Manipulating Bilevel Feature Space for Category-Aware Image Exploration

Kazuyo Mizuno* Hsiang-Yun Wu† Shigeo Takahashi‡

The University of Tokyo

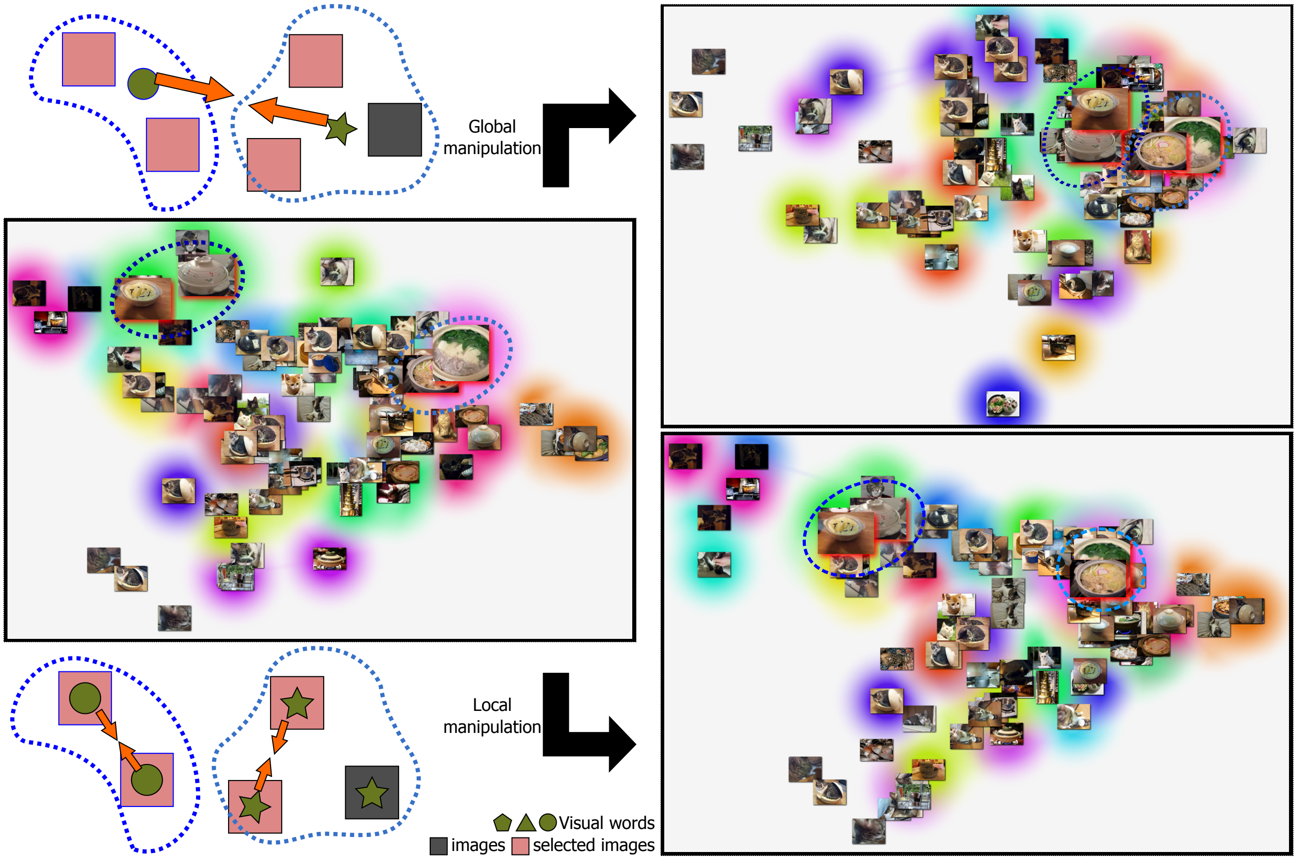


Figure 1: Manipulating feature space for categorizing images. Global manipulation allows us to edit the closeness between image categories on the screen space as shown at the top right, while local manipulation controls the distances between images within each category as depicted at the bottom right. Here, different background colors are assigned to image categories where images in each category are tied with edges of the same color.

Abstract

The demand for interactively designing the image feature space has been increasing due to the ongoing need for image retrieval, recognition, and labeling. Although conventional methods provide an interface for locally rearranging such a feature space, category-level global manipulation is still missing and thus manually rearranging the overall image categorization usually requires a time-consuming task. This paper presents a novel approach to exploring images in the database through the manipulation of bilevel feature space representations, where the upper- and lower-level representations characterize the global categories and local features of the images, respectively. In this approach, the upper-level space describes similarity relationship among the underlying categories extracted from the bag-of-features model, while the lower-level space encodes the closeness between a pair of images within the same category. The key idea behind this approach is to associate the relationship between the two feature spaces with a two-layered graph representation and project it onto 2D screen space using pivot MDS for user manipulation. Experimental results are provided to demonstrate that our approach allows users to understand the entire structure of the given image dataset and reorganize the layout according to their preferences both locally and globally.

Keywords: Image exploration, feature space manipulation, bag-of-features, dimensionality reduction

Index Terms: I.3.6 [Computer Graphics]: Methodology and Techniques; H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems

*e-mail: mizunok@visual.k.u-tokyo.ac.jp
†e-mail: yun@visual.k.u-tokyo.ac.jp
‡e-mail: takahashis@acm.org

1 Introduction

Recent development of web infrastructure has resulted in the rapid increase in the number of images accessible to common users of the Internet. The demand for managing such large-sized image datasets promoted many commercial image database systems including Flickr and Google Image Search. Nonetheless, seeking a more effective means of retrieving specific images from the image database systems still poses a variety of interesting problems from a technical point of view. One of the common solutions is to encode images in a feature vector format and then plot the respective images in the feature space for estimating the similarity between them. Dimensionality reduction has often been employed to project the high-dimensional feature space on the two-dimensional (2D) screen so that we can visualize the underlying structure of the image dataset. In practice, this scheme makes it easy for us to edit the layout of the images through the screen space manipulation for more effective image retrieval.

