Qualitative Organization of Photo Collections via Quartet Analysis and Active Learning

Yuan Gan
State Key Lab for Novel Software Technology, Nanjing University

Yan Zhang*
State Key Lab for Novel Software Technology, Nanjing University

Zhengxing Sun
State Key Lab for Novel Software Technology, Nanjing University

Hao Zhang
Simon Fraser University

ABSTRACT

We introduce the use of *qualitative analysis* and *active learning* to photo album construction. Given a heterogeneous collection of photos, we organize them into a hierarchical categorization tree (C-tree) based on qualitative analysis using quartets instead of relying on conventional, quantitative image similarity metrics. The main motivation is that in a heterogeneous collection, quantitative distances may become unreliable between dissimilar data and there is unlikely a single metric that is well applicable to all data. Our qualitative analysis utilizes multiple distance measures and applies them where reliable comparisons are possible. Then from the C-tree, we develop an active learning scheme for fine-grained photo scene classification, allowing the selection of representative photos for layout construction which better reflects user intent. Finally, the selected photos are laid out in a comic-like arrangement based on a style template library and layout optimization. Experiments demonstrate that our method is efficient, user-centered, and produces photo albums that are more preferable in comparison with previous approaches.

Keywords: C-tree; Active Learning; Comic-like Photo Collage

Index Terms: Computing methodologies— Computer graphics— Image manipulation— Image processing

1 INTRODUCTION

Photo albums are treasure chests of stories, events, and moments for every person and every family. A typical photo album consists of a series of pages, each of which has a uniform size and dimension, and on each page, one or more related photos are displayed in a certain layout. An enjoyable browsing through a photo album can be mainly attributed to the quality of photo selection, for the whole album as well as the assignment of photos to individual pages, and proper photo arrangement on each page. Photos on the same page should be well related to spark interesting connections and fond memories. One may also wish to quickly locate a particular photograph in his or her mind during browsing, perhaps the one that captures a unique moment. Last but not least, a well-made photo album must exhibit visual clarity and aesthetics with its chosen photo layout.

In the era of digital and mobile photography, people are taking many, perhaps too many, photos whenever and wherever they want, leading to an explosion of personal photo collections. Even when photos are organized in file directories by time, event, or other themes, the number, as well as *heterogeneity*, of photos per directory may still be too great for a user to appreciate browsing photos one at a time. Photo albums alleviate this problem as they allow viewing collections of photos at a time. In addition, heterogeneity of the photo collections is conveniently addressed by assigning clusters of well-related photos to different pages of an album. The challenge is then how to make photo albums from very large and diverse photo collections. Manual photo selection and arrangement, the way photo albums were traditionally made, is too tedious. With much advance in image analysis and manipulation, efforts have been made to automate the process [14, 19, 21, 27, 32, 33].

Given a large collection of photos, the core technical problems for photo album construction include image classification and representative photo selection, which are followed by a layout optimization for the set of selected photos. To solve the key classification problem, all current methods have relied on computing *numerical* or *quantitative* similarity distances between images. When the input collection is diverse however, such distances may become unreliable, especially between dissimilar images, and there is unlikely a single metric that is well applicable to all data items in the collection.

Inspired by the work of Huang et al. [9], we introduce the use of *qualitative* analysis to photo/image organization. The key idea is to utilize *multiple* similarity distances when comparing and organizing diverse data and more importantly, such distances are employed only when they can be reliably compared. One situation for reliable comparisons is when there is clear contrast between small and large distances. For example, we can be quite certain that some data pairs are much closer (i.e., more similar) to each other than from others. Four such data items can form a *quartet*, as shown in Figure 1(b), where each quartet reflects a *topological* relation only it does not specify any numerical distance between the data. Given a large number of reliable quartets, we can construct a categorization tree (or C-tree) to hierarchically organize the input photo collection based on reliable estimates of similarities between the photos.

In addition to the use of qualitative photo organization, we also involve *humans in the loop* to improve the quality of the photo albums constructed. After all, album making has always been an endeavor with a personal touch, hence it is unlikely that a fully automated tool can cater to all user desires. To this end, we incorporate active learning into photo classification and representative photo selection, where users provide "must-link" and "cannot-link" constraints to steer the photo classification. Finally, we optimize a comic-like layout for each album page, out of the selected representative photos, based on an album page template library we construct.

Our contributions are three-fold. First, we introduce the use of qualitative analysis to hierarchically organize a photo collection using a C-tree. Second, we develop an active learning scheme based on the C-tree for image classification, recommending representative images to allow the users to select their favorites to keep in the album. Third, we design a photo collage display using comic-like layouts based on a template library, allowing concise photo layouts following aesthetically designed layout patterns.

