Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 298524, 12 pages
http://dx.doi.org/10.1155/2014/298524

Research Article

Scene Consistency Verification Based on PatchNet

Jinjiang Li, Xiaoqing Guo, Zhen Hua, and Zhiyong An

School of Computer Science and Technology, Shandong Institute of Business and Technology, Yantai, Shandong 264005, China

Correspondence should be addressed to Jinjiang Li; lijinjiang@gmail.com

Received 23 April 2014; Revised 16 June 2014; Accepted 17 June 2014; Published 9 July 2014

Academic Editor: Liang Lin

In the real world, the object does not exist in isolation, and it always appears in a certain scene. Usually the object is fixed in a particular scene and even in special spatial location. In this paper, we propose a method for judging scene consistency effectively. Scene semantics and geometry relation play a key role. In this paper, we use PatchNet to deal with these high-level scene structures. We construct a consistent scene database, using semantic information of PatchNet to determine whether the scene is consistent. The effectiveness of the proposed algorithm is verified by a lot of experiments.

1. Introduction

Every object is in a particular scene, and the same object in different scenes will influence our perception. For example, we think that it is normal when an Antarctic penguin is in a world of ice and snow but not grassland (see Figure 1). With the development of computer network and multimedia, the technology of synthetic image is increasingly mature and is widely applied. As important media of modern information communication, digital synthetic image is developing in an unprecedented rate. How to effectively analyze, organize, and manage huge image data has been a research hotspot of multimedia technology. Among them, how to judge the consistency of image scene is a common problem in computer vision. Traditional manual classification and label management of the image have been difficult to meet the practical needs, since it will cost so much human resources and time resources. So how to employ the computer to automatically determine the scene consistency of an image becomes important.

Scene analysis [1] is one of the important research contents in image understanding, and it reflects the inclusion relation between scene and objects which has very strong cognitive structure. In some papers, they analyzed the target in the scene well to complete overall scene recognition, such as literature [2]. Researches of biology and psychology show that human visual perception will get global features of the scene firstly. We can finish scene classification without target analysis and then guide the image understanding according to the prior knowledge and local vision information. So the scene analysis provides the whole mechanism of prior knowledge for image understanding. A remarkable characteristic of human visual perception is that it can quickly grasp the expression of the meaning of a complex image; Potter [3] proved that observers can also identify the semantic category of each image by only observing a group of fast image flow through experiments. The visual and semantic information that is obtained by fast image observation (about 200 ms) is called image gist [4]. When taking pictures, the photographer always tries to put the target and features that can reflect the image gist or semantics in the center of the image. This habit makes the most similar targets for shooting have the same shooting angle in images, which means that these images have spatial similarity. For example, in many penguin images, the upper part is blue sky and under the sky is the snowcapped mountain; the penguins are standing in the snow or on the rock. That contains the context environment of the object that appears in the image.

As shown in Figure 1, obviously, we can find that the scene in (a) and (b) is very harmonious, but the scene in (c) is not consistent. But how does the computer judget? Lalonde and Efros [5] study the problem of understanding color compatibility using image composites as well as the natural images color statistics of a large dataset by looking at differences in color distribution in unrealistic and realistic images and then apply their findings to two problems which include recoloring image regions and classifying composite images. Different

(a) (b) (c)

Figure 1: Consistent and inconsistent scenes.

with literature [5], our approach has semantic information, and it is more accurate for scene identification. We first construct an image database which contains different kinds of objects (see Section 3.1). In order to simplify the algorithm, the images that we selected have clear object and are in simple background. Based on the human visual saliency features, we make saliency detection and segmentation (see Section 3.2) for the images in our database. Then we will construct the PatchNet structure for these images. For a given image, we get the salient object and its sketch (see Section 4.1). The sketch will be used for searching images that are with similar sketches in the image database, and then we will compare the contextual information (see Section 4.2) between them to identify its scene consistency.

2. Related Works

We need to do many works to determine whether the scene is consistent in an image, including saliency detection, image segmentation, and PatchNet constructing. So far, many people have done lots of research on related works mentioned above.

2.1. Visual Saliency Detection and Segmentation. In general, salient regions always keep to common criterions [6] as follows.

(1) *Local Difference.* From the local perspective, salient regions always have obvious color and brightness difference with surrounding areas.

(2) *Global Rarity.* From an overall point of view, characteristics of salient area are always with low frequency in the global scope. Instead, some characteristics of background region always appear frequently in the global scope region.

(3) *Clustering Center.* Salient region usually has a clustered distribution center. That is to say, salient region and objects in the image have obvious characteristics of clustering, rather than dispersed.

