Journal of Information Hiding and Multimedia Signal Processing ©2021 ISSN 2073-4212
Ubiquitous International Volume **12**, Number **2**, June **2021**

Stuck Query Point Processing Of Multi-point Query For Image Retrieval With Relevance Feedback

An Hong Son
Viet-Hung University
Hanoi, Viet Nam
sonanhongvh@gmail.com

Quynh Dao Thi Thuy
Faculty of Information Technology
Posts and Telecommunications Institute of Technology
Hanoi, Viet Nam
thuyquynhtn90@gmail.com

Quynh Nguyen Huu
Faculty of Computer Science and Engineering
Thuyloi University
Hanoi, Viet Nam
quynhnh@tlu.edu.vn

Received October 2020; revised February 2021

Abstract. *Many previous techniques were designed to retrieve semantic images in the entire feature space but with low precision because of bypassing some points in stuck the multi-point query. In this article, we propose a semantic-related image retrieval method, which can retrieve semantic images spread in the entire feature space with high precision, and the method is based on dealing with stuck points in multi-point queries called STUP (Stuck Query Point Processing). Our method combines user feedback with information that is about similarity among the neighbors to choose "the best" alternative point for stuck query points. We also provide experimental results based on two feature databases to demonstrate the accuracy of our method.*
Index Terms: Content-based image retrieval (CBIR), Semantic-related image, similarity-induced semantic relevant image cluster, Stuck Query Point.

1. **Introduction.** Content-based image retrieval (CBIR) has received a lot of attention over the past decade due to the demand for the efficiency of processing an enormous amount of multimedia data, which rapidly grows. Many CBIR systems have been developed, including QBIC [14], Photobook [33], MARS [32], Netra [28], PicHunter [12], Blobworld [8], VisualSEEK [44], SIMPLIcity [47] and some other systems [5, 7, 10, 18, 20, 26, 31, 41, 46]. In a typical CBIR system, the low-level visual image features (color, texture, and shape) are extracted automatically for the goal of indexing and describing images. Besides that, in CBIR's system, two vectors are considered close if two images that correspond to them are similar. To search for the desired images, the user inserts a sample image and the system returns a set of similar images based on the extracted features. Humans have a natural tendency to use high-level features(concepts) such as keywords, written descriptions to explain images, and the similarity between them while the huge number of features automatically extracted using computer vision techniques, are low-level features[40]. Although there are a big number of complex algorithms designed to describe color features, shape, and texture, these algorithms are not semantic-related image models, and they also have lots of limitations when facing big-size images' database

[29]. The above experiments on the CBIR system point out that low-level content often fails when describing semantic-related high-level concepts in user understanding [52]. Thus the efficiency of the system is far from user expectations.

Searching for semantic-related images in the content-based image retrieval systems mentions searching the images that are semantic-related according to user understanding [45, 27, 2, 51, 26, 13, 19], such as green cars and red cars are semantic-related. The approach based on relevance feedback compared with content-based image retrieval, is an active research area for the past few years. Some good experiments that follow this approach can be found in [45, 32, 38, 9, 24, 20, 3, 22, 30, 16, 5]. When CBIR systems present a set of images considered to be similar to a given query image, the user can retrieve the most relevant images for that given query and the systems adjust the query using relevant images that user has selected. CBIR techniques based on relevance feedback do not require the user to provide the initial query correctly but require the user to construct the optimal query by evaluating whether images are relevant. Relevance feedback techniques belong to the two following approaches [34]: single-point and multi-point query shift. A technique that is classified as a single-point shift approach if the shift query at each repeated time only has one point. In reverse, it is a multi-point shift approach. Techniques that belong to the single-point shift approach [32, 38] represent a new query by a single point and adjust the weights of feature components to find an optimal query point and distance function. In this case, a single point is calculated by the average weight of all relevant images in the feature space. While techniques that belong to multi-point query shift approach [9, 24, 20], based on previous analysis, explain that semantic-related images usually spread in different clusters so single-point techniques are not efficient to get sematic-related images which exist in different clusters so using multi-point techniques to present a new query is necessary. Thus we concentrate on retrieving images using the multi-point technique in this article, and we also focus on dealing with things in the multi-point technique which cause low efficiency.

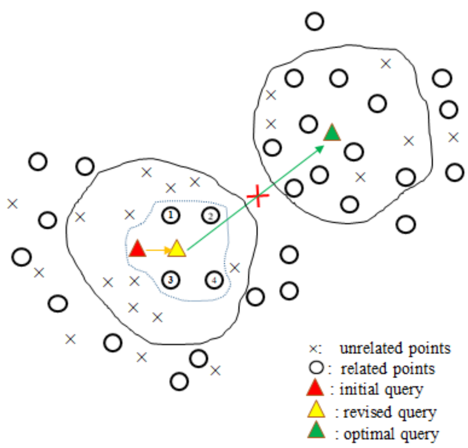

Figure 1. A limitation of current retrieval methods. Query points (revised query points are stuck)

In retrieval systems using previous relevance feedback, to reach an optimal query point from an initial query point, shifting between lots of zones is required. Query points on the path from an Initial query point to an optimal query point are revised query points. We call the zone which contains initial query points is the initial zone, which contains optimal query points is the optimal zone, which contains the revised query points is the revised zone. The process of shifting query points through revised zones can have one zone (or many zones) when query points (or revised query points) get stuck and it causes low retrieval precision. Figure 1 illustrates a possible scenario when executing the initial query, the system returns a list of 15 points (points in the initial zone on figure 1). Among these 15 points, we have 4 points called 1, 2, 3, 4 are related to the initial query. Calculating the centroid of the 5 points (including points 1, 2, 3, 4, and an initial query point). Based on this initial query point, the system returns once again the list of 5 related points of the initial query. This process happens again and again but it never reaches and is shifted to the optimal zone which has optimal points. Thus, in this case, the system is only able to return 5 related images (related points in the initial zone on figure 1), and not able to return 10 related images (related points in the optimal zone on figure 1) as expected. As a result, the system never reaches any optimal query points without even knowing it. We call the points (or revised query points) in this case "stuck query points". Our solution is to take a greater neighbor value k, however, choosing a suitable value for k is difficult.

