

Research Article

Design Pattern Retrieval and Style Analysis for Content Creation of Comic Figures

Bor-Shen Lin and Shang-Te Tsai

Department of Information Management, National Taiwan University of Science and Technology, Taipei 106, Taiwan

Correspondence should be addressed to Bor-Shen Lin; bslin@cs.ntust.edu.tw

Received 7 November 2013; Accepted 29 December 2013; Published 9 March 2014

Academic Editor: Teen-Hang Meen

Copyright © 2014 B.-S. Lin and S.-T. Tsai. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Placement of objects within a constrained space is a common challenge for designers; it is associated with decisions regarding the furnishing of a space with furniture, collocation of dressings, flower arrangement, and design of comic figures. Though many design elements can be shared on the Internet in the current age of technology, it is still not easy to compare or search for design patterns based on these elements. Thus, it is difficult for designers to efficiently retrieve similar patterns designed by others, to compare them, or to learn from them. This paper proposes the architecture of representing, comparing, retrieving, and analyzing the design patterns of digital contents for design support. This scheme can help the designers to explore the huge space of design patterns efficiently, to analyze and summarize the design styles quickly, and to improve design skills and stimulate imaginations effectively during the process of learning or creating. The proposed scheme has been verified with a design support system for the content creation of comic figures. It is generally applicable to the creation of digital contents and shows potential for applications in the fields of design and education.

1. Introduction

Placement of objects within a constrained space is a common challenge for designers; it is associated with decisions regarding the furnishing of a space with furniture, collocation of dressings, flower arrangement, and design of comic figures [1–8]. In such design tasks, users' goals are usually vague or implicit and require clarifications again and again through self-exploration or through the interaction between the customer and the designer for multiple turns. This is mainly because personal tastes and design styles are often implicit, not easy to describe, and difficult to match. For the collocation of dressings, for example, a lady might have simply rough ideas about what she needs. She might have some preferences of choices on color, style, or accessory due to fashion or simply would like to buy a shirt to match a coat or a pair of shoes. In such cases, it often takes much time for the clerks to suggest the candidates of dressings because the preferences of the customers can seldom be described explicitly. If the patterns of collocation can be represented

digitally and retrieved efficiently, the cycles of customer-designer interaction could be reduced effectively and the services might be delivered more efficiently.

With the advances of computer and network technologies, more and more design elements are digitized and become accessible on the Internet, such as photos, icons, sounds, or 3D models. It could be expected that design elements and patterns of various types, such as hair, garment, or comic styles, will be produced, shared, and reused in the future [9–12]. In some researches, for example, design patterns can be represented with specific data structure [13, 14] so as to be reusable for further design or creation. However, the lack of effective comparison and search methods for high-level structural patterns makes it difficult for the designers to refer to similar patterns designed by others, to compare them, or to learn the design skills or styles from them. Provided that the patterns can be compared and retrieved responsively through network, the learning of design can be facilitated largely, and the cocreation or resource sharing among designers becomes feasible [15–21]. In addition, the

style analysis for the design patterns is still a complex problem nowadays. Traditionally, the design styles can be analyzed or summarized only by experienced designers or experts manually through observing a large amount of samples. If the styles can be analyzed and summarized automatically, the tutors could instruct and inspire the learners more efficiently.

In this paper, the architecture for representing, comparing, retrieving, and analyzing structural design patterns of digital contents was proposed, as shown in Figure 1. Based on this scheme, design support systems can be built to assist the content creation and facilitate the sharing of patterns among the designers. Designers, for example, can view patterns that are similar to their own works and see how the elements used in their own works are utilized differently and innovatively by others. These observations can help the designers to improve their skills and stimulate their imaginations for creation. Since design patterns can be clustered, main styles of clusters can be analyzed and summarized automatically. Consequently, designers may browse patterns in an orderly manner, and they can obtain a broader view of design styles quickly. The proposed scheme has been verified with a design support system for the content creation of comic figures; thus, the design elements, which include objects of various types, can be searched, edited, and assembled into patterns. Subsequently, these patterns can be compared, retrieved, and clustered, while the design styles can be analyzed and summarized efficiently. This scheme is generally applicable to the creation of digital contents, and it shows potential for applications in the fields of design and education.

The organization for the rest of this paper is as follows. Section 2 describes how to create the digital contents of comic figures using shared design elements in a network environment. Section 3 illustrates how a pattern can be represented as the feature, based on which similarity and distance can be measured and the comparison of patterns can be performed. Section 4 depicts how the patterns are clustered with agglomerative clustering or k -means clustering and how the style of each cluster can be analyzed and summarized automatically. Section 5 shows how the proposed scheme can be applied to the retrieval and style analysis for the design patterns of comic figures so as to improve the process of learning or creating. Section 6 finally concludes the contributions of this paper.