2 RELATED WORK

Image organization and classification. Researchers generally classify images into different groups based on labeled training data. Numerical distances between images based on or trained on different levels of image features can be defined [15, 24] for subsequent learning [5, 11]. However, when comparing dissimilar images, numerical

*Corresponding author. E-mail: zhangyannju@nju.edu.cn

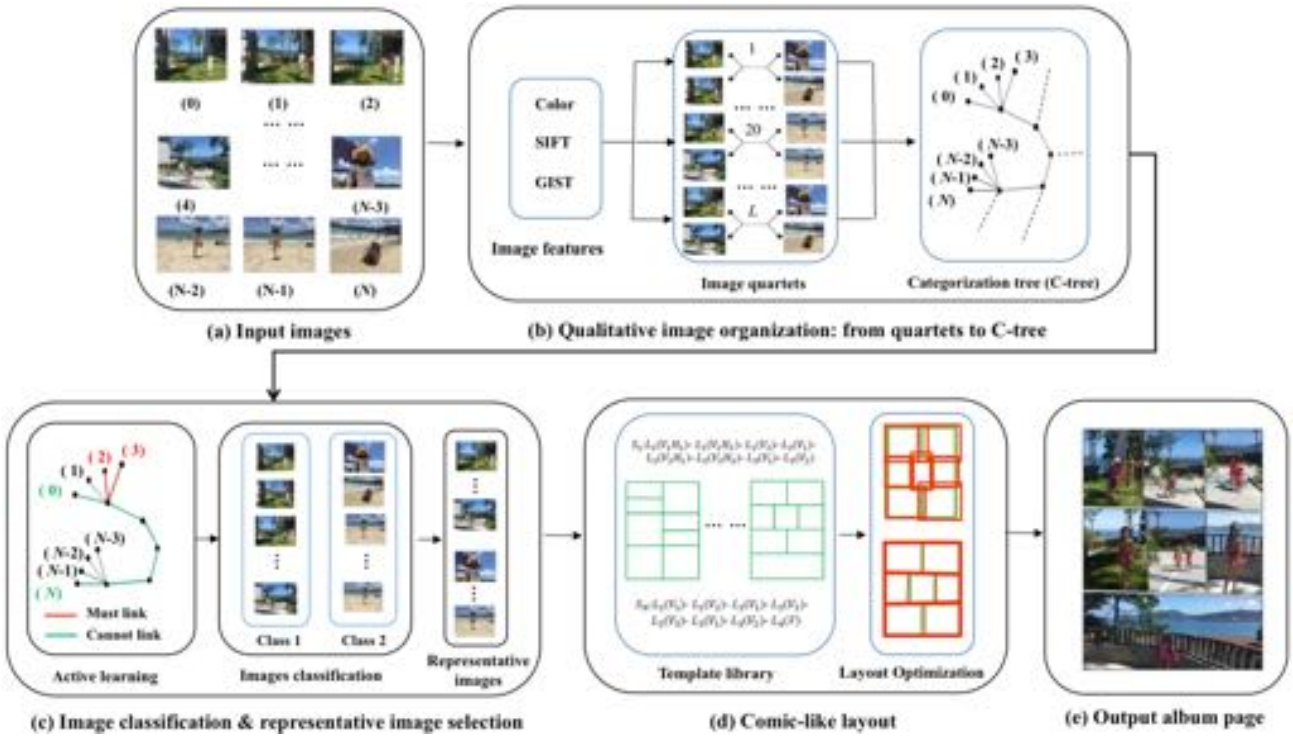


Figure 1: Overview of our method. (a) Input image collection. (b) Image organization based on quantitative analysis from reliable quartets. Each quartet consists of two photo pairs that are close within pairs but far between pairs, based on one or more image features, e.g., color, GIST, etc. A hierarchical C-tree is constructed from the quartets. (c) Image classification and selection of representative images via active learning based on the C-tree. (d) Comic-like layout collage generated from the representative photos. (e) Final output of an album page.

distances may easily become unreliable, leading to classification errors. By contrast, given unlabelled photos, we adopt the idea of qualitative analysis [9] to create quartets of images only when the numerical distances between images are reliable. We construct a C-tree for the images, and deduce the distance between images from the high-level topology information of the C-tree instead of the distance from low-level features of images to classify images into different groups. We then adopt active learning, with human-in-the-loop, to achieve more reliable image organization and classification.

Image collage. Collages are sometimes used to display a group of images, which stitches several images into one page to achieve the summary of images in a compact layout. Image collage starts from the perspective of image browsing [7, 8, 14, 19, 20, 27, 33], focusing on the space utilization of canvas or the compact and smooth sense of image stitching, while ignoring the search of desired images, making it not applicable for such a task. While little attention is paid to the problem of image organization and classification. If we input a large set of images, the effect of collage will be greatly affected. In this paper, for any source of photos we can organize them according to scenes and select representative images. Then we display representative images in a comic-like layout to compensate for the shortcomings of the above methods.

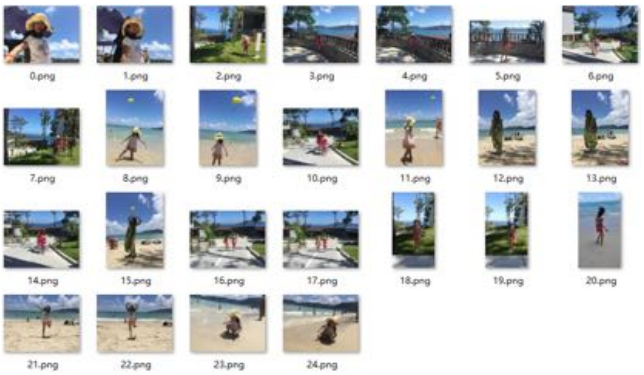
Comic-like layout. Some methods display keyframes in comic-like layout in video summary [6, 25, 31, 32]. These methods are simple and compact for users to browse. These methods usually take several comic templates to show the layout. However, since the number of templates that are manually defined is limited, the layout result is usually monotonous and lacks diversity. Others generate comic layouts based on a learning method [1], which improves the diversity of layout styles. However this requires extensive manual workload to label the comic layout in addition to a variety of learning methods to optimize the generation process, leading to enormous calculations and a complex process. Our method creates a template library based on universal and intuitive rules of comics and conducts a simple optimization for deformation of layouts, involving simple calculations and generating layout results with diverse styles.