Although this style of image exploration is intuitive, it still suffers from a problem of the *semantic gap*, which is defined to be the gap between the semantics of images recognized by the users and their relative positioning within the feature space. The primary reason for this problem is that existing approaches only allow us to change local arrangement of images in the feature space, which is not necessarily coincident with the image semantics in our mind. This means that we usually have our own categorization of the images and try to rearrange the images in the feature space to best meet our expectations. Nonetheless, editing local configuration of the image feature space is not still sufficient for this purpose and thus new high-level operations for organizing the overall categorization are required.

This paper presents an approach to manipulating both the local and global arrangements of images by projecting the overall feature space onto the 2D screen space. For this purpose, we incorporated bilevel feature space representation so that we can associate the global categories and local features of the images with the upper- and lower-level graph structures, respectively. Here, the upper-level graph characterizes the relationships among the underlying image categories extracted from the bag-of-features model, while the lower-level graph shows local closeness between a pair of images within the same category. The two graphs are arranged in a hierarchical fashion so that we can describe how each image belongs to a specific set of image categories. The entire feature space of the images can efficiently be transformed into 2D screen space by taking advantage of a dimensionality reduction technique called pivot MDS, where we employ representative feature vectors of the image categories as the pivots for accelerating the computation of large-scale eigenproblems. Experimental results are also provided to demonstrate that how our approach helps users explore specific images in the feature space through local and global operations. Figure 2 shows the overview of the proposed approach.

Figure 1 presents how we can manipulate the relative positions of images using the proposed approach. Here, global manipulation allows us to edit the closeness between image categories on the screen space as shown at the top right of the figure, while local manipulation basically controls the distances between images within each category. In our system, different background colors are assigned to the images by referring to their primary categories while those in the same category are tied with edges of the same color. This color-based rendering scheme gives us a perceptually plausible guide to identify the images of the same category, and further helps us perform category-aware manipulation of the entire feature space as well as local manipulation of image arrangement.

The remainder of this paper is structured as follows: Section 2 provides a brief survey on existing techniques related to our approach. Section 3 describes the representation of the bilevel feature space together with its projection onto the 2D screen space. Section 4 explains implementation details on how we can interactively explore images in the feature space through the screen space manipulation. After presenting experimental results together with discussions in Section 5, we conclude this paper and refer to possible future extensions in Section 6.

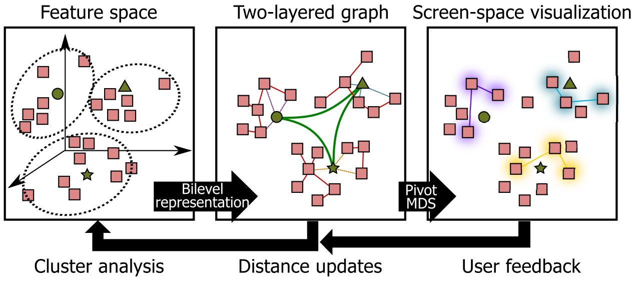


Figure 2: Overview of the proposed approach.

2 Related Work

Filling the semantic gap in the data exploration has been a very important subject, and visualization often helps us effectively explore the feature space of the data, as done in interactive analysis of text semantics for example [6]. In this section, we focus on the problem of image exploration in the feature space, and provide a brief survey on two relevant research topics: visualizing image datasets and dimensionality reduction.

2.1 Visualizing Image Datasets

As the number of images increases in the dataset, more sophisticated visual metaphors should be invented to effectively visualize the underlying structure of the corresponding image feature space. This indeed allows us to understand the semantic configurations of the overall feature space, and further explore a specific set of images through the screen-space interactions.

Eler et al. [5] and Paiva et al. [19] proposed a tree structure named *similarity-tree* for image dataset, which successfully clarified the hierarchical structures hidden behind the image datasets. While the similarity-tree representation helps us explore the image feature space, it is still straightforward representation of hierarchical relationships among images and often not intuitive enough for users to search for specific images of interest. In particular, the spatial relationship between images in the screen space dost not faithfully reflect their similarity. This means that a pair of image nodes are not necessarily close to each other even when they are within a small neighborhood on the screen space, because they may be connected via a parent node at long distance. Furthermore, the computational cost for laying out the similarity trees is relatively high and thus not appropriate for interactive environments in our setting. Kennedy and Naaman [13] and Heath et al. [10] identified a new type of relationship between a pair of images if they contain similar objects, and visualized such relationships by drawing the associated graph layouts. The extracted graph representation makes it possible to compute the dissimilarity between images, and thus provides an effective means of retrieving images that are similar to the key image. However, these methods are again computationally expensive and thus cannot be directly employed for our purpose. Fan et al. [7, 8] proposed an approach to extracting hierarchical structures of the image regions by employing both feature space analysis and textual annotations. This approach successfully identifies image semantics by referring to the extracted hierarchical structures while the quality of the image classification strongly depends on the accuracy of the textual annotations.

On the other hand, Thomee et al. [28, 27] implemented a browser for interactively exploring images in the feature space through a graphical user interface. This approach can effectively incorporate feedback from users for intuitive manipulation of the image feature space. Nonetheless, the proposed framework cannot restrict the updates in the local configuration of the feature space, and can unnecessarily modify the layout of images in which users are not interested at all. Another approach to browsing large image datasets has been proposed by Brivio et al [2], where they employed Voronoi diagrams as the underlying layout of the images. While this provides an effective interactive environments, it does not allow users to edit the configuration of the feature space, unfortunately.

