# Semantics of Directly Manipulating Spatializations

Xinran Hu, Lauren Bradel, Dipayan Maiti, Leanna House, Chris North, and Scotland Leman

**Abstract**—When high-dimensional data is visualized in a 2D plane by using parametric projection algorithms, users may wish to manipulate the layout of the data points to better reflect their domain knowledge or to explore alternative structures. However, few users are well-versed in the algorithms behind the visualizations, making parameter tweaking more of a guessing game than a series of decisive interactions. Translating user interactions into algorithmic input is a key component of Visual to Parametric Interaction (V2PI) [13]. Instead of adjusting parameters, users directly move data points on the screen, which then updates the underlying statistical model. However, we have found that some data points that are not moved by the user are just as important in the interactions as the data points that are moved. Users frequently move some data points with respect to some other "unmoved" data points that they consider as spatially contextual. However, in current V2PI interactions, these points are not explicitly identified when directly manipulating the moved points. We design a richer set of interactions that makes this context more explicit, and a new algorithm and sophisticated weighting scheme that incorporates the importance of these unmoved data points into V2PI.

**Index Terms**—Visual to parametric interaction, visual analytics, statistical models

◆

## 1 INTRODUCTION

Visual to Parametric Interactions (V2PIs) provide data analysts a powerful tool for exploring large, high dimensional datasets via two-dimensional visualizations. V2PI enables users to inject their expertise by directly manipulating the position of the data points in the visualization, and the algorithm quantifies the visual interaction into parametric feedback and reconfigures the visualization accordingly. Recent advances of V2PI concepts (e.g., [11][10][13][14][3]) have shown promising results, but open research challenges exist concerning the interpretation of the user's spatial interactions.

We have observed that users move objects toward or away from each other to express similarity or dissimilarity between those objects [8][7][2][9][5]. *However, the semantics of such spatial movement can be ambiguous.* In this paper we focus on the ambiguity associated with the meaning of the relationship of moved points to unmoved points. The question is whether the unmoved points were considered in the placement of the moved points. For points that are left unmoved, users may be communicating either: 1) they moved points with respect to (towards or away from) the unmoved points and thus intended to expressed similarity or dissimilarity to the unmoved points, or 2) the movement had no relationship to the unmoved points and thus they intended to express no new information about the unmoved points. Without users being explicit about their true intentions, such information may be misinterpreted.

We investigate how to enable users to better express their intended meaning through explicit interactions. Specifically, a user could tag certain unmoved points as being spatially contextual within the interaction in order to distinguish between the two possible interpretations above. We incorporate this notion of tagging, or *highlighting*, unmoved points to give users greater control over expressing their mental models of the data using the V2PI concept. We then compare and contrast these capabilities with previous implementations of similar algorithms, noting that these algorithms can be explained as a special case of the new method presented here.

In this paper, we use a weighted Multidimensional Scaling (MDS) algorithm as the method of visualization and discuss how to fully exploit the visual interactions to derive parametric feedback that appropriately captures the user's mental model of the data as precisely as possible. While we only present a specific case for V2PI, weighted MDS is particularly flexible and relatively simple. Specifically, we improve the current V2PI-MDS algorithm's ability to exploit the intended meaning of the user's explicit interactions including highlighted unmoved data points. Using a highlighting tool and a pairwise weighting scheme, we build an improved V2PI system that reduces ambiguity in spatial movements as much as possible. After presenting the new V2PI-MDS algorithm, we demonstrate the effectiveness of highlighting unmoved points and the pairwise weighting scheme. We conclude that our updated form of V2PI is an improvement over previous efforts.

## 2 V2PI BACKGROUND

Good visual analytics [19] algorithms that foster sensemaking should 1) utilize a communication scheme close to the user's natural communication techniques, and 2) provide the maximal amount of assistance in analytical reasoning. Potential means of communication include Surface Level Interactions and Parametric Level Interactions. However, both schemes only emphasize one of the aforementioned properties. Surface Level Interactions (SLIs) (see Figure 1a) allow the user to alter the visualization through, for example, rotating, zooming, highlighting and repositioning objects. These interactions are intuitive and require little background knowledge concerning the display-generating model to perform. In fact, the underlying visualization model(s) is not altered through SLIs. This separates mathematical analysis from analytic reasoning and confines sensemaking to the visual scope.

In contrast, Parametric Level Interactions (PLIs) (see Figure 1b) are performed directly on the parameters of the underlying visualization model. Visualizations reconfigure according to user-specified parameters. For instance, interactive Principal Component Analysis (iPCA) [12] allows users to adjust visualizations by changing the contributions of data dimensions. The advantages of PLIs include high mathematical integrity and potentially rich information provided for sensemaking. However, these advantages can only be realized if users know which model parameters to change. For example, implementing an intuitive idea (e.g., moving two objects closer together) in the visual display through PLI requires a full understanding of the relationship between the model parameters and the visualization. In the case of iPCA, this relates to adjusting the eigen-space responsible for the projection; a formidable task in even moderately sized datasets. It is seldom, if ever,

• *Xinran Hu is with Virginia Tech. E-mail: huxinran@vt.edu.*
• *Lauren Bradel is with Virginia Tech. E-mail: lbradel1@vt.edu.*
• *Dipayan Maiti is with Virginia Tech. E-mail: dipayanm@vt.edu.*
• *Leanna House is with Virginia Tech. E-mail: lhouse@vt.edu.*
• *Chris North is with Virginia Tech. E-mail: north@vt.edu.*
• *Scotland Leman is with Virginia Tech. E-mail: leman@vt.edu.*