The above limitations are one of the motivations for us to propose the retrieval method with relevance feedback handling stuck query points in multi-point query STUP. Instead of accepting poor results.at a stuck point in the multi-point query, our method selects "the best" alternative point compared with the current point and then continues retrieving at this alternative point. This method is able to achieve semantic-related images that spread in different visual feature clusters with high precision. By using experiments based on a feature database, which contains 10800 images, we will prove the accuracy of the proposed method.

The rest of the article is organized as follows. In part 2, we put our experiment in the context of relevance experiments. the retrieval method to deal with stuck query points is mentioned in part 3. Part 4 is about analyzing experimental results. Finally, conclusions and future research directions are shown in part 5.

2. **Related Works.** Relevance feedback (RF) is a powerful tool commonly used in text-based retrieval systems [39]. The tool is introduced in the middle of the 1990s, with the intention of including users into the retrieval process, thus alleviate the burden of semantic distance between what is described by queries (low-level features), and what is understood by users. RF has significantly improved the efficiency of the CBIR system as it continuously learns through interaction with users [36]. Efforts have been recognized in reducing semantic distance [43] between low-level features and high-level concepts, there are experiments which concentrate on retrieval methods based on relevance feedback. In this section, we will examine previous techniques that used the point-shift approach and multi-point query. We will also show in brief the image retrieval technique based on clusters because it is applied in our research.

The initial approach [14, 44] of content-based image retrieval is unsuitable for query and retrieval models based on the user's perception of visual similarity. To solve this problem, some relevance feedback techniques [4, 5, 6, 7, 22, 34, 37, 50, 48, 53] have been proposed. They attempt to set the cohesion between semantic concepts, low-level image features, and the user's visual perception from their own feedback. There are two components to learn relevance feedback: a distance function and a new query point. The distance function is changed by learning the weights of feature components and the new query point is obtained by learning the ideal query point where users search. Shifting query points were applied to image retrieval systems such as MARS [37] and MindReader [22]. These systems present query as a single point in the feature space and try to shift this point towards the positive points and away from the negative points. This idea originated from the Rocchio algorithm [35], which has been successfully used in document retrieval. In this approach, the weight technique assigns a weight to each dimension of a query point. It assigns the greater weights to more important dimensions and the smaller weights to less important ones. Mar uses the Euclidean distance which has weights, it handles ellipsoids which has the main axis is aligned with the coordinate axis. MindReader, on the other hand, uses a generalized Euclidean distance, which allows the axes to be rotated so that it works well with ellipsoids of any direction. There is also the existence of other methods about query point smoothing using multi-point relevance feedback have also been suggested. The query expansion approach [34] of Mars sets local clusters to related points. In this approach, local clusters are combined all together to generate a broad border that covers all the query points. On the other hand, the query point shift approach [37, 22] bypasses these clusters and considers all related points equally. To attain better results, we have recently proposed a semantic-related retrieval technique [45]. The two approaches above can generate a super ellipsoid or convex shapes that use local clusters in the feature space to overlay all query points for single queries. However, neither of the two approaches succeeds in recognizing the suitable zones for complex queries. [50] mentioned FALCON to facilitate the learning of concave and discrete query points in the vector space as well as the arbitrary measure space. However, the proposed dissimilar aggregation functions depend on special experiences, and this model assumes that all related points are query points.

In [11], a method named 'CLUE' is presented to reduce the semantic distance in CBIR. Unlike other CBIR systems, it attempts to retrieve semantic-related image clusters. By a given query image, a set of images in the database, which is similar to the query, is selected to be the query's neighbor. Based on the assumption that sematic-similar images exist in the same cluster, it clusters these database images into different semantic clusters. Our method makes full use of the information about the similarity between images in the database. However, CLUE's limitation is the poor semantic similarity between members in each cluster, and not all images in the database are considered. But this method has the strength of providing a semantic-related cluster without requiring user feedback.

In general, the above image retrieval techniques either get some neighbor images of query images or get query images that exist in the entire feature space but they cannot guarantee high precision in the case of stuck query points.

3. **The image retrieval method handles stuck query points.** As covered in the previous section, methods using the multi-point query shift approach meet a similar problem about query points (revised query points) get stuck, called stuck query points. In this section, we go deep into detail about how to solve stuck query points, in particular: we first propose some ideas for handling stuck query points. Next, we show the method's diagram with one point in the multi-point query. After that, we describe how to find neighbor sets and semantic-related clusters deduced from similarity. Finally, some proposed retrieval algorithms are presented.

3.1. **Handling stuck query points.** This section will give a solution in case that query points get stuck in certain adjustment zones. One of the solutions is to find an alternative query point. Thus, follow this solution, the problem of solving a stuck query point becomes the problem of finding a replacement for that stuck query point. The idea of solving this problem in our method is as follows: for each related point in an alternative query point's neighborhood in the previous response iteration (the first response iteration is an initial query point), finding "semantic-related clusters deduced from similarity"(See section 3.3) which contain that point(this cluster does not contain other related points). The reason, which is for finding a replacement for a stuck point by using semantic-related clusters deduced from similarity, is to alleviate the user burden in giving feedback about which candidate leads to the optimal query (normally, user burden increases with the number of related points in the neighborhood of revised query points that exist in the previous response iteration). Among the "semantic-related clusters deduced from similarity" which have just been found (each point will have a "semantic-related clusters deduced from similarity"), selecting the cluster which has the biggest size. Next, calculating the centroid of the cluster which has just been selected and then using this point to be the replacement for the stuck query point. Figure 2 illustrates the idea like that. Firstly, finding the "semantic-related cluster deduced from similarity" of point 1 on figure 2(this cluster does not contain points 2, 3, 4), this neighborhood does not have any semantic-related points so the cluster corresponding to point 1 contains only point 1. Then, doing the similar steps to find the "semantic-related clusters deduced from similarity" for points 2, 3, and 4, the result is that the "semantic-related cluster deduced from similarity", which contains point 2, only has a member is point 2, the result is that the "semantic-related cluster deduced from similarity", which contains point 3, only has a member is point 3, the "semantic-related cluster deduced from similarity", which has point 4, is a cluster that contains 5 members. Thus, the "semantic-related cluster deduced from similarity", which has point 4, is the biggest size so this cluster's centroid is selected to be the alternative point for the stuck query point. To continue, a question has been raised about how to get the "semantic-related cluster deduced from similarity" corresponding to each related point (points 1, 2, 3, 4 belong to the initial zone on figure 2)? the basic idea to solve this problem is as follows: on a certain related point, selecting r its neighbors to generate a hierarchical clustering tree by using the method NCut [42]. On this hierarchical clustering tree, the cluster corresponding to the most left leaf is a sematic-related cluster derived from the most similarity with the point which is being considered and contains that point, this cluster is the "semantic-related cluster deduced from similarity".

