EUROGRAPHICS 2008 / A. Vilanova, A. Telea, G. Scheuermann, and T. Möller
(Guest Editors)

Volume 27 (*2008*), *Number 3*

Concurrent Viewing of Multiple Attribute-Specific Subspaces

Robert Sisneros, C. Ryan Johnson and Jian Huang

University of Tennessee at Knoxville

Abstract

In this work we present a point classification algorithm for multi-variate data. Our method is based on the concept of attribute subspaces, which are derived from a set of user specified attribute target values. Our classification approach enables users to visually distinguish regions of saliency through concurrent viewing of these subspaces in single images. We also allow a user to threshold the data according to a specified distance from attribute target values. Based on the degree of thresholding, the remaining data points are assigned radii of influence that are used for the final coloring. This limits the view to only those points that are most relevant, while maintaining a similar visual context.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Display Algorithms

1. Introduction

It is common for scientific visualization production tools to provide side-by-side images showing various results of significance. This is particularly true for applications involving time-varying datasets with a large number of variables. However, application scientists would often prefer to have these results summarized into the fewest possible images. In this paper, we are interested in developing a general scientific visualization method that addresses this issue. In particular, we summarize multivariate data for previewing, i.e. we provide general information about a dataset that a user has a limited knowledge of.

Our method produces single images that fuse key aspects of multiple attributes of the data. From those images, one can discover prominent inter-variable relationships, even though the renderings are independent among variables. Our approach also produces simple statistical evaluations of the data that can be used to effectively navigate the dataset. For instance, which timesteps are important to examine, and what particular variables in those timesteps deserve special attention.

In the remainder of this paper, we first describe the related work of this research in Section 2. In Section 3, we present our design of the overall system, followed by implementation details in Section 4. Finally, our results and discussion are provided in Sections 5 and 6 respectively, and then concluded in Section 7.

2. Related Works

When dealing with multivariate data, the most straightforward approach is to treat each variable separately. However, that approach is too simplistic and still leaves much to be desired. The visualization research community has undertaken the task of finding more capable methods. In particular, current literature usually falls within two categories: fusing multiple variables into single images [BHW06, KML99, REHL03, HA04], or viewing relationships between variables [KMM*01, STS06]. A great survey of many of such techniques has been presented by Wong and Bergeron in [WB97].

Creating images from many variables is a hard problem, as shown in [Tay02]. It is often impossible without somehow reducing the data of each variable. One way to do this is to focus on only the salient regions of each, namely by feature detection and extraction. If features can be identified and highlighted, it becomes much easier to render features of different variables concurrently. [LPK05, WH01, WBS*05] have presented a number of successful example methods based on feature extraction. A common limitation of these methods, however, is the a priori requirement to have features accurately defined before a visualization can be created. To address this limitation, Jänicke et al. recently developed a region detection method that is both application-independent and that doesn't rely on user input [JWSK07]. Their approach extends local statistical complexity of cellu-