2. Content Creation of Comic Figures

In Figure 1, the design patterns are composed by the designers with the shared design elements in a library through a web application. Here in this paper a web application for composing comic figures is taken as an example for illustration. A library of design elements, which contains about 8,000 colored images, has been built using image editing tools and is shared and used for design. The images can be classified into a few types according to their contents, including the heads of famous stars or politicians, the bodies with different poses, actions or dressings, the shadows, the props, and the background scenes, as shown in Figure 2. In addition, each image was tagged manually with several keywords. The image

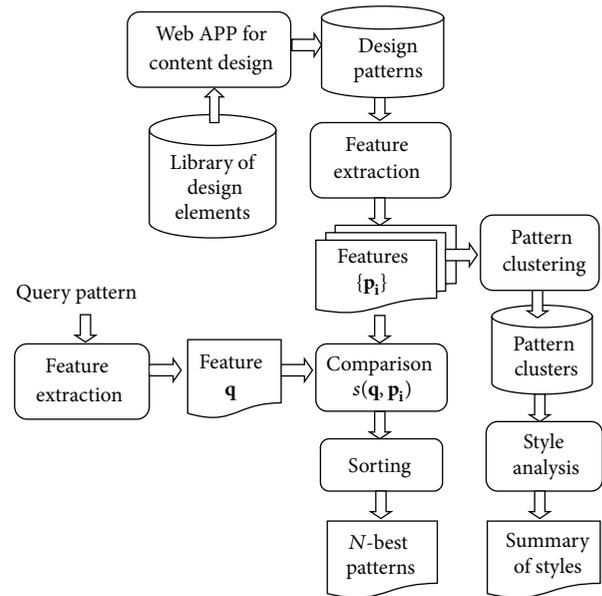


FIGURE 1: Architecture for design pattern retrieval and style analysis.

of Michael Jackson, for example, could be tagged with “pop,” “rock,” and “star,” while the image of Obama could be tagged with “U.S.,” “president,” and “politician.” In addition to the shared elements in the library, images can be imported for design from other sources, such as the camera, the drawing panel, and the web sites like Google or Flickr.

A web application based on the client and server architecture is further provided for the designers to compose the comic figures with the elements. An example of the user interface of the web application for assembling and editing the comic figures is shown in Figure 3. During composition, the designers can adjust the attributes of the design elements such as the size, location, transparency, font text, and font type, while adding the special effects such as rotation or mirror. In addition, the designer can search for desired elements with the keywords, such as *iron man*, or according to the categories, such as *politician*. Each output pattern contains a set of layered image objects of various types with respective attributes.

Figure 4 shows a few output patterns of comic figures composed with the web application. As can be observed in this figure, the example on the top right contains the head of Obama, the body with a nice suit, a luxury car, two shadows, a background scene containing the White House, and a caption with the title of “Yes! We Can.” The example on the bottom left of Figure 4, on the other hand, contains two photos, two cartoon figures, a prop of crown, and a caption. These patterns were composed and saved as the records in the database and used for retrieval and clustering later on.

3. Feature Representation and Pattern Retrieval

To compare the design patterns, the feature of each pattern, which includes the attributes of various objects, needs first



FIGURE 2: Samples of design elements with various types of content, including heads, bodies, shadows, props, fonts, and background scenes.



FIGURE 3: Example of user interface for composing comic figures.

to be extracted and represented in a structural form. This is accomplished in the procedure of “feature extraction” as shown on the top or left part of Figure 1. With the features, the similarities between the query pattern and the target patterns in the database can be measured and then sorted to obtain the

most similar patterns, as depicted in Figure 1 and discussed as follows.

3.1. Feature Extraction. In order to compare the design patterns, the attributes for every pattern need to be extracted and represented structurally. As can be seen from the examples in Figure 4, every pattern contains a variable number of heterogeneous image objects with respective attributes, so conventional feature representation based on the popular bag-of-words model [22] for information retrieval appears too simple and imprecise to be applicable. In this paper, the keywords of the image objects and the attributes of the image objects assigned during composition, such as the location, size, and transparency, were extracted to represent the feature of the pattern. A pattern can then be described with a set of heterogeneous objects with variable attributes.

Figure 5 shows an example of attributes extracted from an image object. In Figure 5, the attribute table on the right hand side corresponds to the head object on the left hand side. On the other hand, the caption on the left hand side has the attributes of *x*, *y*, *size*, *font text*, *font type*, and *font color*. Since each pattern contains a variable number of heterogeneous objects, it is necessary to consider the types of the objects when matching two patterns, which will be discussed in the following section.



FIGURE 4: Example design patterns containing the elements of various types from different sources.



Attribute	Value
id	P0320081016195102
Type	Picture/head
Size	50
x	19 px
y	57 px
Transparency	0
Special effect	No
Rotation angle	0
Mirror effect	No
Tags	Uncle Mau (role name) Cape no. 7 (movie name)

FIGURE 5: An example of extracted attributes corresponding to the head object.

3.2. *Similarity Measure and Pattern Retrieval.* Similarity or dissimilarity measures are critical for quite a few tasks on information retrieval and machine learning [23, 24]. Because the feature of a comic pattern in this study is structural and contains a set of heterogeneous objects, it is inappropriate to compute the similarity between two patterns based on

the classical cosine similarity. Assume the features of two patterns are \mathbf{p} and \mathbf{q} , where $\mathbf{p} = \{p_i\}$, $\mathbf{q} = \{q_j\}$, and p_i 's and q_j 's are the image objects in the two patterns, respectively. The numbers of elements in \mathbf{p} and \mathbf{q} and the types of image objects, p_i and q_j , may be different, as depicted in Figure 6(a). Different symbols in Figure 6(a) stand for different types of

