ContourDiff: Revealing Differential Trends in Spatiotemporal Data

Zonayed Ahmed*, Michael Beyene†, Debajyoti Mondal‡, Chanchal K. Roy§,
Christopher Dutchyn¶, and Kevin A. Schneider‖
Department of Computer Science, University of Saskatchewan, Saskatoon, Saskatchewan, Canada
Email: *zonayed.ahmed@usask.ca, †michael.beyene@usask.ca, ‡d.mondal@usask.ca,
§chanchal.roy@usask.ca, ¶dutchyn@cs.usask.ca, ‖kevin.schneider@usask.ca

***Abstract*—Changes in spatiotemporal data may often go unnoticed due to their inherent noise and low variability (e.g., geological processes over years). Commonly used approaches such as side-by-side contour plots and spaghetti plots do not provide a clear idea about the temporal changes in such data. We propose ContourDiff, a vector-based visualization over contour plots to visualize the trends of change across spatial regions and temporal domain. Our approach first aggregates for each location, its value differences from the neighboring points over the temporal domain, and then creates a vector field representing the prominent changes. Finally, it overlays the vectors along the contour paths, revealing differential trends that the contour lines experienced over time. We evaluated our visualization using real-life datasets, consisting of millions of data points, where the visualizations were generated in less than a minute in a single-threaded execution. Our experimental results reveal that ContourDiff can reliably visualize the differential trends, and provide a new way to explore the change pattern in spatiotemporal data.**

***Index Terms*—Spatiotemporal Data, Change Detection, Contour Plot, Vector Overlay**

I. Introduction

The increasing availability of geosensors and remote sensing technologies have created an abundance of spatiotemporal data [5]. Most of such data are collected periodically over time. Common geospatial data visualization approaches include techniques from a variety of sources such as statistical plots and charts, exploratory data analysis, cartography, and image processing [35]. A combination of these approaches enables us to connect particular phenomena to a spatial reference system [14]. Understanding temporal changes in geospatial data (e.g., soil moisture data or rive-bank erosion) provides valuable information about geological and hydrological processes [30]. However, the inherent noise and slow rate of change [18] makes it challenging to visualize the change pattern in such data. For example, a side-by-side contour plot of the data does not help detecting changes unless the changes are noticeably large.

This motivated us to examine ways to visualize the differential trend in spatiotemporal data. A *differential trend* at a spatial point P is the aggregated value differences of P from its local neighbourhood. Throughout the paper, we consider two types of trends: increasing and decreasing, and visualize them using two vectors. The *increasing (resp., decreasing)* trends are obtained by considering neighbors that have a higher (resp., lower) value than that of P. Fig. 1 illustrates the elevation of three data points p, q, r at different timestamps $t_1, \ldots, t_k$. At a timestamp t_k, the increasing and decreasing trends at point q are shown using the blue and black arrows, respectively. The direction of the black arrow is leftward, which indicates that historically the left neighbour p has had a lower elevation than q. The length of the arrow corresponds to the average value difference over all timestamps. Since the length of the black arrow is larger than that of blue arrow, we know that the average slope between p and q is sharper than the slope between q and r.

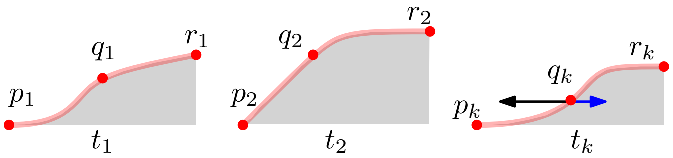
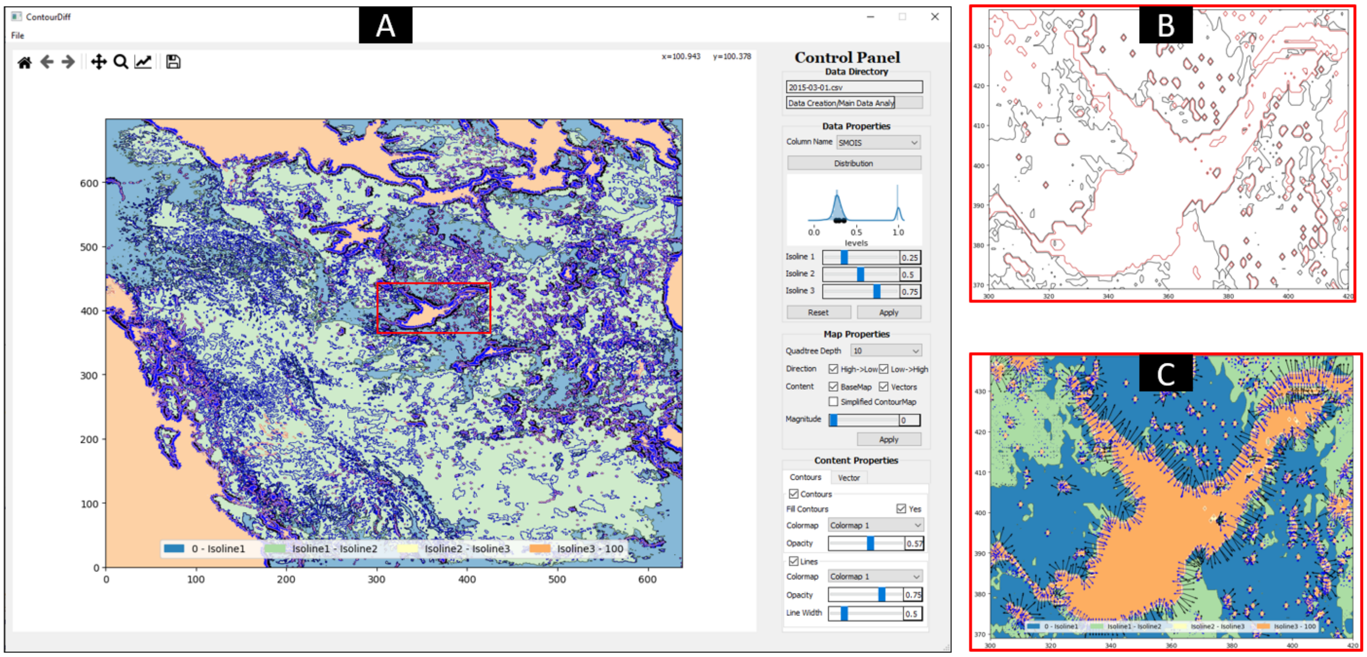


