INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT (IJSREM)
VOLUME: 07 ISSUE: 12 | DECEMBER - 2023 SJIF RATING: 8.176 ISSN: 2582-3930

Pixel Visualizer

Md Saif Ahmad
Department of Computer Science and Engineering
Chandigarh University
Mohali, Punjab, India
asifayan8581@gmail.com

Sunny Kumar
Department of Computer Science and Engineering
Chandigarh University
Mohali, Punjab, India
sunnykr21129@gmail.com

Manmohan Kumar
Department of Computer Science and Engineering
Chandigarh University
Mohali, Punjab, India
manmohan02304@gmail.com

Akshit Chauhan
Department of Computer Science and Engineering
Chandigarh University
Mohali, Punjab, India
chauhanakshit2002@gmail.com

Vansh Sadana
Department of Computer Science and Engineering
Chandigarh University
Mohali, Punjab, India
vanshsadana9822@gmail.com

Er. Deepika Sharma
Department of Computer Science and Engineering
Chandigarh University
Mohali, Punjab, India
deepika.e15915@cumail.in

***Abstract*—Pixel-based visualization techniques have gained recognition because of their ability to provide huge records even as they provide specified insights. However, their effectiveness is restricted while coping with sparse datasets and random data distribution. Single pixels, whether or not remoted in vacant regions or amidst a complex array of differently coloured pixels, maybe without difficulty overlooked. Additionally, in the elaborate information, tendencies and memorable styles can be concealed. This paper explores diverse procedures for reinforcing pixel-primarily based visualizations, inclusive of strategies like halos, history colouring, distortion, and hatching. The assessment considers their efficacy in improving person pixels, figuring out traits, and highlighting extensive styles. Examples from diverse domains, specifically record evaluation, genome analysis, and geospatial evaluation, exhibit the wide applicability of these strategies and the pointers derived from the evaluation.**

Keywords— detailed insights, sparse datasets, vacant regions, trends, notable patterns, enhancing techniques, halos, hatching, efficacy, genome analysis, applicability.

I. INTRODUCTION

The process of classification involves categorizing a digital image by assigning each pixel to one of several land cover classes or themes. This categorization is utilized to create thematic maps representing the current land cover in an image, as suggested by Lillesand and Kiefer in 1994. Typically, multispectral data is employed for classification, with the spectral patterns present in the data for each pixel forming the basis for numerical classification. The primary goal of image classification is to identify and represent the features in an image, such as unique grey levels or colours, in terms of the type or objects of land cover they represent on the ground. Image classification is a crucial aspect of digital image analysis[1]. While having an aesthetically pleasing image with a variety of colours depicting different terrain features is enjoyable, it is rendered ineffective without understanding the meaning behind the colours, as stated by PCI in 1997.

In the realm of digital imaging, dealing with pixel saturation is a common challenge. Let's consider the scenario of an 8-bit quantized red–green–blue (RGB) colour camera. A pixel value for the R, G, or B channel is considered saturated when it reaches its maximum value, which is 255 in this case. When pixels become saturated, crucial information about the scene is lost. Additionally, if not managed meticulously, saturated pixels can result in image artefacts[2]. This issue is especially critical in colour imaging, where saturation in one colour channel alters the relative RGB values. Such situations are likely to arise when the illuminant exhibits a pronounced colour cast or when the camera design incorporates significant variations in gain across different colour channels. Notably, the saturation of responses to image highlights, often achromatic, can be particularly conspicuous.

II. METHODS OF IMAGE CLASSIFICATION

A. Supervised Method

In supervised image classification, we identify classes of interest, referred to as "training sites," within the chosen image. The image processing software is then employed to establish the statistical and geometrical classification of reflectance for each information class. This stage is commonly known as the "signature analysis of an image," involving the development of characterizations, ranging from simple measures like the mean or range of reflectance in each band composite to comprehensive analyses of covariance, variances, and means across all bands. Once the geometric and statistical characterizations are obtained for each information class, the image is classified by assessing the reflectance for each pixel value and determining which signatures it most closely resembles.

B. Unsupervised Method

Unsupervised classification is an approach that involves analyzing a large number of unknown data and subsequently categorizing them into classes based on inherent groupings in the image values. This method, known as Unsupervised Classification, differs from supervised classification in that it doesn't require analyst-specified training data. The key principle is that values corresponding to a particular cover type should exhibit proximity in the measurement space, while data from distinct classes should be reasonably well separated.