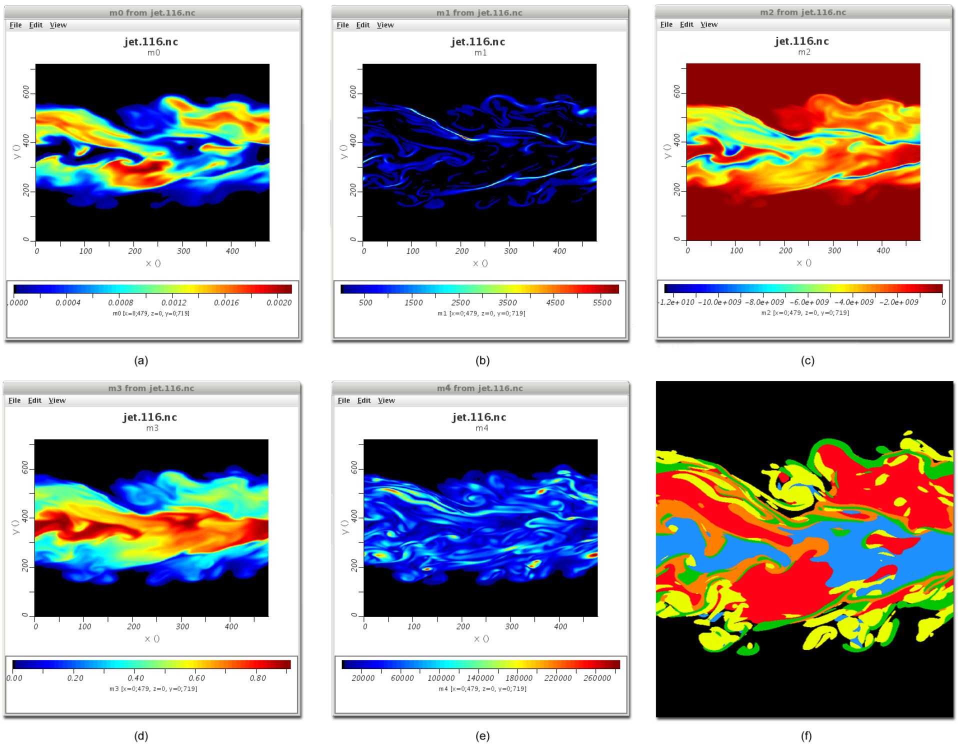


Figure 1: *Images created by ncBrowse (http://www.epic.noaa.gov/java/ncBrowse/) from single variables of the jet combustion dataset (a)-(e). Image created by our method fusing high valued ranges of each of the single variable images (f).*

lar automata to perform region detection in applications that can be described by PDEs.

Woodring and Shen proposed chronovolumes as a technique for displaying time varying volumetric data [WS03]. By treating different timesteps as different variables on the same voxel, their approach also provides some inspiration to multivariate data visualization. Images created by chronovolumes contain a combination of all timesteps of interest, while maintaining a context of surrounding earlier and later timesteps. This is done by adding an integration through time to the raycasting pipeline. They also propose a similar coloring scheme in which they simply assign each timestep its own color. They suggest 20 as a limit on the number of discernible colors. This is an existing example that demonstrates the power of attribute-specific subspaces. However, in this case, it is specific to the dimension of time.

Multivariate visualization is a unique area due to the large number of variables. This is compounded when timesteps are considered as independent variables and the combinatorial space to explore is exponentially large. This special research need of multivariate visualization calls for methods to provide summarizing previews of data. From this respect, our method is based on logical operators that can be combined in a customizable manner. These basic operators provide a way to visually gauge relative strength of variables, and hence guide a user's attention to a much reduced subset of an otherwise large and incomprehensible multivariate dataset. In [dSB04], dos Santos and Brodlie introduce a conceptually similar approach to handle multivariate and multidimensional data. In their work, a filtering, or subsetting, operation is used to select a subset of all variables and dimensions to create a smaller space from the original high dimensional space. Woodring and Shen further extended the idea to include set operations [WS06]. A set can be specific in to any variable as a boolean relationship. Sets can be combined using set operations in an effort to select voxels of interest to the user.

In both of these representative works, selection choices are decided and applied uniformly for the entire dataset. Our work is different in that the variable or attribute we show is eventually a per point selection. In addition, we do not consider spatial dimension for subsetting. In other words, the subspaces we create are dependent on both attributes and points.

Finally we must note that many pioneering researchers have studied how to provide succinct and informative visualization of multivariate data through non-photorealistic and illustrative approaches. There, the focus of visualization research has been to develop innovative visual cues inspired by artistic as well as real world domains [HE02]. Common technical approaches in the domain of non-photorealistic rendering include texture synthesis [LEQ*07, WBS*05], glyphs [HE02] and stippling effects [LMT*03], etc. However, in those works, the foci are features that are well understood (including the definition of the features and the most effective visual analogies for those features). The starting point of our work is different in that our focus is to develop basic operators that can be used to define attribute-specific subspaces and to demonstrate their usefulness. Current sophisticated and powerful illustrative rendering techniques can directly leverage our definition of attribute-specific subspaces for better results.

3. The Approach

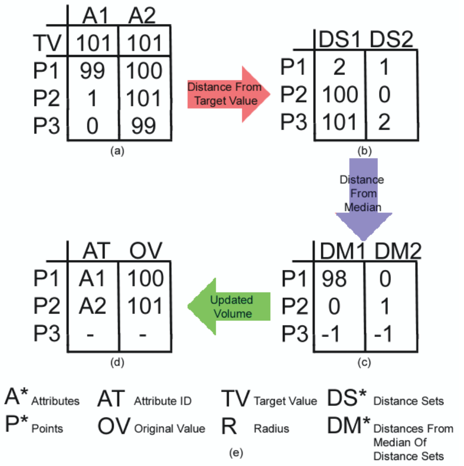
Our prototype system is named massSIV (multiple attribute specific subspace visualization). We intend for massSIV to be generally applicable, however, it is currently designed primarily for scientific simulations that produce volume data. We take as input a general dataset that has, at each location, an associated attribute set. Each attribute must have a value at each location (point). Also, the user supplies a set of attribute target values that allow us to determine for any value at any location, whether or not that value is considered a "good" one, i.e. close to its target value. Our goal is to then provide a rendering of the dataset that is colored based on all attributes. We first split the colorspace, to associate with each attribute its own color. Then, for every location, we select a single value, the one closest to its attribute target value. However, if no values are close to their target values, we may threshold that point. We then redistribute the open spaces created by thresholding to the remaining points, giving each an area of the final space that it may color. After a coloring is chosen, any set of points with the same color is an attribute subspace. The selection of a volume's attribute set is arbitrary, therefore there may be a wide variation, and hence flexibility, in the combination of attribute subspaces.