(a) Surface level interaction

(b) Parametric level interaction
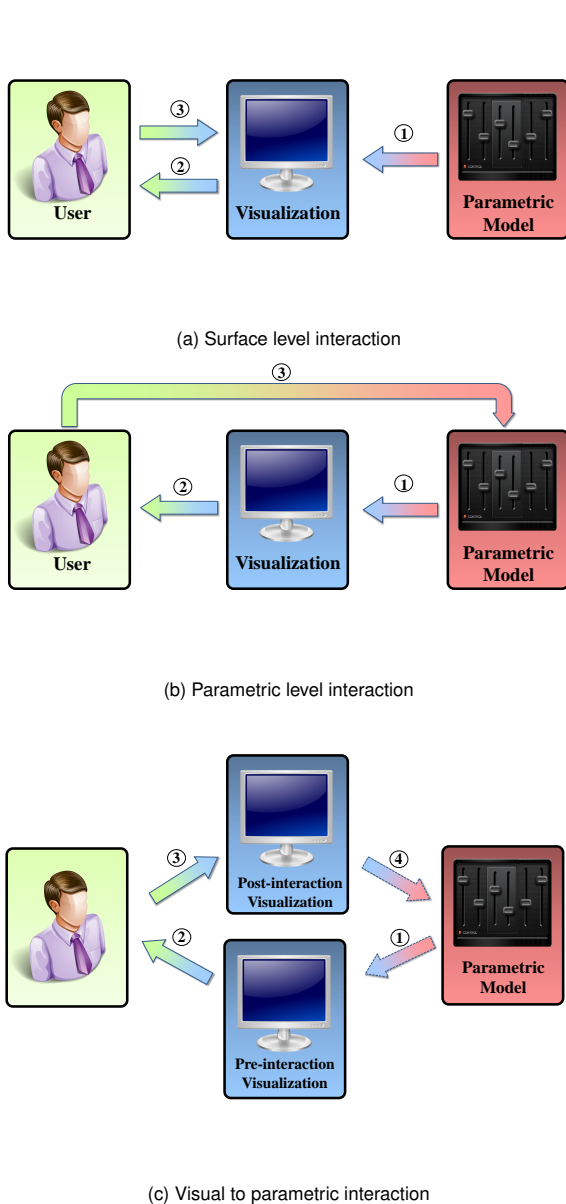
(c) Visual to parametric interaction

Fig. 1. (a) shows the process of surface level interactions. Step 1: the visualization of data is created by the underlying visualization model. Step 2: the visualization is presented for user inspection. Step 3: the user provide visual feedback with the visual display. Only Step 2 and 3 are iterated. (b) shows the process of parametric level interactions. Step 1: the visualization of data is created by the underlying visualization model. Step 2: the visualization is presented for user inspection. Step 3: the user provide parametric feedback with the visualization model. All steps are iterated. (c) shows the process of visual to parametric interactions. Step 1: the initial visualization is created by the underlying visualization model. Step 2: the visualization is presented for user inspection. Step 3: the user provides visual feedback. Step 4: visual feedback is parameterized as parametric feedback. All steps are iterated.



(a) The explicit interactions.    (b) Without any unmoved objects.

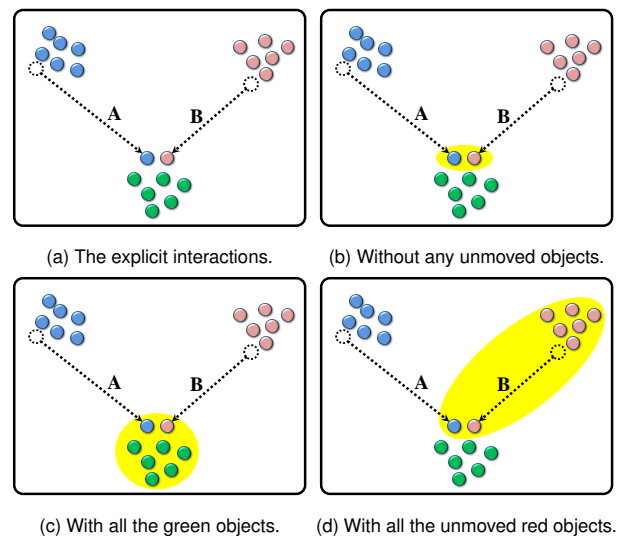(c) With all the green objects.    (d) With all the unmoved red objects.

Fig. 2. (a) shows an explicit interaction of moving two objects (A and B) closer. (b)(c)(d) provide multiple interpretations of this interaction. The user may be attempting to: (b) explore the similarity between the two moved objects; (c) explore the similarity between the two moved objects and the third cluster at the bottom; (d) explore the similarity between the two moved objects and the dissimilarity with the cluster in the upper right corner.

the case that a user has a thorough understanding of the visualization model. In practice, the user relies on trial and error (e.g., tentatively adjusting parameters and hoping that the visualization responds in a desirable manner). This can be very frustrating for users, especially for those without strong mathematical backgrounds.

Visual to Parameter Interactions (V2PIs) combine the advantages of SLIs with PLIs. When V2PIs are available, users express cognitive feedback concerning the data and visualizations via SLIs that are characterized parametrically by V2PI machinery. Then, based on parametric feedback, visualizations update, just as they would with PLIs. The methods used to parameterize feedback and update visualizations are shielded from users - allowing them to focus on their analyses, instead of fiddling with parameters. However, V2PI algorithms are only useful if the parametric feedback is compatible with the user's mental model of the data. In this paper, we design improved V2PI interactions that produce more compatible parametric feedback by giving users more explicit control.

## 3 INTERACTION DESIGN

In V2PI, visual interactions can involve data points in two ways: explicitly and implicitly. For example, when moving data point A near unmoved data point B in an MDS visualization, the data point A is explicitly involved in the interaction. Whereas, data point B is implicitly involved because the user mentally considered it in the interaction but did not specify it in the user interface. Previous implementations of the V2PI-MDS [10] only considered explicit points in the interactions, yet both are important for interpreting the user's mental model of the data. The challenge is defining what these interactions mean, acknowledging various interpretations of implicit points, and potentially using both simultaneously to update visualizations.

Arguably, the intended meaning of moved points in interactions is clear and defined in [13]. However, two valid meanings could be inferred from unmoved points: 1) the unmoved points are implicitly involved in the interaction, or 2) the unmoved points are *not* involved in the interaction. In previous observations, we have found that users frequently apply each of these intents [8][2][9][17][16]. Sometimes they specifically move points nearer to or further from other unmoved points (such as contextual landmarks), and sometimes unmoved points are completely ignored as they organize other points on the screen. Assuming case (1) for all the unmoved points increases the computational cost of MDS significantly and does not accurately match the user's understanding of the data. Assuming case (2) may risk "losing" useful information the user supplied. Furthermore, the ambiguity of implicit points in interactions may confound the interpretation of explicit interactions. Examples of this ambiguity are shown in Figure 2.

