DATA LAYOUT AND LEVEL-OF-DETAIL CONTROL FOR FLOOD DATA VISUALIZATION

Sayaka Yagi Takayuki Itoh
Ochanomizu University
Mayumi Kurokawa Yuuichi Izu Takahisa Yoneyama Takashi Kohara
Toshiba Corporation

ABSTRACT

Analysis of weather and water level is very important to protect our inhabitable area from the damages of floods. Visualization of numeric information related to the weather and water level is also important to analyze, monitor, and alert the danger. Here, we think it is important to geographically represent such numeric information, so that we can realize combinational analysis of multivariate numeric information and its geographic distribution. This paper introduces our prototype software for visualization of flood data including river stage and precipitation amount on the map. This software adequately and quickly places rectangles which represents numeric information according to users' interaction. Our implementation provides two types of representation of the numeric information. Users can switch the two types according to their preferences, usages, and characteristics of datasets. This paper introduces two use cases of the software that demonstrates the effectiveness of the software.

1. INTRODUCTION

It is important to analyze past weather and water level that damaged our inhabitable area due to the floods. Visualization of numeric information including river stage and precipitation amount is useful for the analysis of such past floods. Here, we think it is important to geographically represent the distribution of the numeric information, so that we can realize combinational analysis, monitoring, and alert of the danger of floods. Recent visual analysis research realized various analysis and knowledge discovery including geographic problems. We think visual analysis technique is also useful for analysis and monitoring of flood data.

We have developed a visual analysis tool for flood problem around rivers, which displays numeric information of river stage and precipitation amount on the map. We think that the following are the requirements for the visualization of numeric information of river floods:

- Combinational visualization of river-related numeric information (e.g. stage, flow quantity) and weather-related numeric information (e.g. precipitation amount).
- Visualization of long-range (e.g. one week) time-varying numeric information.
- Switch of wide and local area visualization.
- Representation for easier focus of dangerous times and places.

Based on the above assumption of requirements, we developed prototype software that displays small rectangular charts of the numeric information adequately and quickly placed onto the map. This visualization tool is useful to look over the numeric information associating with the geographic features and distributions. Also, the tool provides two types of representations of the numeric information: polyline charts and color bars. Polyline chart is better to finely read the values; however, it occupies more spaces. We therefore think our implementation of polyline charts is preferable to locally observe the focused regions. On the other hand, our implementation of color bars is better to quickly look over the distribution of the values and find danger regions. Also, it occupies less spaces comparing with the polyline charts. We therefore think our implementation of color bars is preferable to globally look over the information. Users can easily find dangerous regions and focus on there by switching the two types.

2. RELATED WORK

There have been many studies on visualization of geographic numeric information. Lundblad et al. [1] presented a Web-based visual application for road weather data representing on the map. However, the application does not directly display time-variation of the weather information on the map. Sanyal et al. [2] represented weather uncertainty on the map by layers of glyphs and ribbons. Again, this technique does not directly display time-variation of the weather information on the map.

Rectangle layout for display of polyline charts of color bars in our implementation is somewhat similar to a famous visualization problem of data labeling [3]. Data labeling of geographic information on the map is a relatively complicated problem because of geographic constraints; however, there have been also many techniques [4]. One of the recent novel works is presented by Speckmann et al. [5], which circularly place the symbols that displays geographic statistics around the map.