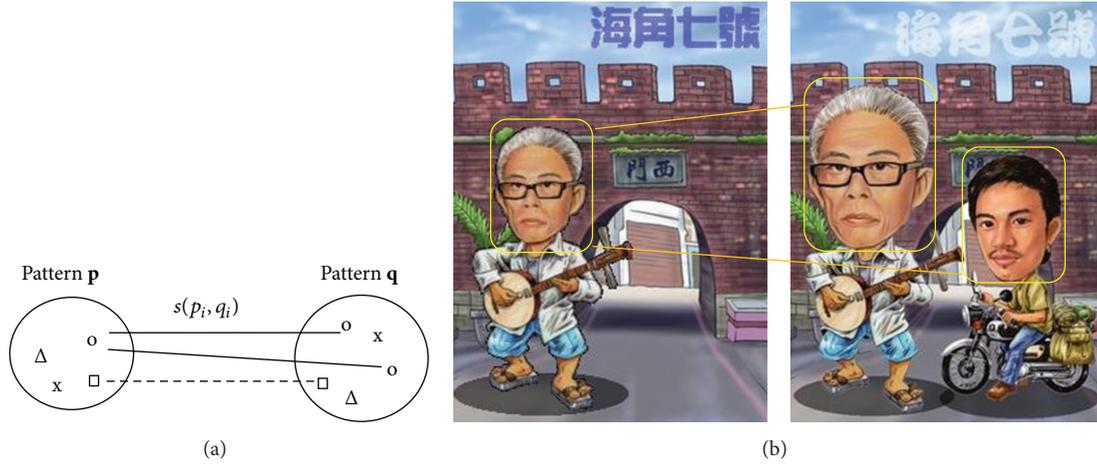


FIGURE 6: Match between two patterns.

image objects, such as head, body, or font. Since there might be more than one object for a type, for an object in one pattern it is reasonable to find the most matched object of the same type in the other pattern. The similarity between the two matched objects then contributes to the similarity of the two patterns. In this way, the similarity between the two patterns \mathbf{p} and \mathbf{q} can be accumulated for all matched pairs of objects. Figure 6(b) is further used to illustrate how two patterns are compared. The pattern on the left hand side of Figure 6(b) contains one head and one body, but the pattern on the right hand side contains two heads (elder and younger) and two bodies. It is intuitive to find the most matched head object on the right hand side of Figure 6(b) for the head object on the left hand side. When two patterns are compared, every object in one pattern is used to find the most matched object of the same type in the other pattern individually, as shown in Figure 6(a), and the similarities of all matched pairs are accumulated into the similarity of the two patterns.

However, how should the similarity between two objects be defined to measure the degree that they are similar to each other? Intuitively, the similarity should be 0 if p_i and q_j are of different types and could be the degree of similarity for all attributes between the two objects, otherwise, as follows:

$$s(p_i, q_j) = \sum_k \text{hit}(p_i[k], q_j[k]) \quad (1)$$

if type (p_i) equals type (q_j),
0 otherwise.

Here p_i and q_j are the objects in the two patterns, respectively, k signifies the attribute index, and $p_i[k]$ and $q_j[k]$ signify the values of attribute k for p_i and q_j , respectively. In addition, $\text{hit}(p_i[k], q_j[k])$ gives a positive score that indicates that the values of attribute k meet the criteria of similarity for that attribute and is 0 elsewhere. In other words, the score is positive when the two objects are similar enough on that attribute and 0 if they are not. The criteria of similarity can be defined flexibly according to the characteristic of the

TABLE 1: Criteria for attribute similarity.

Attribute	Criteria for similarity
id	The same image
Location	Distance less than 30 pixels
Size	Difference smaller than 10 pixels
Transparency	Difference of alpha smaller 30 (0–255)
Color change	Difference for specific band of RGB is lower than 30 (0–255)
Name/tags	Contains the same keyword
Mirror	Mirror effect applied for both
Font size	Difference smaller than 8 points
Font type	The same font type
Font text	Contains the same keyword
Font color	Euclidean distance in RGB space smaller than 30 (0–255).

attribute, as shown in Table 1. For the attribute *location*, for instance, the criteria could be whether the two objects are close enough spatially, while that for the attribute *font color* could be whether the difference between two font colors in color space is small enough. In addition, higher weights of similarity could be assigned to more important attributes. In this study, doubled weight is imposed on the attribute *id* because the image itself is very distinguishable visually. For the attribute *tags* containing a list of keywords, the number of common keywords is taken into account so as to increase the weight of the attribute. In this way, the similarity between two objects can be measured numerically by accumulating the similarities for all attributes.

With the similarity of objects, $s(p_i, q_j)$, depicted in (1), the most matched image object q_i^* in \mathbf{q} corresponding to the object p_i in \mathbf{p} can be computed as

$$q_i^* = \operatorname{argmax}_{q_j \in \mathbf{q}} s(p_i, q_j), \quad (2)$$

where the object q_i^* and the object p_i are of the same type. Table 2 shows an example for illustrating the computation of similarities between the head object on the left hand side of Figure 6(b) and the two objects (elder head and younger head) on the right hand side of Figure 6(b), respectively. As can be seen from Table 2, the total similarities for the elder head and the younger head are 5 and 2, respectively, so the elder head is the most matched object, and the similarity score 5 is accumulated into the similarity between the two patterns in Figure 6(b).