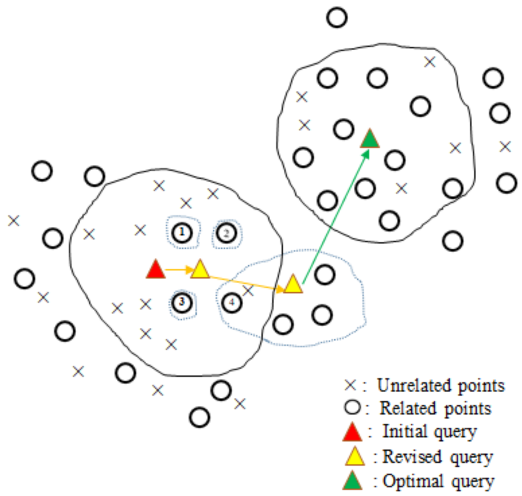

FIGURE 2. Illustrating the solution to get out of the stuck zone

3.2. **The method's diagram.** Figure 3 points out the proposed method's texture with a point in the multi-point query. The retrieval process starts when users input a query image Qinitial. .Query image Qinitial. will be mapped into a multi-dimensional feature space. Before that, database images were also mapped into this multi-dimensional feature space with the same steps as initialization query images. In a multi-dimensional space, the set of points, which are close to the query point, is selected to become the neighbor of the query point. These points create a set of results which are returned to users later. Next, users respond to this set of results to attain a relevance collection. Calculating the centroid of points in the relevance collection to get candidates for the revised query Qrevised . In the case that a candidate for the revised query does not get stuck, this candidate will be selected to become a revised point Qrevised and the process is iterated until it satisfies user requirements or encounters any stuck revised query points. In the case that a candidate for the revised query gets stuck, activating a block to handle stuck query points called STUP (Stuck Query Point Processing). This block works as follows: first, finding a set of points if these points are close to a certain related point in the relevance collection, and then we have a neighbor collection for a point. Finding the set of neighbors for other points in the relevance collection, each neighbor collection has a related point in the relevance collection. Next, finding the "semantic-related clusters deduced from similarity" (see section 3.3) on each neighbor collection that has just been found, selecting the biggest size cluster. The centroid of this cluster is the candiadate for the revised query point.

3.3. **Finding the semantic-related clusters deduced from similarity.**

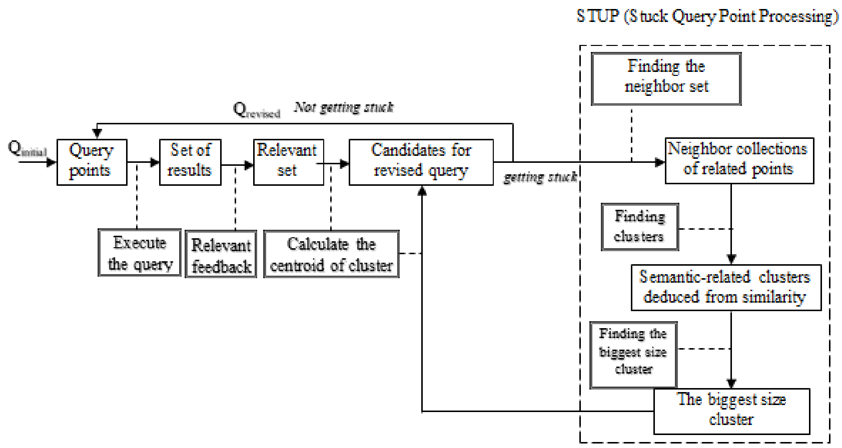

FIGURE 3. The proposed method's texture

Finding neighbor sets. To mathematically define the neighbors of a point, we first need to select a distance measure. For images, distance can be defined by a similar measure (a greater value implies a smaller distance), or by a similar measure(a smaller value implies a smaller distance). Because algebraic operators can convert a similar measure to a dissimilar one, with no loss of generality, we assume that the distance between two images is determined by a measure that is neither similar nor symmetrical called $d(i, j) = d(j, i) >= 0$. To make it simple about mathematical signs, we call d(i, j) is the distance between i and j. There are 2 methods, which can be applied to get neighbors of a query point, called the fixed radius method and the closest neighbor method. If a distance is a measure, both methods generate similar results under suitable parameters. However, for distances that do not have measures, neighbor collections, which are selected by two methods, are different and not dependent on parameters. The reason is the dissatisfying conditions in triangle inequality. Our method can be applied to all distances, which have or do not have measures so we use the nearest neighbor method. The following is a description of a method that is about selecting a set of neighbor images for a query image point i. Firstly, the nearest neighbor method selects k nearest neighbors of i to become seeds. Then, r nearest neighbors for each seed found. Finally, the nearest neighbors, which are selected, are all distinct neighbors among its seeds and neighbors, that is, different neighbors in k(r+1) neighbors. So the number of blocked neighbors is k (r + 1).

Clustering through partitioning. In the STUP block, we use a graphical representation of neighbors. The reason for this choice is that the graphical representation emphasizes the pair relationship which helps find the ―semantic-related clusters deduced from similarity‖.

A collection has n neighbors described by an undirected graph which has weight G=(V,E): weight $G = (V, E) : V = \{1, 2, \ldots, n\}$ represents nodes, edges $E = (i, j) : i, j \in V$, which are created between 2 nodes and a non-negative weight $w_{i,j}$ of an edge (i, j), show the similarity between 2 nodes is a distance function. For a distance d(i, j) which is given between i and j. The weights can be organized to be a matrix W, named affinity.