3. FLOOD DATA VISUALIZATION

We assume that the datasets visualized by our software have the following structure:

- $D=\{R,W\}$, where D is the whole dataset, R is a set of river-related information (e.g. stage), and W is a set of weather-related information (e.g. precipitation amount).
- $R=\{r_1, ..., r_m\}$, where r_i is the i-th observation point of the river-related information, and m is the number of them.
- $W=\{w_1, ..., w_l\}$, where w_i is the i-th observation point of the weather-related information, and l is the number of them.
- $r_i=\{r_{i1}, ..., r_{in}, h_i, u_i, v_i\}$, where r_{ij} is the value at the j-th time step of the i-th observation point, n is the number of the time steps, h_i is the dangerous level of the values of r_{ij}, u_i is the longitude of r_i, and v_i is the latitude of $r_{i.}$
- $w_i=\{w_{i1}, ..., w_{in}, u_i, v_i\}$, where w_{ij} is the value at the j-th time step of the i-th observation point, n is the number of the time steps, u_i is the longitude of r_i, and v_i is the latitude of $r_{i.}$

We do not assume that observation points need to measure both river-related and weather-related information. We assume that only river-related information is measured at several observation points, as well as only weather-related information is measured at several other observation points.

Our visualization software displays a map of the target region in the left side of a window. It overlays rivers approximated by polygonal lines and rectangles which displays polyline charts or color bars. Rectangles for river-related information are painted in brown, while rectangles for weather-related information are painted in blue. Our implementation enables shift and zoom of the map according to mouse operations. Users can switch polyline charts and color bars by the selection of tabs in the right side of the window.

Figures 1 and 2 are examples by polyline charts and color bars. The left side of the window displays the rectangular polyline charts or color bars on the map. The right side of the window shows timeline display of the information or color map for color bars. The below section describes the implementation detail.

3.1 Data Representation by Polyline Charts and Timeline Area

Figure 1 shows an example that applies polyline charts in rectangles on the map. The left side of the window displays river stages in blue rectangles, and precipitation amounts in brown rectangles. The right side of the window displays timeline of such values, where the horizontal axis denotes time steps. River stages and precipitation amounts are represented as horizontally long bars which are vertically aligned. Our implementation displays precipitation amounts in the upper part, and river stages in the lower part. Hue of the color bars depicts the values of precipitation amounts or river stages.

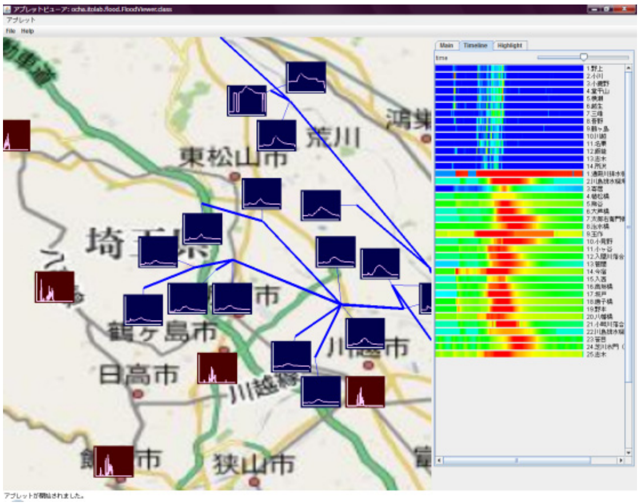


Figure 1. Example of polyline charts on the map.

Our implementation calculates the shapes of polyline charts and colors of timeline color charts as follows. It consumes the dangerous level values of the river stages h_i for each observation point. Also, it calculates the maximum value of the precipitation amounts per unit time w_{max}. Then, it calculates the relative values of river stages or precipitation amounts, r_{ij}' or w_{ij}', by the following equations:

$$r_{ij}' = r_{ij} / h_i,\ \ r_{ij}'=1 \text{ (if } r_{ij}'>1) \qquad (1)$$

$$w_{ij}' = w_{ij} / w_{max},\ \ w_{ij}'=1 \text{ (if } w_{ij}'>1) \qquad (2)$$

Our implementation calculates the heights of polyline charts at each time step from r_{ij}' or w_{ij}'. Also, it calculates the hues of timeline color charts from r_{ij}' or w_{ij}', so that 1.0 corresponds to red, and 0.0 corresponds to blue.