(4) *High-Level Semantics.* According to the experience of human observation, regions with some high-level features are always regions of interest, such as face recognition.

According to the above, many scholars have put forward variety of salient region detection algorithms and segmentation algorithms. Oliva and Torralba [7, 8] proposed the contextual guidance model, whose thought is that information processing has two parallel pathways: the global pathway and the local pathway. The formation of saliency map in the local pathway is mainly by extracting physical characteristics of the local scene, such as direction, brightness, and color. The highlighted area is the visual advantage region. A classical model—the Itti model [9, 10]—belongs to the local pathway. The model extracts and integrates some low-level visual features of the image. Then using two-layer neural network which is named winner-take-all, the attention points in the image are found according to decreasing sequence. It combines with multiscale image to get a saliency map and first proposed the visual attention model based on saliency for rapid analysis of the scene. The global pathway is mainly expressing the overall scene statistical information and task requirements, and it can activate the existing knowledge and experience, thereby guiding the attention.

Shi et al. [11] proposed a generic and fast computational framework called PISA (pixelwise image saliency aggregating). It is holistically complementary saliency cues based on color and structure contrasts with spatial priors. Han et al. [12] build a model for the texture, edges, and color of images by Markov random field framework and then grow saliency region by the seed value growth method in saliency map. Hou and Zhang [13] proposed a simple saliency detection method, which is independent of the object categories, object characteristics, or other forms of prior knowledge. It analyzed the logarithmic spectrum of input image, extracts spectral residual of image spectral domain, and puts forward a method to quickly establish the corresponding saliency map in spatial domain. Li et al. [14] proposed a visual saliency detection algorithm from the perspective of reconstruction errors based on the background templates. Zhu et al. [15] proposed a new method based on segmentation for saliency detection method based on multiscale super pixels. This method combined significant global and local clues. It also extracted super pixels in multiscale and displayed the normal distribution of each super pixel with its associated pixels in CIE-Lab spatial. After generating a full resolution and high quality saliency map based on significant comparison, Cheng et al. [16] employed a new iterative approach named GrabCut [17] to highlight the segmentation target. Mehrani and Veksler [18]

made use of data sets in which images have been labeled by people to study and detected saliency regions. The images which have been segmented by learning classifier will be optimized through a binary graph-cut method. This algorithm also used an iterative segmentation framework, since that the manual annotation of salient regions is of low efficiency and tedious. In order to solve the problem mentioned above, Fu et al. [19] employed a saliency cut method to segment the background and target automatically. Bagon et al. [20] proposed a good image segmentation method, "component segmentation," and it could make an image assemble itself with its own patches easily. The method produced high quality results, and it allowed patch conversion. However, looking for a good segmentation requires a complex, iterative optimization procedure. Therefore, we cannot guarantee that it can converge to a good solution. The full resolution saliency map can effectively retain the clear boundary and keep more original image frequency content than other methods. But if the most pixels in saliency regions or the background are very complex, the regions we get from detection may be background regions instead of saliency regions. In view of the characteristics mentioned above, Achanta and Süsstrunk [21] introduced a saliency region detection method. It overcame the disadvantages of full resolution methods and retained its advantages at the same time. This method utilizes the characteristics of brightness and color which is easy to be realized and has high calculation efficiency.

2.2. Patch Based Image Editor. Patch based method has been widely used in a variety of image and video editing tasks, such as image denoising, super resolution, image texture synthesis [22], and image stitching and image restoration [23]. Barnes et al. [23] proposed an algorithm that can quickly search for the nearest neighbor patch to match the missing part of the image. The algorithm is mainly relying on the natural correlation of images, and it allows us to find patches that match with the missing part from the image itself. Some patches which can match well with missing part can be found by random sampling. One of the advantages of patch sampling plan is that it can provide more precise control which has a great effect on image reconstruction, such as literature [24] and literature [25]. Cho et al. [25] proposed image "patch transformation" which divides the image into a number of misaligned small patches, and these patches can restructure the image. Hu et al. [26] proposed an image editing method based on PatchNet. The method divides the image into different patches based on whether the patch is a part. The location of each patch can form a contextual map. We can get the similar patch based on the comparison of contextual map.

2.3. Sketch and Shape Matching. Shape matching has an important application in the research of computer vision, such as image retrieval and object recognition. Researchers have done a lot of related work and proposed kinds of methods about shape matching. Zhang et al. [27] proposed a robust face sketch generation algorithm which can compose a face sketch image with many faces that from different pictures. These pictures are from more than one training set and they are in different illumination with different poses. The algorithm applies multiscale Markov random field (MRF) model for synthetizing sketch patch of local face. Klare and Jain [28] proposed a feature based face shape matching method. The method used SIFT function descriptor features directly and calculated the similarity between pairwise face sketch and match images. A combination of the two ways mentioned above is also used. Cao et al. [29] proposed an index structure and its corresponding original contour matching algorithm. They calculated the similarity of sketch and natural images. The method considers the image storage cost, the retrieval accuracy, and retrieval efficiency.