III. PROPOSED METHODOLOGY

The proposed method for image classification relies on comparing pixel values between two images by examining the difference for each corresponding pixel. If the difference for each pixel is zero, the images are considered identical; otherwise[3], they are deemed different. In cases involving more than two images, the approach extends to checking pixel value differences for each image[4]. Utilizing these pixel values, image classification into different classes can be achieved. Images sharing the same pixel values are grouped into the same class, while others are assigned to different classes. If the pixel value differences are equal or close to the pixel values of existing classes, the images are classified accordingly. This method specifically applies to grayscale images, where colours are represented as shades of grey. It is crucial to differentiate grayscale images from colour images due to the lower information requirements for each pixel in grayscale images.

IV. ALGORITHM EVALUATION

The visual properties discussed are evaluated based on their effectiveness in enhancing pixel-based visualizations. It is important to note that certain techniques, such as distortion or hatching, are significantly influenced by the pixel size and may not be suitable for very small pixels.

A. Multidimensional Visualization Techniques

The visualization of data with inherent two- or three-dimensional semantics predates the use of computers for creating visualizations. Tufte, in his widely recognized books [5], [6], illustrates numerous examples of visualization techniques employed over many years. With the advent of computer technology in visualization, new techniques have emerged, and existing ones have been expanded to accommodate larger datasets and interactive displays. However, for much of the data stored in databases, there isn't a standard mapping into the Cartesian coordinate system due to the lack of inherent two- or three-dimensional semantics.

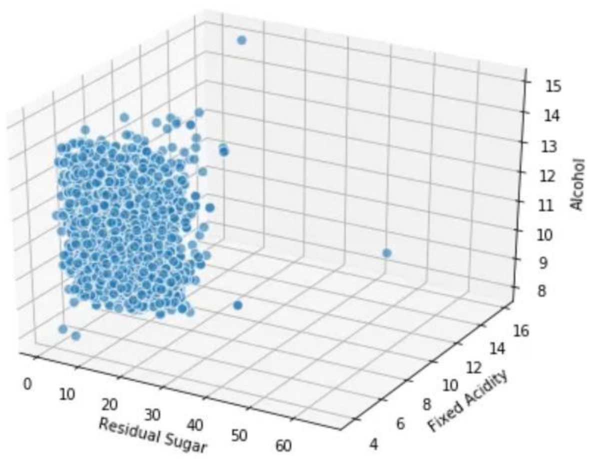

Fig. 1. Multidimensional Visualisation Technique.

Relational databases and their structures are generally viewed as multidimensional data sets, with the attributes of the database serving as dimensions[7]. The classification of techniques for visualizing multidimensional data sets involves three orthogonal criteria: the visualization technique, the distortion technique, and the interaction technique (refer to Fig. 1). Here, orthogonality implies that any visualization technique can be combined with any distortion or interaction technique. Visualization techniques are further categorized into geometric projection, icon-based, pixel-based, hierarchical, and graph-based techniques.

B. Pixel-Oriented Techniques

Pixel-oriented techniques operate on the fundamental concept of mapping each data value to a coloured pixel, presenting data values related to a specific dimension (attribute) in separate subwindows. These techniques typically use one pixel per data value, allowing the visualization of substantial amounts of data on current displays, up to approximately 1,000,000 data values. The screen is divided into multiple subwindows for data sets with m dimensions, with each dimension having its own subwindow. Some pixel-oriented and query-dependent techniques introduce additional windows for overall distance. Within the windows, data values are arranged based on the overall sorting, which can be data-driven for query-independent techniques or query-driven for query-dependent ones. Detecting correlations, functional dependencies, and other relationships between dimensions is facilitated by examining corresponding regions in multiple windows.

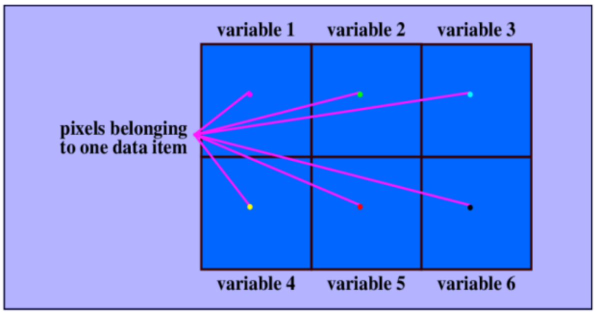

Fig. 2. Structured blocks of data Variables.