Fig. 1. Illustration for differential (increasing and decreasing) trends.

Our idea is simple yet powerful to summarize temporal change information on a single visualization (e.g., see Fig. 2). This can also be seen as a generalization of image gradient [6], which is a fundamental image processing technique to extract information based on the color or intensity changes in a single image. To the best of our knowledge, this is the first attempt to adapt the image gradient approach to visualize changes in spatiotemporal data.

We implemented a system — ContourDiff, to examine whether the vector overlay of differential trends on a contour plot can provide enriched information. ContourDiff first computes a vector field over the spatial domain and overlays that along the contour lines of the region of interest. However, unlike the uniformly spread vector or glyph overlay (e.g., in wind speed or ocean current visualization [8], [31]), we plot the directions along the contour lines to keep the visualization free of visual clutter.

We evaluated ContourDiff by examining the visualizations using real-life datasets (e.g., motion capture data [13] and Weather Research and Forecasting model output [21]). Furthermore, we design a standard user interface to enable user interactions such as to check data distribution and choose contour thresholds, or to edit vector mapping and map content properties. We also provide an interactive panel that deploys



Control Panel

Data Directory

2015-03-01.csv

Data Creation/Main Data Analy

Data Properties

Column Name: SMOIS

Distribution

Isoline 1: 0.25
Isoline 2: 0.5
Isoline 3: 0.75

Reset | Apply

Map Properties

Quadtree Depth: 10

Direction: ☑ High->Low ☑ Low->High

Content: ☑ BaseMap ☑ Vectors
☐ Simplified ContourMap

Magnitude: 0

Apply

Content Properties

Contours | Vector

☑ Contours

Fill Contours: ☑ Yes
Colormap: Colormap 1
Opacity: 0.57

☑ Lines

Colormap: Colormap 1
Opacity: 0.75
Line Width: 0.5

Fig. 2. (a) An overview of the ContourDiff system. The contour plot depicts the soil moisture data over Western Canada. The vectors along the isolines show the historical value differences observed from 2013 to 2015. (b) A zoomed in view of the isolines around the Great Slave Lake. (c) Each blue (resp., black) vector reveals the increase (resp., decrease) in soil moisture value from the origin to the vector direction.

zooming and panning mechanisms, interchanges different states of the map on user command, and saves the image. The overview of the whole system is portrayed in Fig. 2. Our experiments with real-life datasets reveal the effectiveness of visualizing differential trends in spatiotemporal data.

II. Related Work

A. Common Geospatial Visualizations

Box plots are commonly used to visualize the data distribution through summary statistics. Surface box plots generalize the standard box-plots in two and three-dimensions, where the surfaces are computed based on the three quartiles at each two-dimensional point. Surface box plots and various other extensions [11] have been proposed to portray the distribution of geophysical variables over a long period of time. Potter et al. [19] proposed a hybrid summary plot for two-dimensional data distributions that incorporates a collection of descriptive statistics and density information.

Geospatial data over time are often visualized using a series of contour plots and heatmaps. Such plots commonly use various color schemes to leverage humans' natural ability to differentiate colors [34]. A rich body of literature examines effective ways of visualizing geospatial data using colormaps. Color blending [32] widely appears in analyzing correlations among a pair of variables over a spatial domain. Spaghetti plot is an approach to represent ensemble data in an inter-twining time-series manner. Spaghetti plots [23] are conventionally used by meteorologists to examine the uncertainty exhibited by spatial ensembles. Sometimes the ensemble data are overlaid on color maps or contour plots to have a deeper understanding of the ensembles. Potter et al. [20] presented an approach to examine the uncertainty in ensemble data by presenting an isoline for each member of the ensemble data.

However none of the mentioned charts or visualizations can readily visualize differential trends. In addition, potential attempts to blend contour maps from different timestamps and overlay isolines, become unreadable due to visual clutter as the number of timestamps grow.

B. Visualizing Time-varying Dataset

Separate plots for each timestamp or small multiple plots are difficult to explore when the number of timestamps is large. Therefore, various approaches such as animation, and timestamp maps have been proposed in the literature. Slocum et al. [24] proposed MapTime that explores spatiotemporal datasets associated with point locations through animated maps in automatic or user-controlled frame rate. One limitation of animation is that change blindness can cause people to not recognize minor changes [17]. Moreover, memorizing different states of changes can be tiresome, even if users can interact by jumping to different timestamps. Andrienko et al. [2] proposed a time window technique for the animation of object movement, where route fragments passed by the moving objects are shown specifying time lengths. Tempo-Vis [1] also provides time-based interactive exploration using a time-slider by which users can navigate through different time points and explore social network activities. However, animation or such interactive approaches are difficult to adapt for identifying subtle differences even between two consecutive windows.