3.1. Attribute Set Selection

A typical time-varying dataset contains a set of variables associated with each point. The obvious choice for an attribute set is to simply use the variables already associated with each point, and render a single timestep. Reduction of the attributes allows us to see salient regions for all variables at once. We can also focus the attribute set on the timesteps of a dataset. Here, the attribute set for each point would be a single variable's value across many timesteps. In this case, we show, for a variable, in which timesteps interesting features arise. If interest lies with a certain variable, this could be used to help determine which timesteps would be most valuable in that variable's evaluation. For this reason, our program outputs all attributes ranked by the percentage of the image they color. Similarly, we could reduce the spatial dimensionality of a dataset. The attribute set would be, for a 2-D slice of the volume, a certain variable's value across the remaining dimension. The resulting rankings could then be used to determine a slice of the volume to focus on. In the following sections we assume the attribute set is a point's variables.

3.2. Creating Attribute Subspaces

Now that the dataset has an attribute set, i.e. each point has associated with it a set of values, we create attribute subspaces. Each value at every point corresponds to a value of one of the attributes in the attribute set. As stated, we select a single value that is closest to its attribute target value. However, this is not relative to values of a point, but to all the values of an attribute. We do this by calculating for each attribute, the set of distances from each value to that attribute's target value. Then, for every value of a point, we calculate how far away from the median of the distance set that value is. We then select a point's single value to be the one from the initial set of values that is both close to the target value, and farthest from the median. By doing this, we ensure that a value is "good" relative to the entire set values of an attribute, i.e. we say that value has a high impact on an attribute at that point.



	A1	A2
TV	101	101
P1	99	100
P2	1	101
P3	0	99

(a)

	DS1	DS2
P1	2	1
P2	100	0
P3	101	2

(b)

	DM1	DM2
P1	98	0
P2	0	1
P3	-1	-1

(c)

	AT	OV
P1	A1	100
P2	A2	101
P3	-	-

(d)

Figure 2: *(a) Simple 3 point dataset. (b) The distance from target values (distance set) for each attribute. (c) The distance from median of distance set: notice P1's second value is closer to its target value, but its first value is chosen because it is much closer relative to how close the rest of the points' first values are to A1's target value. (d) The new volume, the starting point for the redistribution of open spaces. (e) Key for Figures 2 and 3.*

Along with a point's single value, we also store which attribute that value came from, in the form of an integer attribute ID. The final dataset we create will contain only such attribute IDs, however, based on neighboring points' values, a point's current attribute ID may change. We call the set of points colored by a specific attribute an attribute subspace of the volume. We realize that often it is impossible to define a set of target values since it is unknown which values are interesting. If no target values are specified we take the highest values for each attribute to be the target values. We also allow for average values to be used for the target values, in the case that the high values of attributes are outliers. Figure 2 details a simple example of creating attribute subspaces.

3.3. Thresholding

We consider salient regions to be those in which attribute values are near attribute target values. For this reason, thresholding in massSIV is not done by removing points below or above a certain value, but by removing points that contain no values that are near their attribute target values. A high level of thresholding then focuses on only those values closest to the attribute target values, and therefore the most salient regions. The user may specify a distance from the target values to use as a threshold. By default, we threshold all points that contain no values that are closer than the median of the distance set from the attribute target values. In Figure 2 P3 is thresholded.

3.4. Redistribution of Open Spaces

Although thresholding highlights the points closest to attribute target values, we would also like to maintain a regional context in the case of few remaining points. To achieve this, we redistribute the open space created by thresholding to the remaining points. We give each remaining point its own radius of influence. Firstly, for each attribute we calculate an attribute volume percentage. This is the number of remaining points associated with that attribute divided by the total number of remaining points. We then normalize the values for each attribute, making each point, P, the value that corresponds to a percentage of its attribute, $norm(P)$. A point's volume, volume(P), is then that point's percentage of its attribute's volume percentage. To find a point's radius, we simply set a point's volume equal to the volume of a sphere and solve for the radius:

$$r \;=\; \sqrt[3]{\frac{3 * volume(P)}{4\pi}} \tag{1}$$

We denote the radius of a P as $radius(P)$. An example of radii calculations is provided in Figure 3.

