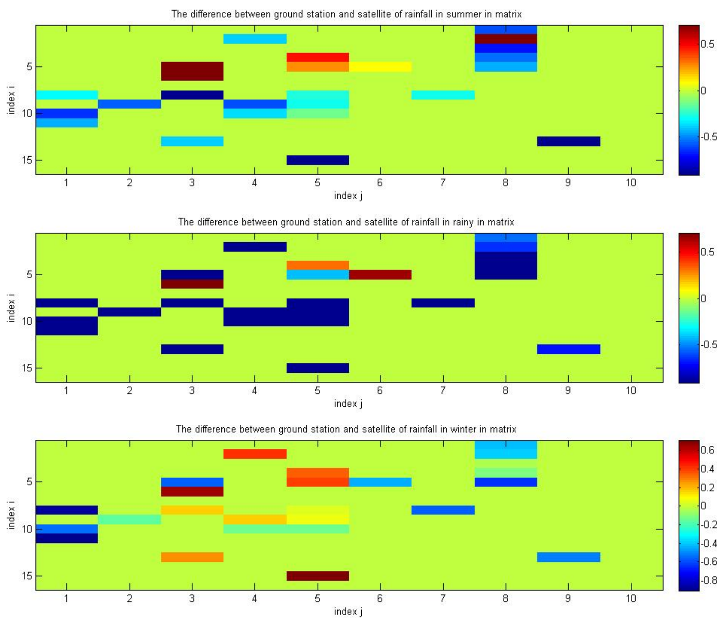

Figure 5: The difference between rainfall data from station and from satellite. The x and y axes represent the grid coordinates (i, j). Each row represents data from summer, rainy and winter seasons, from top to bottom, respectively.

2 Estimation formulation

In this section, we present an estimation formulation of $\hat{y}$ from *one-day* data. The extension of this formulation to consider data throughout the year can be shown in a straightforward way, but is omitted as future work. Suppose that each y_{ij} and z_{ij} is available throughout the entire grid and they can be written in vector as:

$$\begin{aligned} \hat{y} &= (\hat{y}_{11}, \dots, \hat{y}_{m1}, \hat{y}_{12}, \dots, \hat{y}_{m2}, \dots, \hat{y}_{1n}, \dots, \hat{y}_{mn}) \\ z &= (z_{11}, \dots, z_{m1}, z_{12}, \dots, z_{m2}, \dots, z_{1n}, \dots, z_{mn}) \end{aligned}$$

The variable x is obtained by stacking x_{ij} for only $(i,j) \in I_g$ into a column vector. For example, if 3×3 grid has ground station at $I_g = \{(2,1),(2,3)\}$, x will have two entries given by $x = (x_{21}, x_{23})$. An illustration of vectorization process is provided in the appendix.

We can show that from (1), $\hat{y}$ can be expressed as:

$$\hat{y} = Ax + bQz, \tag{2}$$

where A and b are weighting parameters to be estimated, and Q is a mapping matrix that selects elements of z corresponding to the equation (1). We note that A contains $a_{ij,n}$ parameters from (1) and it has a zero structure depending on the ground station locations. To simplify the estimation formulation, we propose a way to permute $\hat{y}$ such that its components are grouped into the grids associated with locations having ground stations and the grids associated with locations without ground stations. To this end, define $Qz = \tilde{z}$ and a permutation matrix P such that $P\hat{y}$ is arranged into

$$P\hat{y} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \end{bmatrix}$$

where $\hat{y}_1$ is a vector consisting $\hat{y}_{ij}$ for $(i,j) \in I_g$ and $\hat{y}_2$ contains $\hat{y}_{ij}$ for $(i,j) \notin I_g$. Note that the length of $\hat{y}_1$ is the cardinality of I_g. This goes the same for $P\tilde{z} = (\tilde{x}_1, \tilde{x}_2)$ and $PA \triangleq \tilde{A} = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}$.