TransGraph [7] provides a visual mapping of volumetric data into a 2D plane that abstracts data evolution over time in different levels of detail. Weber et al. [29] presents a new approach for the visualization of time-series data with periodic behavior using SpiralGraphs. An interactive framework for the 3D visualization of the time-series of Web graphs has been proposed by Itoh et al. [10] where Intra-graph, Inter-time, and Inter-facet relations were used enabling users to explore the evolution of time sequences.

C. Directional Representations

Some geospatial data are inherently directional, e.g., such as wind flow and ocean current. Vector mapping is commonly used to visualize such geographic features [27]. For example, Auer et al. [3] used the gradient tensor field to visualize large scale trends on the background and strongly expressed features at their position as local visualization. They used earthquake and climate datasets to show the trends of change based on the most persistent maxima of their dataset. Kersting and Döllner [12] specified a visual mapping of vector data by scene graph and project rendered 2D textures onto georeference geometry such as terrain surfaces. Telea and Wijk [26] presented a clustering approach for a given vector field with arrow count and generates compact vector field visualization using a hierarchical simplification of vector data.

Wittenbrink et al. [31] visualized uncertainty in an environmental dataset using a number of vector glyphs such as uncertainty glyphs or arrow glyphs. The glyph length or area was scaled based on the magnitude of their data points to show the direction of uncertainty for ocean current and wind dataset. Sanyal et. al. [23] used circular graduated glyphs over a regular grid to visualize ensemble uncertainty in meteorological data. GlyphSea [16] is an interactive application that uses glyphs to encode the orientation of vectors in seismology and astrophysics data. Hlawatsch et al. [9] proposed Flow radar glyphs to visualize time-dependent processes in a static way that show the distribution of uncertainty, directional information and its temporal behavior.

Although our visualization uses vectors to depict the relationships among different spatial locations, our context is different for the following reasons.

- First, the data points in our dataset do not need to be directional.
- Second, the magnitudes associated with the directions are of two types, increasing or decreasing, based on whether the associated location has a higher value than that of its neighbors.
- Third, the directions are not ‘flows’, rather they depict historical relations among different spatial locations.

III. Challenges

The driving question of our research has been whether differential trends can be visualized effectively over the spatial domain using a vector based approach.

Since we are interested in visualizing historical trends over the most recent contour plot, it is natural to first create a contour plot for the most recent timestamp t, and then compute a vector field. Fig. 3 illustrates such a visualization, where the aggregated differential trends (vectors) over all timestamps have been plotted them on the contour plot.

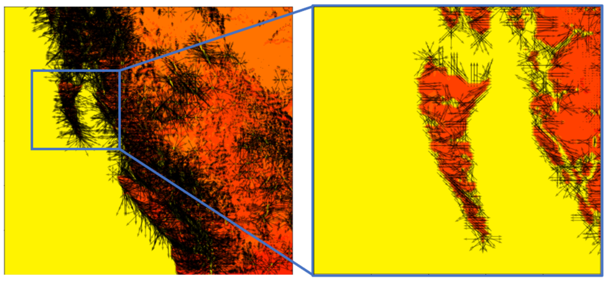


Fig. 3. Direct vector overlay creates a cluttered representation of vectors over the contour map.

One can notice several challenges associated with this initial vector representation:

- The geographic isolines may themselves be wiggly and staggered. Thus the vectors generated may cross themselves. Even if the image is zoomed in to visualize a small 100×100 grid, the vectors are not clearly visible, tangled, and overlapped. This suggests that the vector rendering should be adaptive to the spatial resolution to avoid overlapping.
- The data may be noisy, and thus may create vectors with random directions, which suggests that a reasonably larger neighborhood and timestamp may be chosen.
- The changes may be increasing or decreasing and spanning over the whole region under consideration, and hence appropriate measures need to be taken to reveal a discernible pattern.
- Moreover, slow computation of both contour and vectors poses difficulty to interact with the visualization.

IV. ContourDiff

In this section, we describe the methodology of how the challenges mentioned in Section III have been tackled and discuss the implementation details to produce a ContourDiff visualization. The workflow in Fig. 4 illustrates an overview of various implementation steps.

A. Vector Field Computation

1) Processing the Dataset: Let N be the number of timestamps in the dataset. We preprocess each timestamp, T_1, T_2, ..., T_N to create a primary structure that facilitates the computation of the background contour map and overlaid vectors. For a geospatial variable V (represented as an one dimensional non-negative matrix), we create a tuple (x, y, v), where x and y are the screen coordinates computed from the latitude and longitude information and v is the normalized value: $\frac{(V-\max(V))}{(\max(V)-\min(V))}$, where $max(V)$ and $min(V)$ is respectively the maximum and minimum values calculated from variable V.