Thus, we propose a new interface and algorithm that allows users to explicitly state the meaning of the unmoved points with minimal additional effort. Effectively, we turn some of their implicit actions

| Notation | |
|---|---|
| $\mathbf{D} = \{\mathbf{d_1} ; \ldots ; \mathbf{d_n}\}$ | the dataset matrix. |
| $\mathbf{d_i} = \{d_{i1}, \cdots, d_{ip}\}$ | the $i$th object in the high-D dataset. |
| $\mathbf{R} = \{\mathbf{r_1} ; ; \ldots ; \mathbf{r_n}\}$ | the 2-D representatives of $\mathbf{D}$. |
| $\mathbf{r_i} = \{r_{ix}, r_{iy}\}$ | the 2-D representative of $\mathbf{d_i}$. |
| $\omega = \{\omega_1, \omega_2 \cdots, \omega_p\}$ | the weight parameter of the weighted MDS model. |
| $\delta_{ij}^{(\omega)}$ | the high-D pairwise distance given weight $\omega$. |
| $\gamma_{ij}$ | the 2-D pairwise distance before user feedback. |
| $\tilde{\gamma}_{ij}$ | the 2-D pairwise distance after user feedback. |
| $c_{ij}$ | the weight for the $i$th and $j$th data pair in the weighted stress function. |

into explicit feedback. By highlighting certain unmoved points, users communicate that moved points were moved with respect to the highlighted points and the proximity between highlighted points should be preserved. Dragging and highlighting interactions divide the points into three sets.

- **Moved set**: includes all the points dragged and repositioned by the user.

- **Highlighted set**: includes all the points highlighted by the user.

- **Untouched set**: includes all the points neither highlighted nor dragged by the user.

Technically, our approach can have multiple different Highlighted sets, each of which can be treated independently. This could be useful for enabling more complicated cases where multiple different relationships are expressed. However, the default behavior is to simplify with only one highlighted set at a time, because users will typically decompose complex relationships into multiple simple relationships.

To parameterize feedback from highlighting points, we also introduce a pairwise weighting scheme. The pairwise weighting scheme protects the algorithm from being dominated by any large clusters and fosters flexible data exploration. As we will demonstrate, the influence of a set of points in the parameterization increases quadratically as the number of points within the set increases. Thus, an unweighted quantification approach would introduce bias towards large sets. The pairwise weighting scheme keeps the influences of different sets comparable, preventing large sets from dominating the parameterization.

## 4 V2PI-MDS

Our new V2PI-MDS algorithm with the highlighting tool and pairwise weighing scheme is based on an iterative four-step process (see Figure 1c). The initial visual representation of the dataset is created by the weighted MDS algorithm. The user can then interact with the visualization according to his/her expert judgement and understanding of the dataset. The visual feedback is analyzed and quantified as an adjustment of the weight parameter in the weighted MDS algorithm. The visualization is then reconfigured in accordance with these adjustments. Users continue through this process iteratively until they are satisfied with the visualization. More details of this new V2PI-MDS are described below.

1. **Weighted MDS model and initial visualization**
   Given a high dimensional dataset $\mathbf{D}$ with $n$ observations and $p$ dimensions, the MDS model first computes the pairwise distances for every possible pair of objects. Each pairwise distance quantifies the dissimilarity between two objects. Large distances imply "less similar" than small distances and vice versa. Users can

choose any legitimate distance metric. One possible choice is the weighted Euclidean distance. The weighted Euclidean distance parameter $\omega$ adjusts the contributions of data dimensions in the overall distance. The default setting for $\omega$ is a uniform weight vector with equal weight on all dimensions ($\omega_i = \frac{1}{p}$ for $i \in \{1, \ldots, p\}$). The MDS algorithm that uses the weighted Euclidean distance is also referred to as "weighted MDS" [4][18].

The initial visualization is created by searching through a set of two-dimensional projections that maintain the corresponding high-dimensional pairwise distances as well as possible. The high-dimensional pairwise distances here is computed with a uniform weight vector. The pairwise distances in the two-dimensional visualization should resemble the high-dimensional pairwise similarities and dissimilarities as best as possible.

$$\underset{\mathbf{R}}{\operatorname{argmin}} \sum_{0 \leq i < j \leq n} \left| \delta_{ij}^{(\omega)} - \gamma_{ij} \right|. \tag{1}$$

The formula above is called the stress function, which measures the total discrepancy between all of the high-dimensional pairwise distances, $\delta_{ij}^{(\omega)}$, and their low-dimensional counterparts, $\gamma_{ij}$ for all i and j pairs.

2-3. **User inspection and interaction**
As the initial visualization of the dataset is created, the comparison between the initial visualization and the user's mental image of how the data should be displayed begins. In the new V2PI-MDS algorithm, we expand the interaction capabilities of the tool by allowing highlighting in addition to moving points. If the current distance between two objects contradicts the user's mental model, the user can move both objects or highlight one object and move the other object to adjust the distance. In the case of using multiple highlight sets, if the current distance between two objects is what the user believes it is, the user can highlight both objects to confirm the distance. If an object is unmoved and unhighlighted, the new V2PI-MDS algorithm relies on the assumption that all the pairwise distances associated with this point are irrelevant to the user's mental model and ignored. Because the interactions in this algorithm are performed directly on objects, V2PI is also referred as Observation-level Interaction [10].

4. **Cognitive feedback to parametric feedback**
While the initial visualization relies on a uniform weight, these weights can be adjusted through user feedback. An updated weight vector of the weighted MDS model, $\tilde{\omega}$, is acquired by minimizing the following stress function:

$$\underset{\tilde{\omega}}{\operatorname{argmin}} \sum_{0 \leq i < j \leq n} c_{ij} \left| \delta_{ij}^{(\tilde{\omega})} - \tilde{\gamma}_{ij} \right|, \tag{2}$$

The stress of each $i$ and $j$ pair is adjusted by a non-negative weight $c_{ij}$ and values of all $c_{ij}$, $i, j \in \{1, \ldots, n\}$ is referred to as the **pairwise weighting scheme**. The pairs that are not compatible with user's mental model of the data receive larger, positive weight and the pairs less compatible with user's mental model receive smaller or zero weight. In our new V2PI-MDS algorithm, data points are categorized into three sets, moved, highlighted and untouched set. Each cell in the Table 1 represents a group of inter-set or intra-set object pairs. For example, cell **(i)** represents all the pairs of two moved objects and cell **(ii)** represents all the pairs of one moved and one highlighted objects. The $c_{ij}$ for each $i$ and $j$ pair is given by the equation in the cell. In this default setting, $c_{ij}$ equals to the reciprocal of the number of pairs in the same cell. Therefore the combined weight of each cell is 1 except the last column. Those are the pairs associated with at least one un-touched object and they are ruled out of the quantification. Note, the user can override the default setting to change the emphasis on different pairs.