3. Constructing Scene Database

In order to complete the image consistency matching, we established a database which has numbers of images, saliency maps of these images, and their PatchNets. For the creation of the database, we have three steps: (1) searching for various kinds of images from the Internet (such as "Flickr"); (2) saliency extraction and segmentation of images in the database; (3) building the PatchNet of images

3.1. Query by "Flickr". Firstly, we established an image library which contains 6000 images with different kinds of scenes and objects. We used about 60 words (such like "zebra," "penguin," "desert," and "surfing") to grab images from the "Flickr"; then we strictly screen these images in order to make it more in-line with our classification. Part of the image database can be seen in Figure 2. For these images, we will make saliency detection and segmentation further (see Section 3.2) then construct the PatchNet (see Section 3.3).

3.2. Salient Region Extraction and Cut. In Section 2, we introduced some methods about image saliency detection, including the content-based bottom-to-up method. Seeing that the salient region segmentation algorithm has become mature, we do not have our algorithm. After comparing different kinds of algorithms, we decided to use a saliency detection method that is based on the global contrast [30]. The result can be seen in Figure 3; the mask is the salient region of Figure 1(b). The algorithm extracted saliency region based on regional contrast and evaluated the global difference of the image and the space consistency at the same time. It generated full resolution image easily and efficiently. We will introduce the algorithm simply as follows.

First we use graph-based image segmentation method [31] to divide the image into several regions r_i, and we express the value of salient region by the following equation:

$$S(r_k) = \sum_{r_k \neq r_i} w(r_i) D_r(r_k, r_i), \quad (1)$$

where $w(r_i)$ refers to the weight of the region r_i, the value of the weight is the number of pixels in the region, and $D_r(\cdot, \cdot)$