Polyline chart is relatively better to finely read the values of river stages or precipitation amounts. Also, timeline color charts makes easier to understand the causal correlation between river stages and precipitation amounts. However, the polyline charts occupies relatively more spaces to finely read the values. It may cause to overdraw too much geographic information if it displays many rectangles of polyline charts. To solve the problem, our software provides another solution described in the below section.

3.2 Data Representation by Color Bars

Figure 2 shows an example that applies color bars in rectangles on the map. Brightness in the color bars depicts the values of river stages or precipitation amounts, and hue depicts the time. The right side of the window displays the correspondence relationship between the time and hue. As Figure 2 shows, color bar is comprehensive to briefly read the values by using relatively thin rectangles. We think this representation can solve the problem discussed in Section 3.1 that polyline charts may occupy the large part of display spaces.

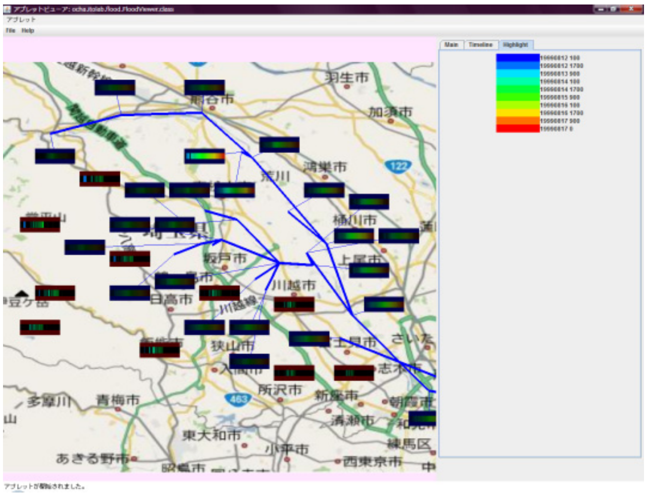


Figure 2. Example of color bars on the map.

We expect that users will find bright parts of the color bars, and then find important regions, which have larger river stage or precipitation amount values. They can also quickly understand which time they have larger values from the hues.

Our implementation of color bars applies level-of-detail control, which non-uniformly assigns subregions to each time interval. Figure 3 shows the mechanism of non-uniform subregion assignment. It assigns wider subregions to time intervals which have larger values, and narrower subregions to time intervals which have smaller values. This non-uniform assignment emphatically displays important time intervals, and prevents that important subregions are collapsed in small display spaces. We expect users can easily focus on important times and regions by applying this representation.

(a) uniform assignment (b) non-uniform assignment

Figure 3. Subregion assignment in a color bar.

Our implementation calculates widths of the subregions by the following equations. It firstly calculates the brightness B from r_{ij}' or w_{ij}'. It then calculates the weight a_t by the following equations:

$$a_t = 3.0B + 0.5 \quad (3)$$

$$B = \frac{v_t - v_{\min}}{v_{\max} - v_{\min}} \quad (4)$$

Here, v_t denotes the value of river stage or precipitation amount at the time t. v_{max} and v_{min} denote the maximum or minimum value of river stage or precipitation amount.

It then calculates the width of the subregion w_i from a_i by the following equation:

$$w_t = \frac{a_i}{\sum a_i} \times (x_{\max} - x_{\min}) \quad (5)$$

Here, x_{max} and x_{min} denote the x-coordinate values at the left-end and right-end of a color bar.

3.3 Rectangle Layout and Level of Detail Control

The presented software calculates positions of rectangles of polyline charts or color bars when viewing configuration is changed according to shift or zoom operation. Here we implemented a rectangle placement algorithm that satisfies the following conditions:

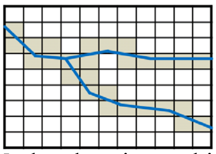
- Never overlap rectangles each other.
- Never overdraw rectangles on the rivers.
- Attempt to minimize the distance between a rectangle and corresponding position on the map.
- Attempt to minimize the movements of rectangles.
- Calculate positions in a small computation time.