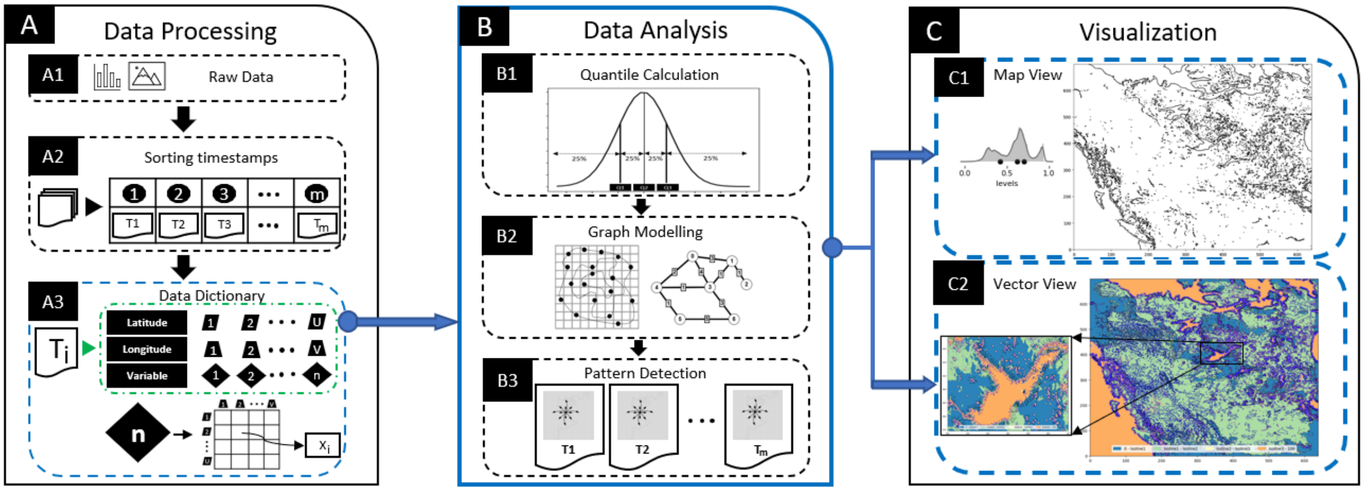


Fig. 4. A complete illustration for the workflow to compute a ContourDiff visualization.

2) Computing the Background Plot: For the background of our visualization, we choose contour plot of the most recent timestamp. A contour map is generated based on the normalized values of a geospatial variable of interest. The reason for using such a contour map is that it provides an intuitive idea for the underlying data distribution. We show the data distribution on the system interface (Fig. 2A), which allows users to interactively choose the desired isoline thresholds. If the user does not want to choose thresholds, then we use lower quartile (25^{th} percentile), median (50^{th} percentile) and upper quartile (75^{th} percentile) as the default isoline thresholds. While one reason to choose the quartile is that it mimics the standard box plot, the other is that it places the contour lines sufficiently far apart so that the vectors don't overlap much the contour lines. After rendering the contour map for the latest timestamp, we overlay the vectors computed from this and all the other timestamps on top of the generated map.

We construct a graph G from the previously generated contour map by extracting information from the contour lines. Each vertex of G can be defined as $[l, p, x, y]$, where x, y are the coordinates of a point on a polygonal path p of isoline l. Two vertices (points) of G are adjacent if they appear consecutively on the polygonal path. We set the weight of an edge of the graph G to be the Euclidean distance between the end points. We remove the isolated vertices and small paths from G to remove visual clutter.

3) Neighbor Selection and Vector Creation: For every spatial location s, we first computed the rate of change $\frac{dv}{ds}$, where v is a function that varies over the spatial domain and ds corresponds to the change in the spatial domain. We then categorized them into increasing and decreasing changes. Finally, we aggregated them separately to form two representative vectors. More specifically, while considering a neighborhood of s, we used k-hop neighbors, where $k \in \{1, 2\}$. For $k \leq 2$ and T timestamps, we obtain $24\dot{T}$ vectors for each data point. We then group the vectors in two groups based on whether s has a smaller value than its neighbors or not. Finally, we aggregate the vectors in these two groups separately to obtain the increasing and decreasing differential trends at s.

4) Speeding Up the Computation: We leveraged the matrix computation libraries to gain computational speed, hence formulated the vector computation process using shifted matrices. To compute the vectors for the 1-hop neighbors, we pad the $M \times N$ matrix with 1 row and 1 column of zeroes and create a $(M + 2) \times (N + 2)$ matrix. We shift the matrix in all eight directions to obtain an $M \times N$ matrix and then subtract the original matrix from the shifted matrix. For 2-hop neighbors, we pad the original matrix with 2 rows and 2 columns of zeroes and create a $(M + 4) \times (N + 4)$ matrix. But we shift this matrix in 16 directions to find $M \times N$ matrix in each direction.

Let $\vec{V_X}(t_i)$ be a vector representing an increasing trend for the data point X at a timestamp t_i. After the shifting and subtraction, we get a matrix of $M \times N$ vectors in 24 directions for X in a timestamp. For each data point, we aggregate all these 24 vectors to compute $\vec{V_X}(t_i)$, i.e.,

$\vec{V_X}(t_i) = \sum_{i=0}^{23} \vec{V_X}(P_i) \times I(P_i)$, where $I(P_i)$ is an indicator variable which is set to 1 for the increasing trend, and 0 otherwise. Similarly, we also compute the decreasing trends.

We compute the aggregated vector $\vec{V_X}$ by aggregating all the 2D vector components for all timestamps t_1, t_2,…,t_n, as follows. $\vec{V_X} = \sum_{i=1}^{n} \vec{V_X}(t_i)$.