The zero structure of A, denoted by the index set I_A, and matrices P, Q can be computed beforehand and it is explained in the appendix. After the permutation on $\hat{y}$, the zero structure of PA is changed, and denoted by the index set $I_{\tilde{A}}$. As a result, it follows that

$$P\hat{y} = PAx + bP\tilde{z} = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} x + b \begin{bmatrix} \tilde{z}_1 \\ \tilde{z}_2 \end{bmatrix} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \end{bmatrix}$$

To define an estimation formulation, we select choosing A and b in a least-squares sense. We intend to fit the predicted rainfall $\hat{y}$ to the values observed from the ground and satellite stations. A quadratic loss, repesenting goodness of fit, is denoted by f and consists of two terms: ground station and satellite fitting terms. The variables A, b would be chosen in a way that $\hat{y}$ is close to ground station data when available and close to satellite data when not. Furthermore, the satellite is assumed to be less accurate so that it is reasonable to penalize the satellite term with γ. In other word, the parameter γ is a positive number which trades off between the fitting error from station and satellite data. The cost objective can be expressed as

$$\begin{aligned} f(A,b) &= \|\hat{y}_1 - x\|_2^2 + \gamma \|\hat{y}_2 - z_2\|_2^2 \\ &= \|A_1 x + b\tilde{z}_1 - x\|_2^2 + \gamma \|A_2 x + b\tilde{z}_2 - z_2\|_2^2 \end{aligned} \tag{3}$$

Note that the zero structure of A is changed by the permutation. We propose to use the following estimation formulation:

$$\begin{aligned} \min_{A_1, A_2, b} \quad & \|A_1 x + b\tilde{z}_1 - x\|_2^2 + \gamma \|A_2 x + b\tilde{z}_2 - z_2\|_2^2 \\ \text{subject to} \quad & a_{ij} = 0, \quad \forall i,j \notin I_g \\ & a_{ij} \geq 0, \quad \forall (i,j) \\ & b \geq 0, \end{aligned} \tag{4}$$

with variable $A = (A_1, A_2)$ and its entries are a_{ij}, and the variable $b \in \mathrm{R}$. The constraints $a_{ij} \geq 0$ and $b \geq 0$ suggest that we consider only the nonnegative weighting in the interpolation.

3 Result of rainfall prediction

3.1 Sample prediction result on 9 September 2012

A satellite data is in a grid starting at longitude 90 and latitude -15. An adjacent cell represents distance of 0.25 degree. Locating corresponding satellite data, station has the same grid size of 0.25 degree and overall dimension of 16×10. Vectors x, z are reshaped to grid form X, Z of the dimension 16×10 to visualize the result.

As an example, the data on rainy day of 9 September 2012 is considered with $\gamma = 0.5$. From Figure 6 and Figure 7, grid matrix X and Z share the same tendency. A dark color represents a small value, conversely a light color represent a large value. On the z axis of the following plot is rainfall magnitude in centimeter. x, y axes are a grid belongs to latitude of 15.5 to 19.5, and longitude of 99.0 to 101.5.

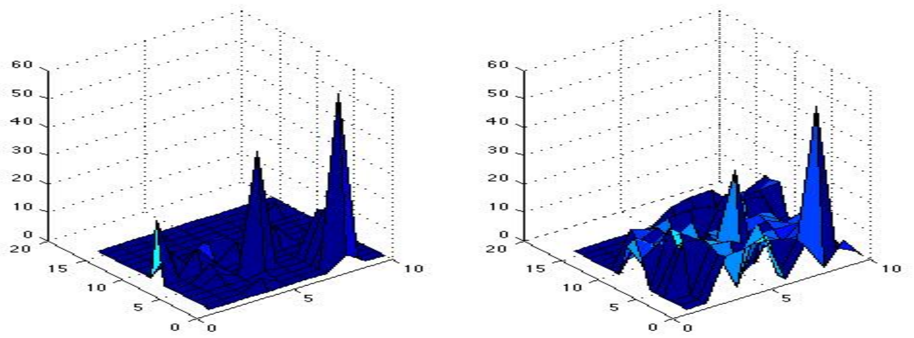

Figure 6: Rainfall from ground station in form of grid matrix X. Original and weighted version are shown on the left and right respectively.