3 METHOD

For many images from different scenes, we make photo albums by three steps: image organization based on quantitative analysis, image classification and selection with C-tree and active learning, and comic-like layout collage generated from the representative photos. The brief process is shown in Figure 1.

3.1 Qualitative Photo Organization

Given hundreds of diverse images (as shown in Figure 1(a)), we have not been able to entirely automate the album creation process based on user intent, and therefore, we involve users in the process. To minimize their efforts, we organize the images according to the scenes to realize a pre-classification first. The difficulty in the scene organization is how to make the computer understand the scene information of the image by feature analysis, especially for a diverse image set. In fact, when comparing two dissimilar images, the numerical distance values are usually not reliable. For instance, a numerical distance between an image by the sea and an image in the park is most likely less informative than a distance between two images by the sea or between two images in the park. In a large image collection possessing rich variations, it can be extremely difficult to use a single distance measure that will allow a meaningful analysis. By contrast, we take the idea of the 3D shape classification method [9] to create a C-tree, as shown in Figure 1(b), for image set, then deduce the distance between the images.



(a)

(b)

Figure 2: An example image set and four potential quartets under the GIST feature. (a) Input image set; (b) The above two quartets are reliable and the bottom two are discarded.

Quartet creation. The organization of heterogeneous 3D shapes [9] involves some features which are normally used for distinguishing the 3D shapes. Accordingly, to organize diverse images, we primarily select three visual features, Color [22], SIFT [17], and GIST [2] to characterize the scenes of images. They can classify low-level similarity but not high-level similarity.

A quartet contains four images with two pairs $(A,B|C,D)$ and follows the constraint that two images in one pair are similar, but images in different pairs are dissimilar. We adopt the similar method in [3,9] to create the quartets for images under each feature and then combine them together. First, we construct an undirected graph with four nodes A,B,C,D, where each node denotes an image and the weight of edge denotes the distance between images under a certain feature. Then the largest three edges are deleted. If the graph is no longer connected, they cannot form a quartet. Otherwise, we denote the bridge edge between two pairs as $d3$ and the other two edges as $d1$ and $d2$, if $d3/d1$ and $d3/d2$ are both greater than a certain threshold R, then these images can form a reliable quartet.

If R is too low, the number of quartets will be too large, which will increase the computation of C-tree and reduce the reliability of quartets. If R is too high, the number of quartets will be too small, which will affect the accuracy of C-tree. For 60 images, we get about 8100 quartets under the Color feature with R = 2.4, about 6400 quartets under the SIFT feature with R = 3.2 and about 8700 quartets under the GIST feature with R = 2.2, for a total of 23200 quartets for subsequent C-tree construction. Figure 2(b) shows some quartets during generation of an example image set (as shown in Figure 2(a)) under the GIST feature.

C-tree construction. After obtaining the quartets, we construct the C-tree, which is an unrooted tree maximally conforming to the topology of the input quartets. Each leaf node of the tree denotes an image. The internal node is a parent or ancestor of other nodes and represents information of scene organization. If two leaf nodes share the same parent or ancestor, they are likely to belong to the same scene in the following classification, if two internal nodes share the same parent or ancestor, they are likely to belong to similar scenes, so we could organize images hierarchically based on the relationship between images in the C-tree.

For image set in Figure 2(a), we construct the corresponding C-tree as shown in Figure 3(a). It reveals that the C-tree can organize images by scenes and the topology information in quartets is kept inherently in the structure of the C-tree. The quartet that has been labeled in green in Figure 2(b) can be found in the C-tree in Figure 3(a) with the paths are labeled green as well.

3.2 Photo Classification and Selection

Since we have obtained a C-tree for input images in the previous section, which shows the overall relationship of input images. In order to better deliver user intent, we use active learning method [13, 28–30] to perform the fine classification of the input images. Then, for images in each scene, we perform branch sampling based on the C-tree to recommend representative images for the user. Meanwhile, the user can switch or reject any samples selected by the automatic algorithm to choose the images they prefer.

Image classification based on active learning. We observed that the process of creating C-tree and divisive hierarchical clustering [18] are very similar, both of them recursively split one big set into singletons to form a hierarchical structure. We decomposed the C-tree into several clusters to classify images. The root node of C-tree is an internal node, and all images reside in leaf nodes.

We denote T_x as a subtree rooted at node x. Given a C-tree with root r, T_r, we can compute the depth of every node. We define the degree of separation (number of traversed edges), DoS [9] as the distance between two nodes. Let N_k be the number of album pages specified by the user, where one album page corresponds to one cluster. We start the splitting process with $d = 1$, where d represents node depth, to get a node set N_d. For each node n in N_d, we have a subtree T_n. All leaf nodes in T_n become a cluster, so we can get $|N_d|$ clusters in all. If $|N_d| = N_k$, then the algorithm terminates, as we have the desired number of clusters. If $|N_d| > N_k$, we merge the two nearest nodes in N_d by adding one to another node's child until $|N_d| = N_k$. If there are more than one pair of nodes having the minimum distance, the pair which would generate a smaller cluster after merging will have priority. Finally, if $|N_d| < N_k$, we continue the loop by adding 1 to d at a time. Figure 3(b) is the initial clustering result corresponding to the C-tree in Figure 3(a).