By a graphical representation, clustering can be expressed as a graph partition problem. Here, we use the method NCut [42] for a cluster of points, the reason is that the NCut method is relatively powerful in generating balanced clusters [49] and has an approximate solution with acceptable computational complexity.

The NCut method attempts to organize nodes (points) into groups so the similarity is high in the same group, and/or the similarity between the groups is low. A graph G=(V, E) with the affinity matrix W, a simple way to estimate the cost of partition nodes into two discrete collections A and B ($A \cap B = \emptyset$ and $A \cup B = V$) is the sum of weights of edges that connect 2 collections. In graph theory, this cost is called a cut, a cut is assumed the similarity measure between clusters and is determined as follows:

$NCut = \frac{Cut(A,B)}{Assoc(A,V)} + \frac{Cut(A,B)}{Assoc(B,V)}$ (**)

With $Cut(A,B) = \sum_{i\in A, j\in B} w(i,j)$ and $Assoc(A,V) = \sum_{u\in A, t\in V} w(u,t)$

An unbalanced cut will result in a large value NCut. Finding a partition that bisects with the minimum value NCut is a NP-complete problem. This problem has an approximate solution by solving general eigenvalues problem [42] as follows:

Dividing a partition of nodes of graph V into 2 collections A and B, x is a vector and contains $|V|$ members with i^{th} member $x_i = 1$ if $x_i \in A$, $x_i = -1$ if reverse. $d(i) = \sum_j w(i,j)$ is the sum of weights between I and the others nodes. The fomula (**) is rewritten as follows:

$NCut = \frac{Cut(A,B)}{Assoc(A,V)} + \frac{Cut(A,B)}{Assoc(B,V)} = \frac{\sum_{x_i>0,x_j<0} -w_{ij}x_ix_j}{\sum_{x_i>0} d_i} + \frac{\sum_{x_i<0,x_j>0} -w_{ij}x_ix_j}{\sum_{x_i<0} d_i}$

With W is a matrix Affinity $n \times n$, $D = diag[s_1, s_2, \ldots, s_n]$ is a diagonalizable matrix with $s_i = \sum_{j=1,\ldots,n} w_{ij}$, D-W is the Laplace matrix, 1 is vector $N \times 1$, $k = \frac{\sum_{x_i>0} d_j}{\sum_i d_i}$. So $4[NCut(x)]$ is rewritten as follows:

$4[NCut(x)] = \frac{(1+x)^T(D-W)(1+x)}{k.1^T.1} + \frac{(1-x)^T(D-W)(1-x)}{(1-k).1^T D.1} = \frac{[x^T(D-W)x+1^T(D-W)1]}{k(1-k)^T D.1} + \frac{2(1-2k)1^T(D-W)x}{k(1-k)1^T D.1}$

Set: $\alpha(x) = x^T(D-W)x; \beta(x) = 1^T(D-W)x; \gamma = 1^T(D-W)1; M = 1^T D1$, we have :

$[4[NCut(x)] = \frac{(\alpha(x)+\gamma+2(1-2k)\beta(x)}{k(1-k)M}$

$= \frac{\alpha(x)+\gamma+2(1-2k)\beta(x)}{k(1-k)M} - \frac{2(\alpha(x)+\gamma)}{M} + \frac{2\alpha(x)}{M} + \frac{2\gamma}{M}$

$= \frac{\alpha(x)+\gamma+2(1-2k)\beta(x)-2k(1-k)(\alpha(x)+\gamma)}{k(1-k)M} + \frac{2\alpha(x)}{M} + \frac{2\gamma}{M}$

$= \frac{(1-2k+2k^2)(\alpha(x)+\gamma)+2(1-2k)\beta(x)}{k(1-k)M} + \frac{2\alpha(x)}{M} + \frac{2\gamma}{M}$

$= \frac{\frac{(1-2k+2k^2)}{(1-k)^2}(\alpha(x)+\gamma)+\frac{2(1-2k)}{(1-k)^2}\beta(x)}{\frac{k}{1-k}M} + \frac{2\alpha(x)}{M} + \frac{2\gamma}{M}$

$= \frac{\frac{(1-2k+2k^2)}{(1-k)^2}(\alpha(x)+\gamma)}{\frac{k}{(1-k)}M} + \frac{\frac{2(1-2k+k^2-k^2)}{(1-k)^2}\beta(x)}{\frac{k}{(1-k)}M} + \frac{2\alpha(x)}{M} + \frac{2\gamma}{M}$

Set $b = \frac{k}{1-k}; \gamma = 0$, we have: $\frac{(1+b^2)(\gamma(x)+y)}{b.M} + \frac{2(1-b^2)\beta(x)}{b.M} + \frac{2b\alpha(x)}{b.M} - \frac{2b\gamma}{b.M}$ (***) Replace α, β, γ into (***) we have:

$4[NCut(x)] = \frac{(1+b^2)(x^T(D-W)x+1^T(D-W)1)}{b.1^T.D.1} + \frac{2(1-b^2)1^T(D-W)x}{b.1^T.D.1} + \frac{2.b.x^T(D-W)x}{b.1^T.D.1} - \frac{2.b.1^T(D-W).1}{b.1^T.D.1}$

$= \frac{(1+x)^T(D-W)(1+x)+b^2(1-x)^T(D-W)(1-x)-2b(1-x)^T(D-W)(1+x)}{b.1^T.D.1}$

$= \frac{[(1+x)-b(1-x)]^T(D-W)[(1+x)-b(1-x)]}{b.1^T.D.1}$

Set y = (1+x) – b(1-x) whereas $y^T D.1 = \sum_{x_i>0} d_i - b\sum_{x_i} d_i = 0$ so we have: $y^T Dy = \sum_{x_i>0} d_i + b^2 d_i = b\sum_{x_i<0} d_i + b^2 \sum_{x_i<0} d_i$

$= b(\sum_{x_i<0} d_i + b\sum_{x_i<0} d_i)$

$= 4b1^T D1$

At this point, $NCut(x)$ is rewritten as follows: $NCut(x) = \frac{y^T(D-W)y}{y^T Dy}$

$min_{\mathrm{x}} NCut(x) = min_{\mathrm{y}} \frac{y^T(D-W)y}{y^T Dy}$ with $y_i \in [1; -b); y^T D1 = 0$

Applying Rayleigh quotient [17] in special case. Finding $min_x NCut(x)$ becomes the job of finding λ_{min} of the follow equation :

$(D-W)\overrightarrow{\mathrm{y}} = \lambda D\overrightarrow{\mathrm{y}}$ (****)

The second general eigenvector corresponds to the second smallest general eigenvalue of (****) used to partition the graph. The NCut method can be applied recursion with the goal of receiving more than 2 clusters. Here, the subgraph with the maximum number of nodes is recursively partitioned. The process ends when the preset number of clusters has been reached or the NCut value exceeds a certain threshold.