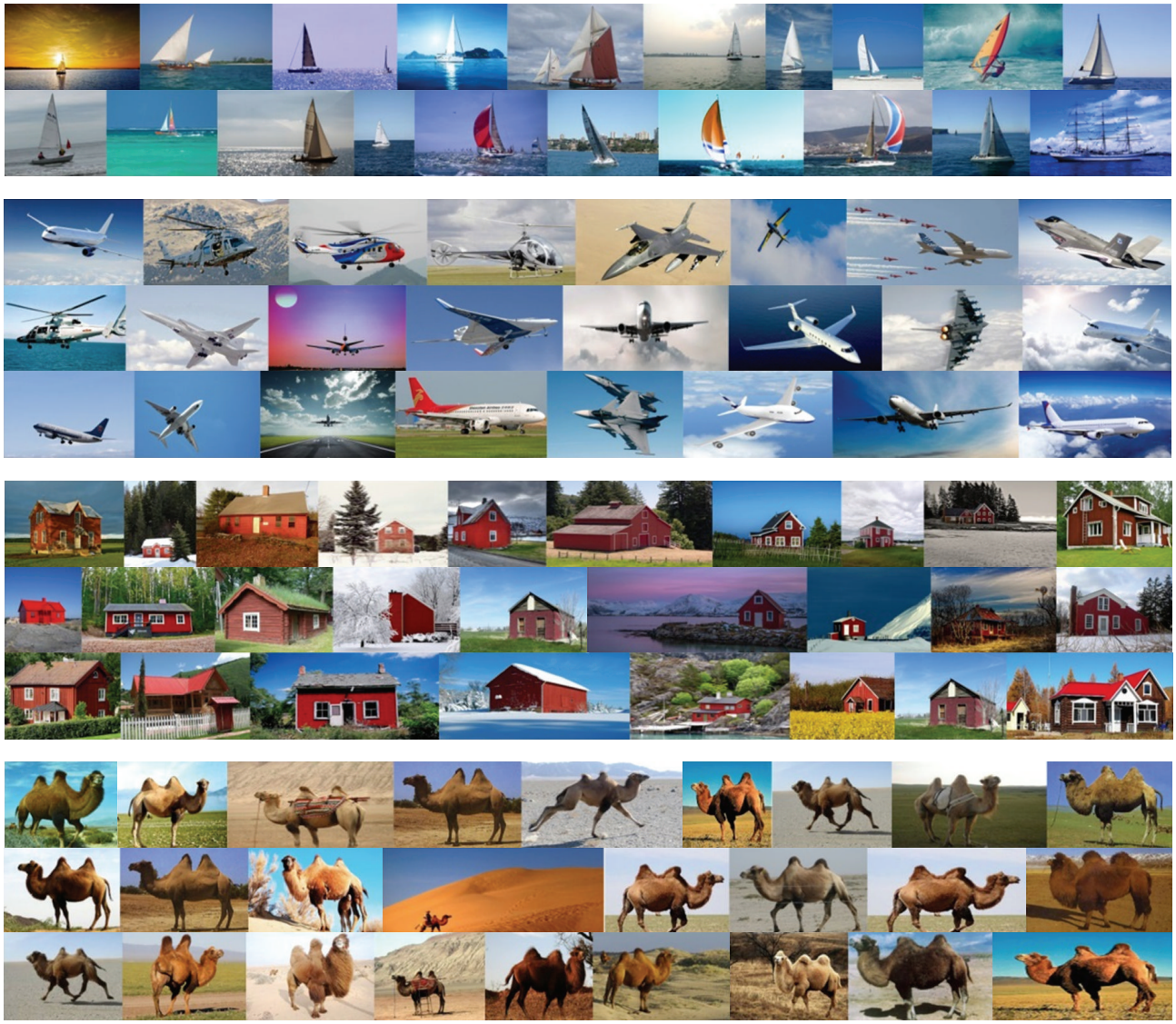


Figure 2: Scene dataset.

Figure 3: Salient region extraction.

refers to the color distance between two regions that are defined as follows:

$$D_r(r_1, r_2) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} f(c_{1,i}) f(c_{2,j}) D(c_{1,i}, c_{2,j}). \quad (2)$$

In order to increase the influence between regions that have close relationship and correspondingly reduce influence between nonclose regions, literature [30] introduces a spatial weight for formula (1), which is defined as follows:

$$S(r_k) = \sum_{r_k \neq r_i} \exp\left(-\frac{D_s(r_k, r_i)}{\sigma_S^2}\right) w(r_i) D_r(r_k, r_i), \quad (3)$$

where $D_s(r_k, r_i)$ refers to the spatial distance between two regions and σ_S affects the space weight. The bigger σ_S is, the farther region will do more contribution to the saliency of the current region. The Euclidean distance between two centroids of the two regions will be used to express the spatial distance. Here, we normalized pixel coordinates to [0, 1], and let $\sigma_S^2 = 0.4$.

For a color image $I := \{I_i\}$, I_i is the pixel in RGB color space. We want to segment image I which means that we need to determine the opacity value $\alpha := \{\alpha_i\}$ of each corresponding pixel. In order to realize unsupervised segmentation, we

(a) (b) (c)

Figure 4: Nodes and contextual map.

can construct a gauss mixture model (GMM) to distinguish the color distribution of foreground and background. This can avoid setting thresholds manually and extracting binary mask directly [32]. We use the GrabCut [17] to solve the problem of image segmentation. The optimization of the Gibbs energy function is shown as follows:

$$\min_{\alpha} E(\alpha, G, I) = \min_{\alpha} (U(\alpha, G, I) + V(\alpha, I)), \quad (4)$$

where $U(\alpha, G, I)$ evaluates the adapted value of opacity value α with respect to data I that under G. $V(\alpha, I)$ evaluates the smoothness of α. When $\alpha = 1$, it represents the foreground, and when $\alpha = 0$, it represents the background. More details can be gotten from literature [17].

3.3. Building a PatchNet. In order to make the image editing more time saving and more close to reality, Hu et al. [26] proposed a new interactive image editing method based on database. The method can effectively express the location relationship between the representative patch and the other patches in the image to realize the image editing. In this paper, we applied it to the scene consistency detection and obtained good results.

The construction of PatchNet consists of three main steps: (1) determining the representative patch; (2) determining the real nodes and its corresponding image region; (3) forming the contextual map.

3.3.1. Finding Representative Patches. Since patches of natural image are fundamental elements for visual pattern modeling and recognition, Lin et al. [33] proposed an approach for representing and recognizing objects with a massive number of local image patches. But the patches are something different from the representative patches we mentioned. In simple terms, a representative patch is a patch that can represent a coherent region of the image. For instance, in Figure 4(a), the blue sky is a representative patch. Each patch is denoted by P which is associated with the mask that is denoted by "m." In Figure 4(b), each region in different color can be seen as an "m" and each mask may overlap with each other and may contain disjoint parts. The algorithm in literature [26] using a pixelwise occupies map Q to mark pixels that are unoccupied by any masks.