After the initial clustering, we find out that most images can be reasonably classified according to the scenes, but there exist some images which are so ambiguous. Therefore, to refine the classification results, and to enable users to participate in the classification process, we adopt active learning method in this paper. Active learning is a special case of semi-supervised method by which samples can be dynamically selected and labeled. By interactively adding constraints, user intent can be learned and more information can be used to refine the clusters.

Since we have obtained initial clusters, by interactively adding few *must-link* or *cannot-link* constraints, the C-tree is updated and so are the clusters, all ambiguous images will be reassigned into correct class finally. To minimize the efforts required from the user, we suggest pairs of images that are likely to improve classification when constrained. Those images should have low confidence of belonging to their clusters, and we take the silhouette index as their confidence [30]. Let $S(x)$ denote the silhouette index of image x, its value ranges from -1 to 1. If the value is close to -1, this image is closer to another cluster than it belongs to, which means it might be handed over to the user. If the value is close to 1, this image has strong confidence in current cluster. The higher the silhouette value, the more believable its cluster will be. The image with the greatest

confidence value will be the center of the cluster.

First, we select m images with the smallest confidence value as a set S_m. We adopt silhouette index, the distance between x and one cluster is measured by the average distance between image x and images in that cluster. C_x is the cluster that x belongs to, O_x is the closest cluster except C_x. Then we form a triplet $T(x,A(x),B(x))$ for each image x in S_m, where $A(x)$ is the center of cluster C_x, and $B(x)$ is the center of cluster O_x. Then we present these m triplets for the user, and ask them to judge whether $A(x)$ or $B(x)$ is more similar to x. If the user chooses $A(x)$, then image x has already been assigned to the right cluster. If the user chooses $B(x)$, then we need to reassign x.

In fact, a triplet consists of two constraints, one *must-link* and one *cannot-link* because when the user chooses one, the other will be separated from x. In every iteration, if the user chooses more than one $B(x)$, we will reconstruct the C-tree to obtain new clusters by updating quartets set. We augment the quartets set with new quartets $(x,b|c,d)$, where b is the image in O_x, c,d are the images in C_x. We remove the quartets $(p,q|r,s)$ from the quartets set, where p is x and q is the image in C_x or r is x and s is the image in C_x. The iteration will be repeated until the user chooses all $A(x)s$ for m quartets, which means all images are in the right class according to user expectations. In experiment, we set $m = 2N_k$.

One loop of the active learning is shown in Figure 4. The node with label 20 marked in red in Figure 3(a) has the lowest confidence value. Figure 4 (a) illustrates the triplets suggested by our algorithm according to the initial clustering result in Figure 3(b). The triplet marked in red means that B(x) is chosen by the user, so we update the C-tree as shown in 4(b), the corresponding classification result is in Figure 4(c), it shows that image with label 20 is in the right place.

Selecting representative images. After the fine classification of the input images, we can determine themes of the album, where images of one page share the same theme. However, every class might still contain too many images while the space for comic layout on each page is limited. Again, to minimize the effort required from the user, we will suggest representative images for each theme. We believe that representative images not only have strong confidence but also are well separated in each class.

In the C-tree, we call the leaf nodes with the same parent as a *branch*. For the classification result, every class actually corresponds to a subtree of the C-tree, so every class consists of several branches. Images in the same branch are very similar and in different branches are not. For one class, we choose one image with the greatest confidence value from each branch at a time until the number of images reaches n. Then given the representative images, the user can switch or reject anyone through the interactive interface according to their preferences. If the user does the *switch* operation, we'll reselect a new representative image with the greatest confidence value in the same branch except the switched one. If the user does the *reject* operation, then the whole branch will be rejected.

In practice, we find that around 10 photos per page tends to obtain a good balance between the amount of information and clarity and aesthetics of the layout, hence we set $n = 10$ in our experiments. To layout the representative images in a collage, we randomly select one image as the first one, then sort the others according to the distance to the first.

3.3 Photo Collage in Comic Layout

Once all representative images are confirmed, we start to make the photo album. We choose comic-like layout for the album due to its beauty and diversity. A single page will be made for each scene. First, based on the heuristic rules defined by us, we create a comic-like template library through enumeration. We then select a template to make an automatic collage based on the principle of keeping the maximum information of the images.

Template library. To avoid monotonous layout and meet the needs of different users, variety in templates is essential. Some researchers have constructed active templates by heuristic rules, such as the tree structure based on the comic [16, 23]. But the space divided by these methods is usually messy and uncoordinated. In order to make the space tidy, we constructed the corresponding template library with enumeration. Comparing to the tree structure method, we make the divided space clearer, and significantly reduce the time of calculation.

It is well-known that a page of comic layout is often made up of several layers, each composed of different size panels, and there are many rules in the process of comic layout to limit the number of templates, so we can generate the initial templates with enumeration. In this paper, we construct the template library primarily by the following heuristic rules:

- **R1:** Due to limited space in the comic, to achieve aesthetic effects, we stipulate that each page can be only divided into at most 4 layers and each layer can be only divided into at most 3 panels;
- **R2:** Since the comic has a specific reading sequence, for two panels, P_i and P_j, in the comic layout, if $i < j$, panel P_i must be on the left of or above panel P_j. Therefore, the layer number of panel P_i must be not larger than that of panel P_j;

Since each layer can only contain at most 3 panels, the positions of the panels in each layer will be limited, which we can enumerate easily. All possible placements in each layer of the panel are shown in Figure 5, where J represents the number of panels contained in each layer. We primarily use a string to represent the location of each panel. For example, V_2H_1 indicates that the panel is on the upper right corner of the corresponding layer.