Let u_X be the vertex in the weighted graph G that corresponds to the location X. The vector $\vec{V_X}$ represents the differential trends acting on its adjacent edges. Therefore, while calculating the final vector, we scale the magnitude of the vector by the average weight w of the edges adjacent to u_X. Thus the final vector is formulated as $\vec{V} = w\vec{V_X}$.

B. Vector Rendering and Quadtree

Hierarchical data structure [22] and adaptive representation of the spatial information [4] are common ways to deal with large spatial data. Since a large number of vectors correspond

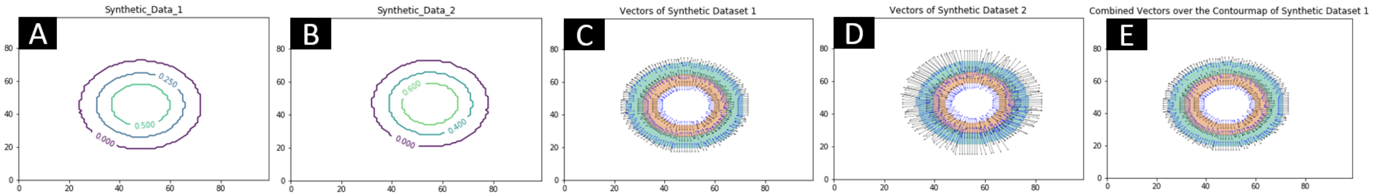


Fig. 5. Illustration for the ContourDiff visualization on a synthetic dataset. Figures 5A indicates design of a synthetic data with contour thresholds 0, 0.25 and 0.5. Figure 5B shows design of similar data at a different timestamp with changed contour thresholds 0, 0.4 and 0.6. Figures 5C and 5D indicates generated vectors of synthetic data 1 and synthetic data 2 respectively. Finally Figure 5E shows vectors generated from both dataset overlapped on top of generated contourmap from dataset 1.

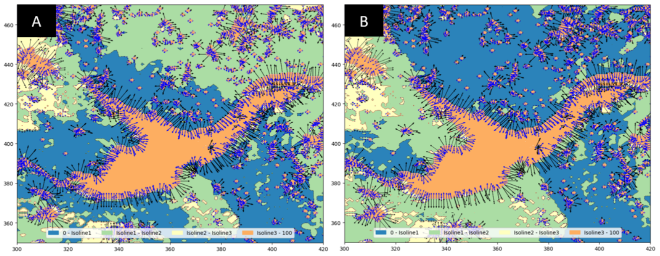


Fig. 6. ContourDiff Visualization for (A) Soil Moisture and (B) Soil Liquid Water over the Grate Slave Lake in Canada.

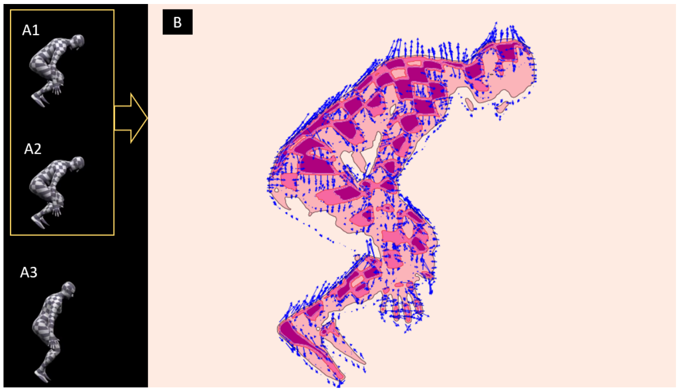


Fig. 7. Detecting small change from timelapse photos of a human motion

to a large rendering time, we maintain the vectors in a *quadtree* data structure [22]. We allow users to choose the leaf size of the quadtree (Fig. 2A). By choosing a lower resolution, users can decrease the number of vectors to be rendered and can achieve fast interaction speed (in a few seconds).