3.5. Finalizing the Attribute Subspaces

To maintain a sense of spatial locality, we have a point's influence over another point dissipate within it's area of influence, the farther away the two points are. We achieve this with a simple linear function. For two points, P_1 and P_2, $dist(P_1,P_2)$ is the euclidean distance between those points. For a point P_2 that is within P_1's area of influence, P_1's influence on P_2 is then:

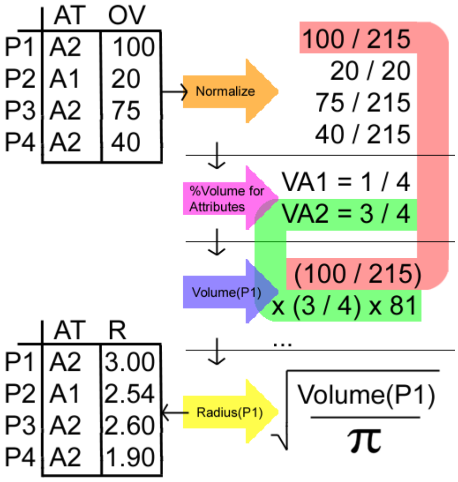


Figure 3: *Detailed calculation of P1's radius of influence. In this example, there are only 4 points remaining after thresholding a a 2-D 9x9 dataset (81 is the total area of the space). Note: Equation 1 is for radius calculation in 3-D.*

$$f(P_1,P_2) = norm(P_1)\left(1 - \frac{dist(P_1,P_2)}{radius(P_1)}\right) \tag{2}$$

This value is calculated between each point, P, and all other points that contain P within their radii. P then receives the value of the point containing P with the maximum value of Equation 2. The final output of our program is a spatial dataset where each location is reduced to a single integer tag that corresponds to one attribute, and therefore one color. In the case that a point falls within no other points' radii of influence, that point is assigned the value zero, which in our rendering is represented by black (2-D): the background color, or clear color (3-D): fully transparent. This dataset is the completed set of attribute subspaces.

4. The Renderer

2-D images created from our resulting datasets are simple bitmaps created from a 2-dimensional slice of the volume, with each point colored by its attribute color. To visualize the volumes resulting from our method, we have implemented an hardware-accelerated volume renderer for interacting with tag volumes. The unordered and discrete nature of tags makes it impossible to use traditional linear interpolation for determining a sample's tag, as it may introduce false intermediate tags between segments. Nearest neighbor interpolation, on the other hand, produces correct results but the voxel resolution is immediately apparent and can detract from the visualization.

Most previous approaches to improved boundary filtering of tagged volumes rely on several passes [TSH98] or a maximum of four tags [FHN*05] for correct interpolation. To avoid these limits, we have developed an interpolation scheme based on a sample's proximity to a gradient-aligned neighbor. We evaluate this proximity using the sample's linearly interpolated tag to choose between the sample's nearest tag and the gradient neighbor's nearest tag. This method requires two textures to be sent to the GPU. The first texture contains only the tags and uses a nearest neighbor filter. The second texture contains both the tag gradients and the tags, using a linear filter. For the gradient calculation, we found central differencing to produce gradients of poor quality across tag boundaries [FHN*05]. Instead, we compute the tag gradients using linear regression over a 3x3x3 neighborhood [NCKG00].

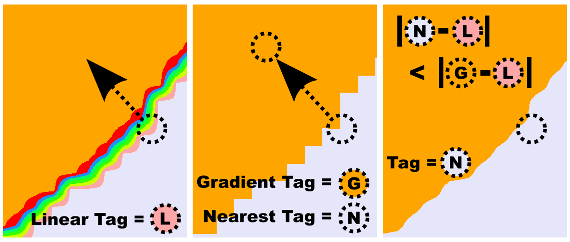
For the rendering, view-aligned slices are rendered in back-to-front order, and a fragment program is invoked for each fragment across a slice. To determine a sample's tag t at subvoxel resolution, we use the gradient to locate the gradient-aligned neighbor, i.e., the nearest neighboring voxel to the sample along the gradient direction. In most instances, the value of the tag around the sample will range from the nearest tag t_n to the gradient tag t_g, with a smooth boundary lying halfway between t_n and t_g. We use the linear tag t_l to locate this halfway point. To compute t smoothly, then, we examine the difference between t_l and the other two tags and set t to the closer. Figure 4 illustrates an example where t_n is closer to t_l. As shown in the following equation, if t_l is closer to the nearest tag, $t = t_n$. Otherwise $t = t_g$.

$$t = \begin{cases} t_n & : \quad |t_n - t_l| < |t_g - t_l| \\ t_g & : \quad otherwise \end{cases}$$

The computed tag then indexes into a colormap to retrieve the sample's color and opacity. A simple interface allows users to alter each tag's color and opacity. With a volume size of 120x480x512, a 512x512 viewport, and a sampling rate of 1.0, we achieve a frame rate of 8-11 fps on a Dell Precision 470 workstation using an NVIDIA Quadro FX 3450 GPU.