The procedure of rectangle placement in our implementation is as follows. Figure 4 illustrates the procedure.

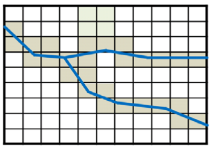
1. Divide the map area by a grid, and draw polygonal segments approximating the geometry of rivers. The implementation supposes that rectangles of polyline charts or color bars are drawn in larger rectangular subregions formed by a set of grid subregions.
2. Mark rectangular subregions which the polygonal segments pass through. Gray subregions in Figure 4 are the marked subregions.
3. Define the order of rectangle placement.
4. Select an unplaced rectangle in the order defined in 2. Apply the following procedure to the selected rectangle:
   4-1) Determine the ideal position of the rectangle.
   4-2) Attempt to place the rectangle at the rectangular subregions. The implementation attempts around the ideal position.
   4-3) Decide the position of the rectangle where a set of unmarked subregions.
   4-4) When an adequate position is found, it marks all subregions which are overdrawn by the rectangle to indicate the position of the current

rectangle. Green subregions in Figure 4 depict the marked subregions.

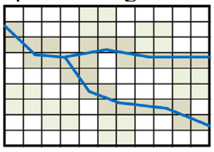
5. Repeat 4. for all rectangles. The implementation may give up placing several rectangles if it cannot find any adequate positions.
6. Draw segments connecting the observation points on the rivers and corresponding rectangles, after dealing with all the rectangles.



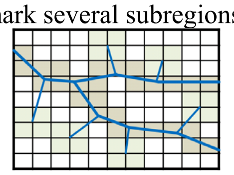
(a) Mark subregions which rivers pass through.



(b) Place a rectangle and mark several subregions.



(c) Place rectangles one-by-one.



(d) Draw segments connecting rectangles and

Figure 4. Algorithm for rectangle placement on the map.

Here, our early implementation defined the ideal positions of rectangles just as the positions observation points defined as (u_i,v_i) for the i-th observation point. However, we observed that this setting made movement of rectangle larger, and felt that visualization results were therefore unstable. We modified the definition of the ideal positions as dividing point between the positions of observation points and the position where the rectangles are previously placed, as the following equation:

$$x_i' = t\,u_i + (1-t)\,x_i \qquad (6)$$

$$y_i' = t\,v_i + (1-t)\,y_i \qquad (7)$$

Here, (x_i,y_i) is the position where the i-th rectangle is previously placed, (x_i',y_i') is the ideal position of the i-th rectangle, and t is a constant value satisfying $0<t<1$.

This algorithm adaptively controls the number of displayed rectangles according to the resolution of the map and density of the rectangles, and consequently realizes the level of detail control which displays important regions of measurement data as much as possible. On the other hand, this algorithm may occur different placement results due to the order of rectangle placement. Therefore, we need to implement consistent order of the placement so that important rectangles should be placed first.

4. CASE STUDY

We applied datasets published by Water Information System of Ministry of Land, Infrastructures and Transport [6]. The system publishes the following information on the Web:

- Name of rivers
- Names, longitude, latitude, and dangerous level of observation points
- River stage or precipitation amount at each time step

4.1 Example (1)

We visualized a dataset during August 12 to 17 in 1999, along Arakawa River in Saitama Prefecture, Japan, including 25 observation points for river stages and 14 observation points for precipitation amounts. We had a flood as a result of heavy rain by a tropical cyclone.

Figure 5 is an example of visualization of all the observation points by color bars on the map. This example displays 41 color bars as thin rectangles; however, geography is still comprehensive because the thin rectangles do not overdraw the geographic information too much.

The example depicts that river stages were relatively high for a long time in the north-east parts of the river, because color bars were relatively bright. The result suggests that especially we needed to pay attention to the flood of north-east parts of the river.