Determining the semantic-related clusters deduced from similarity:

We define a "semantic-related cluster deduced from similarity" as the cluster has "many" members in semantic-related clusters deduced from similarity. The definition of "Many" here depends on which algorithm determines the semantic-related cluster deduced from similarity [11]. Their method makes full use of the information about similarity between neighbors (the similarity between each neighbor and related points, and the similarity between neighbors and each other). The input requirements for the algorithm include a related point in the set of related points, and the neighbor collection of related points (covered in the previous section), the number of clusters($M >= 2$) and a threshold for the value $NCut(0 \leq T \leq 1)$ are requested in the recursive process NCut. Recursively apply the NCut algorithm for larger graph or subgraph until the number of clusters is equal to M or the value of NCut is greater than T. The result of this process is a hierarchical tree, each leaf of the tree corresponds to a cluster, and the cluster corresponding to the leftmost leaf node of the tree is "semantic-related cluster deduced from similarity".

3.4. **Algorithm.** We propose the image retrieval method that can handle stuck query points in relevance feedback. The retrieval algorithm, which handles stuck query points called STUP (Stuck Query Point Processing), is presented as follows:

Stuck Query Point Processing Algorithm

```
Input:
        Set of images: S
        Initial query Images: Q_initial
        Number of images returned at each iteration: m
Output:
      Set of results of optimal query: Result(Query_opt)

1.iter←1;
2. Result(Q_initial) ←<1, Q_initial, W, D, S, m>;
3. Relevant(Q_initial,n)←Feedback(Result(Q_initial));
4. Candidate_Query_revised←Centroid(Relevant(Q_initial,n));
5. repeat
  5.1 If (iter≠1)
      Candidate_Query_revised←Centroid(Relevant(Query_revised,n));
  5.2 loop do
    5.2.1 If (Candidate_Query_revised are not stuck)
          Query_revised←Candidate_Query_revised;
          Stop;
    5.2.2 For (each i∈Relevant(Candidate_Query_revised,n)) do
          NN_i ←Nearest_Neighbtbor(i,r);
          src←Find_Similarity_Induced_Semantic_Relevant_Cluster(NN_i);
            AddQueue(src,L);
            Sort(L); // sort L in ascending order of cluster size
    5.2.3 c←Remove(L);
    5.2.4 Candidate_Query_revised←Centroid(c);
  5.3 Result(Query_revised) ←<1, Query_revised, W, D, S, m>;
  5.4 Relevant(Query_revised,n)←Feedback(Result(Q_revised));
  until (User stops responding);
6. Result(Query_opt) ←<1, Query_revised, W, D, S, m>;
7. Return Result(Query_opt);
```

FIGURE 4. Retrieval algorithm handles stuck query points with relevance feedback.

The retrieval algorithm handles stuck query points with relevance feedback STUP on figure 4 works as follows:

Each image in a set of images S is presented as a point in the multi-dimensional feature space. When a user input an initial query $Q_{initial}$ on the query interface by template, the algorithm uses the same procedure as the data base image to make image $Q_{initial}$ become a point in the multi-dimensional feature space. The initial query $< 1, Q_{initial}, W, D, S, m >$ is implemented (step 2), here the number 1 represents a query point in the space S, W is a set of weights related to $Q_{initial}$, D is a distance function and k is the number of points which are retrieved in the space S of each iteration. The results after implementing the initial query is assigned to Result($Q_{initial}$). On the set of Result($Q_{initial}$) which is returned by the initial query, users respond on a graphical interface by the function Feedback(Result($Q_{initial}$)) to get the set of n related points and store in Relevant($Q_{initial}$, n) (step 3). The centroid of related points Relevant(,n) is calculated by the function (Relevant($Q_{initial}$, n)) and assigns to the revised query candidate Candidate $\_Query_{revised}$ (step 4). For the first iteration (step 5), inter=1, in case that the revised query candidate Candidate$_Query_{revised}$ is not stuck (step 5.2.1), Candidate$_Query_{revised}$ is chosen to become revised query and exists in the do loop (step 5.2) via the stop command. In reverse of step 5.2.2, for each i that belongs to the relevance set of revised queries Relevant(Candidate$_Query_{revised}$,n)), calculate the neighbor r of i via the function Nearest_Neightbor(i,r) and assign to NNi . On the neighbor collection NNi , finding similarity induced sematic relevant image cluster via the function $Find\_Similarity\_Induced\_Semantic\_Relevant\_Image\_Cluster(NNi)$. After that, adding semantic-related clusters deduced from similarity into the queue L and sorting this queue in descending order of cluster size. Then, selecting the first cluster in the queue L to assign to c and calculating the cluster c‘s centroid to assign to a revised query candidate Candidate$_Query_{revised}$, and back to the beginning of the loop do (step 5.2). Next, implementing revised query $< 1, Query_{revised}, W, D, S, m >$ (step 5.3) to get the set of results Result($Query_{revised}$). Users repond on this set of results Result($Query_{revised}$) via the function Feedback(Result($Query_{revised}$) to attain a set of related points and then storing them in Relevant($Query_{revised}$,n)(step 5.4). In case that the user requirements are met, the result of the query implementing $< 1, Query_{revised}, W, D, S, m >$ (step 6) is the result of the optimal query Result($Query_{opt}$), and this problem‘s output is the returned result. In reverse, the centroid of Relevant ($Query_{revised}$,n), which we had in step 5.4, will be calculated via the function Centroid Relevant ($Query_{revised}$,n)) to assign value to the revised query candidate Candidate $Query_{revised}$ (step 5.1)

4. **Experimental results.** In this section, some empirical results are performed to show the performance of the proposed method. We evaluate the performance of our proposed method with relevant feedback on two benchmark datasets, SIMPLIcity [47], and INRIA Holidays image dataset [23]. The reason that we use the INRIA Holidays image dataset [23] for the evaluation is because some of the state of the art methods are based on the deep features [**?**, 1] were implemented on this dataset.