2.2 Dimensionality Reduction

Visualizing datasets in high-dimensional space has been an important technical problem in the last decade. Dimensionality reduction schemes usually provide us with powerful solutions to this problem in the sense that they project such high-dimensional data onto low-dimensional space where we can visually inspect the underlying structure of the datasets. One traditional scheme for dimensionality reduction is *multidimensional scaling* (MDS) [29, 14], which provides the arrangement of data samples in the low-dimensional space by maximally respecting the distance between every pair of samples in the original high-dimensional space. Indeed, MDS has often been employed for browsing images by projecting high-dimensional feature space onto the 2D screen space, where various types of similarities were incorporated for the purpose of image retrieval [21, 16, 23]. A *self-organizing map* (SOM) is another popular technique to transform data samples in high-dimensional space specifically to 2D, and several SOM-based techniques for image browsing and retrieval have been developed [15, 26].

Recently, user-driven dimensionality reduction approaches have been proposed so that users can interactively manipulate the configuration of the high-dimensional feature space according to their preferences. Mamani et al. [18] employed a sophisticated framework called *local affine multidimensional projection* [12] for this purpose, and realized a visualization system that allows users to edit the structure of the high-dimensional feature space by interactively updating the local configuration of images through the 2D screen space. One of the most relevant work has been done by Paulovich et al. [20], where they developed an approach to interactively manipulating the feature space through the piecewise Laplacian-based projection. Their method is similar to our method in that they employed a neighborhood graph for defining local similarity among images in the feature space, and allows users to edit the graph connectivities through the 2D screen space manipulation. Nonetheless, their method primarily focused on manipulation of local configurations in the feature space. On the other hand, our proposed approach provides an effective means of editing both the local and global configurations while retaining the consistency among them, by incorporating bilevel representation of the feature space.

3 Constructing the Bilevel Feature Space

As described earlier, we provide global operations for rearranging clustered sets of the images in the feature space, as well as local ones for adjusting the closeness of a specific image to its neighbors. In particular, the global manipulation will allow us to change the configuration of the feature space by referring to the underlying categorization of the images. For this purpose, we employ bilevel representation of the feature space and prepare a two-layered graph structure for that representation, as shown in Figure 3. In the rest of this section, we explain three steps for the construction of the initial representation of the bilevel feature space: categorizing images by their features, constructing the two-layered graph, and projecting the feature space onto the screen space.

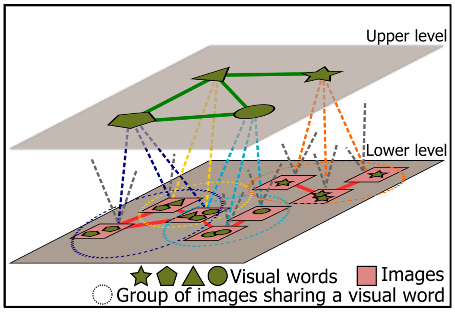


Figure 3: Bilevel feature space representation and its associated two-layered graph.

3.1 Categorizing Images by Their Features

For initializing the bilevel feature space, we have to extract an underlying categorization of the images so that we can compose category-based hierarchy over the image dataset. In this work, we introduced the bag-of-features model [25, 3], which has been recently popular for image recognition and retrieval since it can significantly accelerate the associated image exploration and retrieval and also improve their accuracy. The construction of the bag-of-features model commonly begins with extracting SIFT features [17] from each image in the dataset, where the SIFT descriptor is known to be robust enough for affine transformations and variation in view, lighting and occlusion conditions. Since a SIFT feature is usually encoded as a 128-dimensional feature vector, we plot each SIFT feature as a point sample in the 128-dimensional Euclidean space first of all. We then apply the k-means clustering to the set of feature vectors and extract the center of each cluster as the *visual word*, which has been considered as a representative of some category implicitly defined in the bag-of-features model. In practice, we employed the visual words as the nodes of the upper-level graph in the proposed feature space representation, as shown in Figure 3. As for the number of clusters, we empirically set $k = 100$, while k can be adjusted according to the characteristics of the given image dataset.

According to the bag-of-features model, we can represent each image as a histogram with respect to the visual words we have extracted. This can be accomplished by projecting each SIFT feature contained in that image to the closest visual word in the 128-dimensional feature space. In practice, the histogram of the image can be composed by counting the occurrences of each corresponding visual word, as shown in Figure 4. This encoding based on the vector quantization now transforms the 128-dimensional SIFT feature vector of each image into a new k-dimensional vector representation with respect to the k extracted visual words, which usually allows us to enjoy a sparse vector representation of the image by associating it with a small number of representative features, i.e., the visual words. Note that, in our implementation, common weights known as *tf-idf* (term frequency - inverse document frequency) [25] has been applied to the histogram representation, in order to take into account the frequency of each visual word in the entire set of images. We use this histogram representation of the images as the fundamental for constructing the bilevel feature space.

3.2 Constructing the Two-Layered Graph

Our next step is to construct the two-layered graph by incorporating the bag-of-features model we have composed, as shown in Figure 4. Actually, by the presence of the two-layered graph, we can describe the relationships among image categories with the upper-level graph while associate local arrangements of images with the lower-level graph. This type of graph data structure allows us to

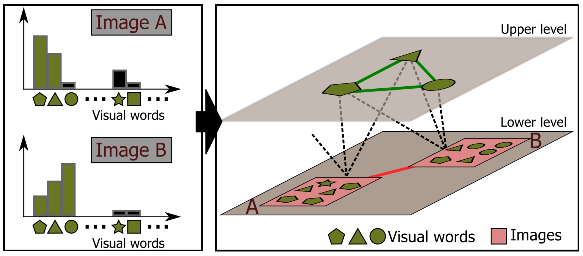


Figure 4: Histogram representations of images.

perform fuzzy clustering of images in the feature space and thus implicitly represent semantic polysemy of the images also. The construction of the two-layered graph consists of three parts, i.e., constructing the upper-level graph, connecting edges between the two graphs, and constructing the lower-level graph, each of which will be detailed in what follows.

