Analysis of Local Data Patterns by Local Adaptive Color Mapping

Sebastian Mittelstädt
Universität Konstanz

Andreas Stoffel
Universität Konstanz

Tobias Schreck
Universität Konstanz

Daniel A. Keim
Universität Konstanz

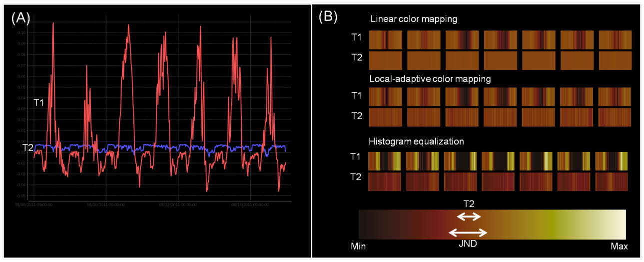

Figure 1: Boosting of peak points in a pixel-based time series visualization. (A) shows two time series in a line chart. (B) illustrates the pixel-based visualization of the same data with different color mappings (each rectangle holds 24 hours of data). The trend of T2 is invisible in the linear representation, which is recovered by the other color mappings. However, the non-linear histogram-based algorithm transfers some value ranges that were orange into the yellow and white ranges of the color map, which biases global comparison.

Abstract

Color, after position, is among the most effective visual variables to encode information. It is pre-attentively processed by the visual system, and if used appropriately, supports detection and correlation of patterns. Several global color mapping schemes (such as linear, non-linear and histogram-based) exist that support certain analysis tasks. However, static global schemes map data with a small local variation (within a data set of high variation) to small color differences. Often, these color differences are below the noticeable difference threshold of user perception or the display device. As a consequence, valuable information may be lost since data points or structures cannot be adequately perceived and correlations or other patterns of interest may be missed. Existing techniques to avoid this effect either require user interaction or are based on specific assumptions about the data. We introduce a novel automatic algorithm for *local-adaptive color mapping* that is applicable to dense data and is based on the idea to locally modify the color mapping to enhance the visibility of structures. This technique emphasizes patterns of interest within locally chosen color-ranges such that (1) the visibility of local differences is enhanced and (2) the introduced global distortion of the color mapping is kept small. This allows the perception of relevant patterns while approximately maintaining global comparability across the whole data set.

1 Introduction

Color mappings yet do not consider the surround of pixels in visualizations and cannot guarantee the visibility of structures in the final visualization. This is one of the general problems of visualizations, for example, to enhance the visibility of structures such as streets on maps or veins in the human body but also in abstract data visualizations where sets of pixels form important structural information. Assuming we can identify requirements of a concrete visual analysis task, we may be able to define an appropriate color scale for the particular case, which embeds a adapted *scaling* of data values. Data transformations can support the analysis of local data properties of interest.For instance, square root normalization spreads data concentrated in lower intervals of the input data domain to a larger region in the color-scale. It thereby improves the distinction of data values in this area. However, such techniques require a-priori understanding of the data and its relevant scales. They are static in that they apply one scheme to the whole data set. And also, they introduce non-linearity, harming comparison tasks on the global and absolute scale where perceptual linearity is desired. For example, in Figure 1 (A), the periodic patterns of time series T1 and T2 are visible in the line charts. However, the patterns are invisible in the linear pixel-based display (Figure 1 (B)). If we apply a histogram-based algorithm, the trend of T2 becomes visible by mapping the original orange color tones into yellow and red. If we want to compare these values, one would intuitively say, that T1 is far higher than T2. Indeed, the averages are about the same.

Given these shortcomings of static color-mapping schemes, we propose a novel algorithm for dense data displays to dynamically adapt colors based on local data properties. This approach retains the visibility of local details without using additional visualization space and aims at keeping the global distortion small. The technical achievements are as follows: 1) A color boosting algorithm that locally adapts the color mapping for important data structures and guarantees the visibility of important data points; 2) The method preserves global metric quantities of the data and provides an informative overview without interaction.

Our method thereby introduces contrast effects in order to enhance the visibility of structures. Typically, this is one of the major problems in data visualizations. A solution to compensate for contrast effects was recently presented in [4]. In this paper, however, we present how contrasts can be used to enhance dense-data visualizations that use color for encoding metric quantities of numerical data. Note, that this algorithm can be applied to any image or data visualization to enhance the visibility of known structures.

2 Related Work

Generally, there is no color map that meets all the requirements of data visualizations equally. The challenge is to create an appropriate color map for a specific task and specific data properties, as proposed in the literature. General guidelines on selecting color maps can be found in [1, 7, 8]. Approaches that use histogram equalization optimize the visual distinction of data points by equalizing the distribution of the color scale in the image [2, 6]. Also, lenses [3] could be used as an interactive tool to emphasize local visualization aspects, but this depends on the user to actively spot areas of interest in the data. In [5] the authors discuss several approaches to enhance the visibility of data points. E.g., *Boosting by halos and color* visually highlights interesting data points or borders by changing the color of their surrounding pixels in order to increase the contrast. The techniques in [5] are applicable predominantly to sparse data. We consider dense data and therefore, these approaches are not ideal as we want to preserve the metric quantities of the data and avoid overwriting of quantities.

Data: Structures sorted according importance
Result: Boosted structures
for *Structure s in structures* **do**
  high = s.copy(); low = s.copy();
  while *!visible(high) && !visible(low)* **do**
    for *i=0; i < high.size; i++* **do**
      high[i].color ++; low[i].color - -;
    end
  end
  if *visible(high)* **then**
    visibleStructures.add(high);
  else
    visibleStructures.add(low);
  end
end
return *visibleStructures*;

Algorithm 1: Guarantees the visibility of important structures. The variable *high[i].color* denotes the colormap level of a pixel within the structure that is increased by the boosting.