5. Results

We reduce multiple images, each representing a data attribute, into a single image, while maintaining discernible features. However, the root of our approach is not in highlighting the most salient regions of the dataset, but to do so at every point. Just as single variable renderings are independent of each other, so are attribute subspaces. However, through thresholding, we do focus on the areas of the volume where attributes have a high correlation to the attribute target values. In the extreme thresholding case, we show only the single points that correspond to the values closest to the attribute target values. Thresholding does highlight regions of interest in a more global sense. Using a point's radius of influence, we also accentuate the area of the volume likley containing events surrounding those highest impact points. We show results from three datasets, the first is a 480x720x120 jet combustion dataset containing five variables, we use only the last timestep (timestep 116). The second, a 256x128 climate dataset containing 7 variables, again we only use one timestep (corresponds to the year 2000). Lastly, a 128x64 climate dataset containing one variable: a CO_2 measurement taken hourly over several years, we use the first hour of Jan. 1 for ten years, 1890-1899.



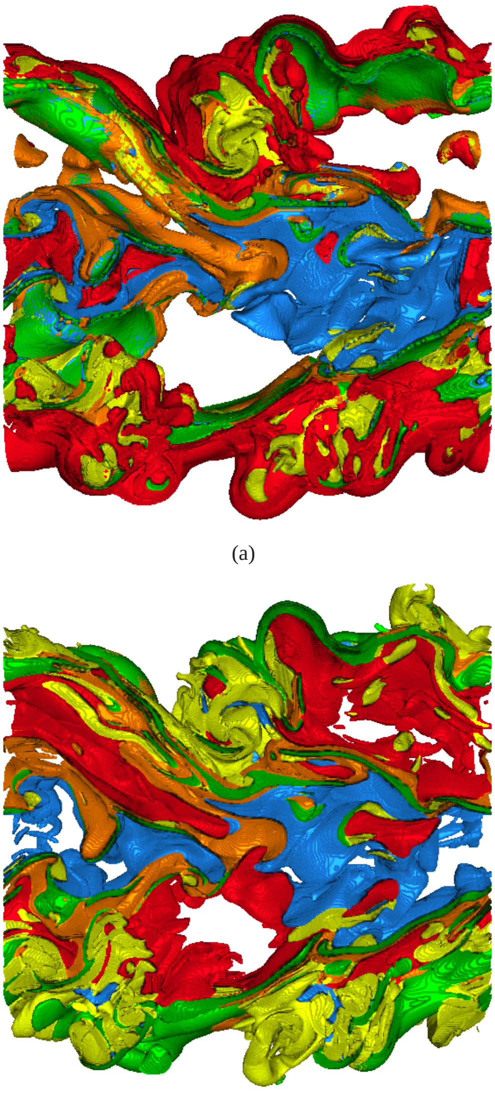
(a) Linear Filter (b) Nearest Filter (c) Final Tag

Figure 4: *To avoid the inaccuracy of linear filtering and the coarse appearance of nearest filtering, tags are interpolated at subvoxel precision with a hybrid filtering approach. (a) The tag and gradient are retrieved using linear interpolation for the encircled fragment. (b) The tag is retrieved for the same fragment using nearest filtering. Additionally, the nearest tag along the gradient is retrieved. (c) The final filtered tag is determined by using the linear tag to find the closer of the gradient's and fragment's nearest tags. In this case, the nearest tag is chosen.*

5.1. Concurrent Views

The typical use of our program is to provide a single image showing salient regions of multiple attributes as apposed to showing multiple side-by-side images. In Figure 1 we show the single variable renderings, of a single slice of the volume, of the jet combustion dataset along with the single image resulting from our approach corresponding to the same slice. We did not change the natural coloring of ncBrowse, and in those images, the points with high values are colored in shades of red that clearly stand out. For this reason, we ran our program with the variables' highest values used as the attribute target values. The regions of high values in each single variable image correspond to one of the five colors in the image resulting from our approach.

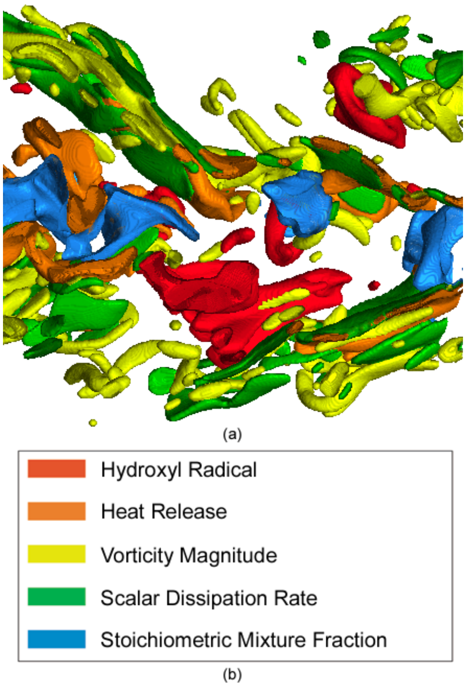
Also from the jet combustion dataset, we show 3-D renderings of the dataset that is the output from our program. In Figure 5, we show the results of our program from using both maximum values as attribute target values and average