To achieve this objective, several design challenges must be addressed. The first challenge involves mapping data values to colours, which requires careful engineering to ensure an intuitive and effective mapping. The second important consideration is the arrangement of pixels within the subwindows, a decision influenced by the data and visualization task. As discussed in Section 4, this arrangement problem can be formally described as an optimization problem, with different techniques optimizing various variants. The third question pertains to the shape of subwindows. For data sets with numerous dimensions, the rectangular shape may result in distant subwindows, making it challenging to identify interesting relationships between dimensions(refer to Fig. 2). In Section 5, we introduce a visualization technique that addresses this problem more effectively.

C. Pixel Arrangement

The significance of a well-considered arrangement is heightened by the pixel displays' density, as only an effective arrangement enables the identification of clusters and correlations among dimensions. The challenge of arranging pixels within subwindows becomes more pronounced, requiring careful consideration for effective visualization. This issue becomes particularly pertinent when dealing with large datasets,[8] where a suboptimal arrangement might hinder the recognition of meaningful patterns.

In addressing the arrangement problem, it is essential to differentiate between data sets that inherently possess a natural ordering of data objects, such as time-series data, and those without inherent ordering, as seen in the case of query responses. For datasets with a natural ordering, the arrangement must be designed to leverage the temporal sequence effectively. On the other hand, in datasets lacking inherent ordering, the arrangement becomes a more nuanced challenge, requiring careful consideration of the data distribution and the objectives of the visualization task.

Furthermore, the ordering of subwindows for dimensions (attributes) plays a crucial role in effective pixel-oriented techniques. This is particularly evident in applications where no intrinsic order exists for dimensions. Developing strategies to order subwindows in a manner that aligns with the inherent characteristics of the data is a crucial aspect of optimizing pixel-oriented visualization techniques[9]. In summary, achieving a balance between the density of pixel displays, effective arrangement, and appropriate ordering is imperative for uncovering meaningful insights and relationships within multidimensional datasets.

V. PIXEL VALUE SCHEME

This section introduces the proposed steganographic scheme for colour images. Initially, each colour pixel undergoes decomposition into its corresponding colour components, namely R, G, and B. Subsequently, pairs are formed by combining (R, G) and (G, B), although other ordered pairs are also viable.

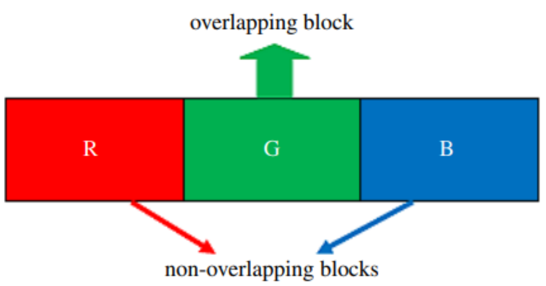

Fig. 3. Pixel Block of Colour Image.

However, our implementation focuses on using pairs like (R, G) and (G, B). These pairs create two consecutive overlapping blocks, as illustrated in Figure 3.

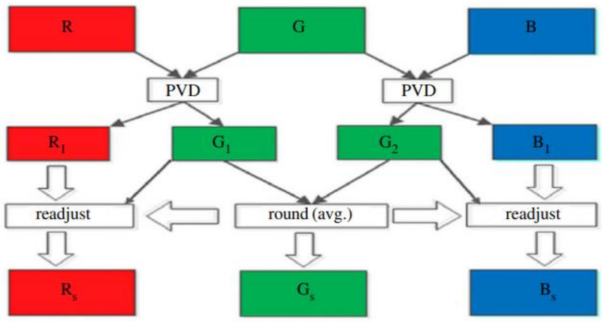

Fig. 4. Schematic Diagram of data embedding procedure.

In our scheme, variable secret message bits are embedded based on the difference of each pair using PVD (Pixel Value Differencing). Following the embedding process[10], the intermediate colour components are readjusted to achieve the final stego-colour components.

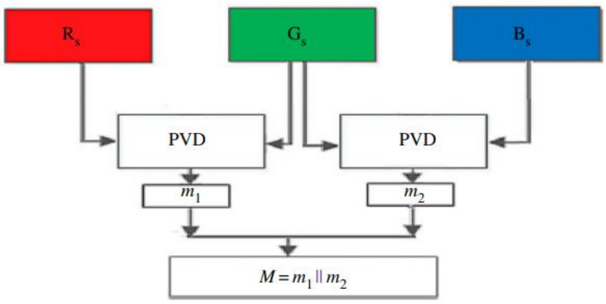

Fig. 5. Schematic Diagram of data extraction procedure.

To mitigate potential distortion that might be noticeable in the natural colour image due to data hiding, we employ a suitable threshold value. The data-hiding capacity within each colour pixel is constrained by this threshold value, ensuring that the stego-image maintains high visual quality[11]. Figure 4 depicts the overall embedding process, while the decoding process is illustrated in Figure 5.