The similarity between the two patterns, \mathbf{p} and \mathbf{q} , can thus be computed by accumulating the similarities for all most matched pairs of objects as follows:

$$s(\mathbf{p}, \mathbf{q}) = \sum_{p_i \in \mathbf{P}} s(p_i, q_i^*). \quad (3)$$

Based on this definition, the similarities between a query pattern with the feature \mathbf{q} and the target patterns with the features \mathbf{p}_i 's can be computed one by one, as shown in the middle of Figure 1. All the similarities can be further sorted to obtain the most similar patterns for the query pattern, which can be formulated as

$$P = \text{nbest}_i s(\mathbf{q}, \mathbf{p}_i), \quad (4)$$

where nbest denotes the function of finding N -best patterns that have the highest similarities with the query pattern. Note that the similarity defined in (3) is asymmetric, which is valid for pattern retrieval since all the similarities can be accumulated based on the query pattern. However, when applied to the pattern clustering, the similarity or the distance should be symmetric, so the following definitions are used instead:

$$s'(\mathbf{p}, \mathbf{q}) = \frac{1}{2} [s(\mathbf{p}, \mathbf{q}) + s(\mathbf{q}, \mathbf{p})], \quad (5)$$

$$d(\mathbf{p}, \mathbf{q}) = 1 - \frac{s'(\mathbf{p}, \mathbf{q})}{s'_{\max}}. \quad (6)$$

The processing stages for retrieving the N -best patterns are summarized as a flowchart shown in Figure 7. The query pattern and target patterns are first extracted from the database and represented as features, denoted as \mathbf{q} and \mathbf{p}_i 's, respectively. The similarities between query feature (\mathbf{q}) and the target features (\mathbf{p}_i 's) can be computed according to (1)–(3) one by one and sorted according to (4) such that N -best patterns for the query pattern can be obtained.

4. Pattern Clustering and Style Analysis

Traditionally, design style usually refers to a prominent group of similar patterns in which implicit rules of attributes exist. To analyze the design styles therefore implies to find out the groups of design patterns and to summarize the styles over them. In this section, clustering algorithms were used to generate the clusters of design patterns, based on which the summarization for design styles could be performed, as shown on the right hand side of Figure 1.

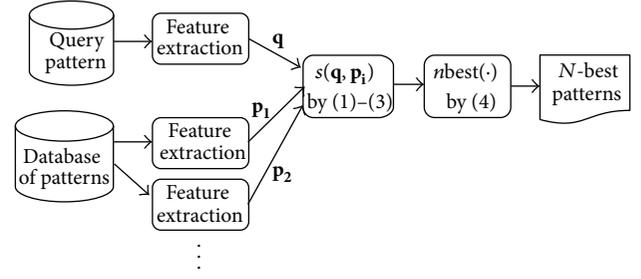


FIGURE 7: Flowchart of retrieving N -best patterns for a query pattern.

TABLE 2: Example for the computation of similarities.

Attribute	Elder head	Younger head
id	2	0
Location	0	0
Size	0	1
Transparency	1	1
Color change	0	0
Name/tags	2	0
Mirror	0	0
Font size	0	0
Font type	0	0
Font text	0	0
Font color	0	0
Total	5	2

4.1. Clustering Design Patterns. Since the distance between two patterns has been defined in (6), it is not difficult to conduct agglomerative clustering for all the patterns [25, 26]. In agglomerative clustering, each pattern initially belongs to a cluster individually. The distances among all pairs of clusters are then computed so as to decide the closest pair of clusters and merge them. This procedure is executed iteratively, and smaller clusters are merged gradually to produce larger clusters according to the distances, as shown in Figure 8(a). Finally, all clusters are merged into one cluster, and a tree structure called *dendrogram* is constituted, as shown in Figure 8(a). Afterwards, a threshold of distance can be set to split the dendrogram into clusters. For the case shown in Figure 8(a), for instance, the threshold can split the dendrogram into three clusters, $\{\mathbf{p}_1, \mathbf{p}_2\}$, $\{\mathbf{p}_3, \mathbf{p}_4, \mathbf{p}_5\}$, and $\{\mathbf{p}_6\}$, respectively.

In agglomerative clustering, the distance between two clusters could typically be the maximum distance, the minimum distance, or the average distance, among the distances of all pairwise patterns [25]. Here in this study the maximum distance is utilized, as illustrated in Figure 8(b), because it can strictly constrain the similarity for the patterns within every cluster and produces more compact clusters [26]. In addition, the threshold of distance stands for the maximum allowable distance among patterns within every cluster and can be adjusted to obtain different numbers of clusters flexibly. The limitation of agglomerative clustering is that, it may require a

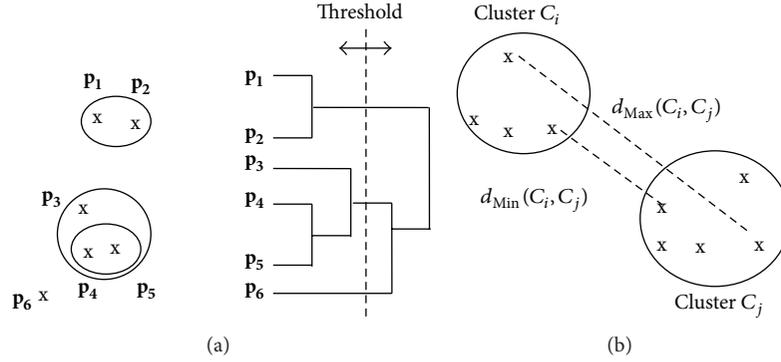


FIGURE 8: Agglomerative clustering (a) agglomeration and dendrogram (b) distance computation.

lot of computations for computing the pairwise distances and is therefore suitable for moderate amount of data only.