3.2.1 Constructing the Upper-Level Graph

Let us denote the upper-level graph G_u in this paper. As mentioned previously, we represent each extracted visual word as the node of G_u, which also serves as a representative of some image category in our approach. For connecting the nodes within G_u, we would like to find some meaningful correlation between every pair of the visual words. For that purpose, we define a distance metric between the nodes v_i and v_j in G_u as follows:

$$\begin{aligned} d_g(v_i,v_j) &= rL_{\min}+(1-r)L_{\max}, \text{ where} \\ r=\frac{|J|}{|I|} &\quad \text{and} \quad J=\{s\in I \mid H_i(s)>0 \text{ and } H_j(s)>0\}. \end{aligned} \tag{1}$$

Here, I represents the entire set of input images, $H_i(s)$ is the histogram value of the i-th visual word v_i contained in the image $s \in I$, and $|J|$ is the number of images contained in the image set J. We also set $L_{\min} = 1.0$ and $L_{\max} = 2.0$ by default in our setting. This means that we define the closeness between v_i and v_j by counting the number of images that contain both the i-th and j-th visual words as the representative features. In the actual construction of G_u, we connect each node with its l-nearest neighbors by employing Eq. (1), where l is set to be 5 in our implementation.

3.2.2 Connecting Edges Between the Two Graphs

Suppose that we represent the lower-level graph as G_l, where its node corresponds to an image of the given dataset in the proposed two-layered graph representation. After having constructed the upper-level graph G_u, we try to connect edges between the nodes of G_u to those of G_l. This is because in our approach we would like to associate each image with a specific number of visual words that are matched with major features contained in that image.

For finding edges emanating from an image node s in G_l, we first search for the visual word v_1 that has the maximum histogram value of the image s, and insert an edge from s in G_l to v_1 in G_u. We then choose a visual word v_2 if it has the largest histogram value of the image s among those adjacent to v_1, and connect s and v_2 with an edge. We perform the iteration for finding the next visual word among those adjacent to the already selected visual words in G_u, until we can insert m edges from s to the nodes in G_u. Note that the m edges incident to s may be reconnected through interactive manipulation of the feature space, which will be described later in Section 4.

3.2.3 Constructing the Lower-Level Graph

Finally, we consider how to connect nodes within the lower-level graph G_l in our approach. In practice, we seek n-nearest neighbors of each image node as its adjacent nodes in G_l, in a similar way to in G_u. Nonetheless, we use a different metric for evaluating the distance between image nodes in G_l. This is because every image node now largely depends on a specific number of visual word nodes in G_u, and thus we should respect such association with the underlying image categorization when positioning the image node. This leads us to the idea of locating each image node in G_l using a barycentric coordinate system with respect to its adjacent visual word nodes in G_u. Our approach employs the formulation by Rustamov et al. [22] for this purpose, where we can define the distance metric between the image nodes s_i and s_j in G_l as

$$d_l(s_i,s_j) = \sqrt{-\frac{1}{2}(\boldsymbol{b}(s_i)-\boldsymbol{b}(s_j))^{\mathrm{T}}\boldsymbol{D}(\boldsymbol{b}(s_i)-\boldsymbol{b}(s_j))}. \tag{2}$$

Here, $\boldsymbol{D}$ is the matrix of squared distances with respect to the visual word nodes in G_u, and $\boldsymbol{b}(s_i)$ conforms to the barycentric coordinates of the image node s_i. Note that the (i,j)-component of $\boldsymbol{D}$ can be calculated as the squared distance between the visual words v_i and v_j using Eq. (1). Furthermore, the i-th barycentric coordinate of an image node s can be obtained by

$$\boldsymbol{b}_i(s) = \begin{cases} \dfrac{H_i(s)}{\sum_{j\in A} H_j(s)} & (j \in A) \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

where A is a set of visual word nodes in G_u that are adjacent to s in G_l. Again, the parameter n can be adjusted by users, while it is empirically set to be 3 in our implementation.

Note that we can use Eq. (2) also for evaluating the length of an edge between the image node s in G_l and visual word node v in G_u. More specifically, we can rewrite the barycentric coordinates Eq. (3) for the visual word node v_j as follows:

$$\boldsymbol{b}_i(v_j) = \begin{cases} 1 & (i=j) \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

3.3 Projecting the Feature Space onto the Screen Space

The last step of initializing the bilevel feature space is to project it to 2D so that we can allow users to interactively explore the images in the dataset and rearrange their arrangement through the screen space. Indeed, the problem of visualizing high-dimensional data has been a common problem and not limited to image dataset, and thus various dimensionality reduction methods are proposed for general use. The aforementioned *multidimensional scaling* (MDS) can be thought of as one classical solution of this dimensionality reduction problem, where it transforms high-dimensional point samples onto low-dimensional space using mutual distances among the the samples in the original high-dimensional space.

In this classical MDS setting, however, the associated computation takes more time as the number of samples increases, since we have to solve the eigenproblems of the size equal to the squared number of samples. Landmark MDS [4] accelerates the computation by first calculating the low-dimensional layouts of the smaller set of representative samples, and then successfully locates other samples in the low-dimensional space by referring to the positions of these representatives. Multilevel MDS called *Glimmer* [11] has also accelerated this type of distance-based dimensionality reduction process by taking advantage of GPU functionalities.