Also, the example clearly represents the spatial difference of rain. It depicts that it was rainy for a long time in the west part, because color bars of precipitation amounts marked by an orange rectangle were relatively bright. On the other hand, it depicts that it was rainy for a short time in the east part, because color bars marked by pink circles. Also, it depicts that we had a strong short rain before the long rain at several observation points marked by dotted black circles, because we can observe bright blue in the color bars.

As mentioned above, this example well demonstrates the effectiveness of color bars in our software. We can focus on characteristic observation points just by looking at the brightness of color bars, and understand when we had dangerous levels of rain stages or strong rains just by looking at the hues of the bright parts of them. We can zoom into a focused region, and then switch to polyline charts, if we would like to finely read the values of river stage or precipitation amount.

Figure 7 is an example of visualization of all the observation points by timeline color charts. The upper part displays precipitation amounts, and the lower part displays river stages. The river stages arrange in the order from the upper stream to the lower stream for each arm of a river. The example in Figure 7 depicts that river stages in the upper stream got higher just several hours after a strong rain, and then river stages in the lower stream got higher.

As mentioned above, we observed a short rain at several observation points marked by dotted black circles in Figure 5. Such short rain can be also observed in Figure 7, as marked by pink circles. We observe that river stages got higher just at several observation points marked by pink circles in Figure 7, just after the short rain.

Also, we can observe that river stages quickly got lower after the rain at several observation points, while river stages were high for a long time at other several observation points. We think such observation points might be dangerous if we had another rain. Especially we need to remark that river stages were extremely high for a long time at two observation points (1) and (2) in Figure 6. Also, we need to remark that river stages marked as (3) in Figure 7 got low earlier, even though its corresponding observation point was located at a lower stream of the river. As mentioned above, we think that timeline color chart is useful to understand temporal relationships among observation points.

On the other hand, timeline color chart is not sufficient to understand geographic relationship among observation points. The two observation points marked as (1) and (2) in Figure 7 were separately displayed because they are in the different arms of the river; however, they were geographically close, as marked by a green rectangle in Figure 5. We think that timeline color chart is useful to analyze temporal relationships among observation points; however, it is not always useful to analyze their geographic relationships. We think color bar shown in Figure 5 is useful to briefly observe and understand relationships among observation points both temporally and geographically.

4.2 Example (2)

We also visualized another dataset during July 2 to 9 in 2000 at the same region in Japan, including 28 observation points for river stages and 14 observation points for precipitation amounts. We had a flood as a result of heavy rain by a cyclone.

Figure 6 is an example of visualization of all the observation points by color bars on the map. Figure 8 is an example of visualization of all the observation points by timeline color charts. Numeric characters, (1), (2), (3), and (4), in Figures 6 and 8 depict corresponding observation points.

Figure 6 depicts that river stages were high for a long time at four observation points marked as (1), (2), (3), and (4). We can observe that the four observation points are not necessarily close by looking at Figure 6. The example shown in Figure 6 also depicts that river stages or precipitation amounts were relatively high at the particular time step represented in yellow at other observation points.

At the same time, Figure 8 also depicts river stages were high for a long time at the four observation points, (1), (2), (3), and (4). The figure also depicts that river stages or precipitation amounts were relatively high at the particular time step. We subjectively feel that we can observe this feature more clearly by looking at Figure 8.

From this example, we would like to suggest using this software as follows. Firstly look at the whole region with color bars on the map, and briefly find temporal and geographic relationships among observation points. Then, focus to interested regions or times, and switch to display polyline charts and timeline color charts to finely read the values.

5. CONCLUSION

This paper presented our prototype software for visualization of flood data including river stages and precipitation amounts. The paper also demonstrated our software with real datasets of past flood damage.

Our potential future issues include the following:

[New design to draw numeric information] Our current design applies similar polyline charts or color bars to both river stages and precipitation amounts, and distinguishes them just by background colors. We do not think it is a sufficient design. We would like to implement additional drawing functions to display numeric information by other metaphors, including colored arrow lines on the rivers.