We can obtain a series of strings using the above enumeration with each string corresponding to the initial template of a comic layout. A group of data containing 8 input images can generate about 120 initial templates according to the enumeration definition. Figure 6 shows the generated strings and corresponding templates for 8 input images by our method. S_1, S_i, and S_j represent the strings obtained by enumeration, and $L_N(\cdot)$ in the strings represents the layer position of the current panel in the template. The content in the bracket following L_N corresponds to the given strings in Figure 5, representing the panel position. T_1, T_i, and T_j represent the corresponding templates.

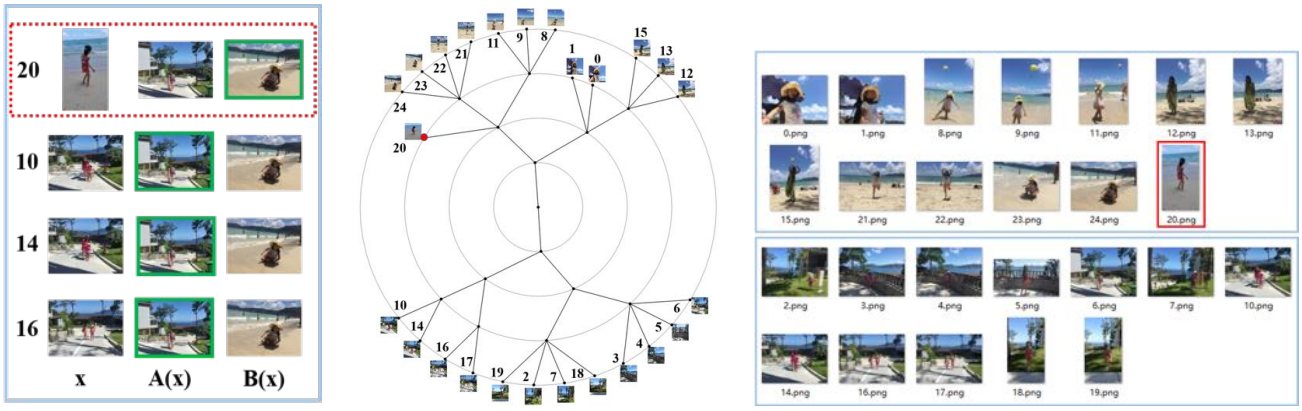
Appropriate template selection. After obtaining the template library, we hope to select the best template for each set of representative images. When people take photos, they do not just pick up the camera and press the shutter at will. They usually resort to some photographs skills, considering shooting angle, composition rules and so on, to achieve the best overall effect. So it is essential to keep the original appearance of images while making albums. However, the size of images and panels can not be exactly the same. The following strategy is used to keep more information.

For each image, we first calculate the saliency map [4], and set the face regions detected by the classical approach [26] to maximize the saliency value. Then we find the *saliency center*, of which the sums of saliency values on both sides are the same in horizontal and vertical directions, and scale the representative images to keep the area of the page and the sum of them equal. Next, the images are placed into each panel of the template in order, on the premise that the saliency centers of images and the corresponding panel centers are aligned.

In order to select the most appropriate template, we define the penalty function as follows: for the group of images to be collaged, we set the obtained initial template library as $\{T_m\}_{m=1}^{M}$. For any template T_m, we represent the layers as L_i ($i \in [1,4]$) and panels in L_i as $P_{i,j}$ ($j \in [1,3]$), and the rectangle of the corresponding panel with scaled representative images in it as $I_{i,j}$. The $Area(\cdot)$ is the area of panel or image, and $Height(\cdot)$ is the height of the panel.

(a) (b)

Figure 3: The corresponding C-tree and initial clustering results for the input images in Figure 2(a). (a)The C-tree constructed by all quartets. (b) Two clusters via our method based on the C-tree when $d = 1$, corresponding to the subtree T_x and T_y.



(a) (b) (c)

Figure 4: Image classification based on active learning. (a) The interaction results, images chosen by the user are in green box (b) Updated C-tree (c) Classification results after active learning.

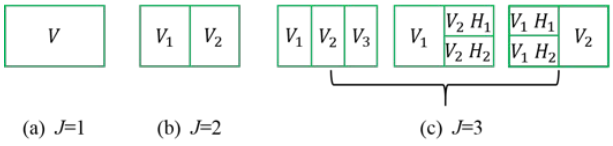


Figure 5: Different cases of panels in one layer.