(a)

(b)

Figure 5: *Rendering of the jet combustion dataset with the attributes' average values (a) and attributes' maximum values (b) used for the attribute target values. Attribute color key included in Figure 6*



(a)

(b)

Figure 6: *(a)Rendering of the jet combustion dataset with attributes' maximum values used for attribute target values, and all but 4.7% of points thresholded. (b) Attribute coloring key for combustion dataset images.*

variable values as the attribute target values. Figure 5(b) is simply the 3-D version of Figure 1(f), and therefore a summary of 5 3-D single variable renderings.

5.2. Highlighting Salient Regions

We present two examples of using thresholding to highlight salient regions. In Figure 6, the output from our program ran on the jet combustion dataset is colored by only 4.7% of the number of points in the initial dataset. The radii of influence highlight regions associated with the points not thresholded, which is especially important for those small regions. This dataset was also created using the attributes' maximum values as the attribute target values.

In Figure 7 we show our program's default output on the 7-variable climate dataset (2-D). To clearly illustrate the thresholding/area of influence relationship, we present the second image of Figure 7. This image has the maximum level of thresholding, all but 5 points are thresholded. In this image we highlight the areas surrounding those points, this is an indication of the general areas that are likely to be of interest for those attributes.

Figure 7 is a representative example of the utility of mass-SIV. Specifically, this visualization shows the most extreme variable among the 7 variables on each geographical location. The target audience for such a viewing are those who do not already have an in-depth understanding of the problem domain; our image illustrates the following information. The black regions represent places where all 7 variables are not sufficiently close to the target values, and have been thresholded. Examples of such regions include the flat lands in Africa, north-east of China, central regions in U.S. and parts of Russia. The specific humidity is high over the trop-

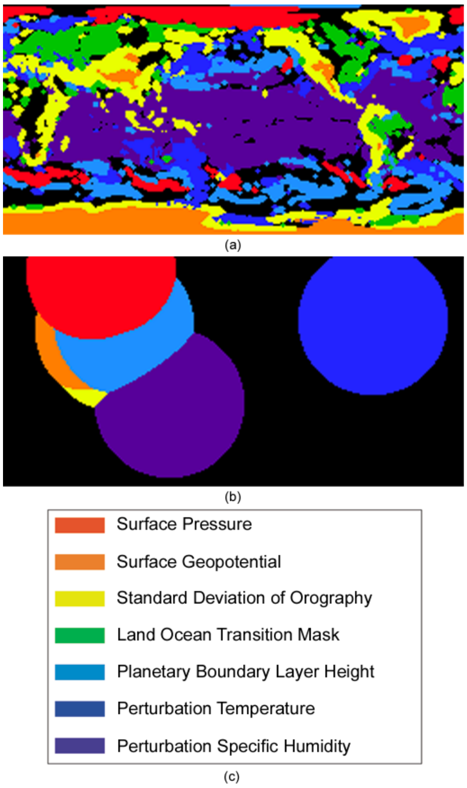
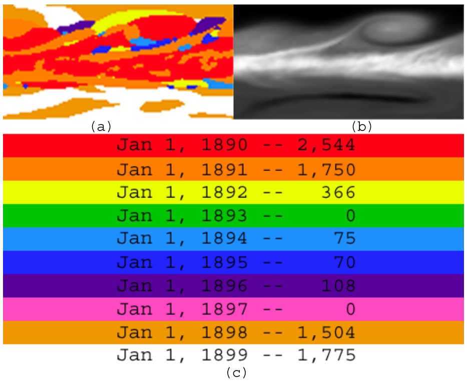

(a)
(b)
🟥 Surface Pressure
🟧 Surface Geopotential
🟨 Standard Deviation of Orography
🟩 Land Ocean Transition Mask
🟦 Planetary Boundary Layer Height
🟦 Perturbation Temperature
🟪 Perturbation Specific Humidity
(c)

Figure 7: *Climate dataset with maximum attribute target values. Default (a) and full (b) thresholding. (c) Attribute coloring key.*

ical oceans because the humidity is high in these warm regions over water. Temperatures are high in the South-Pacific (South-Pacific warm pool) and off the coast of Europe due to the Meridional Overturning Circulation. PBL (planetary boundary layer) height is low only over the oceans where other factors are not dominant. Stddev of orography is high in mountainous regions, often forming a "bulls eye" around surface geopotential, where the altitude is high enough to dominate the stddev of orography. Surface pressure is low in the Arctic and high in the southern oceans due to wind circulation patterns. We can see the height of ice cap on Antarctica delineated by the abrupt "jump" in surface geopotential. The same can be seen around Greenland. In other locations around the globe, only geographic regions such as Tibet, American Rockies and Andes show such abnormal surface geopotential due to their great altitudes.