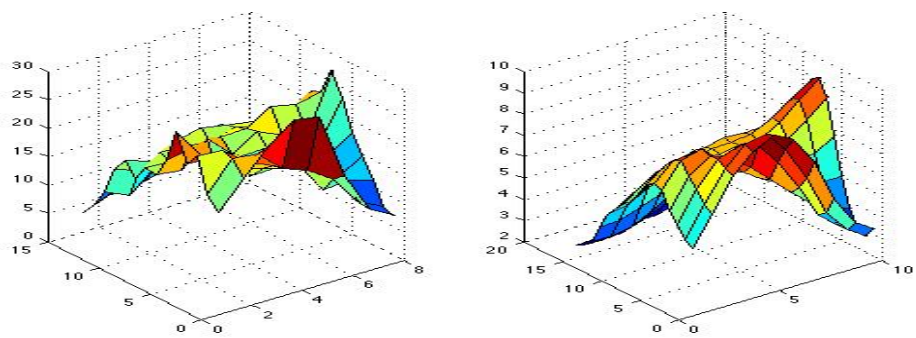

Figure 7: Rainfall from satellite prediction in form of grid matrix Z. Original and weighted version are shown on the left and right respectively.

In Figure 6 and 7, original rainfall data are on the left and the weighted version are on the right. The plot shows that the data from x is preferred to z, as a consequence of higher degree of freedom. The ground station data x would be less preferred by the optimization process when the model is extended to be dynamic.

On the right side of Figure 7, an original rainfall data from satellite is weighted with $b = 0.0407$. If $\hat{y}$ is a result from equally weighted 9 surrounding grid of z, the value of parameter should be $b \approx \frac{1}{9}$.

3.2 Predicted $\hat{Y}$

The predicted $\hat{y}$ is a combination of weighted x and z discussed in section 3.1. The vector $\hat{y}$ are reshaped back to the grid matrix $\hat{Y}$ to visualize the result. In the Figure 8, $\hat{Y}$ has a large fluctuation. From Figure 6 and 7, $\hat{y}$ use z as an background, which remain its value of 1/3. Then, the sharper fluctuation is obtained from weighted x. A peak position of $\hat{Y}$ are similar to those of X and Z.

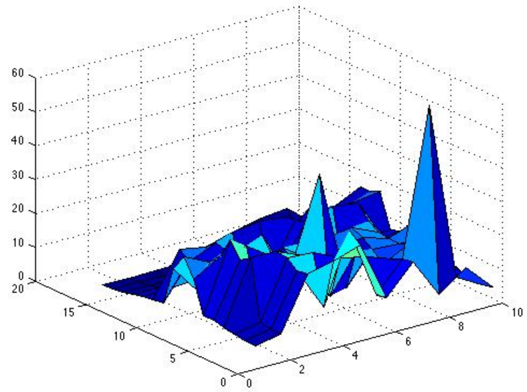

Figure 8: Grid matrix $\hat{Y}$ interpolated from x and z

3.3 Effect on varying penalty parameter

The same result of data on 9 September is evaluated with different γ. Using 50 value of γ from logarithmic space range from 10^{-3} to 10^{3}, the variation is portrayed in Figure 9.

The cost objective according to (3) is recalled into two parts as followed:

$$f_1(A, b) = \|A_1 x + b\tilde{z}_1 - x\|_2^2, \quad (5)$$
$$f_2(A, b) = \|A_2 x + b\tilde{z}_2 - z_2\|_2^2 \quad (6)$$

The parameter γ is a penalty term multiplying to $f_2(A, b)$. If gamma is large, $f_2(A, b)$ will be small. It follows from Figure 9 that when one cost function is small the other cost function is large.