C. Visualization Interface

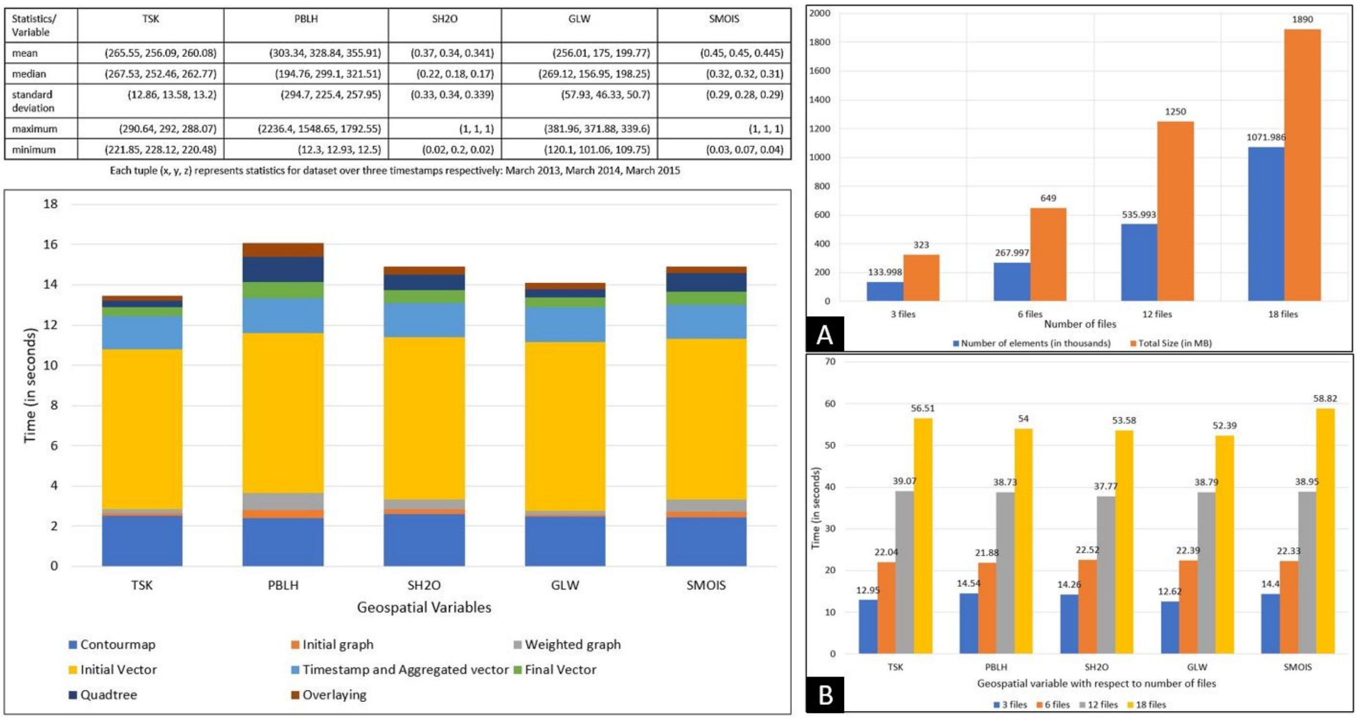
We created a standard graphical interface by following the principal for grouping interactive controls with similar functionalities, to provide users with an intuitive workflow [28]. The contour map is the primary visualization view (Fig. 2A), which is colored using standard perceptual color maps. We design vectors as 'free-flow', i.e., vectors can be presented from 0 to 360 degree angle. The control panel provides options to choose colors for both the contour plots and the vectors. We also provide several direct interaction features such as zoom, pan, change navigation and image-save capabilities (Fig. 2C).

The control panel has been organized according to the task process. The first set of controls (from top-to-bottom) are for selecting dataset, where users can choose thresholds for the contours as necessary. The next set of controls are for controlling vector overlay. One may choose to plot only increasing or decreasing, or both trends. Users can also choose quadtree leaf size to adjust rendering speed. The last set of controls are for choosing colormaps, or scaling the vectors using different types of data mappings (e.g., logarithmic, linear, exponential).

D. Case Studies

Case Study 1 (Trend Validation using Synthetic Data): To validate the system functionality, we examined ContourDiff with different synthetic datasets. Here is an example where we take a two-dimensional standard normal distribution of 40,000 samples to create contour plots for two timestamps, as illustrated in Fig. 5A and Fig. 5B. The corresponding vectors at each timestamps are depicted in Fig. 5C and Fig. 5D, respectively. One can immediately see from the vector sizes that the rate of change along the contour lines is much higher at timestamp-B than that of timestamp-A. Fig. 5E illustrates the differential trends over the contour map of Fig. 5A after aggregating the vectors of Fig. 5A and Fig. 5B. One can see from Fig. 5E that the vectors indicated by blue arrows show the increasing (low to high) changes along the contour lines. The black arrows depict the opposite trend.

Case Study 2 (Geological Trends): In this case we examined the soil moisture and soil liquid water content over Western Canada. Fig. 6A and B illustrate a zoomed in view over the Grate Slave lake area. To validate the differential trend, we worked with a hydrological modeling expert. After familiarizing the expert with ContourDiff using synthetic data, we discussed the real-life dataset. The expert pointed out that the soil moisture is saturated at the water body, which could be the reason that we see major differential trends (large vectors) mainly alongside the water body (Fig. 6A). The expert also pointed out that the decreasing trends (black arrows) are larger towards the land compared to the increasing trends (blue arrows) which are inside the water body. This indicates a sharp drop in soil moisture content as we move away from the water body. He mentioned that such findings may sometimes give valuable information about the hydrological model that generated this soil moisture data, as well as creates



Statistics/ Variable	TSK	PBLH	SH2O	GLW	SMOIS
mean	(265.55, 256.09, 260.08)	(303.34, 328.84, 355.91)	(0.37, 0.34, 0.341)	(256.01, 175, 199.77)	(0.45, 0.45, 0.445)
median	(267.53, 252.46, 262.77)	(194.76, 299.1, 321.51)	(0.22, 0.18, 0.17)	(269.12, 156.95, 198.25)	(0.32, 0.32, 0.31)
standard deviation	(12.86, 13.58, 13.2)	(294.7, 225.4, 257.95)	(0.33, 0.34, 0.339)	(57.93, 46.33, 50.7)	(0.29, 0.28, 0.29)
maximum	(290.64, 292, 288.07)	(2236.4, 1548.65, 1792.55)	(1, 1, 1)	(381.96, 371.88, 339.6)	(1, 1, 1)
minimum	(221.85, 228.12, 220.48)	(12.3, 12.93, 12.5)	(0.02, 0.2, 0.02)	(120.1, 101.06, 109.75)	(0.03, 0.07, 0.04)

Each tuple (x, y, z) represents statistics for dataset over three timestamps respectively: March 2013, March 2014, March 2015

Fig. 8. (left) Time spent for each step of ContourDiff for WRF geospatial variables with three timestamps: March 2013, March 2014, March 2015. (right) Scalability analysis as the number of timestamps grow.

the ability to ask new research questions. Furthermore, he indicated that the soil moisture consists of soil ice and liquid water content. With limited presence of soil ice, we observe a similar differential trend for soil liquid water (Fig. 6B).