VI. GEOMETRY RELATED DATA

Various applications involve geometry-related data, such as weather measurements (temperature, rainfall, wind speed) at multiple locations, connection nodes in telecommunications, and internet node loads across different locations. The visualization of such information necessitates representing data values (e.g., air pollution) and their spatial locations. An intuitive approach is to represent data values as coloured pixels on a screen, corresponding directly to their spatial locations. However, since spatial data locations are not uniformly distributed in a rectangular space, the display may be sparsely populated in some regions and experience overplotting in others. Take, for example, air pollution, where cities with more than 10,000 inhabitants cluster in specific regions like North America, Europe, and Asia, creating sparsely inhabited areas.

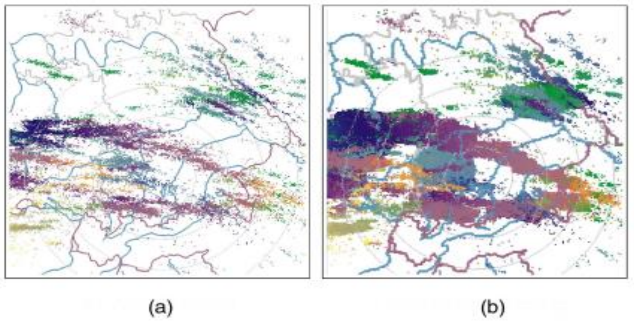

Fig. 6. Lightning Strike data (a). With Overlap (b). Without Overlap[12]

When presented on a world map, a significant portion of the screen corresponding to oceans remains unused, resulting in a loss of potentially important information. To address this, a simple yet intuitive idea is to present data values that cannot be accurately positioned on the screen at nearby unoccupied positions. This approach allows for the utilization of as many pixels as necessary while still reflecting the spatial nature of the data. In Fig 6, a dataset of lightning strikes in southern Germany is presented, demonstrating the avoidance of overlapping data points on unoccupied pixels, preserving spatial information and preventing information loss.

VII. IMPLEMENTATION

All techniques discussed in the preceding subsections, with the exception of certain grouping variants, have been implemented as integral components of the VisDB system. In addition to our pixel-oriented techniques, the VisDB system also accommodates the parallel coordinates technique developed by Inselberg and Dimsdale at IBM (Inselberg 1991; Inselberg and Dimsdale 1990) and the stick figure technique developed by Picket and Grinstein at the University of Massachusetts, Lowell (1988). The overarching objective of all our techniques is to present as many data items as possible on the display, allowing each data value to be represented by a single pixel.

However, it is essential to note that the parallel coordinate and stick figure techniques require more than one pixel per data value and typically result in visualizations with overlapping data items. To make these techniques suitable for large databases, we extended them by incorporating colour coding based on overall distance[13]. In the case of overlapping elements, we prioritize drawing the most relevant data items over the less relevant ones. Colouring and drawing based on overall distances facilitate the identification and comparison of the most relevant data items, a critical aspect when dealing with extensive datasets.

While the parallel coordinate and stick figure techniques serve their purpose, they are most effective for datasets with a limited number of items. In a broader context, we advocate the use of multiple techniques at different stages of the data exploration process. Depending on the user's goals and the characteristics of the data, a combination of query-independent and query-dependent techniques may be employed[14]. For instance, Peano-Hilbert or Morton visualizations may be suitable for unstructured or unknown data structures, while the recursive pattern technique with specific parameter settings is advantageous when the data structure is known. As exploration progresses, users may transition to query-dependent techniques like the snake-spiral technique for a more focused search or the snake-axes technique for additional information about the data. Once interesting clusters or functional dependencies are identified, grouping, parallel coordinate, or stick figure techniques can be utilized for targeted exploration within those portions of the dataset.

VIII. SUMMARY AND CONCLUSION

Pixel-oriented visualization techniques have proven to be valuable in exploring and analyzing large databases to uncover compelling data clusters and their characteristics. However, a notable observation is that many of these techniques appear to be ad hoc solutions, lacking a formal basis. This paper aims to address this gap by demonstrating that beneath the surface of developing and designing pixel-oriented techniques lie several significant optimization problems that demand careful consideration.

In the course of our investigation, we elucidate the formal definitions of these optimization problems, shedding light on the intricacies involved in creating effective pixel-oriented techniques. By delving into the optimization of different criteria, we illustrate how distinct variants of pixel-oriented techniques emerge[15]. This approach not only provides a systematic foundation for these visualization methods but also allows for a more structured understanding of their design principles. In essence, our work contributes to establishing a formal framework for pixel-oriented techniques, elevating them from ad hoc solutions to systematically optimized tools for database exploration and analysis.