1. **Coverage of layer L_i.** We want the coverage between adjacent panels of every layer to be as small as possible. The coverage of L_i with J panels is:

$$E_o(L_i) = \sum_{j=1}^{J-1} (Area(I_{i,j}) \cap Area(I_{i,j+1})) \tag{1}$$

2. **Degree of separation of layer L_i.** Similarly, the gap between adjacent panels of every layer should be as small as possible. The degree of separation of layer L_i with J panels is:

$$E_g(L_i) = Area(L_i) - Area(L_i) \cap (\bigcup_{j=1}^{J} Area(I_{i,j})) \tag{2}$$

3. **Coverage between layer L_i and L_{i+1}.** We also want the coverage between adjacent layers to be as small as possible. The coverage between layer L_i and layer L_{i+1} is:

$$D(L_i, L_{i+1}) = Area(\bigcup_{j=1}^{J} I_{i,j}) \cap Area(\bigcup_{k=1}^{K} I_{i+1,k}) \tag{3}$$

Based on these constraints we define the penalty function for template T_m as follow:

$$P(T_m) = \alpha \sum_{i=1}^{NL} (\aleph(E_o(L_i)) + \aleph(E_g(L_i))) + \beta \frac{\sum_{i=1}^{NL-1} D(L_i, L_{i+1})}{PAGE}, \tag{4}$$

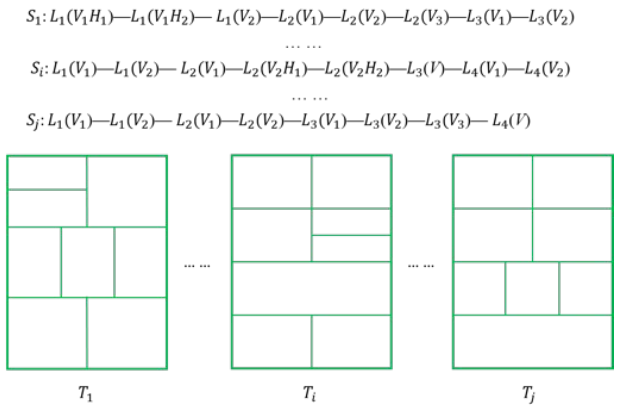


Figure 6: Example of templates corresponding to string representations: T_1, T_i, T_j correspond to S_1, S_i, S_j, respectively.

Figure 7: Layout optimization. (a) Best template. (b) Initial layout. (c) Template after optimization. (d) Final layout.

where $P(T_m)$ is the penalty function for template T_m. NL is the maximum number of layers of T_m, $\aleph$ is the Min-Max normalization operator for $E_o(L_i)$, $E_g(L_i)$ and $E_h(L_i)$. $PAGE$ is the area of the comic page, α, β are corresponding weights for local constraint and global constraint, we set $PAGE = 1200 * 1600$ $pixels$, $\alpha = 0.4$, $\beta = 0.6$ by experiment.

Based on this penalty function, we can perform a matching search in the template library to find the best template which has the minimum penalty. For seven representative images selected from the bottom class in Figure 4(c), the most appropriate template is shown in Figure 7(a) and the initial layout without optimization is shown in Figure 7(b). Besides, the user also can browse the other templates in the interface, we sort all other templates according to $P(T_m)$.

Layout optimization. After obtaining the best template, there may still be gaps between the images and corresponding panels. For compact and diverse layouts, we optimize the deformation of the panels and then fine-tune image sizes to fill the gaps. Throughout the process, we keep the aspect ratio of every image and align the saliency center of them to the center of the corresponding panel. If we were to adjust the shape of the panel, we would only need to make a corresponding adjustment on each section line in the comic layout. The goal of our layout optimization is to minimize:

$$Y(T_m) = \sum_{i=1}^{NL} \sum_{j=1}^{J} (Ratio(P_{i,j}) - Ratio(I_{i,j}))^2, \quad (5)$$

where $NL \in [1,4]$ is the maximum number of layers of T_m, $J \in [1,3]$ is the number of panels of each layer, and $Ratio(\cdot)$ is the aspect ratio of the panel (if the panel is an irregular quadrilateral, it calculates the aspect ratio of its minimum bounding box).

We optimize the objective function by the PSO [12] algorithm, take the parameters of each section line in the template as particles.

Table 1: Datasets and timing statistics (in seconds):N_I is the number of images in the set; N_K is the number of classes; T_F is the time of calculating features and creating quartets; T_C is the time of constructing C-tree; T_L is the time of comic-like layout.

	Number		Time		
	N_I	N_K	T_F	T_C	T_L
Set#1	116	5	90	2	10
Set#2	182	6	94	2	12
Set#3	250	7	100	3	14

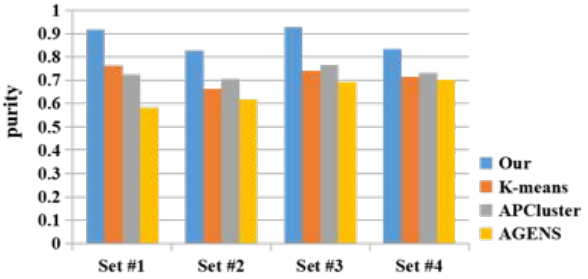


Figure 8: Comparison of several clustering methods.

After the objective function reaches a minimum value or the iteration reaches a specified number, we fine-tune the size of the images to fill the gaps to obtain the final comic-like layout. Last but not least, in order to let the final album page look like a real comic page, we make a simple shift on each deformed line and set the layer spacing to 15 pixels and panel spacing to 8 pixels. Compared with Figure 7(a), panels have been somewhat deformed while maintaining the primary relationship in Figure 7(c) after optimization. In that case, different images with the same template will generate different layout. Figure 7(d) is the corresponding final comic-like layout.

4 Results and Evaluation

In this section, we introduce the experiments and analyze our method. We carry out user evaluation to verify the effectiveness of our method. Limitations will be discussed at the end.