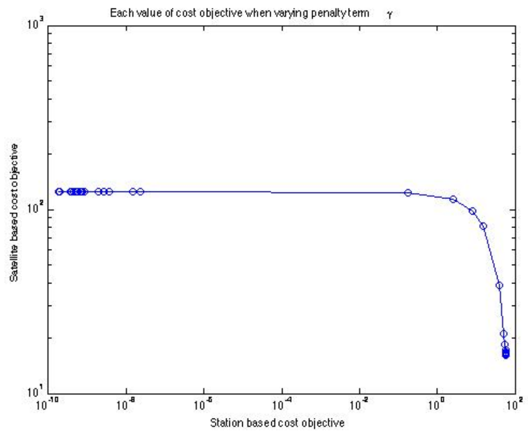

Figure 9: Plot of trade off curve between cost objective $f_1(A, b)$ and $f_2(A, b)$ when varying penalty parameter γ.

3.4 Examples of extreme cases

Within year, there are days when both source of data highly disagree. Two examples of prediction on those extreme cases are provided below:

On October 1, the station grid has its maximum near 6 and has few spike. The satellite grid shows different tendency and has range below 1.5, which is 4 times lower than those of station. With $\gamma = 0.3$, $\hat{Y}$ is predicted in a way that heavily based on x data and ignore z. It follows from Figure 12 and 10 that X and $\hat{Y}$ is approximately the same. As in Figure 11, the value of satellite grid is further suppressed to be below 1.

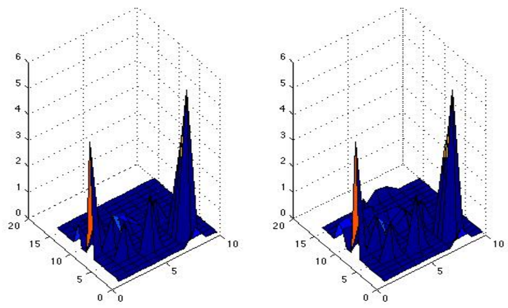

Figure 10: Station grid with some spike

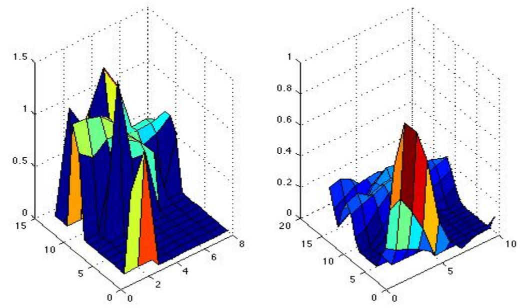

Figure 11: The location of station

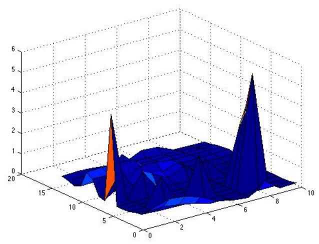

Figure 12: A predicted value of grid $\hat{Y}$

On February 1, the station measurement is all 0, shown in Figure 13. It follows from Figure 14 that there are low but non-zero value for satellite data. The optimization yield very low value for b in order of 10^{-8}, which leads to approximately all 0 entries of predicted $\hat{Y}$.

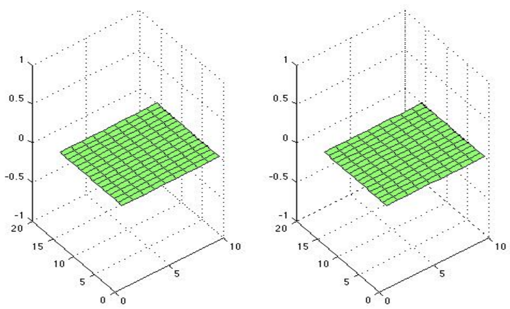

Figure 13: Station grid with no rainfall

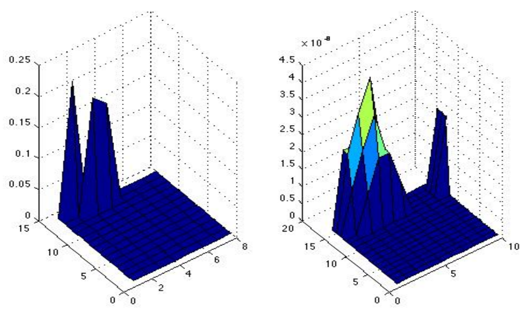

Figure 14: Satellite grid is weighted to be close to 0

Appendix

Programming code

The following `m-file` script yields and estimated value of y. The inputs are station and satellite data `X` and `ZExcess` in grid form. The label `Excess` is a satellite grid matrix with additional entries extended 1 grid from its border. This is added due to the evaluation of $\sum z_{ij}$ at the border grid.