5.3. Narrowing Exploration Space

Lastly, we give an example of how massSIV can be used for dataset exploration. In Figure 8 we show the output of our method run on a CO_2 measurement climate dataset over ten timesteps. This is a single variable dataset, so each color represents the values at a different timestep. Our program outputs the size of each attribute space, which suggests which timesteps contain the most impactful points. We also show the gray-scale image corresponding to the largest attribute subspace (red). This information can be used to determine which timestep a user should render, via a favorite single variable renderer.



```
Jan 1, 1890 -- 2,544
Jan 1, 1891 -- 1,750
Jan 1, 1892 --   366
Jan 1, 1893 --     0
Jan 1, 1894 --    75
Jan 1, 1895 --    70
Jan 1, 1896 --   108
Jan 1, 1897 --     0
Jan 1, 1898 -- 1,504
Jan 1, 1899 -- 1,775
```

Figure 8: *Concurrent rendering of 10 timesteps of the CO_2 measurement dataset (a). A sample gray-scale image of Jan. 1, 1890, the highest ranked attribute (b). The rankings of all 10 timesteps (c).*

6. Discussion

To provide single summarizing images/volumes we employ a data reduction in attribute space. As with any approach utilizing simplification or compression, there are many concerns as to whether or not the results are oversimplified. To this end, our intention is for massSIV to be used in an application driven manner, so that the simplification can aid users in excluding areas not of interest.

Another issue is the sensitivity of target values. In our admittedly limited range of tested target values, we had few problems with sensitivity. However, there are cases in which this could be an issue, the following discussion can be used as a guide for appropriate uses of our approach. In cases of simple attributes, like the climate dataset's land ocean transition mask (1 for land, 0 for ocean), even changes in other attributes' target values could drastically alter its representing attribute subspace.

There are other cases of target value sensitivity, but they are relatively predictable. For instance, an attribute with a large variance would certainly differ (possibly greatly) with changes in target values. The inverse of this is shown in Figure 5, stoichiometric mixture fraction only slightly changes between its average and maximum value, i.e. it has low standard deviation. Also, although we don't provide renderings based on attribute relationships, attribute relationships could cause target value sensitivity. Areas where attributes are closely related represent possible areas where target values would be sensitive. However, this should only be an issue if those attributes are proportional throughout the volume, i.e. the attributes would have to be similarly close to their distance sets.

7. Conclusion and Future Work

In order for a user to handle a problem space of overwhelming complexity, a concise but accurate big picture is of critical importance. Many subsequent analysis tasks depend on this capability. Unfortunately, prevalent current practices often lack generality and rely on past experiences. This is indeed a hard problem of data analytics that calls for much concerted research efforts. Our approach is a simple first step.

Our method can be regarded as a point-based classification approach. The added components include data filtering, radii of influence and statistical evaluations. Through these, we have achieved interesting results. Concepts central in our approach are: allowing abstraction of the data space by not constraining the set of values associated with each data point, data reduction by selecting a single value from that set to represent each point and the redistribution of open space resulting from data reduction. By integrating these concepts into a system, we have developed a straightforward way to provide summarizing views of multivariate datasets.

The interesting results we have achieved using this specific instance of the implementation leads us to believe that this is a promising domain for future research. In particular, we intend to augment our approach to display areas of attribute relationships. Also, our color selection is entirely arbitrary, a direction for research is a comprehensive study of perceptually salient color selection.

8. Acknowledgements

The authors gratefully acknowledge the primary source of funding for this work, provided through the Institute of Ultra-Scale Visualization (www.ultravis.org), under the auspices of the DOE Scientific Discovery through Advanced Computing (SciDAC) program. This work was also supported in part by an NSF grant CNS-0437508 and a DOE Early Career PI grant awarded to Jian Huang (grant no. DOE DE-FG02-04ER25610). We are indebted to the anonymous reviewers for their insightful suggestions. Wesley Kendall has been instrumental in helping us to create all illustrations used in the paper. We would also like to thank Forrest Hoffman of the Oak Ridge National Laboratory for the many in-depth discussions that helped us to evaluate our visualization results and improve our system.

References

[BHW06] BAIR A., HOUSE D. H., WARE C.: Texturing of layered surfaces for optimal viewing. *IEEE Trans. Vis. Comput. Graph. 12*, 5 (2006), 1125–1132.