[Order of rectangle placement] Our current implementation just places rectangles in the order of description in the dataset files. We think there are better definitions of ordering of observation points. For example, we would like to preferentially place rectangles corresponding to observation points which had dangerous values. Or, we would like to preferentially place rectangles corresponding to observation points at users' interested regions.

REFERENCES

[1] P. Lundblad, J. Thoursie, M. Jern, "Swedish Road Weather Visualization", *14th International Conference on Information Visualisation*, 313-321, 2010.

[2] J. Sanyal, et al., "Noodles: A Tool for Visualization of Numerical Weather Model Ensemble Uncertainty", *IEEE Transactions on Visualization and Computer Graphics*, 16(6), 1421-1430, 2010.

[3] J.-D. Fekete, C. Plaisant, "Excentric Labeling: Dynamic Neighborhood Labeling for Data Visualization", *ACM SIGCHI Conference: Human Factors in Computer Systems*, 512-519, 1999.

[4] A. Skupin, "A Cartographic Approach to Visualizing Conference Abstracts", *IEEE Computer Graphics and Applications*, 22(1), 50-58, 2002.

[5] B. Speckmann, K. Verbeek, "Necklace Maps", *IEEE Transactions on Visualization and Computer Graphics*, 16(6), 881-889, 2010.

[6] Water Information System, http://www1.river.go.jp/

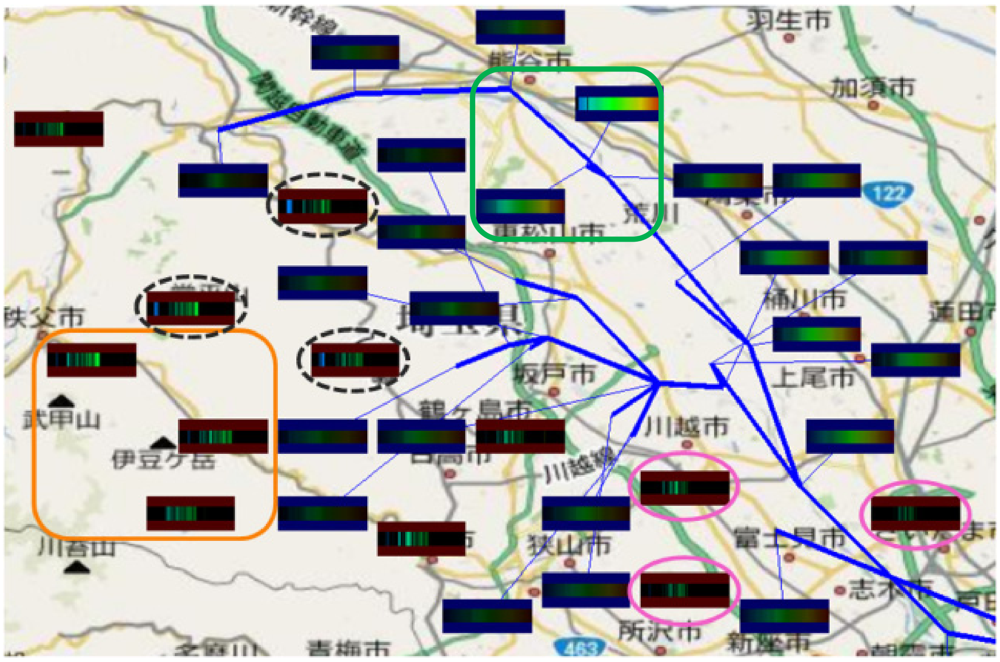


Figure 5. Example of color bars (1).