Table 1. The default pairwise weighting scheme. $n_m$ and $n_h$ are the numbers of objects in the moved and highlighted set, respectively.

| moved | highlighted | untouched | |
|---|---|---|---|
| $\frac{2}{n_m(n_m-1)}$ **(i)** | $\frac{1}{n_m n_h}$ **(ii)** | 0 | **moved** |
| | $\frac{2}{n_h(n_h-1)}$ | 0 | **highlighted** |
| | | 0 | **untouched** |

We note the optimization in Equation (2) mimics the optimization which yields the spatial layout in Equation (1). The notable difference is that the optimization in (2) takes place over the weight vector, whereas, the optimization in (1) is over the low dimensional space. Because the user is supplying additional information about the *desired* relationships between objects, we hold that information constant (denoted as $\hat{\gamma}_{ij}$) and optimize the weight vector to best explain these adjustments. Without consideration of the pairwise weights, the solution for $\tilde{\omega}$ results from a straight forward inversion of the optimization in Equation (1).

The minimization of the stress function accomplishes two goals simultaneously:

(1) An updated weight vector is found that depicts the user's definition of dissimilarity. Dimensions with larger weights play a more significant role in defining the dissimilarity in the user's semantic understanding of the data.

(2) An updated visualization that is more compatible with the user's mental model of the dataset.

## 5 SYSTEM EVALUATION

In this section, we apply V2PI-MDS algorithm. We conducted an expressive study [11] of a dataset downloaded from **nutritionaldata.com** to demonstrate the effectiveness of highlighting and pairwise weighting scheme by comparing the new V2PI-MDS algorithm to the previous implementation of V2PI.

A cereal grain dataset was created with 60 observations that describe three grains: wheat, rye and triticale (a crossbreed of wheat of rye). For each grain, there are 20 observations with measurement of 14 nutrients. The goal is to learn the *nutritional properties triticale inherits from rye*. We used the V2PI-MDS algorithm to investigate this topic and understand how triticale and rye are similar nutritionally. In specific, we are going to move some triticale objects towards unmoved rye objects and run V2PI-MDS to interpret our interaction. The V2PI-MDS algorithm should upweight dimensions that make triticale and rye similar. By examining the algorithm's updated weight vector, nutrients with large weights are designated as the algorithm's solution to the question.

Before the study, we also acquired some acknowledged answers and we will compare these acknowledged answers with the algorithm's solution. The criteria for comparison are:

(1) The number of nutrients that match acknowledged answers.

The acknowledged answers include *protein*, *vitamin B6*, *magnesium*, *phosphorus* and *potassium* and *Folate (Folic Acid)*. *Vitamin C* is also considered as a correct answer because both triticale and rye have zero vitamin C. However, this factor is not mentioned in the acknowledged answers. We examine how many of these nutrients also appear in the algorithm's solution.

(2) The weights of the aforementioned seven nutrients.

A result is more favorable when the aforementioned seven nutrients receive a larger weight. We have created side-by-side bar graphs

to compare the weight of the seven matched nutrients versus the others.

### 5.1 Highlighting Tool

We demonstrate how highlighting impacts the algorithm's interpretation process. We conduct three comparative experiments with the same movement interactions but different highlighting interactions. In this section, to focus on the advantage of the new highlighting scheme independent of the new pairwise weighting scheme, we used a simple pairwise weighting scheme that weighs all moved or highlighted pairs equally and ignores all other pairs (i.e., $c_{ij} = 1$ if and only if the $i$th and $j$th objects are either moved or highlighted).

Figure 3 and 4 show the initial visualization and the movement interactions. In the initial visualization, all the nutrients contribute equally to the overall distances and the three well-separated clusters represent the three types of grain. To learn how rye and triticale are similar nutritionally, we move four triticale objects close to the rye cluster. This indicates that triticale is more similar to rye than shown currently. For now, we do not highlight any unmoved object. The selected moved points are displayed in Figure 5a. Based only on the explicit interactions in Figure 5a, V2PI-MDS upweights and downweights certain dimensions to match the new distances supplied by us. A new visualization is generated and shown in Figure 5b and the finalized dimension weights are shown in Figure 5c. The top seven nutrients suggested by the algorithm are *fat*, *Potassium*, *vitamin B6*, *vitamin C*, *carbohydrate*, *Sodium* and *protein*. Although four nutrients coincidentally match the answers, their ranks are relatively low (No. 2,3,4,7). The combined weight of the seven nutrients of interest is 59.66%.
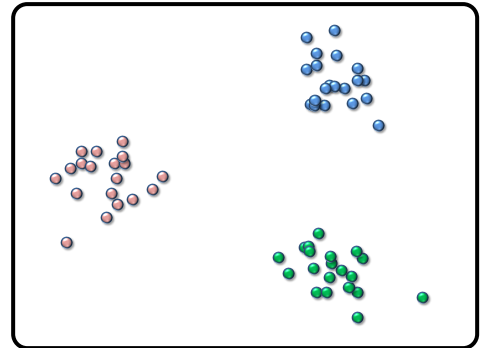


Fig. 3. The initial visualization of the cereal grain dataset. Wheat, rye and triticale are plotted as red, blue and green, respectively.
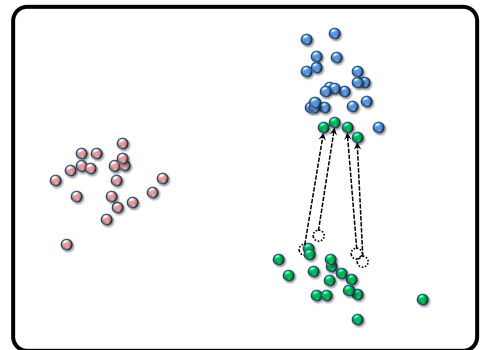


Fig. 4. Our visual interactions. Four triticale objects are moved toward the rye cluster. The dashed hollow markers show the moved objects' previous locations and the arrows show their movements.