Case Study 3 (Detecting Small Change): ContourDiff is able to interpret small changes in image or numeric data. We show this in Fig. 7, using the motion capture images of a human gradually standing up [13]. We primarily use images of Fig. 7A1 and Fig. 7A2 to produce the image Fig. 7B. Image Fig. 7A3 shows the person approaching the final state. Although it is challenging to interpret detailed body movement from almost identical images, ContourDiff visualization could reveal both the significant change (head and shoulder) and an overall understanding of the movement (upward motion).

V. Performance Analysis

The machine used to develop and evaluate ContourDiff has an Intel(R) Core(TM) i7-8700 Processor with 3.2 GHz processing speed with 16 GB RAM and 1920×1080 display resolution. To examine the scalability of the system, we used Weather Research and Forecasting (WRF) model output data [15]. We used different climatic parameters (monthly data from 2013 to 2015): Surface Skin Temperature (TSK), Planetary Boundary Layer Height (PBLH), Soil Liquid Water (SH2O), Ground Long Wave (GLW), Soil Moisture (SMOIS). Each timestamp contained 699× 639 datapoints with a spatial resolution of $4km$ over western Canada.

Fig. 8(left) shows the time taken at different stages of ContourDiff, where computing the vectors is clearly the most time consuming step. For PBLH, the time is highest as the values have the most standard deviation for all 3 datasets. SMOIS and SH2O has similar characteristics, and thus total time is also similar. Fig. 8A illustrates the growth of data (data points and data size) as the number of timestamps (files) grows. Since each file contains the same number of elements, increase of file size linearly increases the number of elements. Fig. 8B illustrates the time spent for each dataset as the number of timestamps grows. We can observe that ContourDiff scales well with the increasing number of timestamps, i.e., it can process over 1 million data points under a minute.

VI. Limitations and Future Work

Since we investigated the feasibility of visualizing differential trends, our effort was primarily invested on identifying the challenges, examining the computational aspects, and developing a general system to explore the potential of ContourDiff in revealing changes pattern in real-life datasets. With the insights gained, it would be interesting to further examine the ContourDiff approach targeting specific application domain with expert evaluation and controlled user studies. It would also be interesting to compare ContourDiff against side-by-side visualization or animation by controlling data variability. Although ContourDiff can process millions of data points in a minute, the vector computation can be done in parallel. Hence we plan to use GPU to leverage its parallel processing capability to speed up computation. Additionally, it would be interesting to conduct a usability study for our interface to find bottlenecks and improve the current system.

VII. Conclusion

We propose ContourDiff, a vector-based visualization approach that overlays vectors over contour map to analyze the trends of change across spatial regions and temporal domain. We also provide an interactive system for the users to explore ContourDiff visualization. We evaluated our system using both synthetic and real-life datasets, which shows the potential of the system to detect small differential trends even in noisy data. While ContourDiff lays out the ground work for visualizing differential trends, it would be interesting to examine how the system can be improved further for various domain specific data analysis. It would also be interesting to investigate our work in the context of deep learning based approaches for change detection [25], [33]. We believe the approach will inspire future research on visualizing differential trends in geospatial data and changes in timelapse images.

Acknowledgment

This work is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), and by two CFREF grants coordinated by GIFS and GIWS.

References

[1] J.-w. Ahn, M. Taieb-Maimon, A. Sopan, C. Plaisant, and B. Shneiderman. Temporal visualization of social network dynamics: Prototypes for nation of neighbors. In *International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction*, pages 309–316. Springer, 2011.

[2] N. Andrienko, G. Andrienko, and P. Gatalsky. Visualization of spatio-temporal information in the internet. In *Proceedings 11th International Workshop on Database and Expert Systems Applications*, pages 577–585. IEEE, 2000.

[3] C. Auer, J. Kasten, A. Kratz, E. Zhang, and I. Hotz. Automatic, tensor-guided illustrative vector field visualization. In *2013 IEEE Pacific Visualization Symposium (PacificVis)*, pages 265–272. IEEE, 2013.

[4] B. L. Cheeseman, U. Günther, K. Gonciarz, M. Susik, and I. F. Sbalzarini. Adaptive particle representation of fluorescence microscopy images. *Nature communications*, 9(1):1–13, 2018.

[5] C. Dempsey. Where is the phrase "80% of data is geographic" from?'. *GIS Lounge*, 28, 2012.

[6] R. Gonzalez and R. Woods. *Digital Image Processing*. Pearson Education, Inc., 2008.

[7] Y. Gu and C. Wang. Transgraph: Hierarchical exploration of transition relationships in time-varying volumetric data. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2015–2024, 2011.

[8] C. G. Healey and J. T. Enns. Large datasets at a glance: Combining textures and colors in scientific visualization. *IEEE transactions on visualization and computer graphics*, 5(2):145–167, 1999.

[9] M. Hlawatsch, P. Leube, W. Nowak, and D. Weiskopf. Flow radar glyphs—static visualization of unsteady flow with uncertainty. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):1949–1958, 2011.

[10] M. Itoh, M. Toyoda, and M. Kitsuregawa. An interactive visualization framework for time-series of web graphs in a 3d environment. In *2010 14th International Conference Information Visualisation*, pages 54–60. IEEE, 2010.