In our approach, we employ the pivot MDS [1] for efficiently projecting bilevel feature space onto the 2D screen space for our purpose. The pivot MDS is a more advantageous method than the

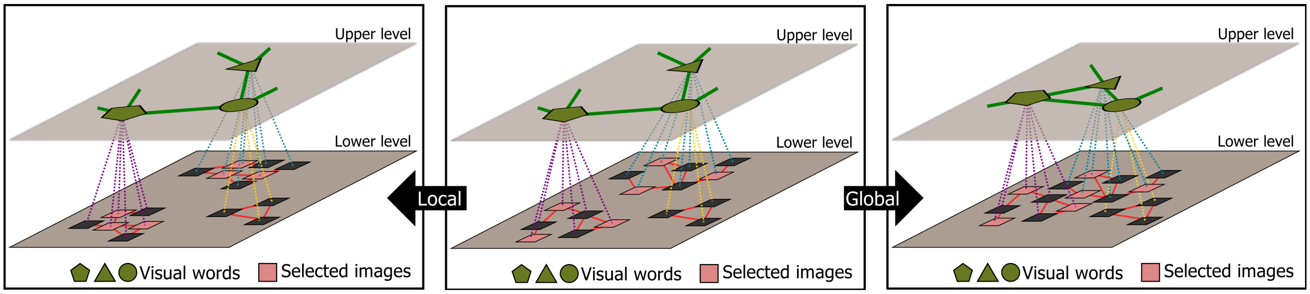


Figure 5: Local and global operations for manipulating feature space.

landmark MDS in that it has been designed to alleviates potential errors in the landmark MDS by taking into account distances of representative samples from other nonrepresentative samples. Note that the representative samples are called *pivots* in [1], and we employ this terminology also in this paper. For projecting the bilevel feature space in our approach, we just employ the visual word nodes in the upper-level graph G_u as the pivots, while we compute the shortest path between every pair of nodes by referring to the distance associated with the edges in the composite two-layered graph, on the assumption that we can freely traverse between the upper-level graph G_u and lower-level graph G_l through the edges in between. This significantly accelerates the projection of the bilevel feature space onto the 2D screen space, which is important for realizing an interactive environment for image exploration.

4 Manipulating the Feature Space

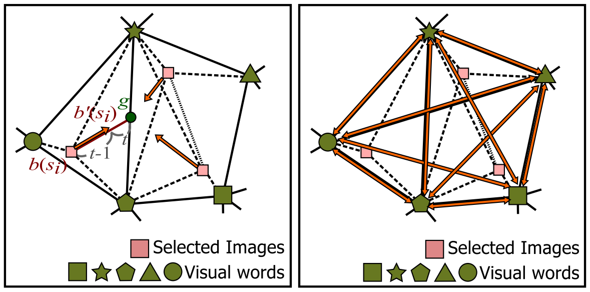
Now we are ready to rearrange the images in the feature space using both global and local manipulations that become available from the bilevel representation of the feature space. Our basic strategy here is to select a set of images we want to manipulate by clicking them on the screen space, and then adjust their displacement by mouse motion. This is accomplished by updating the distance among visual word nodes and image nodes in the two-layered graph we constructed previously as shown in Figure 5 (cf. Section 3). In practice, with the local operations we basically update the lengths of edges incident to image nodes, so that we can manually control the correlation among a specific set of images to make them either closer to or further from each other. On the other hand, the global operations enable us to adjust the lengths of edges between visual word nodes and thus modify the image categorization by increasing/decreasing the correlation between particular pairs of categories. The remainder of this section describes the details of these two types of operations for rearranging the feature space of the images.

4.1 Local Manipulation of the Feature Space

Editing the local arrangements of images helps us adjust the predefined similarity among a specific set of images within a small neighborhood of the feature space. This means that the local manipulation of the feature space allows us to increase the closeness among images to categorize them into the same group, or to decrease the closeness to make them apart from each other in the feature space. Figure 6(a) shows how we can achieve this local manipulation by updating the barycentric coordinates of each target image node.

Suppose that we would like to control the closeness among a set of user-selected images $s_j\,(j \in B)$. In our implementation, we first compute the average barycentric coordinates of the visual word nodes that have connection with the user-selected image nodes as

$$\boldsymbol{g} = \frac{1}{|C|}\sum_{i \in C} \boldsymbol{b}(v_i), \tag{5}$$



(a) (b)

Figure 6: Updating the distance between the nodes in the two-layered graph when manipulating the feature space. (a) local manipulation. (b) Global manipulation.

where C represents the set of the visual word nodes and $|C|$ is the number of nodes in C. The closeness among the selected images is controlled by updating $\boldsymbol{b}(s_j)$ to $\boldsymbol{b}'(s_j)$ as

$$\boldsymbol{b}'(s_j) = (1-t)\boldsymbol{b}(s_j) + t\boldsymbol{g} \quad (j \in B). \tag{6}$$

Here, t is a parameter that can be controlled by users, where we can increase and decrease the closeness among the image set by setting $0 < t(< 1)$ and $t < 0$, respectively in Eq. (6). Note that after updating the barycentric coordinates of the images $s_j(j \in B)$ using Eq. (6), we recalculate the weights of edges between image and visual word nodes and also reconnect edges among image nodes.

4.2 Global Manipulation of the Feature Space

On the other hand, global manipulation of the feature space allows us to design the categorization of the images by altering the similarity between the visual words. This means that the global operation just updates the distance between the visual word nodes selected by users, as illustrated in Figure 6(b).

In this case, we first ask users to select a set of images in the feature space through the screen space. We then extract a set of visual word nodes in the upper-level graph G_u that are adjacent to the nodes of the selected images. For each edge between the visual word nodes v_i and v_j in the set, we update the length d_{ij} to d'_{ij} as:

$$d'_{ij} = \begin{cases} \max\{L_{\min}, \min\{d_{ij} - t(d_{ij} - L_{\min}), L_{\max}\}\} & i, j \in V \\ d_{ij} & \text{otherwise} \end{cases} \tag{7}$$

where V is an index set of the extracted visual word nodes. Note that t is a parameter again that can interactively be edited by users, where we can decrease the distance between the visual word nodes