In addition to agglomerative clustering, k -means clustering is the most commonly used clustering algorithm. It is popular mainly because it is simple and efficient. Also, it works fairly well for quite a few applications. Conventional k -means clustering is based on square-error criterion,

$$E_k = \sum_{x_i \in C_k} (x_i - m_k)^2, \quad (7)$$

and the centroid of each cluster is computed as

$$m_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i. \quad (8)$$

However, for the structural design patterns studied in this paper, it is impossible to compute the centroid for a cluster of patterns according to (8), because each pattern contains a set of objects whose attributes are perhaps categorical instead of numerical (e.g., *font type* or *tags*). To deal with this issue, here a *pseudo centroid* based on the minimization of average distance is proposed. Note that the conventional centroid in (8) is a point that minimizes the within-cluster variance and the average distance. Though it is infeasible to find the average for nonnumerical data, it is possible to compute the pairwise distance of patterns according to (3). Hence, the pseudo centroid for a cluster can be defined as the pattern that achieves the minimum average (or total) distance in a cluster, as follows:

$$\hat{m}_k = \operatorname{argmin}_i \left(\sum_{\substack{j \\ p_j \in C_k}} d(\mathbf{p}_i, \mathbf{p}_j) \right). \quad (9)$$

Accordingly, the k -means clustering can be modified and conducted for the structural patterns in this study. Note the pseudo centroid here is not the *average* of the patterns in fact, but a pattern that is selected from the cluster and regarded as close to the *center* of the cluster relatively. Therefore, the criterion of minimizing the square error for k -means clustering does not hold for the modified version.

4.2. Style Analysis. After the clustering was performed, a few clusters of patterns were produced. Since every cluster contains a set of patterns that are similar to one another, it is possible to collect the statistics of the attributes for each cluster so as to find out the key attributes that contribute mostly to the similarity of the patterns within the cluster. This is actually the issue of summarization in data mining. However, because some attributes of image objects here are categorical variables (e.g., *font type* or *tags*) instead of continuous or numerical variable, it is infeasible to summarize a cluster according to the means, variances, or ranges for the attributes. In this section, an analysis approach based on the attribute similarity was proposed, based on which summarization can be performed.

As depicted in (1), the similarity between objects can be accumulated for all attributes such that the similarity between two patterns can be obtained, as illustrated in (3). To summarize the style on a cluster, the similarity between pairwise patterns within the cluster in (2) and (3) could be computed once again to collect the statistics of the attributes. Let \mathbf{p} and \mathbf{q} denote two patterns within a cluster, C . The statistics of the attributes can then be collected based on the procedure of Algorithm 1, in which q_i^* is the most matched object in pattern \mathbf{q} for the object p_i in pattern \mathbf{p} , as depicted in (2), and k is the attribute index for the objects p_i and q_i^* . It can be seen from Algorithm 1 that the count of attribute k is increased when the attribute values of p_i and q_i^* meet the criteria of similarity for the attribute. Moreover, when the attribute is *name*, *tags*, or *font text* that contains perhaps multiple keywords, the content strings need to be further parsed such that the statistics of the keywords can be collected to summarize the main topics for the cluster.

5. Experiments and Analysis

5.1. Retrieval of Similar Patterns. The web application described in Section 2 was used to compose 709 comic figures by 66 designers. These patterns were extracted from the database, represented as the features according to the attributes of objects, and used for retrieval. Query patterns were also selected randomly from these patterns and used to

```

for every  $\mathbf{p}, \mathbf{q} \in C$  and  $\mathbf{p} \neq \mathbf{q}$ 
  for every  $p_i \in \mathbf{p}$  and  $q_i^* \in \mathbf{q}$  as the most matched pair
    for each attribute  $k$  for  $p_i$  and  $q_i^*$ 
      if ( $\text{hit}(p_i[k], q_i^*[k]) > 0$ )
        count  $[k] += \text{hit}(p_i[k], q_i^*[k])$ 
        if ( $k$  is name, tag or font text)
          parse the strings of  $p_i[k]$  and  $q_i^*[k]$  and
          collect the statistics of the keywords
        end if
      end if
    end for
  end for
end for
end for

```

ALGORITHM 1: Procedure of collecting the statistics of attributes and keywords for a cluster of patterns.

TABLE 3: Average of means and standard deviations for pairwise similarities for all clusters.

Clustering	K-means partitioning clustering						Agglomerative clustering	
	50	100	150	200	250	300	253 (6.0)	305 (6.5)
Number of clusters	50	100	150	200	250	300	253 (6.0)	305 (6.5)
Average of means	5.34	6.40	6.95	7.29	7.63	8.04	8.09	8.4
Average of standard deviations	1.81	1.61	1.40	1.36	1.17	0.97	0.61	0.49

retrieve 10-best patterns that are most similar to the query pattern, according to the flowchart depicted in Figure 7. The retrieval results are shown in Figure 9. In both Figures 9(a) and 9(b), the top-left icons are the query patterns and the rest icons are the corresponding retrieved patterns. As can be seen from this figure, the proposed scheme is able to find out visually similar patterns, which contain either common elements or the elements with similar attributes of size, location, name, tags, and so on. In addition, it can be seen that the icons with higher similarity look more similar to the query pattern in general. The patterns in the upper row, for example, are more similar than those patterns in the lower row visually. This implies that the proposed similarity measure based on the attributes can achieve quite adequate discriminative capability for the structural patterns.

Moreover, it can be observed that a few elements are commonly used in the 10-best patterns when compared with the query pattern. Therefore, the designers can learn efficiently from the retrieval results how the elements used in their works are utilized differently or innovatively by other designers. For example, they may learn how a character (head object) can be accompanied with different background scenes, props, or bodies, and what their relative locations are. This may help the designers to improve the design skills and inspire them for creation. It can also encourage the exchange of ideas and facilitate the sharing of such resources as the patterns in a design support system.