IX. ACKNOWLEDGEMENTS

We express our gratitude to our project supervisor, Er. Deepika Sharma, for providing valuable guidance and consistent encouragement throughout the project. His support and insightful suggestions proved immensely beneficial in the successful completion of this endeavour.

Our sincere thanks extend to all contributors who played a vital role in the development and triumph of the Pixel Visualizer project. The realization of this project would not have been possible without the collaboration, dedication, and support of numerous individuals and organizations.

Foremost, we extend heartfelt thanks to our team of skilled researchers, developers, designers, and engineers. Your expertise, creativity, and countless hours devoted to Pixel Visualizer have been crucial in bringing this project to fruition. Your commitment to innovation and excellence has significantly shaped the success of the project.

We also acknowledge the invaluable guidance and mentorship provided by our advisors and mentors throughout the project's lifecycle. Their wisdom and insights have played a pivotal role in guiding us and ensuring the project's overall success.

Our appreciation goes to the users and beta testers who provided essential feedback, contributing to the refinement of the Pixel Visualizer tool. Your input has played a crucial role in enhancing its usability and functionality.

Furthermore, we are thankful for the support from academic and research institutions, as well as funding agencies, that have backed this project. Their financial and logistical support has been instrumental in advancing the research and development efforts.

REFERENCES

[1] A. Inselberg, The Plane with Parallel Coordinates, Special Issue on Computational Geometry,º The Visual Computer, vol. 1, pp. 69-97, 1985.

[2] D.A. Keim, ªVisual Support for Query Specification and Data Mining,º Aachen, Germany: Shaker-Publishing Company, 1995.

[3] X. Zhang and D. H. Brainard, ''Method and apparatus for estimating true colour values for saturated colour values in digitally captured image data, U.S. patent 6,731,794 (May 4, 2004)

[4] J. A. S. Viggiano, ''Minimal-knowledge assumptions in digital still camera characterization. I: Uniform distribution, Toeplitz correlation,'' in Proceedings of the IS&T/SID 9th Color Imaging Conference (Society for Imaging Science and Technology, Springfield, Va., 2001), pp. 332–336.

[5] E.R. Tufte, The Visual Display of Quantitative Information. Cheshire, Conn.: Graphics Press, 1983.

[6] Chan CK, Cheng LM. 2004 Hiding data in images by simple LSB substitution.PatternRecognit.37,469–474 (doi:10.1016/j.patcog.2003.08.007)

[7] A. Inselberg, The Plane with Parallel Coordinates, Special Issue on Computational Geometry,º The Visual Computer, vol. 1, pp. 69-97, 1985.

[8] P. Longere and D. H. Brainard, ''Simulation of digital camera images from hyperspectral input,'' in Vision Models and Applications to Image and Video Processing, C. van den Branden Lambrecht, ed. (Kluwer Academic, Boston, Mass., 2001), pp.123–150.

[9] Swain G. 2016 Adaptive pixel value differencing steganography using both vertical and horizontal edges. Multimedia. Tools Appl.75, 13 541–13 556. (doi:10.1007/s11042-015-2937-2)

[10] Prasad S, Pal AK. 2017 Data from An RGB colour image steganography scheme using overlapping block-based pixel-value differencing.DryadDigitalRepository.(http://dx.doi.org/10.5061/dryad.21tm5)

[11] G.G. Robertson, J.D. Mackinlay, and S.K. Card, ªCone Trees: Animated 3D Visualizations of Hierarchical Information,º Proc. Human Factors in Computing Systems CHI '91 Conf., pp. 189-194, 1991.

[12] D.A. Keim and A. Herrmann, The Gridfit Algorithm: An Efficient and Effective Algorithm to Visualizing Large Amounts of Spatial Data,º Proc. IEEE Visualization Conf., pp. 181-188, 1998

[13] Robertson, G., Card, S., and Mackinlay, J. (1991), "Cone Trees: Animated 3D Visualizations of Hierarchical Information," in Proceedings of the ACM CHI International Conference on Human Factors in Computing (CHI'9I), pp. 189-194.

[14] Swayne, D. F., Cook, D., and Buja, A. (1992), "User's Manual for XGobi, a Dynamic Graphics Program for Data Analysis," Bellcore technical memorandum.

[15] Wegener, I. (1990), "Bekannte Sortierverfahren und eine HEAPSORT-Variante, die QUICKSORT schlagt,' Informatik Spektrum, 13,321-330.