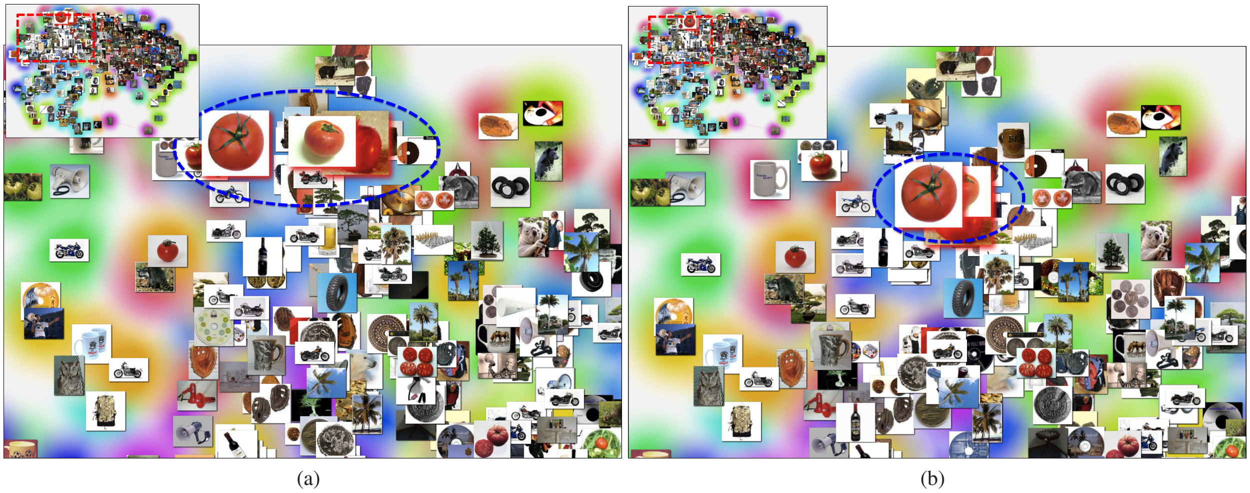


Figure 7: Local manipulation of the feature space: (a) Original layout of images where tomato images are selected. (b) Local manipulation permits us to increase the closeness of the selected two images.

when $0 < t$ and increase it when $t < 0$ while the distance remains to be within the range $[L_{\min}, L_{\max}]$. This process also lets us update the distance matrix among the visual word nodes $\boldsymbol{D}$, which means that we recompute the edge connection among visual word nodes and image nodes in our two-layered graph representation. The illustration on the right of Figure 5 and Figure 6(b) show such cases where the connectivity among the set of extracted visual word nodes is updated with the global manipulation of the feature space.

5 Results and Discussion

In this section, we present experimental results of the proposed approach together with discussion on its possible limitations. Our prototype system has been implemented on a desktop PC with Quad-Core Intel Xeon CPUs (3.2GHz, 8MB cache) and 8GB RAM, and the source code was written in C++ using OpenCV for SIFT feature extraction and matrix computation, and the sqlite and soci library for managing the image database. The interface for visualizing and manipulating the image feature space has been implemented independently with JavaScript.

5.1 Experimental Results

Throughout this experiment, we employed the bag-of-features model to extract 100 visual words from each image dataset, and represented the respective images as a vector of 100 histogram values. In constructing the two-layered graph, we empirically set $l = 5$, $m = 3$, and $n = 3$ as described in Section 3. Indeed, the choice of these parameters comes from the consideration that we should seek compromise between multiple categorization of images and the sparseness of the two-layered graph.

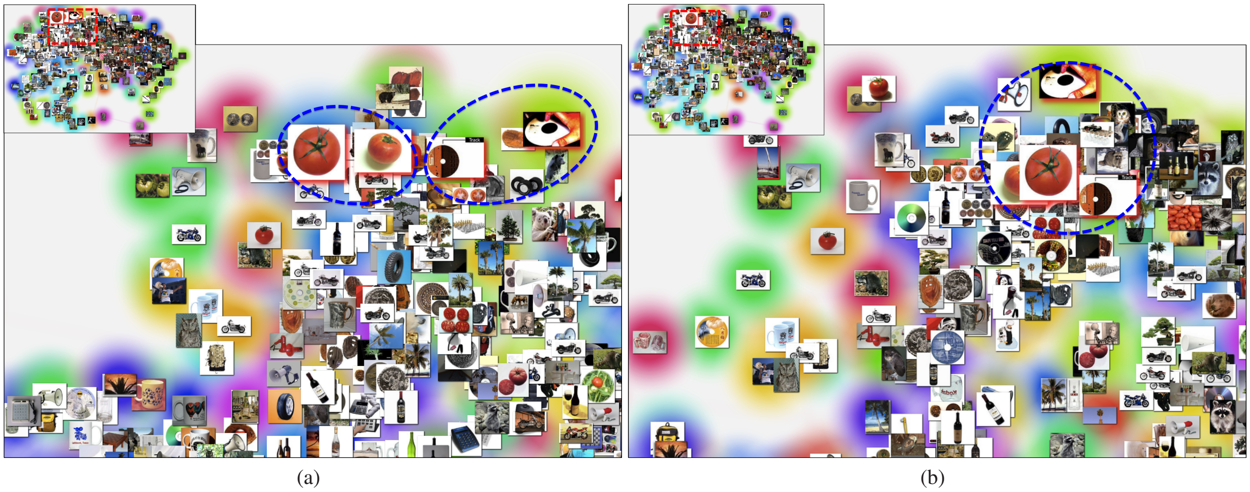
Figure 1 shows an example where we applied local and global manipulation to the image dataset, which has been composed by manually collecting 75 copyright-free photos of cats and containers. As described earlier, we used the global manipulation to increase the closeness between two categories of images of food in a container. On the other hand, the local manipulation just controls the distance between images within each category as shown in the figure. Note that each image has a background color that corresponds to the primary visual word in the sense that the corresponding histogram value is the largest. Furthermore, every image shares an edge with its nearest neighbors within the same category and the edge is rendering in the corresponding color also in our implementation. This color assignment scheme systematically supports our category-aware manipulation of the feature space.