5.2. Analysis of Pattern Clusters. The patterns can be further clustered by the algorithms depicted in Section 4.1. First, the k -means clustering was performed. Note that for k -means clustering, if the clusters are initially selected randomly, its performance is unstable and very sensitive to the initial selection. Therefore, here in this study the largest cluster was

selected to split into two clusters iteratively so as to increase the cluster number gradually. In addition, the clustering is performed based on pairwise similarity, so the mean and variance of pairwise similarities for each cluster can be defined as

$$\begin{aligned}
m_{\text{PS}} &= \frac{1}{C_2^n} \sum_{i \neq j} s(\mathbf{p}_i, \mathbf{p}_j), \\
\sigma_{\text{PS}}^2 &= \frac{1}{C_2^n} \sum_{i \neq j} (s(\mathbf{p}_i, \mathbf{p}_j) - m_{\text{PS}})^2,
\end{aligned} \tag{10}$$

respectively, where n denotes the number of patterns in the cluster, C_2^n with the value of $n(n-1)/2$ denotes the number of unequal pairs of patterns in the cluster, and \mathbf{p}_i and \mathbf{p}_j are two patterns within a cluster. The average of the means and standard deviations for pairwise similarities for all clusters can then be used to measure the overall closeness for all clusters.

In addition, since the space of the structural patterns is sparse, large number of clusters is necessary in order to obtain visually similar clusters. Therefore, k -means clustering was conducted on the 709 patterns for the group number from 50 to 300 with a step of 50, and the results are displayed in Table 3. It can be seen from Table 3 that the average of means increases as the number of clusters is increased from 50 to 300. However, the average of standard deviations decreases slowly because the pairwise similarities within every cluster are not constrained strictly and more outliers are included. Figure 10 further shows two examples of clusters obtained from k -means clustering with 200 clusters. As can be seen from this figure, the k -means clustering tends to produce the clusters with outliers that are visually less similar to the other patterns.



(a)



(b)

FIGURE 9: Two examples of retrieval results for the query patterns on top-left corner.

Furthermore, agglomerative clustering depicted in Section 4.1 was conducted for the patterns, and a strict threshold of distance was selected such that every pairwise similarity within a cluster must be above a similarity threshold. Note that the maximum distance applied during agglomeration as depicted in Figure 8(b) also imposes the constraint of minimum pairwise similarity according to (6). Therefore, agglomerative clustering can constrain all pairwise similarities within every cluster strictly. The averages of means and standard deviations for minimum similarities of 6.0 and 6.5, corresponding to 253 and 305 clusters are 8.09/0.61 and 8.40/0.49, respectively, as can be

seen from Table 3. Such result means that, for structural patterns, agglomerative clustering is more effective than *k*-means clustering in average for obtaining visually similar patterns within every cluster. 46 large clusters with more than three patterns and 173 small chunks can be obtained finally when excluding 86 outliers from the dendrogram for the 305 clusters. A few patterns in four large clusters are displayed in Figure 11, in which every row corresponds to a cluster. As can be observed in this figure, the patterns in every cluster show high degree of similarity and the styles can be easily discriminated visually. The topics of the patterns for the four clusters are mainly the horse-riding dance in a Korean

TABLE 4: Contributions of the attributes to pattern similarity within every cluster.

Attribute	Cluster 1	Cluster 2	Cluster 3	Cluster 4
id	24.36%	20.23%	23.08%	21.71%
Location	5.95%	5.84%	7.69%	7.75%
Size	12.18%	14.01%	23.08%	24.03%
Name	12.18%	13.23%	11.54%	10.85%
Tags	42.78%	40.08%	34.62%	32.56%
Font size	1.42%	0.78%	0	0.78%
Font type	0.85%	0.39%	0	0
Font color	0.28%	5.45%	0	2.33%
Keywords with high frequency	Keywords relevant to a Korean show: Jiang-Nan (role), Pu Jae-Shiang (actor), suits of horse-riding dance, horse-riding dance.	Keywords relevant to the movie "Iron Man": space base, Iron Man II, Iron man.	Keywords relevant to a comic star in a Chinese drama: God of wealth, Stephen Chou (actor), hat for God of wealth.	Keywords relevant to pop music: male singer, A-Shin (singer), Mayday (band), with suit and microphone.

FIGURE 10: Example of clusters generated by k -means clustering with 200 clusters.

show, the scenes in the movies *iron man* and *iron man II*, a comic movie star in a Chinese drama, and a Taiwanese rock band, respectively. The visual clues are so prominent that the designers can make conclusions on the topics of the clusters by themselves instinctively.

In addition, the cooccurrences of image objects with high frequency imply the preference of collocation or the correlation that might be socially and culturally meaningful. The character and the body in the first cluster, for example, are from a Korean show, and the character and the prop in the third cluster are from a Chinese drama. Therefore, such analysis for patterns could help to discover the public images, designers' behaviors, or cultural knowledge that might be delivered implicitly in the designed patterns. Moreover, the main clusters can give the designers a quick and broad view of

the design works in an orderly manner and make them avoid being overwhelmed by a lot of disordered patterns. Therefore, it can also be used for browsing the patterns structurally and systematically, which can potentially improve the user experiences for design support systems.