4.1. Image database and representation of images.

INRIA Holidays image dataset. To illustrate the performance of the proposed method compared to some of the state of the art methods, we experimented on the INRIA Holidays image dataset. This set is a set of high-resolution images and contains some personal holiday images. They include a variety of landscape images such as natural, man-made, water and fire effects, etc. This set also contains 500 image groups, each representing a scene or a separate object. The first image of each group is the query image and there are several images related to the query image among the remaining images. The size of each image in the same subject is usually the same, but the size of the images in different topics is very different.

Figure 5. Some samples in the Image Dataset COREL.

Image Dataset SIMPLIcity. To demonstrate the performance of the proposed method, in addition to experimenting on Image Dataset INRIA, we also conducted experiments on Dataset SIMPLIcity. This is a small data set with a thousand images and 10 categories. Each image in this set is 256 * 384 or 384 * 256. We represent each image by two features, that is, color and edge features. The color feature is represented by the color structure descriptors with a 128-dimensional vector, while the edge feature is the edge histogram descriptors with the 150-dimensional vector. A vector of 278 dimensions, composed of two color and edge features, represents an image. The precision of the Baseline method is calculated based on the Euclidean distance between the 278- dimensional feature vector of the query image and the images in the database.

Figure 6. Some samples in the Image Dataset SIMPLIcity.

4.2. **Automatic feedback scheme and evaluation metric.** To evaluate the effectiveness of the image retrieval methods, we use the precisionscope curve and precision rate [21]. The scope is determined by the top images that are presented to the user. The ratio of the number of relevant images, which are presented to the user, to images is the precision. The precisionscope curve provides an overall performance evaluation of the methods since it describes the precision with many scopes, while the precision at a specific scope value is emphasized by the precision rate.

We use fourfold cross-validation to evaluate the methods. The reason is that in an actual image retrieval system, a query image is often not present in the image database. With this evaluation way, the image database is divided into four subsets of equal size 21 and each of a subset has 25 images. At each retrieval, three subsets are used as the image database, and the other one is selected as the query image set. The average of the results from the fourfold cross-validation is calculated to determine the precisionscope curve and precision rate.

4.3. **Evaluate performance on traditional feature.** We compared our proposed image retrieval method **(STUP)** with three methods, including **CLUE** [11], **SRIR** [45], **DLRPIR** (Discriminative Low-Rank Projection for Image Retrieval) [25]. The objective of this comparison is to demonstrate the precision of STUP. In the real world, users usually only provide feedback samples for the first few loops even for the first one or two. Therefore, the retrieval performance of the first two feedback loops is the most important. In this experiment, the values for the parameters **50, 100, and 150 scopes on the SIMPLIcity dataset .**

TABLE 1. Comparison of average precision of methods in the 50, 100, and 150 scopes on the SIMPLIcity dataset.

Method	Average precision by scopes		
	50	**100**	**150**
CLUE	0.2887	0.3065	0.3199
SRIR	0.3135	0.42658	0.4846
DLRPIR	0.3224	0.47658	0.5125
STUP	**0.5836**	**0.6065**	**0.799**

Table 1 shows the average execution time of the different methods. The average query execution time for these four methods is very fast (less than 0.05 s for the first feedback iteration). The execution time of our STUP method is faster than CLUE, SRIR, **DLRPIR**. Computer configuration for the experiment is 3.1 GHz Dual-Core Intel Core i5 Catalina machine with 8 GB 2133 MHz LPDDR3 memory.

4.4. **Evaluate performance with image retrieval method on deep feature.** In this section, we will compare the performance of our proposed method with two state-of-the-art methods of image retrieval on deep features, named PAC (Pooling Approach based on Co-occurrence) [15] and CBIRCCC (CBIR based on a Constrained-Clustering Concept) [1]. Experiments in this section are done on INRIA Holidays image dataset. Based on the correlation between feature maps that contain useful information, PAC adds co-occurrence information to the image descriptor using a new representation of deep co-accurrence to improve accuracy in image search.The image descriptor, in PAC, uses a new 512-dimensional representation from deep cooccurrence. CBIRCCC [1] incorporates some deep and hand-crafted features based on the probability distribution of a membership score of a given cluster in an unsupervised manner. BIRCCC utilizes the pairwise similarities between the nearest neighbors, so the model can easily avoid including false-positive results in the final result set. Image descriptor, in CBIRCCC, includes 4 features that are OLDFP, BOW, HSV, and NetVLAD. A feature vector represents the image with 512 dimensions. It is important to note that, in the empirical evaluation of this section, the PAC and CBIRCCC methods were performed on a deep feature but did not use user's feedback (see details of PAC in [15] and CBIRCCC in [1]. Meanwhile, our proposed method SCDPIR uses user's feedback on a traditional feature set of 278 dimensions. The precision of our proposed method will increase if we use deep features like in [15, 1], however, we want to demonstrate the effectiveness of the proposed image retrieval method without should be based on good features. The feature set in our method consists of two features: color and edge. The color feature is a description of the 128- dimensional color structure and the edge feature is a 150-dimensional edge schema description. Our proposed SCDPIR method uses a feedback loop and 6 dimensions for subspace. In addition, the values for the parameters k and α are set to 12 and 50, respectively.

The average precision-scope curves of the PAC, CBIRCCC and SCDPIR methods are shown in Figure 7. In this experiment, precision in scopes 1, 2, 3, 4 and 5 is specified. The reason we chose scopes 1, 2, 3, 4, and 5 for experimentation here is because in the Holidays set, the relevant images of a query image are usually very small, ranging from 1 to 4 images. Here, the precision at each scope is the average precision of the 500 queries for that scope. According to the results in Figure 7, the precision of CBIRCCC method is the highest. Also on this figure, the precision of our method and PAC is similar at scopes 1, and 2, however, at the 3, 4 and 5 scopes the precision of our proposed method is higher than that of PAC. The reason that CBIRCCC has the highest precision is because it enhances the impact of the best features for a particular query image when incorporating some hand-crafted and deep features. Although our proposed method does not use the deep feature, it still produces better results than the PAC method because the subspace that is learned in our method is good.