We also employed another image dataset Caltech256 [9] for our experiment. We selected 20 categories in advance and randomly extracted 50 images from each category to collect 1,000 images in total. The proposed local manipulation permitted us to fit similar images into a small space as shown in Figure 7. Here, we first selected tomato images as shown in Figure 7(a), where the selected images were enlarged. We then controlled the closeness between the selected images by mouse motion as shown in Figure 7(b). On the other hand, as shown in Figure 8, we can make two image categories, i.e. tomatos and CDs, closer to each other through the global manipulation of the feature space, by selecting the images contained in those categories. In practice, we focused images of different categories by referring to the background colors as shown in Figure 8(a), and tried to merge the two small categories into one as shown in Figure 8(b). The global manipulation effectively helps us reorganize the image categories inferred by the bag-of-features model. Indeed, in this case, once we merged two categories of images, we could further successfully attract other unselected images that include similar looking round objects, around the merged category as shown Figure 8(b).

We conducted a small user study where we recruited three graduate students as participants for evaluating the prototype system. As a whole, all the participants preferred the combination of the global and local manipulations to local manipulations only, since they could systematically change the arrangement of images by explicitly employing the image categorization. In practice, they were more likely to continue their image exploration with the composite manipulations since they could rearrange the image positions in further detail. The accompanying video contains some interactive sessions with our prototype system. Note that Table 1 provides statistics on computation times at the stage of initializing the two-layered graph, handling local manipulation, and handling global manipulation, according to the size of the image database.

5.2 Discussion

The primary motivation of the proposed bilevel feature space representation is to equip our image database with category-aware manipulation. Toward this goal, we introduced the bag-of-features model so that we can employ the extracted visual words as the representatives of the underlying image categories of the given dataset.



(a)

(b)

Figure 8: Global manipulation of the feature space: (a) Original layout of the images where tomato and CD images are selected. (b) Global manipulation helps us merge the two different image categories and further collect images including round objects around the merged category.

Table 1: Computation times (in seconds).

stage	process	Number of images		
		100	500	1,000
init.	graph construction	0.071	0.639	1.675
init.	distance computation	0.065	0.576	1.866
init.	MDS computation	0.005	0.014	0.018
local	graph updates	0.019	0.086	0.259
local	distance computation	0.006	0.531	1.764
local	MDS computation	0.005	0.011	0.019
global	graph updates	0.094	0.568	1.753
global	distance computation	0.057	0.543	1.789
global	MDS computation	0.005	0.011	0.019

Furthermore, we also use the visual words as the pivots for accelerating the MDS computation, which effectively allows us to project the high-dimensional feature space of the images onto the 2D screen space. Another advantage is that it enables us to allow fuzzy categorization of the images since each image node has connections with multiple visual word nodes in the two-layered graph representation. This naturally lets us assign multiple semantics with each image, along with the provided local and global manipulations through the 2D screen space.

On the other hand, the present approach has several limitations. In our implementation, we label each image with a color according to its primary visual word only. This means that we cannot explicitly elucidate other related visual words of the image in our visualization scheme, which remains to be improved especially for supporting our fuzzy categorization of the images. In addition, the system occasionally makes a set of images unexpectedly far away from each other while handling other selected images with the global manipulations as shown in Figure 9, where we employed the same image set as that in Figure 1. We have learned that this problem arises when some visual word node has a large number of edge connections with the images. This problem can be alleviated by incorporating a new formulation for restricting the maximum degree of each visual word node.

6 Conclusion

This paper has presented an approach to exploring images in the feature space so that we can edit the global relationships among image categories as well as the positions of the images in a local neighborhood through the 2D screen space. We supported both the global and local manipulation of image layouts by introducing the bilevel feature space representation together with the two-layered graph structure. The initial layout of the image set has been calculated based on the bag-of-features model, where we associated the representative features called visual words as the node of the upper-level graph while we represent the respective input images by the nodes of the lower-level graph. The bilevel feature space has been effectively projected onto the screen space using the pivot MDS, where we employ the visual word nodes as the pivot in the process of the dimensionality reduction. The global and local manipulations of the feature space have been achieved by adjusting the lengths of the edges in the two-layered graph, by taking advantage of the barycentric coordinate representation of the image nodes with respect to the visual word nodes. Experimental results were also provided to demonstrate the capability of the global category-aware and local detailed rearrangements of images in the user-driven feature space manipulation.

As future work, we may be able to further sophisticate the definition of the co-occurrence of the visual words, by referring to their geometric relationships in the original SIFT feature space [24]. Increasing the controllability of image layouts with the bilevel feature space representation still remains to be tackled. Our future work also includes the improvement of interface design so that users can fully associate their mental map with the arrangement of images in the feature space.

Acknowledgments

This work has been partially supported by JSPS under Grants-in-Aid for Scientific Research on Innovative Areas No. 25120014.

References

[1] U. Brandes and C. Pich. Eigensolver methods for progressive multidimensional scaling of large data. In *Proceedings of the 14th international conference on Graph drawing*, Springer Lecture Notes in Computer Science, pages 42–53, 2007.

[2] P. Brivio, M. Tarini, and P. Cignoni. Browsing large image datasets through voronoi diagrams. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1261–1270, 2010.