We implemented the whole process in C++ on a PC with Intel Core i5 3.2GHz CPU and 8G memory. In order to validate the effectiveness of our method, we conducted experiments on several sets of images taken by many different devices in different times and places. Each of them contains hundreds of photos, including portraiture, landscapes, and events. The details of the sets and the running time of our algorithm can be found in Table 1. And the user can always get instant feedback of the system during interaction. With the help of C-tree, every round of interaction can be completed quickly within 5 seconds if the user does not hesitate deliberately. On average, image classification needs $10N_k$ seconds for interaction, N_k is the number of album pages specified by the user. Experimentally, constructing C-tree exhibits a performance that is close to $O(mlog(m))$, where m is the number of images. Although we only test on a small database, according to the analysis, it needs 40 minutes for 10,000 images to construct C-tree twice as C-tree will be reconstructed after the interaction.

4.1 Image Clustering

In order to verify the effectiveness of our method based on C-tree for scene classification, we compare the classification results with several conventional clustering methods, k-means, AP [8] and AGENS [10]. To be fair, we use the same features and the initial clustering results without active learning. We select 15 undergraduate students to classify the input image sets as ground truth. For a set of images, after one user sets it up from the personal perspective,

(a) (b) (c) (d)

Figure 9: Two comparisons (top and bottom) of photo collage results generated from the same set of representative images. (a) Our result using comic-like layout. (b) Our result using rigid layout. (c) Autocollage [19] result. (d) Content-Aware Photo Collage [32] result.

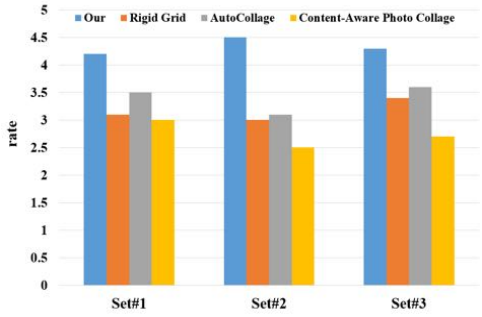


Figure 10: User satisfaction for the albums.

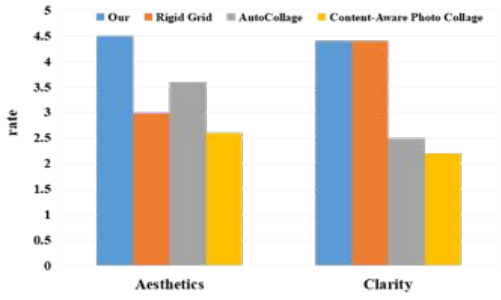


Figure 11: User survey results of average rate of all pages in aesthetics and clarity respectively.

it will be handed over to another user to modify until all of the users are satisfied with the results.

We use purity of clustering as the standard of comparison. The final results of the comparison are shown in Figure 8. It reveals that the purity of our method is higher than that of k-means, AP and AGENS clustering. It proves that the effect of our method is closer to the classification of human visual and more stable with the expansion of images set, which also illustrates the effectiveness of the C-tree organization. Better initial clustering requires less user interaction. The cause of the results is that any single image similarity measure may be unreliable, especially when the images are very different. Our method is carried out on the basis of the C-tree constructed by quantitative analysis, which takes the advantage of reliable quartets from several features. That is why the purity of our method just dropped a little bit when the image set doubled (Set#3 has 250 images and Set#1 has 116 images).

4.2 User Evaluation

User studies are conducted to validate the effectiveness of our layout method for album generation. We select 50 undergraduate students to evaluate the results obtained by our method, rigid layout (similar to our method), AutoCollage [19], and Content-Aware Photo Collage

[32]. User are not aware of correspondence between the results and methods in advance. For fairness, we use the same representative images as the input, and set the page size to be the same as well. The evaluation results are shown in Figure 9.

First, for each image set, we make photo albums using the above methods, where they have the same pages. Then users are asked to rate them with 1 (definitely no) to 5 (definitely yes) according to their preference. The result in Figure 10 shows that the albums generated by our method have the highest score for every image set.

To further understand the advantages of our method, we carry out more user study. For all the pages generated by these four methods, users are asked to rate them in two aspects: aesthetics and clarity. For a better comparison, four pages generated by these methods with the same representative images form a group. The result in Figure 11 shows that the albums generated by our method are preferred by undergraduate students in aesthetics and clarity. In our opinion, the structure of the comic-like layout makes the album collage clearer compared to AutoCollage and Content-Aware Photo Collage. And the template choosing and optimization make the album collage more beautiful than the rigid layout.

4.3 Limitations

Although albums can be effectively made by our method, there are still some limitations. We just combine saliency map with face detection to capture the content of the image. The pages generated by our system will be not good enough when the face detection failed or there are too many objects in the images. Our user studies are preliminary and suggestive, and users graded according to their standard. It is surprising that users would prefer non-rectangular images and further study is needed to verify this result.

5 Conclusion and Future Work

In this paper, we propose a new album management method in a comic-like layout based on scene classification. Firstly, we effectively organize images by a qualitative analysis method of Quartet Analysis to realize the hierarchical organization of the scenes and we use active learning method to classify images by scenes according to user intent. In addition, we propose a new method for automatic collage with comic-like layout based on a template library. This permits a concise collage presentation of the representative images in the hierarchical scenes. The results demonstrate that we achieve the goal of effective organization and clear display of the images. Users can easily produce an album with our method.

In future work, we plan to incorporate high-level semantics, e.g., accurate face recognition, to improve classification. In addition, we will explore the use of C-trees to assist users in photo organization and generating photo album more quickly.