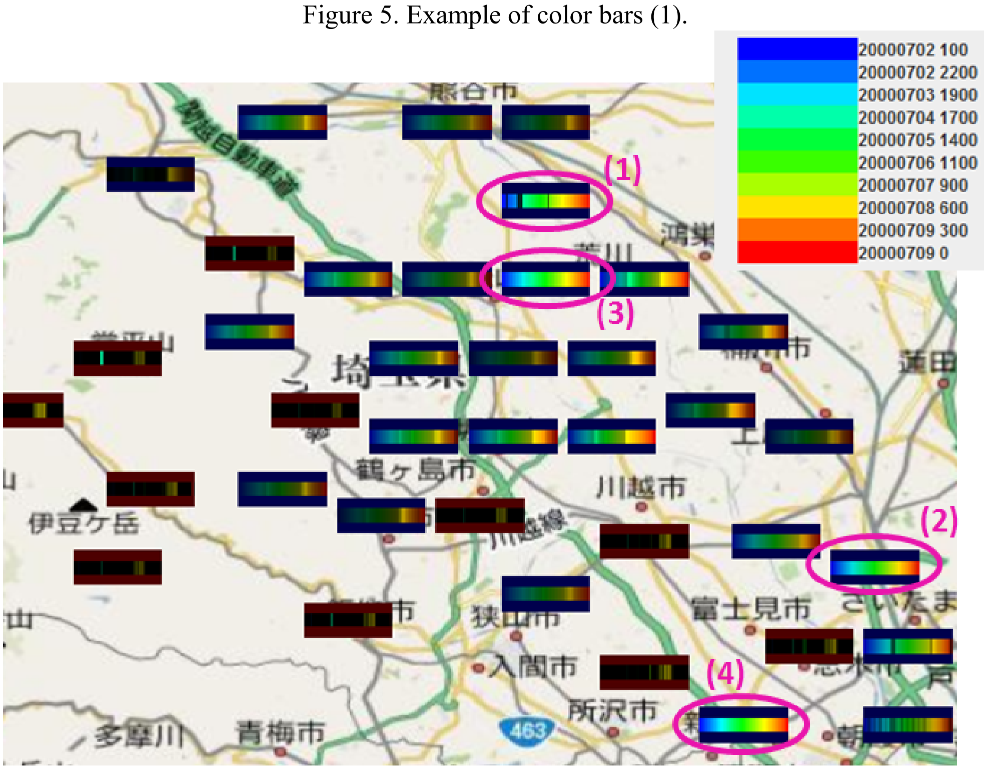


Figure 6. Example of color bars (2).

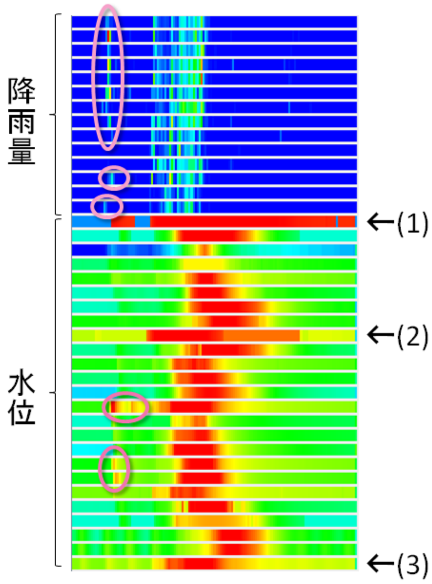


Figure 7. Example of timeline color chart (1).

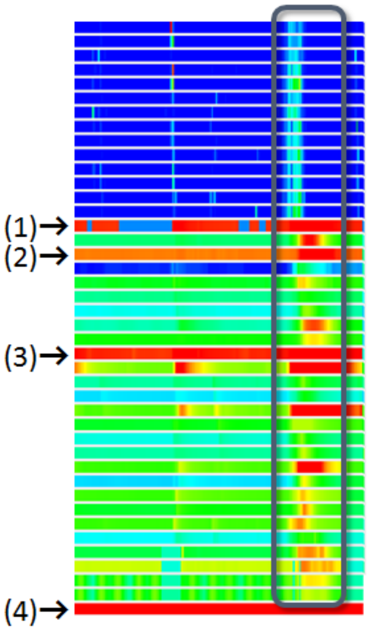


Figure 8. Example of timeline color chart (2).