A matrix `hasStation` is a grid which has binary entries. The matrix `hasStation` has entry equal to 1 when there is a ground station at that grid and 0 otherwise. The matrix P in the report is implemented through function `forwardP` which is simply row swapping according to the linear index of I_g. Conversely, `backwardP` is an inverse version P. Note that PA is not a stacking of A_1 and A_2.

The zero structure of A is determined by `getIndexA12Zero` to be used in the constraint of the convex optimization.

```
% Several differences in notation are
% Ig is a linear index
% s denotes spreading Z i.e. Qz
Ig = find(hasStation);
notIg = find(~hasStation);

% Row swapping
% label 1 indicates having station
% label 2 indicates no station
x = X(:);
[x1, x2] = forwardP(x, hasStation);
Z = cropZ(ZExcess);
Zs= cropZ(spreadGrid(ZExcess));
z = Z(:);
zs = Zs(:);
[z1, z2] = forwardP(z, hasStation);
[zs1, zs2] = forwardP(zs, hasStation);


nGr = length(Ig);
nSat = length(zs2);

% obtain zero structure of A
[indexA1Zero, indexA2Zero] = getIndexA12Zero(Ig, hasStation, rowGrid, colGrid);

cvx_begin

    variable A1(nGr, nGr);
    variable A2(nSat, nGr);
    variable b;

    minimize(norm(A1*x1 + b*zs1 - x1, 2) + gamma*norm(A2*x1 + b*zs2 - z2, 2));
    subject to
        A1(indexA1Zero) == 0;
        A2(indexA2Zero) == 0;
        A1 >= 0;
        A2 >= 0;
        b >= 0;

cvx_end
```

```
% WARNING: Don't forget to map back
% A ~= [A1; A2] so do the others
A = backwardP([A1; A2], hasStation);
zs = backwardP([zs1; zs2], hasStation);


% evaluate y
% y = Ax + bQz = Ax + bzs
weightedx = A*x1;
weightedz = b*zs;
y = weightedx + weightedz;

X_est = vec2grid(weightedx);
Z_est = vec2grid(weightedz);
Y_est = vec2grid(y);
```

The function getIndexA12Zero is simply a loop through each index of I_g and assignment of entries related to that index. The description is explained in the comment block in the following code.

```
function [indexA1Zero, indexA2Zero] = getIndexA12Zero(Ig, hasStation, rowGrid, colGrid)

    N = rowGrid*colGrid;
    h = rowGrid;
    nGr = length(Ig);

    nonzeroAGrid = zeros(N, nGr);

    % each level of for loop do each following
    % (a) run each available ground station position
    % (b) go to that point set point before and after to 1
    % (c) go to a column grid before/after that point repeat b

    for i = 1:nGr
        idx = Ig(i);
        for j = [-1,0,1]
            for k = [-h,0,h]
                if(((idx+k)+j > 0) && ((idx+k)+j < N))
                    nonzeroAGrid((idx+k)+j, i) = 1;
                end
            end
        end
    end

    [A1Zero, A2Zero]= forwardP(~nonzeroAGrid, hasStation);

    indexA1Zero = find(A1Zero);
    indexA2Zero = find(A2Zero);

end
```

An implementation on Qz is not by directly find Q. However, the grid notation is used to simplify the thinking process during the implementation.