For the second experiment, we put the highlighting tool into use and highlight the rest of the unmoved triticale cluster (as shown in Figure

6a) to create the visualization in Figure 6b. The goal of the second experiment is to demonstrate how highlighting interactions can affect the algorithm's interpretation process. Figure 6b and 6c show that both the updated visualization and weight vector and are indeed different from the previous experiment.



(a) The explicit interactions     (b) The updated visualization
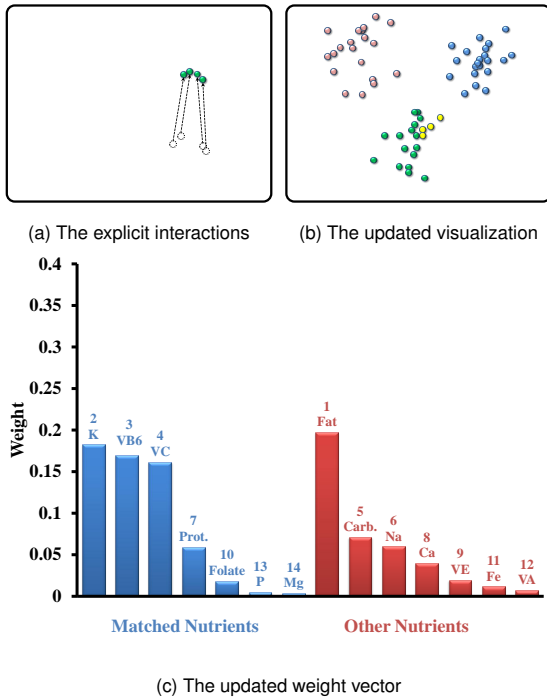


(c) The updated weight vector

Fig. 5. The first experiment when no object is highlighted. (a) shows the explicit interactions in this experiment. The interpretation of our actions for the moved objects is not compatible with the our true meaning of these actions. (b) shows the updated visualization. The cluster structure does not change much. The four yellow points are the four moved triticale objects. (c) shows the side-by-side bar graph of the weight vector. The left half of the plot shows the acknowledged nutrients' weights in decreasing order (blue bars). The right half of the plot shows the other nutrients' weights in decreasing order. The numbers above nutrient names are the overall ranks of their weights.

For the third experiment, we used the correct interaction to answer the question. When we moved the triticale objects toward the rye cluster, we also used the unmoved rye cluster as a reference, i.e. we was supplying new similarities between triticale and rye objects. Thus, rather than highlight the triticale cluster, we should highlight the rye cluster. Figure 7a shows the explicit interactions after highlighting the rye cluster. With the help of the rye cluster, the correct interpretation of the our interaction is now easily seen. The updated weight vector, as shown in Figure 7c, has five matches (*Magnesium*, *vitamin C*, *protein*, *Potassium* and *Folate*) in the top seven nutrients and they all rank high (No. 1, 2, 4, 5, 6). The combined weight of acknowledged nutrients is also significantly higher (77.39%) than the previous experiment. For the nutrients that did not match our expected results, only *Iron* has a substantial weight, the others average only 0.94% of weight. Also in the updated visualization (Figure 7b), the rye and triticale cluster are much closer than in the initial visualization.

The experiments in this section show that when given an appropriate tool to explicitly identify the implicitly involved unmoved points, our mental model of the data can be better captured by the V2PI-MDS algorithm.

## 5.2 The pairwise weighting scheme

In the previous section, we used a simple scheme to specify each $c_{ij}$. The scheme worked because the number of moved objects and highlighted objects was fairly balanced. For unbalanced case, we need smart ways to specify $c_{ij}$. We use three experiments (numbered 4, 5



(a) The explicit interactions     (b) The updated visualization
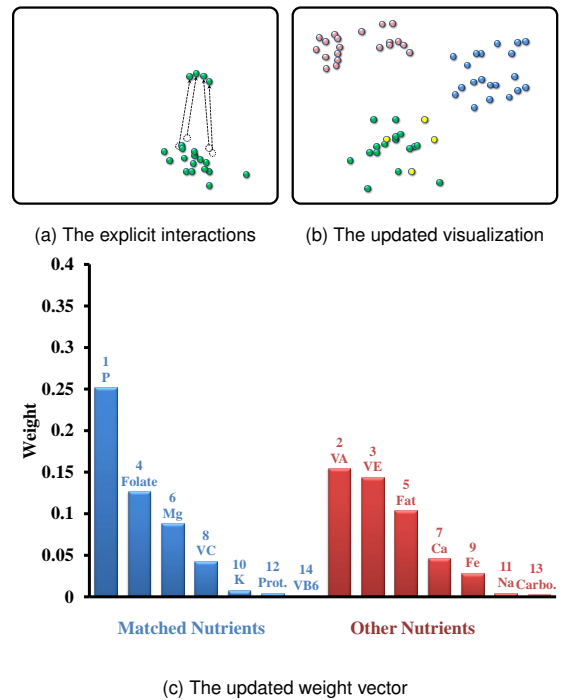


(c) The updated weight vector

Fig. 6. The second experiment when the triticale cluster is highlighted. (a) shows the explicit interactions in this experiment. (b) shows the updated visualization. The cluster structure does not change much. (c) shows the side-by-side bar graph of the weight vector.



(a) The explicit interactions     (b) The updated visualization
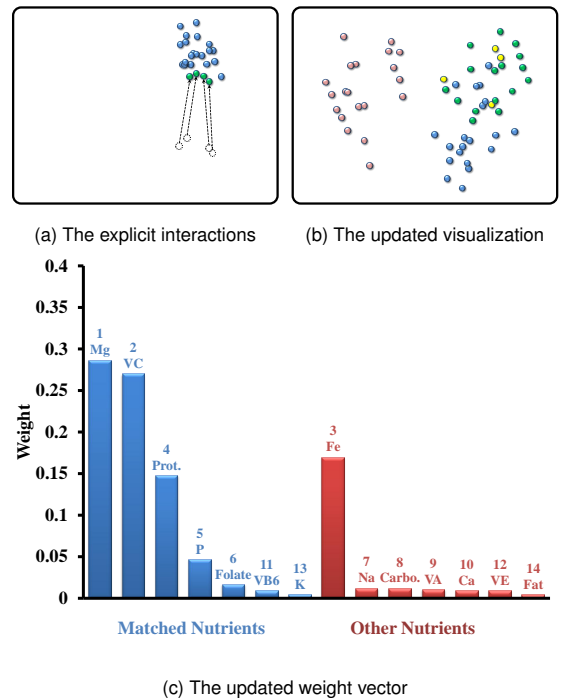