5.3. Automatic Style Analysis. When the style analysis proposed in Section 4.2 was further applied to the clusters, the contributions of the attributes to the similarity for each cluster could be estimated according to the statistics computed from the procedure of Algorithm 1. The statistics of the attributes and the summary of important keywords with high frequency corresponding to the four clusters in Figure 11 are further displayed in Table 4. As can be observed from this table, clusters 3 and 4 have relatively higher ratio of similarity

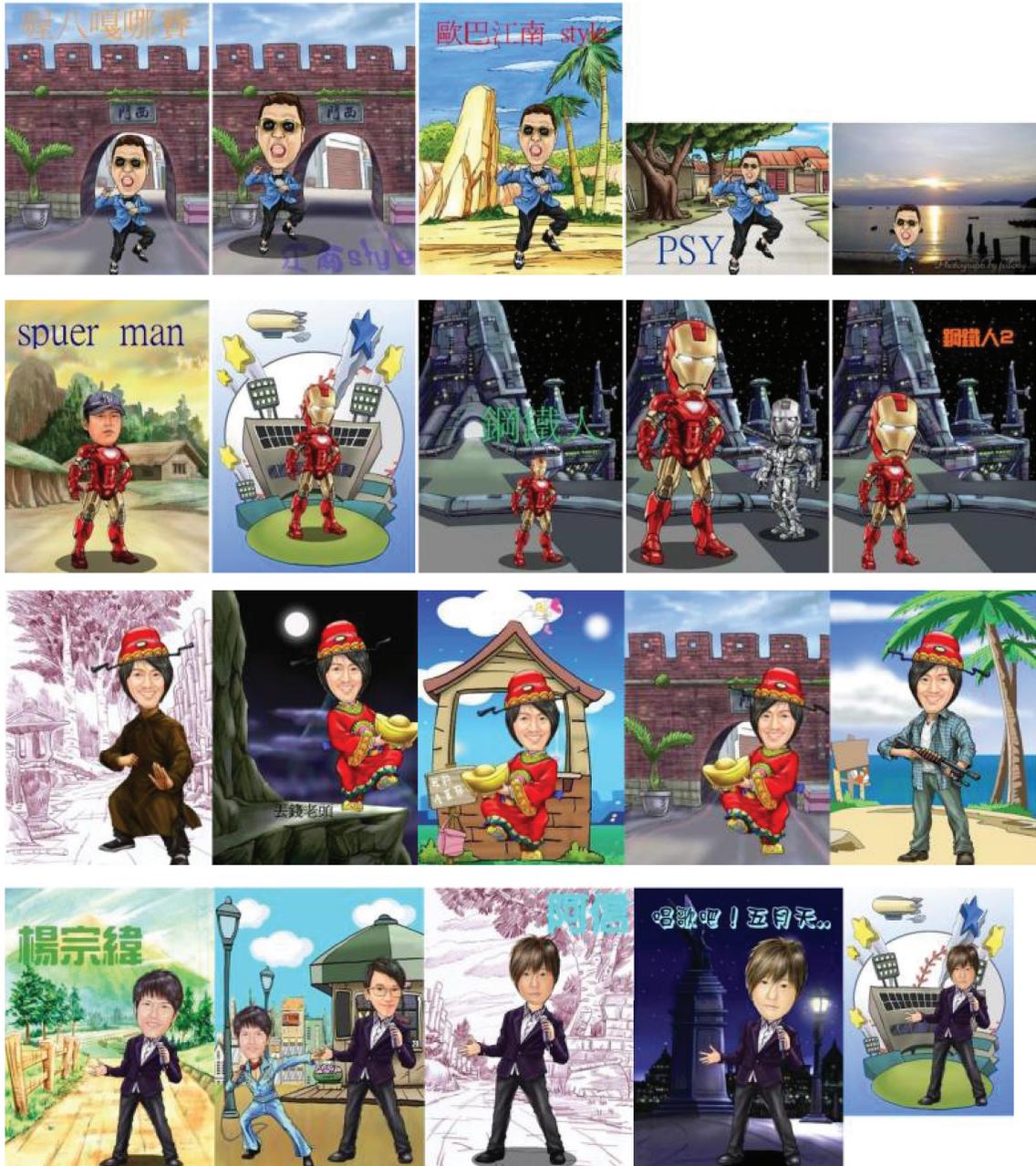


FIGURE 11: Three examples of clusters with a part of patterns.

on the attribute of *size*, and cluster 1 has the highest similarity on the attribute of *tags* due to more common head and body. The most relevant keywords for each cluster can be summarized automatically according to their occurrence counts, as shown in Table 4, and used to indicate the main topics of the cluster. Through the proposed approach, it becomes possible to perform style analysis and summarization on pattern clusters automatically at a significantly lower cost than conventional summarization by experts. The analysis results may be helpful for both the learners and the tutors and applicable to design learning systems.

6. Conclusions

In this paper, the architecture for representing, comparing, retrieving, and analyzing the design patterns of digital contents was proposed. Based on this scheme, design support systems can be built to assist the content creation and facilitate the sharing of patterns among the designers. Designers can view the design patterns similar to their works efficiently and see how the design elements used in their works are utilized differently and innovatively by others. These observations can help the designers to enhance their skills and stimulate their imaginations during