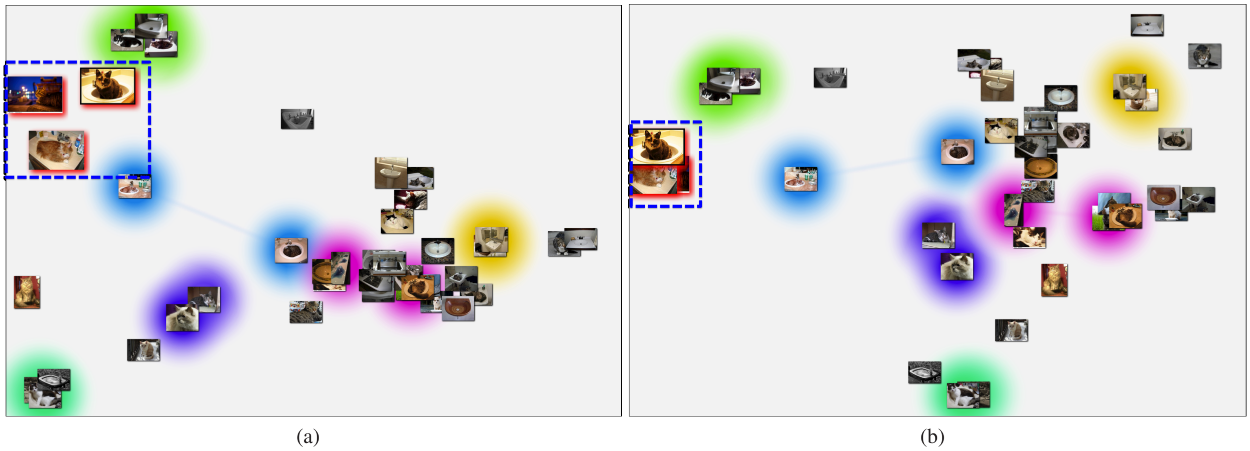


Figure 9: An example of unexpected rearrangement of the images. (a) Two images (a cat in the night and a cat in the washbowl) are selected. (b) Bringing the two images close to each other incurs a drastic change in the layout of other images.

[3] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *ECCV '04 Workshop on Statistical Learning in Computer Vision*, pages 1–22, 2004.

[4] V. de Silva and J. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. In *Advances in Neural Information Processing Systems (Proceedings of NIPS 2012)*, volume 15, pages 721–728, 2003.

[5] D. M. Eler, M. Y. Nakazaki, F. V. Paulovich, D. P. Santos, G. F. Andery, M. C. F. Oliveira, J. B. Neto, and R. Minghim. Visual analysis of image collections. *The Visual Computer*, 25(10):923–937, 2009.

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

[7] J. Fan, Y. Gao, and H. Luo. Hierarchical classification for automatic image annotation. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 111–118, 2007.

[8] J. Fan, Y. Gao, H. Luo, and R. Jain. Mining multilevel image semantics via hierarchical classification. *IEEE Transactions on Multimedia*, 10(2):167–187, 2008.

[9] G. Griffin, A. D. Holub, and P. Perona. The caltech 256. Technical report, California Institute of Technology, 2006.

[10] K. Heath, N. Gelfand, M. Ovsjanikov, M. Aanjaneya, and L. Guibas. Image webs: Computing and exploiting connectivity in image collections. In *Proceedings of the 23rd Computer Vision and Pattern Recognition (CVPR '10)*, pages 3432–3439, 2010.

[11] S. Ingram, T. Munzner, and M. Olano. Glimmer: Multilevel MDS on the GPU. *IEEE Transactions on Visualization and Computer Graphics*, 15(2):249–261, 2009.

[12] P. Joia, F. Paulovich, D. Coimbra, J. Cuminato, and L. Nonato. Local affine multidimensional projection. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2563–2571, 2011.

[13] L. S. Kennedy and M. Naaman. Generating diverse and representative image search results for landmarks. In *Proceedings of the 17th International Conference on World Wide Web (WWW '08)*, pages 297–306, 2008.

[14] J. B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29:1–27, 1964.

[15] J. Laaksonen, M. Koskela, S. Laakso, and E. Oja. Picsom – content-based image retrieval with self-organizing maps. *Pattern Recognition Letters*, 21(13-14):1199–1207, 2000.

[16] H. Liu, X. Xie, X. Tang, Z.-W. Li, and W.-Y. Ma. Effective browsing of web image search results. In *Proceedings of the 6th ACM SIGMM International Workshop on Multimedia Information Retrieval*, MIR '04, pages 84–90, 2004.

[17] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the 7th IEEE International Conference on Computer Vision (ICCV, 99)*, ICCV '99, pages 1150–1157, 1999.

[18] G. M. H. Mamani, F. M. Fatore, L. G. Nonato, and F. V. Paulovich. User-driven feature space transformation. *Computer Graphics Forum*, 32(3):291–299, 2013.

[19] J. G. S. Paiva, L. Florian-Cruz, H. Pedrini, G. P. Telles, and R. Minghim. Improved similarity trees and their application to visual data classification. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2459–2468, 2011.

[20] F. V. Paulovich, D. M. Eler, J. Poco, C. P. Botha, R. Minghim, and L. G. Nonato. Piecewise laplacian-based projection for interactive data exploration and organization. *Computer Graphics Forum*, 30(3):1091–1100, 2011.

[21] K. Rodden, W. Basalaj, D. Sinclair, and K. Wood. Does organisation by similarity assist image browsing? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '01, pages 190–197, 2001.

[22] R. M. Rustamov, Y. Lipman, and T. Funkhouser. Interior distance using barycentric coordinates. In *Proceedings of the Symposium on Geometry Processing*, pages 1279–1288, 2009.

[23] G. Schaefer and S. Ruszala. Image database navigation on a hierarchical MDS grid. In *Pattern Recognition (Proceedings of the 28th DAGM Symposium)*, volume 4174 of *Lecture Notes in Computer Science*, pages 304–313. 2006.

[24] M. Shi, X. Sun, D. Tao, and C. Xu. Exploiting visual word cooccurrence for image retrieval. In *Proceedings of the 20th ACM International Conference on Multimedia*, pages 69–78, 2012.

[25] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proceedings of the 9th IEEE International Conference on Computer Vision (ICCV '03)*, pages 1470–1477, 2003.

[26] G. Strong and M. Gong. Similarity-based image organization and browsing using multi-resolution self-organizing map. *Image and Vision Computing*, 29(11):774–786, 2011.

[27] B. Thomee, M. Huiskes, E. Bakker, and M. Lew. An exploration-based interface for interactive image retrieval. In *Proceedings of 6th International Symposium on Image and Signal Processing and Analysis*, pages 188–193, 2009.

[28] B. Thomee, M. J. Huiskes, E. Bakker, and M. S. Lew. Deep exploration for experiential image retrieval. In *Proceedings of the 17th ACM International Conference on Multimedia*, pages 673–676, 2009.

[29] W. S. Torgerson. Multidimensional scaling: I. Theory and method. *Psychometrika*, 17:401–419, 1952.