(c) The updated weight vector

Fig. 7. The third experiment when the rye cluster is highlighted. (a) shows the explicit interactions in this experiment. (b) shows the updated visualization. (c) shows the side-by-side bar graph of the weight vector.

and 6) to exemplify the role of $c_{ij}$. For experiment 4, consider Figure 8. It is almost identical to the previous experiment, except that the rye cluster has quadrupled in size. The goal of the study and our explicit interactions are the same, also the entire rye cluster are again highlighted as reference. Figure 9 plots the updated weight using the

simple scheme as in previous experiments shown in Table 2. Noted the result differs from Figure 7c. The top 7 nutrients suggested by the algorithm are *Iron*, *Fat*, *vitamin B6*, *Folate*, *protein*, *Phosphorus* and *vitamin E*. Not only does the number of matched nutrients drop from five to four, but their ranks are also lower (No. 3, 4, 5, 6). The matched nutrients' combined weight dropped to less than one half, 48.88%. Besides these issues, the most alarming problem is that the weight vector has not changed much from the uniform weight. The algorithm failed to extract our mental model of the data from the visual interactions.
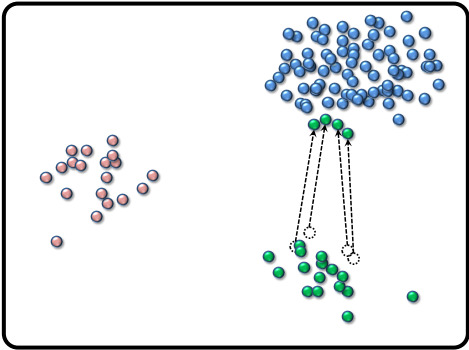


Fig. 8. The initial visualization and the our visual interactions in the experiment in Section 5.2. This experiment has more highlighted rye objects in the cluster. The moving interactions are identical to the experiments in the last section.

Table 2. The simple pairwise weighting scheme used in experiment 4.

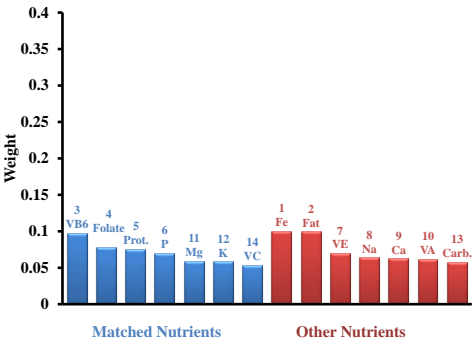| moved | highlighted | untouched | |
|---|---|---|---|
| 1 | 1 | 0 | **moved** |
| | 1 | 0 | **highlighted** |
| | | 0 | **untouched** |



Fig. 9. The updated weight vector when the algorithm uses a simple pairwise weighting scheme that weighs all explicit-to-explicit pairs equally.

There are 84 objects with which the we interacted in experiment 4 (4 moved triticale objects and 80 highlighted rye objects, see Figure 10). There are 6 triticale-to-triticale pairs (moved-to-moved), 320 triticale-to-rye pairs (moved-to-highlighted) and 3160 rye-to-rye pairs (highlighted-to-highlighted). With the simple weighting scheme, these 3486 pairs receive equal, non-zero weights in the minimization process (Equation (2)). The rye-to-rye pairs account for 91% of the total pairs in the stress function and the distances of these pairs are unchanged in the interaction. Therefore their stresses will be minimized by the uniform weight vector. Consequently, the weight vector $\tilde{\omega}$ could not change dramatically as it would increase the stresses of
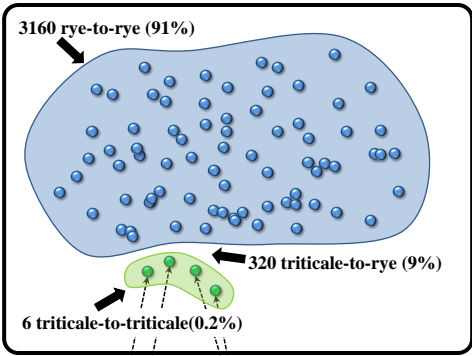


Fig. 10. A close-up of the explicit interactions in the experiment in Section 5.2. The explicit-to-explicit interactions can be categorized into three groups; 1) highlighted-to-highlighted pairs within the blue circle, 2) moved-to-moved pairs within the green circle and 3) moved-to-highlighted pairs between the blue and green circles.

Table 3. The pairwise weighting scheme used in experiment 5.

| moved | highlighted | untouched | |
|---|---|---|---|
| $\frac{2}{n_1(n_1-1)} = \frac{1}{6}$ | $\frac{1}{n_1 n_2} = \frac{1}{320}$ | 0 | **moved** |
| | $\frac{2}{n_2(n_2-1)} = \frac{1}{3160}$ | 0 | **highlighted** |
| | | 0 | **untouched** |

these pairs. The result is $\tilde{\omega}$ becomes unsensitive to the interpretation associated with moved objects.

To increase the sensitivity, we improve the way to specify values for each $c_{ij}$. We allows each $c_{ij}$ to change its value as $i$ and $j$ changes. The weights of pairs in large clusters should be penalized and the weights of pairs in small clusters should be amplified. Specifically, for experiment 5, we use the default setting of pairwise weight shown in Table 3 and assigned $\frac{1}{6}$, $\frac{1}{320}$, $\frac{1}{3160}$ and 0 to moved-to-moved, moved-to-highlighted, highlighted-to-highlighted pairs and the rest of the pairs. The updated weight vector after applying this weight scheme is shown in Figure 11. The top seven nutrients suggested are *vitamin C*, *protein*, *Phosphorus*, *Sodium*, *Calcium*, *Folate* and *Iron*. Although the number of matched nutrient did not increase, their ranks are significantly higher. More importantly, the weight vector is dramatically different from uniform vector and the combined weight of the matched terms is clearly larger than before.