3 Color Boosting

We illustrate the basic idea of our proposed method by the example in Figure 1. In part (A), the local details in time series T2 (blue) are visible in the line chart. However, they are invisible in the linear pixel-based representation (B), as they are below the noticeable difference threshold. We assume that the peak points are important in this application. Therefore, we locally manipulate the color mapping in order to make these details *just* noticeable. Our color boosting algorithm performs two steps in order to reveal local structures: 1) Structure detection: Local structures of interest are detected by an application dependent detector; 2) Color boosting: The color level of these structures is scaled in both directions of the color map until they become just noticeable from their spatially surrounding area.

The structure detection is a modular function within the algorithm and can be replaced by any meaningful approach. For example, connected pixels that form a structure such as pixels encoding a peak value or streets on maps can be defined as structures. A structure is perceivable, if its border pixels are visual distinct to the surrounding. Therefore, the color distance of each border pixel must be greater than the *just noticeable difference* (JND) to its surrounding, excluding the pixels of the same structure. We want to minimize the color distortion in order to preserve global comparability as far as possible. Thus, the algorithm has to maximize the visibility of structures and minimize the distortion of color at the same time.

Our heuristic solves this optimization problem by sorting the structures in a priority queue according to their importance for the current visualization and then sequentially process each structure. The algorithm aims to maximize the number of visible border pixels of a structure, while minimizing the bias of the boosting. This (sub-) optimization problem can easily be tackled by an algorithm, that increases and decreases the color levels of the structure pixels and tests in each iteration whether the border pixels have become visible (see Algorithm 1). As soon as enough border pixels of a structure are visible the algorithm stops and continues with the next structure in the priority queue. As a rule of thumb, we set this to 99% of all border pixels of a structure. A border pixel is considered as visible, if the color distance of the pixel to every of its surrounding pixels is above the JND (Algorithm 2). Color distances can be estimated in the CIELAB color space, where $\Delta E \approx 2.3$ expresses the JND. This algorithm has a complexity of $O(s*p*w*l)$ where s is the number of structures, p the number of pixels per structure, w is the number of neighbors (standard: 8 neighborhood) and l the number of color levels. Since, in the worst case $s*p = n$ (number of pixels) and $w, l << n$, the algorithm has a linear complexity of $O(n)$.

Data: Current structure s
Result: Visibility of s
for *Pixel p in borderPixels(s)* **do**
  for *Pixel n in neighbors(p,!s)* **do**
    if *colorDistance(p,n) < JND* **then**
      visible = false; break;
    end
  end
  cntVisible = visible?cntVisible+1:cntVisible;
  cntInvisible = !visible?cntInvisible+1:cntInvisible;
end
return $\frac{cntVisible}{cntVisible+cntInvisible} > 0.99$

Algorithm 2: Tests the structure's visibility. The function *neighbors(p,!s)* provides the neighboring pixels of p within the 8 neighborhood that are not in the same structure.

4 Conclusion & Future Work

We present a method that locally adapts the color mapping to improve the visibility of interesting pixels and structural information. The algorithm is applicable to any kind of 2D image or visualization when the structures that should be preserved are known. It should be highlighted that there are some general limitations in the local-adaptive color mapping methodology and there are interesting future work items to be addressed. When local adaption is applied, we may see artifacts introduced, since equal data values may be mapped to different color tones and vice versa, based on the local adaption strategy. It is an interesting fundamental question how we can assess the trade-off between the analytical gain of making local structures visible, as compared to the bias introduced by this methodology. It will be interesting to understand in more detail, where the trade-off runs for different application domains and visualization techniques. Further, we aim for defining the problem of structural visibility as perceptual optimization problem since important perceptually processes are not considered in the technique such as contrast sensitivity and color appearance. In the near future, we want to research and extend image-based structure detectors for visual analytics applications. We plan to research an interactive approach where the user marks a local data pattern of interest, and then the systems computes a color mapping which best emphasizes this local structure in the global view. Thereby, we could conveniently parameterize the local-adaptive color mapping on-the-fly.

References

[1] L. D. Bergman, B. Rogowitz, and L. A. Treinish. A rule-based tool for assisting colormap selection. In *Proceedings of the IEEE Conference on Visualization*, pages 118–125. IEEE, 1995.
[2] E. Bertini, A. Girolamo, and G. Santucci. See what you know: Analyzing data distribution to improve density map visualization. *IEEE Symposium on Visualization (Eurographics 2007)*, 2007.
[3] N. Elmqvist, P. Dragicevic, and J. Fekete. Color lens: Adaptive color scale optimization for visual exploration. *IEEE Transactions on Visualization and Computer Graphics*, 17(6), 2011.
[4] S. Mittelstädt, A. Stoffel, and D. Keim. Methods for compensating contrast effects in information visualization. *Computer Graphics Forum*, 33(3), 2014.
[5] D. Oelke, H. Janetzko, S. Simon, K. Neuhaus, and D. Keim. Visual boosting in pixel-based visualizations. *IEEE Symposium on Visualization 2011 (EuroVis)*, 30(3), 2011.
[6] S. Pizer, E. Amburn, J. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. ter Haar Romeny, J. Zimmerman, and K. Zuiderveld. Adaptive histogram equalization and its variations. *Computer vision, graphics, and image processing*, 39(3), 1987.
[7] C. Tominski, G. Fuchs, and H. Schumann. Task-driven color coding. In *Proceedings of the 12th International Conference on Information Visualisation*, pages 373–380. IEEE, 2008.
[8] C. Ware. *Information visualization: perception for design*. Elsevier, 2012.