the process of learning or creating. The proposed scheme has been verified with a design support system for the content creation of comic figures; thus, the design elements, which include objects of various types, can be searched, edited, and assembled into patterns. Subsequently, these patterns can be compared, retrieved, and clustered, while the design styles can be analyzed and summarized efficiently. This scheme is generally applicable to the creation of digital contents, and it shows potential for applications in the fields of design and education.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] K. Honda and F. Mizoguchi, "Constraint-based approach for automatic spatial layout planning," in *Proceedings of the 11th Conference on Artificial Intelligence for Applications*, pp. 38–45, 1995.
- [2] L.-F. Yu, S.-K. Yeung, C.-K. Tang, D. Terzopoulos, T. F. Chan, and S. J. Osher, "Make it home: automatic optimization of furniture arrangement," *ACM Transactions on Graphics*, vol. 30, no. 4, article 86, 2011.
- [3] P. Merrell, E. Schkufza, Z. Li, M. Agrawala, and V. Koltun, "Interactive furniture layout using interior design guidelines," *ACM Transactions on Graphics*, vol. 30, no. 4, article 87, 2011.
- [4] Y.-C. Lin, Y. Kawakita, E. Suzuki, and H. Ichikawa, "Personalized clothing recommendation system based on a modified Bayesian network," in *Proceedings of the 12th International Symposium on Applications and the Internet (SAINT '12)*, pp. 414–417, 2012.
- [5] Y. Kalatidis, L. Kennedy, and L.-J. Li, "Getting the look: clothing recognition and segmentation for automatic product suggestions in everyday photos," in *Proceedings of the International Conference on Multimedia Retrieval (ACM ICMR '13)*, pp. 105–112, 2013.
- [6] L. Bossard, M. Dantone, C. Leistner, C. Wengert, T. Quack, and L. V. Gool, "Apparel classification with style," in *Proceedings of the Computer Vision (ACCV '12)*, vol. 7727 of *Lecture Notes in Computer Science*, pp. 321–335, 2013.
- [7] K. Arai and H. Tolle, "Automatic e-comic content adaptation," *International Journal of Ubiquitous Computing*, vol. 1, no. 1, pp. 1–11, 2010.
- [8] H. Tobita, "Comic computing: Creation and communication with comic," in *Proceedings of the 29th ACM International Conference on Design of Communication (SIGDOC '11)*, pp. 91–98, October 2011.
- [9] J.-K. Chou and C.-K. Yang, "Simulation of face/hairstyle swapping in photographs with skin texture synthesis," *Multimedia Tools and Applications*, vol. 63, no. 3, pp. 729–756, 2013.
- [10] X. Wang and K. Li, "Pattern recognition based on fuzzy cluster for recognizing garment style in the photo," in *Proceedings of the 9th International Conference on Computer-Aided Industrial Design and Conceptual Design: Multicultural Creation and Design (CAIDCD '08)*, pp. 250–254, November 2008.
- [11] C. Zhang, G. Liu, and Z. Wang, "Cartoon face synthesis based on Markov Network," in *Proceedings of the 18th International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS '10)*, December 2010.
- [12] C. Sauvaget and V. Boyer, "Comics stylization from photographs," in *Proceedings of the 4th International Symposium on Visual Computing (ISVC '08)*, pp. 1125–1134, 2008.
- [13] D. Jung, Y. K. Hui, H.-K. Lee, J. Nam, and W. H. Jin, "Development of a packaging storage format for electronic comic services with metadata," in *Proceedings of the 26th IEEE International Conference on Consumer Electronics, The Mobile Consumer (ICCE '08)*, pp. 1–12, January 2008.
- [14] K. Arai and T. Herman, "Method for automatic E-comic scene frame extraction for reading comic on mobile devices," in *Proceedings of the 7th International Conference on Information Technology—New Generations (ITNG '10)*, pp. 370–375, April 2010.
- [15] T. Kohler, J. Fueller, K. Matzler, and D. Stieger, "CO-creation in virtual worlds: The design of the user experience," *MIS Quarterly*, vol. 35, no. 3, pp. 773–788, 2011.
- [16] R. Fruchter, "Internet-based web-mediated collaborative design and learning environment," in *Artificial Intelligence in Structural Engineering*, pp. 133–145, Springer, 1998.
- [17] Ö. Özyurt, H. Özyurt, and A. Baki, "Design and development of an innovative individualized adaptive and intelligent e-learning system for teaching-learning of probability unit: details of UZWEBMAT," in *Expert Systems with Applications*, vol. 4, pp. 2914–2940, 2012.
- [18] D. Romero and A. Molina, "Collaborative networked organisations and customer communities: value co-creation and co-innovation in the networking era," *Production Planning and Control*, vol. 22, no. 5–6, pp. 447–472, 2011.
- [19] M. Gilbert and M. Zachry, "Sharing time: engaging students as co-designers in the creation of an online knowledge sharing application," in *Proceedings of the 30th International Conference on Design of Communication (ACM SIGDOC '12)*, pp. 121–126, 2012.
- [20] I. Alkholi, "Using hypermedia in teaching baroque Interior design as seen in Dutch paintings," *International Journal of Emerging Technologies in Learning*, vol. 5, no. 3, pp. 37–41, 2010.
- [21] J. Xiao, M. Wang, L. Wang, X. Zhu, and J. Ng, "Design and implementation of a cloud-learning service platform," in *Proceedings of the International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT '12)*, pp. 296–300, 2012.
- [22] B. Liu, *Web Data Ming: Exploring Hyperlinks, Contents and Usage Data*, Springer, 2011.
- [23] J. Zobel and A. Moffat, "Exploring the similarity space," *SIGIR Forum*, vol. 32, no. 1, pp. 18–34, 1998.
- [24] R. P. W. Duin and E. Pçkalska, "The dissimilarity space: bridging structural and statistical pattern recognition," *Pattern Recognition Letters*, vol. 33, no. 7, pp. 826–832, 2012.
- [25] M. Kantardzic, *Data Mining: Concepts, Models, Methods, and Algorithms*, Wiley Inter-Science, Piscataway, NJ, USA, 2003.
- [26] E. Alpaydin, *Hierarchical Clustering. Introduction to Machine Learning*, MIT Press, 2004.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