Although this scheme works reasonably well for a wide range of examples, it has drawbacks. It may not be optimal for all applications, including this study. The most compatible interactions in this study were the moved-to-highlighted pairs, and we have already shown that the interpretation of the moved-to-moved pairs is irrelevant to the goal of this study. In fact, the pairwise weighting scheme in experiment 5 puts 50 times more weight on a moved-to-moved pair than a moved-to-highlighted pair. This decision does not suit the goal of the study very well. An important property of our new V2PI-MDS algorithm is the pairwise weights are changeable. Hence, the performance can be further optimized by increasing the weights of moved-to-highlighted pairs and decreasing the weights of moved-to-moved pairs. We consequently changed the weight for moved-to-moved, moved-to-highlighted and highlighted-to-highlighted to 0, $\frac{1}{80}$ and $\frac{1}{3160}$ (see Table 4) for experiment 6. The new result scores six matches and the combined weight of the seven nutrient we are looking for is particularly large (87%) (Figure 12).

Experiments in this section exemplify the impact of the pairwise weighting scheme on capturing our mental model of the data. We showed that our default equations for pairwise weight work much better than the simple scheme in our previous implementation of the V2PI. The improvement can be greater if we enable users to tweak the weight for a particular question.
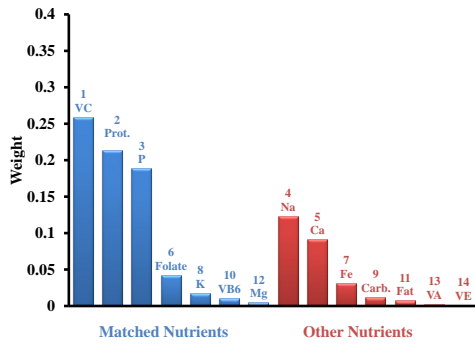
Fig. 11. The updated weight vector when the algorithm uses the pairwise weighting scheme in Table 3.

Table 4. The pairwise weighting scheme used in experiment 6, Note the weights of moved-to-moved and moved-to-highlighted pairs has been decreased and increased, respectively.

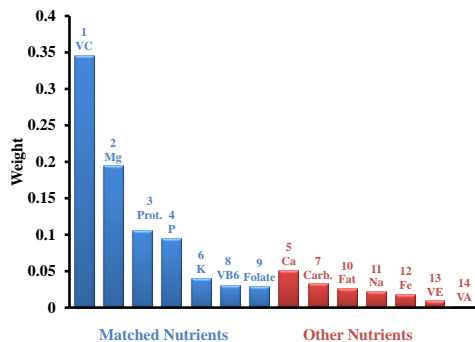| moved | highlighted | untouched | |
|---|---|---|---|
| 0 | $\frac{1}{80}$ | 0 | **moved** |
| | $\frac{1}{3160}$ | 0 | **highlighted** |
| | | 0 | **untouched** |



Fig. 12. The updated weight vector when the algorithm uses a optimized pairwise weighting scheme for this particular example.

# 6 RELATED WORK

Many existing systems generate spatializations from dimension reduction algorithms (e.g., multidimensional scaling (MDS), principle component analysis (PCA) [20]). These systems typically employ parametric level interaction to foster sensemaking. Tools, such as iPCA [12], STREAMIT [1], and IN-SPIRE's "Galaxy View" [22], update the visualizations as users tweak parameters. The updates give users immediate feedback in response to their actions. This allows users without a statistical background to explore the data and learn how their actions impact the visualization. However, without a solid understanding of the algorithm(s) driving the layout, it is possible that the user could miss important insights into their data.

Other systems allow the user to adjust the spatialization in order to update the underlying models. For example, Torres et al. developed a system where users are able to move images towards or away from a single image in order to portray pair-wise similarity or dissimilarity [21]. From there, an image retrieval algorithm determines the features and dimensions shared between the images that the user has determined as being similar. Spatializations of document sets exist that allow users to place objects of interest into the layout. VIBE [15] allows users to define multiple objects of interest in the spatial

Table 5. $|Y_1|$ and $|Y_2|$ are the number of objects in two explicit interacted groups and $N$ is the total number of objects in the dataset

| explicit 1 | explicit 2 | untouched | |
|---|---|---|---|
| 1 | $\frac{N(N-1)}{|Y_1||Y_2|} - 1$ (a) | 1 | **explicit 1** |
| | 1 | 1 | **explicit 2** |
| | | 1 | **untouched** |

layout that correspond to a series of keywords describing a subject matter of interest to the user. Similarly, Dust & Magnet allows users to place a series of Magnets representing keywords into the space and observe how documents are attracted or repelled from the locations of these magnets [23]. iCluster[6] enables users to directly manipulate the cluster contents while the clustering algorithm learns to improve the suggested cluster structure according to the user's changes.

Most similar to our system is Dis-Function [14][3]. Dis-Function allows users to move objects in the data spatialization, which in turn updates the distance function of MDS, which drives the layout. However, there is a key difference in the way user's mental model of the data is interpreted. Dis-Function uses the following formula

$$\underset{\omega^{t+1}}{\text{argmin}} \sum_{0 \leq i < j \leq n} L_{ij} \left( \delta_{ij}^{(\omega^{t+1})} - U_{ij} \delta_{ij}^{(\omega^t)} \right)^2.$$

This stress function is a variant that compares high dimensional distances against high dimensional distances, while in V2PI-MDS and the original MDS algorithm, the stress function always compare high dimensional distances versus their low dimensional counterparts. However, the Dis-function formula can be rewritten in a high-D-versus-two-D representation based on the definition of $U_{ij} = \frac{\tilde{\gamma}_{ij}}{\gamma_{ij}}$.

$$\underset{\omega^{t+1}}{\text{argmin}} \sum_{0 \leq i < j \leq n} L_{ij} \left( \delta_{ij}^{(\omega^{t+1})} - \frac{\delta_{ij}^{(\omega^t)}}{\gamma_{ij}} \tilde{\gamma}_{ij} \right)^2.$$

The motivation of adjusting $\tilde{\gamma}_{ij}$ by $\frac{\delta_{ij}^{(\omega^t)}}{\gamma_{ij}}$ is not clear. However, when the MDS visualization is perfect (i.e., the high-D distances equal their 2D counterparts, $\frac{\delta_{ij}^{(\omega^t)}}{\gamma_{ij}} = 1$), Dis-Function and V2PI-MDS are identical(in this aspect). Dis-Function also uses a different pairwise weighting scheme. The value of $L_{ij}$ is defined in Table 5. These weights emphasize the distance changes between two groups of objects (i.e., cell (a)) that the user identifies. The total weight of these pairs equals the total weight of all other pairs. However, it assumes that all unmoved objects are implicitly involved in the interaction, which adds a significant computational burden. This is less efficient and might not match the user's intended semantic input.