Let $e_{k,\ell} \in \mathbb{R}^{\ell\times 3}$,

$$e_{k,\ell} = \begin{cases} 1 & \forall (i,j) \in (k-1,1),(k,2),(k+1,3) \\ 0 & \text{otherwise} \end{cases}$$

We rewrite Z at surrounding grid as $\mathbf{e}_{i,m}^T X \mathbf{e}_{j,n}$, then Qz can be expressed as:

$$Qz = \text{vec}(\mathbf{1}^T \mathbf{e}_{i,m}^T Z \mathbf{e}_{j,n} \mathbf{1})$$

Consider product of $\mathbf{1}^T \mathbf{e}_{i,m}^T = \begin{bmatrix} 0 & 0 & \cdots & 1 & 1 & 1 & \cdots & 0 \end{bmatrix}$ where 1 are located at row entries $i-1, i, i+1$. Similarly $\mathbf{e}_{j,n}\mathbf{1} = \begin{bmatrix} 0 & 0 & \cdots & 1 & 1 & 1 & \cdots & 0 \end{bmatrix}^T$, where 1 are located at column entries $j-1, j, j+1$. Let $T_m \in \mathbb{R}^{m \times m}$ be a tridiagonal matrix of the form

$$T_m = \begin{bmatrix} 1 & 1 & & & & \\ 1 & 1 & 1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & 1 & 1 & 1 & \\ & & & 1 & 1 & \end{bmatrix}$$

which evaluated by the following function `getTridiag`. A matrix Qz when in grid form can be obtained by `spreadGrid`.

```
function spread_grid = spreadGrid(grid)

    [row, col] = size(grid);
    Tm = getTridiag(row);
    Tn = getTridiag(col);

    spread_grid=Tm*grid*Tn;

end

function T = getTridiag(m)

    % make tridiagonal matrix for spreading Z
    T=toeplitz([1,1,zeros(1,m-2)],[1,1,zeros(1,m-2)]);

end
```

Apart from MATLAB, `javascript` with `node.js` library is used for reading and cleaning a data into a grid database. The script read a file from large satellite database then select the interested portion of data and write into the new small grid database. The `csv` format is chosen due to its cross platform compatibility. All of the MATLAB implementation utilizes the `X` and `ZExcess`, which is read once form such grid database.

Explanation on grid system

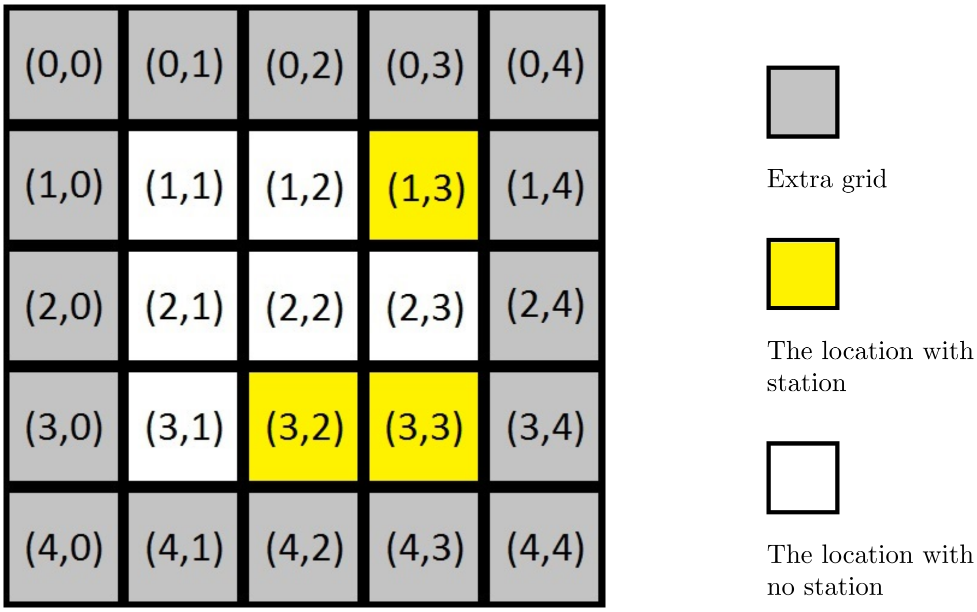

Figure 15: Example of 3×3 grid

From figure 15, we define the rainfall of each grid by the following equations.