References

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

[2] Y. Cao, A. B. Chan, and R. W. Lau. Automatic stylistic manga layout. *ACM Trans. on Graphics*, 31(6):141, 2012.

[3] S. Chen, Z. Sun, and Y. Zhang. Scalable organization of collections of motion capture data via quantitative and qualitative analysis. In *Proc. of ACM Int. Conf. on Multimedia Retrieval*, pp. 411–418. ACM, 2015.

[4] M. M. Cheng, G. X. Zhang, N. J. Mitra, X. Huang, and S. M. Hu. Global contrast based salient region detection. In *Computer Vision and Pattern Recognition*, pp. 409–416, 2015.

[5] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, vol. 1, pp. 1–2. Prague, 2004.

[6] J. Fan, Y. Gao, H. Luo, and G. Xu. Statistical modeling and conceptualization of natural images. *Pattern Recognition*, 38(6):865–885, 2005.

[7] F. Fang, M. Yi, H. Feng, S. Hu, and C. Xiao. Narrative collage of image collections by scene graph recombination. *IEEE transactions on visualization and computer graphics*, 24(9):2559–2572, 2018.

[8] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *science*, 315(5814):972–976, 2007.

[9] S.-S. Huang, A. Shamir, C.-H. Shen, H. Zhang, A. Sheffer, S.-M. Hu, and D. Cohen-Or. Qualitative organization of collections of shapes via quartet analysis. *ACM Trans. on Graphics*, 32(4):71, 2013.

[10] L. Kaufman and P. J. Rousseeuw. *Finding groups in data: an introduction to cluster analysis*, vol. 344. John Wiley & Sons, 2009.

[11] Y. Ke and R. Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *IEEE CVPR*, vol. 2, pp. II–506. IEEE, 2004.

[12] J. Kennedy. Particle swarm optimization. In *Encyclopedia of Machine Learning*, pp. 760–766. Springer, 2010.

[13] Z. Li, J. Liu, and X. Tang. Constrained clustering via spectral regularization. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 421–428. IEEE, 2009.

[14] L. Liu, H. Zhang, G. Jing, Y. Guo, Z. Chen, and W. Wang. Correlation-preserving photo collage. *IEEE Transactions on Visualization & Computer Graphics*, (6):1956–1968, 2018.

[15] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[16] E. Myodo, S. Ueno, K. Takagi, and S. Sakazawa. Automatic comic-like image layout system preserving image order and important regions. In *ACM Int. Conf. on Multimedia*, pp. 795–796. ACM, 2011.

[17] M.-E. Nilsback and A. Zisserman. A visual vocabulary for flower classification. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, pp. 1447–1454. IEEE, 2006.

[18] L. Rokach and O. Maimon. *Clustering Methods*. Springer US, 2005.

[19] C. Rother, L. Bordeaux, Y. Hamadi, and A. Blake. Auto collage, May 5 2009. US Patent 7,529,429.

[20] I. V. Safonov, I. V. Kurilin, M. N. Rychagov, and E. V. Tolstaya. *Automatic Generation of Collage*. 2018.

[21] P. Sandhaus, M. Rabbath, and S. Boll. Employing aesthetic principles for automatic photo book layout. In *Advances in Multimedia Modeling*, pp. 84–95. Springer, 2011.

[22] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[23] T. Tanaka, K. Shoji, F. Toyama, and J. Miyamichi. Layout analysis of tree-structured scene frames in comic images. In *IJCAI*, vol. 7, pp. 2885–2890, 2007.

[24] A. Vailaya, M. Figueiredo, A. Jain, and H. Zhang. Content-based hierarchical classification of vacation images. In *IEEE Multimedia Computing and Systems*, vol. 1, pp. 518–523. IEEE, 1999.

[25] A. Vailaya, M. A. Figueiredo, A. K. Jain, and H.-J. Zhang. Image classification for content-based indexing. *Image Processing, IEEE Transactions on*, 10(1):117–130, 2001.

[26] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *Proc Cvpr*, 1:511, 2001.

[27] J. Wang, L. Quan, J. Sun, X. Tang, and H.-Y. Shum. Picture collage. In *IEEE CVPR*, vol. 1, pp. 347–354. IEEE, 2006.

[28] X. Wang and I. Davidson. Active spectral clustering. In *Proc. of IEEE Data Mining (ICDM)*, pp. 561–568, 2010.

[29] Y. Wang, S. Asafi, O. van Kaick, H. Zhang, D. Cohen-Or, and B. Chen. Active co-analysis of a set of shapes. *ACM Trans. on Graphics*, 31(6):165, 2012.

[30] Q. Xu, K. L. Wagstaff, et al. Active constrained clustering by examining spectral eigenvectors. In *Discovery Science*, pp. 294–307, 2005.

[31] Y. Yang, Y. Wei, C. Liu, Q. Peng, and Y. Matsushita. An improved belief propagation method for dynamic collage. *The Visual Computer*, 25(5-7):431–439, 2009.

[32] Z. Yu, L. Lu, Y. Guo, R. Fan, M. Liu, and W. Wang. Content-aware photo collage using circle packing. *Visualization and Computer Graphics, IEEE Transactions on*, 20(2):182–195, 2014.

[33] L. Zhang and H. Huang. Hierarchical narrative collage for digital photo album. In *Computer Graphics Forum*, vol. 31, pp. 2173–2181, 2012.