# 7 FUTURE WORK

One of the questions we would like to investigate further concerns the values specified for each $c_{ij}$ in Equation (2). Currently, the values are pre-set according to Table 2. Pre-set weights are not necessarily optimal for all scenarios. A simple solution would be running several instances of V2PI-MDS with different weights and letting the user to choose the best one. Another possibility would be to identify a vocabulary of common movement operations, and setting the weights in Table 2 accordingly. More ideally, we would incorporate each $c_{ij}$ into the parameter space and learn their values interactively. Not only could this save us the effort to pre-set each $c_{ij}$ manually, it would also prompt sensemaking and analytical reasoning in a higher level. Also, instead of one moved, highlighted and untouched set, there could be multiple moved or highlighted sets. The refinement of these sets would help V2PI-MDS to interpret more complicated spatial interactions.

In addition, the MDS algorithm uses one weight vector to describe the dissimilarities between all objects. It is possible that a user has different definitions of dissimilarity in different subsets of objects. To handle that, we probably need a more flexible MDS algorithm that uses multiple weight for different pairs.

## 8 CONCLUSION

We have presented a tool that allows users to be more specific in the meaning of their interactions by moving objects and having the option to highlight unmoved objects, resulting in a spatialization algorithm that preserves these semantics. In addition to enabling these interactions, we have introduced a modified pairwise weighting scheme to alleviate the problem of over-emphasizing large clusters. Both modifications have shown promising improvements to their predecessors. The results presented here have helped us to define a more flexible and accurate V2PI-MDS algorithm that handles a rich set of user interactions.

### ACKNOWLEDGMENTS

### REFERENCES

[1] J. Alsakran, Y. Chen, Y. Zhao, J. Yang, and D. Luo. Streamit: Dynamic visualization and interactive exploration of text streams. In *Pacific Visualization Symposium (PacificVis), 2011 IEEE*, pages 131–138. IEEE, 2011.

[2] C. Andrews, A. Endert, and C. North. Space to think: large high-resolution displays for sensemaking. In *Proceedings of the 28th international conference on Human factors in computing systems*, pages 55–64. ACM, 2010.

[3] E. T. Brown, J. Liu, C. Brodley, and R. Chang. Dis-function: Learning distance functions interactively. In *IEEE Conference on Visual Analytics Science and Technology (VAST)*, 2012.

[4] J. D. Carroll and J.-J. Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of eckart-young decomposition. *Psychometrika*, 35(3):283–319, 1970.

[5] D. Carver. Designing the user interface, strategies for effective human-computer interaction. *Journal of the American Society for Information Science*, 39(1):22–22, 1988.

[6] S. M. Drucker, D. Fisher, and S. Basu. Helping users sort faster with adaptive machine learning recommendations. In *Human-Computer Interaction–INTERACT 2011*, pages 187–203. Springer, 2011.

[7] A. Endert, L. Bradel, and C. North. Beyond control panels: Direct manipulation for visual analytics. *Computer Graphics and Applications, IEEE*, 33(4):6–13, 2013.

[8] A. Endert, P. Fiaux, and C. North. Semantic interaction for sensemaking: Inferring analytical reasoning for model steering. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2879–2888, 2012.

[9] A. Endert, S. Fox, D. Maiti, S. Leman, and C. North. The semantics of clustering: analysis of user-generated spatializations of text documents. page 555–562, 2012.

[10] A. Endert, C. Han, D. Maiti, L. House, and C. North. Observation-level interaction with statistical models for visual analytics. In *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on*, pages 121–130. IEEE, 2011.

[11] L. House, S. Leman, and C. Han. Bayesian visual analytics (bava). *FO-DAVA Technical Report2010*, 2010.

[12] D. H. Jeong, C. Ziemkiewicz, B. Fisher, W. Ribarsky, and R. Chang. ipca: An interactive system for pca-based visual analytics. In *Computer Graphics Forum*, volume 28, pages 767–774. Wiley Online Library, 2009.

[13] S. C. Leman, L. House, D. Maiti, A. Endert, and C. North. Visual to parametric interaction (v2pi). *PloS one*, 8(3):e50474, 2013.

[14] J. Liu, E. Brown, and R. Chang. Find distance function, hide model inference. In *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on*, pages 289–290. IEEE, 2011.

[15] K. A. Olsen, R. R. Korfhage, K. M. Sochats, M. B. Spring, and J. G. Williams. Visualization of a document collection: The vibe system. *Information Processing & Management*, 29(1):69–81, 1993.

[16] J. Patten and H. Ishii. A comparison of spatial organization strategies in graphical and tangible user interfaces. In *Proceedings of DARE 2000 on Designing augmented reality environments*, pages 41–50. ACM, 2000.

[17] A. C. Robinson. Collaborative synthesis of visual analytic results. In *Visual Analytics Science and Technology, 2008. VAST'08. IEEE Symposium on*, pages 67–74. IEEE, 2008.

[18] S. S. Schiffman, M. L. Reynolds, and F. W. Young. *Introduction to multidimensional scaling: Theory, methods, and applications*. Academic Press New York, 1981.

[19] J. J. Thomas and K. A. Cook. *Illuminating the path: The research and development agenda for visual analytics*. IEEE Computer Society Press, 2005.

[20] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

[21] R. S. Torres, C. G. Silva, C. B. Medeiros, and H. V. Rocha. Visual structures for image browsing. In *Proceedings of the twelfth international conference on Information and knowledge management*, pages 49–55. ACM, 2003.

[22] J. A. Wise, J. J. Thomas, K. Pennock, D. Lantrip, M. Pottier, A. Schur, and V. Crow. Visualizing the non-visual: Spatial analysis and interaction with information from text documents. In *Information Visualization, 1995. Proceedings.*, pages 51–58. IEEE, 1995.

[23] J. S. Yi, R. Melton, J. Stasko, and J. A. Jacko. Dust & magnet: multivariate information visualization using a magnet metaphor. *Information Visualization*, 4(4):239–256, 2005.