$$\hat{y}_{11} = b(z_{00} + z_{10} + z_{20} + z_{01} + z_{11} + z_{21} + z_{02} + z_{12} + z_{22}) \tag{1}$$

$$\hat{y}_{21} = a_{21}x_{32} + b(z_{10} + z_{20} + z_{30} + z_{11} + z_{21} + z_{31} + z_{12} + z_{22} + z_{32}) \tag{2}$$

$$\hat{y}_{31} = a_{31}x_{32} + b(z_{20} + z_{30} + z_{40} + z_{21} + z_{31} + z_{41} + z_{22} + z_{32} + z_{42}) \tag{3}$$

$$\hat{y}_{12} = a_{12}x_{13} + b(z_{01} + z_{11} + z_{21} + z_{02} + z_{12} + z_{22} + z_{03} + z_{13} + z_{23}) \tag{4}$$

$$\hat{y}_{22} = a_{22,1}x_{13} + a_{22,2}x_{32} + a_{22,3}x_{33} + b(z_{11} + z_{21} + z_{31} + z_{12} + z_{22} + z_{32} + z_{13} + z_{23} + z_{33}) \tag{5}$$

$$\hat{y}_{32} = a_{32,1}x_{32} + a_{32,2}x_{33} + b(z_{21} + z_{31} + z_{41} + z_{22} + z_{32} + z_{42} + z_{23} + z_{33} + z_{43}) \tag{6}$$

$$\hat{y}_{13} = a_{13}x_{13} + b(z_{02} + z_{12} + z_{22} + z_{03} + z_{13} + z_{23} + z_{04} + z_{14} + z_{24}) \tag{7}$$

$$\hat{y}_{23} = a_{23,1}x_{13} + a_{23,2}x_{32} + a_{23,3}x_{33} + b(z_{12} + z_{22} + z_{32} + z_{13} + z_{23} + z_{33} + z_{14} + z_{24} + z_{34}) \tag{8}$$

$$\hat{y}_{33} = a_{33,1}x_{32} + a_{33,2}x_{33} + b(z_{22} + z_{32} + z_{42} + z_{23} + z_{33} + z_{43} + z_{24} + z_{34} + z_{44}) \tag{9}$$

From equation (1)-(9),we can rewrite them to this form:

$$\hat{y} = Ax + bQz, \tag{10}$$

where variables and parameters are define as:

$$\hat{y} = \begin{bmatrix} \hat{y}_{11} \\ \hat{y}_{21} \\ \hat{y}_{31} \\ \hat{y}_{12} \\ \hat{y}_{22} \\ \hat{y}_{32} \\ \hat{y}_{13} \\ \hat{y}_{23} \\ \hat{y}_{33} \end{bmatrix}, A = \begin{bmatrix} 0 & 0 & 0 \\ a_{21,1} & 0 & 0 \\ a_{31,1} & 0 & 0 \\ 0 & a_{12,1} & 0 \\ a_{22,1} & a_{22,2} & a_{22,3} \\ a_{32,1} & 0 & a_{32,2} \\ 0 & a_{13,1} & 0 \\ a_{23,1} & a_{23,2} & a_{23,3} \\ a_{33,1} & 0 & a_{33,2} \end{bmatrix}, x = \begin{bmatrix} x_{32} \\ x_{13} \\ x_{33} \end{bmatrix}$$

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T, \quad z = \begin{bmatrix} z_{00} \\ z_{10} \\ z_{20} \\ z_{30} \\ z_{40} \\ z_{01} \\ z_{11} \\ z_{21} \\ z_{31} \\ z_{41} \\ z_{02} \\ z_{12} \\ z_{22} \\ z_{32} \\ z_{42} \\ z_{03} \\ z_{13} \\ z_{23} \\ z_{33} \\ z_{43} \\ z_{04} \\ z_{14} \\ z_{24} \\ z_{34} \\ z_{44} \end{bmatrix}, \quad and \quad b = b$$