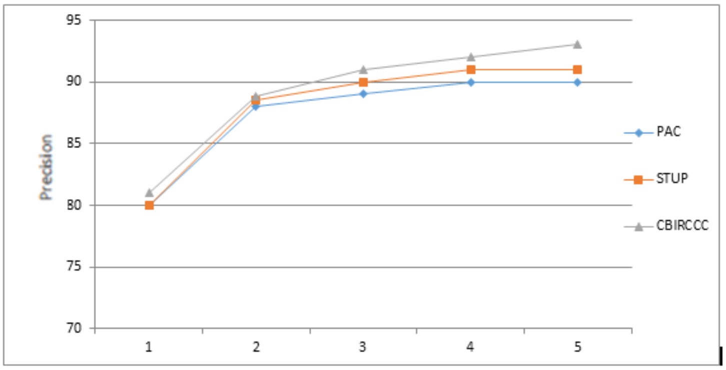

FIGURE 7. Average precision-scope curves of various methods (PAC, CBIRCCC, and STUP) on INRIA Holidays image dataset.

5. **Conclusions and future work.** We concentrated on proposing the method named STUP, solving 2 main problems which are: (1) finding sematic-related images spread in the entire feature space and (2) handling stuck query points in multi-point query while do not need to increase user burden. The solution for the first problem is that we use the multi-point query approach and for the second one, we do not require users to participate in our process, among the points in the previous reponse iteration, we determined the best point via the semantic-related cluster deduced from similarity.

Our experimental results on the feature databases, which have two benchmark datasets, shows that the proposed method STUP provide a higher precision compared with the methods using multi-point approach.

Acknowledgment. The authors would like to thank the anonymous reviewers and the Associate Editor for their comments, which led to improvements of this paper.

REFERENCES

[1] Leulseged Tesfaye Alemu and Marcello Pelillo. Multi-feature fusion for image retrieval using constrained dominant sets. *Image and Vision Computing*, 94:103862, 2020.

[2] Ahmad Alzu'bi, Abbes Amira, and Naeem Ramzan. Semantic content-based image retrieval: A comprehensive study. *Journal of Visual Communication and Image Representation*, 32:20–54, 2015.

[3] Barbara André, Tom Vercauteren, Anna M Buchner, Michael B Wallace, and Nicholas Ayache. Learning semantic and visual similarity for endomicroscopy video retrieval. *IEEE Transactions on Medical Imaging*, 31(6):1276–1288, 2012.

[4] TV Ashwin, Rahul Gupta, and Sugata Ghosal. Adaptable similarity search using non-relevant information. In *VLDB'02: Proceedings of the 28th International Conference on Very Large Databases*, pages 47–58. Elsevier, 2002.

[5] Ilaria Bartolini, Paolo Ciaccia, and Florian Waas. Feedbackbypass: A new approach to interactive similarity query processing. In *VLDB*, pages 201–210, 2001.

[6] Ana B Benitez, Mandis Beigi, and Shih-Fu Chang. A content-based image meta-search engine using relevance feedback. *IEEE Internet Computing*, 2(4):59–69, 1998.

[7] Roberto Brunelli and Ornella Mich. Image retrieval by examples. *IEEE Transactions on Multimedia*, 2(3):164–171, 2000.

[8] Chad Carson, Serge Belongie, Hayit Greenspan, and Jitendra Malik. Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Transactions on pattern analysis and machine intelligence*, 24(8):1026–1038, 2002.

[9] Kaushik Chakrabarti, Michael Ortega-Binderberger, Sharad Mehrotra, and Kriengkrai Porkaew. Evaluating refined queries in top-k retrieval systems. *IEEE Transactions on Knowledge and Data Engineering*, 16(2):256–270, 2004.

[10] Lei Chen, M Tamer Özsu, and Vincent Oria. Mindex: An efficient index structure for salient-object-based queries in video databases. *Multimedia Systems*, 10(1):56–71, 2004.

[11] Yixin Chen, James Ze Wang, and Robert Krovetz. Clue: Cluster-based retrieval of images by unsupervised learning. *IEEE transactions on Image Processing*, 14(8):1187–1201, 2005.

[12] Ingemar J Cox, Matt L Miller, Thomas P Minka, Thomas V Papathomas, and Peter N Yianilos. The bayesian image retrieval system, pichunter: theory, implementation, and psychophysical experiments. *IEEE transactions on image processing*, 9(1):20–37, 2000.

[13] Robert A Davis, Zhongmiao Xiao, and Xiaojun Qi. Capturing semantic relationship among images in clusters for efficient content-based image retrieval. In *2012 19th IEEE International Conference on Image Processing*, pages 1953–1956. IEEE, 2012.

[14] Myron Flickner, Harpreet Sawhney, Wayne Niblack, Jonathan Ashley, Qian Huang, Byron Dom, Monika Gorkani, Jim Hafner, Denis Lee, Dragutin Petkovic, et al. Query by image and video content: The qbic system. *computer*, 28(9):23–32, 1995.

[15] Juan I Forcen, Miguel Pagola, Edurne Barrenechea, and Humberto Bustince. Co-occurrence of deep convolutional features for image search. *Image and Vision Computing*, 97:103909, 2020.

[16] Th Gevers and Arnold WM Smeulders. Content-based image retrieval: An overview. *Emerging topics in computer vision*, 2004.

[17] GH Golub and CF Van Loan. 1989, matrix computations, the johns hopkins university press, baltimore. 1989.

[18] Amarnath Gupta and Ramesh Jain. Visual information retrieval. *Communications of the ACM*, 40(5):70–79, 1997.

[19] Derrall Heath, David Norton, and Dan Ventura. Conveying semantics through visual metaphor. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(2):1–17, 2014.

[20] Kien A Hua, Ning Yu, and Danzhou Liu. Query decomposition: A multiple neighborhood approach to relevance feedback processing in content-based image retrieval. In *22nd International Conference on Data Engineering (ICDE'06)*, pages 84–84. IEEE, 2006.