[dSB04] DOS SANTOS S., BRODLIE K.: Gaining understanding of multivariate and multidimensional data through visualization. *Computers and Graphics 28*, 3 (2004), 311–325.

[FHN*05] F.VEGA, HASTREITER P., NARAGHI R., FAHLBUSCH R., GREINER G.: Smooth volume rendering of labeled medical data on consumer graphics hardware. In *Proceedings of SPIE Medical Imaging 2005* (2005).

[HA04] HELGELAND A., ANDREASSEN O.: Visualization of vector fields using seed lic and volume rendering. *IEEE Trans. Vis. Comput. Graph. 10*, 6 (2004), 673–682.

[HE02] HEALEY C. G., ENNS J. T.: Perception and Painting: A Search for Effective, Engaging Visualizations. *IEEE Computer Graphics and Applications 22*, 2 (mar/apr 2002), 10–15.

[JWSK07] JÄNICKE H., WIEBEL A., SCHEUERMANN G., KOLLMANN W.: Multifield visualization using local statistical complexity. *IEEE Trans. Vis. Comput. Graph. 13*, 6 (2007), 1384–1391.

[KML99] KIRBY R. M., MARMANIS H., LAIDLAW D. H.: Visualizing multivalued data from 2D incompressible flows using concepts from painting. In *IEEE Visualization '99* (San Francisco, 1999), Ebert D., Gross M., Hamann B., (Eds.), pp. 333–340.

[KMM*01] KNISS J., MCCORMICK P., MCPHERSON A., AHRENS J., PAINTER J., KEAHEY A., HANSEN C.: Interactive texture-based volume rendering for large data sets. *IEEE Computer Graphics and Applications 21*, 4 (/2001), 52–61.

[LEQ*07] LU A., EBERT D. S., QIAO W., KRAUS M., MORA B.: Volume illustration using wang cubes. *ACM Trans. Graph. 26*, 2 (2007).

[LMT*03] LU A., MORRIS C. J., TAYLOR J., EBERT D. S., HANSEN C., RHEINGANS P., HARTNER M.: Illustrative interactive stipple rendering. *IEEE Transactions on Visualization and Computer Graphics 9*, 2 (2003), 127 – 138.

[LPK05] LOVE A. L., PANG A., KAO D. L.: Visualizing spatial multivalue data. *IEEE Computer Graphics and Applications 25*, 3 (May/June 2005), 69–79.

[NCKG00] NEUMANN L., CSÉBFALVI B., KÖNIG A., GRÖLLER E.: Gradient estimation in volume data using 4D linear regression. In *Computer Graphics Forum (Eurographics 2000)* (2000), Gross M., Hopgood F. R. A., (Eds.), vol. 19, pp. 351–358.

[REHL03] RILEY K., EBERT D., HANSEN C., LEVIT J.: Visually accurate multi-field weather visualization. In *IEEE Visualization '03* (2003), pp. 279 – 286.

[STS06] SAUBER N., THEISEL H., SEIDEL H.-P.: Multifield-graphs: An approach to visualizing correlations in multifield scalar data. *IEEE Trans. Vis. Comput. Graph. 12*, 5 (2006), 917–924.

[Tay02] TAYLOR R.: Visualizing multiple fields on the same surface. *IEEE Computer Graphics and Applications 22*, 3 (2002), 6–10.

[TSH98] TIEDE U., SCHIEMANN T., HÖHNE K. H.: High quality rendering of attributed volume data. In *IEEE Visualization '98* (1998), Ebert D., Hagen H., Rushmeier H., (Eds.), pp. 255–262.

[WB97] WONG P., BERGERON R.: 30 years of multidimensional multivariate visualization. In *Scientific Visualization - Overviews, Methodologies and Techniques* (1997), pp. 3 – 33.

[WBS*05] WEILER M., BOTCHEN R. P., STEGMAIER S., ERTL T., HUANG J., JANG Y., EBERT D. S., GAITHER K. P.: Hardware-assisted feature analysis and visualization of procedurally encoded multifield volumetric data. *IEEE Computer Graphics and Applications 25*, 5 (2005), 72–81.

[WH01] WALTER J. D., HEALEY C. G.: Attribute preserving dataset simplification. In *IEEE Visualization 2001* (Oct. 2001), pp. 113–120.

[WS03] WOODRING J., SHEN H.: Chronovolumes: A direct rendering technique for visualization time-varying data. In *Eurographics/IEEE TVCG Workshop on Volume Graphics* (2003), pp. 27 – 34.

[WS06] WOODRING J., SHEN H.-W.: Multi-variate, time varying and comparative visualization with contextual cues. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (2006), 909–916.