[11] D. Kao, A. Luo, J. L. Dungan, and A. Pang. Visualizing spatially varying distribution data. In *Proceedings Sixth International Conference on Information Visualisation*, pages 219–225. IEEE, 2002.

[12] O. Kersting and J. Döllner. Interactive 3d visualization of vector data in gis. In *Proceedings of the 10th ACM international symposium on Advances in geographic information systems*, pages 107–112. ACM, 2002.

[13] M. M. Loper, N. Mahmood, and M. J. Black. MoSh: Motion and shape capture from sparse markers. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 33(6):220:1–220:13, Nov. 2014.

[14] A. M. MacEachren and M.-J. Kraak. Research challenges in geovisualization. *Cartography and geographic information science*, 28(1):3–12, 2001.

[15] J. Mai, K. C. Kornelsen, B. A. Tolson, V. Fortin, N. Gasset, D. Bouhemhem, D. Schaefer, M. Leahy, F. Anctil, and P. Coulibaly. The canadian surface prediction archive (CaSPAr): A platform to enhance environmental modeling in canada and globally. *Bull. Amer. Meteor. Soc.*, 101:E341–E356, 2020.

[16] E. McQuinn, A. Chourasia, J. P. Schulze, and J.-B. Minster. Glyphsea: visualizing vector fields. In *Visualization and Data Analysis 2014*, volume 9017, page 90170L. International Society for Optics and Photonics, 2014.

[17] L. Nowell, E. Hetzler, and T. Tanasse. Change blindness in information visualization: A case study. In *Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01)*, page 15. IEEE Computer Society, 2001.

[18] E. Pendall. Fast microbes regulate slow soil feedbacks. *Nature Climate Change*, 8(10):859, 2018.

[19] K. Potter, J. Kniss, R. Riesenfeld, and C. R. Johnson. Visualizing summary statistics and uncertainty. In *Computer Graphics Forum*, volume 29, pages 823–832. Wiley Online Library, 2010.

[20] K. Potter, A. Wilson, P.-T. Bremer, D. Williams, C. Doutriaux, V. Pascucci, and C. R. Johnson. Ensemble-vis: A framework for the statistical visualization of ensemble data. In *2009 IEEE International Conference on Data Mining Workshops*, pages 233–240. IEEE, 2009.

[21] J. G. Powers, J. B. Klemp, W. C. Skamarock, C. A. Davis, J. Dudhia, D. O. Gill, J. L. Coen, D. J. Gochis, R. Ahmadov, S. E. Peckham, et al. The weather research and forecasting model: Overview, system efforts, and future directions. *Bulletin of the American Meteorological Society*, 98(8):1717–1737, 2017.

[22] H. Samet. *Foundations of multidimensional and metric data structures*. Morgan Kaufmann, 2006.

[23] J. Sanyal, S. Zhang, J. Dyer, A. Mercer, P. Amburn, and R. Moorhead. Noodles: A tool for visualization of numerical weather model ensemble uncertainty. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1421–1430, 2010.

[24] T. Slocum, S. Yoder, F. Kessler, and R. Sluter. Maptime: software for exploring spatiotemporal data associated with point locations. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 37(1):15–32, 2000.

[25] J. Tasnim and D. Mondal. Data reduction and deep-learning based recovery for geospatial visualization and satellite imagery. In *Proceedings of the IEEE International Conference on Big Data (Big Data)*, pages 5276–5285. IEEE, 2020.

[26] A. Telea and J. J. Van Wijk. Simplified representation of vector fields. In *Proceedings of the conference on Visualization'99: celebrating ten years*, pages 35–42. IEEE Computer Society Press, 1999.

[27] M. Thöny, M. Billeter, and R. Pajarola. Deferred vector map visualization. In *SIGGRAPH ASIA 2016 Symposium on Visualization*, page 16. ACM, 2016.

[28] D. Todorovic. Gestalt principles. *Scholarpedia*, 3(12):5345, 2008.

[29] M. Weber, M. Alexa, and W. Müller. Visualizing time-series on spirals. In *Infovis*, volume 1, pages 7–14, 2001.

[30] S. J. Winterbottom and D. J. Gilvear. A GIS-based approach to mapping probabilities of river bank erosion: regulated river tummel, scotland. *Regulated Rivers: Research & Management: An International Journal Devoted to River Research and Management*, 16(2):127–140, 2000.

[31] C. M. Wittenbrink, A. T. Pang, and S. K. Lodha. Glyphs for visualizing uncertainty in vector fields. *IEEE transactions on Visualization and Computer Graphics*, 2(3):266–279, 1996.

[32] K. Wu and S. Zhang. Visualizing 2d scalar fields with hierarchical topology. In *Proceedings of the IEEE Pacific Visualization Symposium (PacificVis)*, pages 141–145. IEEE, 2015.

[33] M. Yang, L. Jiao, F. Liu, B. Hou, and S. Yang. Transferred deep learning-based change detection in remote sensing images. *IEEE Trans. Geosci. Remote. Sens.*, 57(9):6960–6973, 2019.

[34] G. M. H. Zahan, D. Mondal, and C. Gutwin. Contour line stylization to visualize multivariate information. In *Proceedings of Graphics Interface (GI)*, pages 28:1–28:7, 2021.

[35] X. Zhang and M. Pazner. The icon imagemap technique for multivariate geospatial data visualization: approach and software system. *Cartography and Geographic Information Science*, 31(1):29–41, 2004.