[21] Dionysius P Huijsmans and Nicu Sebe. How to complete performance graphs in content-based image retrieval: Add generality and normalize scope. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(2):245–251, 2005.

[22] Yoshiharu Ishikawa, Ravishankar Subramanya, and Christos Faloutsos. Mindreader: Querying databases through multiple examples. 1998.

[23] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *European conference on computer vision*, pages 304–317. Springer, 2008.

[24] Deok-Hwan Kim and Chin-Wan Chung. Qcluster: relevance feedback using adaptive clustering for content-based image retrieval. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 599–610, 2003.

[25] Zhihui Lai, Jiaqi Bao, Heng Kong, Minghua Wan, and Guowei Yang. Discriminative low-rank projection for robust subspace learning. *International Journal of Machine Learning and Cybernetics*, pages 1–14, 2020.

[26] Ying Liu, Xin Chen, Chengcui Zhang, and Alan Sprague. Semantic clustering for region-based image retrieval. *Journal of Visual Communication and Image Representation*, 20(2):157–166, 2009.

[27] Ying Liu, Dengsheng Zhang, Guojun Lu, and Wei-Ying Ma. A survey of content-based image retrieval with high-level semantics. *Pattern recognition*, 40(1):262–282, 2007.

[28] WY Ma and BS Manjunath. Scene netra: a toolbox for navigating large image databases. In *International Conference on Image Processing*, volume 1, 1997.

[29] Aleksandra Mojsilovic and Bernice Rogowitz. Capturing image semantics with low-level descriptors. In *Proceedings 2001 International Conference on Image Processing (Cat. No. 01CH37205)*, volume 1, pages 18–21. IEEE, 2001.

[30] David Norton, Derrall Heath, and Dan Ventura. Annotating images with emotional adjectives using features that summarize local interest points. *IEEE Transactions on Affective Computing, Under Review*, 2016.

[31] Virginia E Ogle and Michael Stonebraker. Chabot: Retrieval from a relational database of images. *Computer*, 28(9):40–48, 1995.

[32] Michael Ortega-Binderberger and Sharad Mehrotra. Relevance feedback techniques in the mars image retrieval system. *Multimedia Systems*, 9(6):535–547, 2004.

[33] Alex Pentland, Rosalind W Picard, and Stan Sclaroff. Photobook: Content-based manipulation of image databases. *International journal of computer vision*, 18(3):233–254, 1996.

[34] Kriengkrai Porkaew and Kaushik Chakrabarti. Query refinement for multimedia similarity retrieval in mars. In *Proceedings of the seventh ACM international conference on Multimedia (Part 1)*, pages 235–238, 1999.
[35] Joseph Rocchio. Relevance feedback in information retrieval. *The Smart retrieval system-experiments in automatic document processing*, pages 313–323, 1971.
[36] Yong Rui and Thomas Huang. Optimizing learning in image retrieval. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 1, pages 236–243. IEEE, 2000.
[37] Yong Rui, Thomas S Huang, and Sharad Mehrotra. Content-based image retrieval with relevance feedback in mars. In *Proceedings of international conference on image processing*, volume 2, pages 815–818. IEEE, 1997.
[38] Yong Rui, Thomas S Huang, Michael Ortega, and Sharad Mehrotra. Relevance feedback: A power tool for interactive content-based image retrieval. *IEEE Transactions on circuits and systems for video technology*, 8(5):644–655, 1998.
[39] Gerard Salton. Automatic text processing. addison welsley. *Reading, Massachusetts*, 4, 1989.
[40] Ishwar K Sethi, Ioana L Coman, and Daniela Stan. Mining association rules between low-level image features and high-level concepts. In *Data mining and knowledge discovery: theory, tools, and technology III*, volume 4384, pages 279–290. International Society for Optics and Photonics, 2001.
[41] Heng Tao Shen, Beng Chin Ooi, and Xiaofang Zhou. Towards effective indexing for very large video sequence database. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 730–741, 2005.
[42] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.
[43] Arnold WM Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on pattern analysis and machine intelligence*, 22(12):1349–1380, 2000.
[44] John R Smith and Shih-Fu Chang. Visualseek: a fully automated content-based image query system. In *Proceedings of the fourth ACM international conference on Multimedia*, pages 87–98, 1997.
[45] Quynh Dao Thi Thuy, Quynh Nguyen Huu, Canh Phuong Van, and Tao Ngo Quoc. An efficient semantic–related image retrieval method. *Expert Systems with Applications*, 72:30–41, 2017.
[46] Khanh Vu, Kien A Hua, and Wallapak Tavanapong. Image retrieval based on regions of interest. *IEEE Transactions on knowledge and data engineering*, 15(4):1045–1049, 2003.
[47] James Ze Wang, Jia Li, and Gio Wiederhold. Simplicity: Semantics-sensitive integrated matching for picture libraries. *IEEE Transactions on pattern analysis and machine intelligence*, 23(9):947–963, 2001.
[48] Tao Wang, Yong Rui, and Shi-Min Hu. Optimal adaptive learning for image retrieval. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I. IEEE, 2001.
[49] Yair Weiss. Segmentation using eigenvectors: a unifying view. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 975–982. IEEE, 1999.
[50] Leejay Wu, Christos Faloutsos, Katia Sycara, and Terry R Payne. Falcon: Feedback adaptive loop for content-based retrieval. 2000.
[51] Yilin Yan, Mei-Ling Shyu, and Qiusha Zhu. Negative correlation discovery for big multimedia data semantic concept mining and retrieval. In *2016 IEEE Tenth International Conference on Semantic Computing (ICSC)*, pages 55–62. IEEE, 2016.
[52] Xiang Sean Zhou and Thomas S Huang. Cbir: from low-level features to high-level semantics. In *Image and Video Communications and Processing 2000*, volume 3974, pages 426–431. International Society for Optics and Photonics, 2000.
[53] Xiang Sean Zhou and Thomas S Huang. Comparing discriminating transformations and svm for learning during multimedia retrieval. In *Proceedings of the ninth ACM international conference on Multimedia*, pages 137–146, 2001.