The pixels of the image are all marked unoccupied in Q at first, and we will iteratively perform the following steps before all pixels are occupied.

(1) Choose the pixel location x from all pixels that are unoccupied in Q, while it has the minimal gradient magnitude, and then take it as the center of the patch P_x. Before handling more complex regions, we have to extract regions that have relatively uniform shade firstly.

(2) Use formula (5); partially adjust and reset the center of representative patch P_x:

$$x_{\text{new}} = \frac{\sum_{z \in P_x} g_z z}{\sum_{z \in P_x} g_z}, \quad (5)$$

where z is the pixel position of patch P_x, g_z is the gradient magnitude of pixel z, and x_{new} generally is the local gradient centroid.

(3) Seek out all the image patches which can be represented by P_x and indicated by mask m_x. For every pixel y, calculate $d_c(P_x, P_y)$ which represents the L_2 norm color difference between P_x and P_y. Supposing that $d_c(P_x, P_y) < \delta_{x,y}$, then merge P_y into m_x and mark y that has been occupied by Q. $\delta_{x,y}$ is defined as follows:

$$\delta_{x,y} = k\left(\frac{\overline{g}(x)}{C(x, y)}\right)^m, \quad (6)$$

where k is 2 and m is 0.5; $\overline{g}(x)$ refers to the average gradient value of patch P_x; $C(x, y)$ is the value of the average color difference between P_x and P_y; and the range is $[0, 255]$. Intuitively, the stronger the gradient of P_x is, the larger the threshold is (i.e., $\overline{g}(x)$ is large), so the growth limit in strong texture region is relatively not so strict. Meanwhile, if there is a relatively uniform color in P_x, the value of the threshold mainly depends on the color difference between P_x and P_y to avoid different color regions being merged.

3.3.2. Determining Real Nodes. A node can represent a region of the image and nodes can be divided into real nodes and compound nodes. A real node is indivisible and a compound node can be divided into real nodes or compound nodes. For instance, in Figure 5, yellow and blue areas are indivisible and they are real nodes; the region surrounded by red border is compound nodes. According to Finding Representative Patches, we can find a series of representative patches P_i and

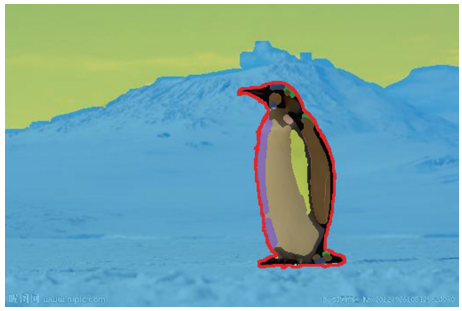


Figure 5: Real and compound node.

each with a mask m_i. Then do mask merging or something else to get real nods N_i^r and each of it has a single, nonoverlap image region Y_i correspondingly, and these can be represented well by P_i.

3.3.3. Graph Construction. Firstly, the real nodes in the top level are found. We can evaluate the visual dominance according to the size of the region; if the maximum region is larger than the threshold, we may put its represented patch P_i into the top of the image. Then the remaining nodes are divided into groups according to the spatial connectivity and the nodes in each group compose a compound node. If the biggest region is too small, then it cannot meet the standard of PatchNet and we give up this image. After finishing the steps above, we gradually expand compound node N_i^C at each level through observing all the nodes that belong to the region it represents. Among these nodes, those who have direct contact with brother nodes will be seen as child nodes. Others can form groups according to the spatial connectivity again, and each group correspondingly has compound child nodes. We process this procedure iteratively until no more compound nodes can be found in the deeper layer. After determining the hierarchical structure, we add an edge line between each of the paired nodes N_a and N_b since they are neighbor nodes in the same level and then calculate the 5∗5 contextual map $M(N_aN_b)$. In $M(N_aN_b)$, in order to compute the pixel value of (i, j), we can count the number of pixels that N_b contains and these pixels are on the positions with an offset of $(i-2, j-2)$ to all the pixels in N_a. At last, normalize the map into $[0, 255]$.

4. Scene Consistency Verification

For an image, we will complete the image scene consistency verification in two steps: (1) detecting and segmenting the salient region of the image and finding similar shape from the database by using sketch match method; (2) constructing the PatchNet of the image and getting the contextual map to do related matching work so that we can determine whether the image scene is consistent.

4.1. Sketch Based Retrieval. The work of this part is inspired by Hirata and Kato [34]. In order to complete the image retrieval, they let users use sketching method to match images in database and that is sketch based image retrieval (SBIR). Cheng et al. [32] provided a simple and effective method for SBIR which is realized by a cascade filter for image retrieval.

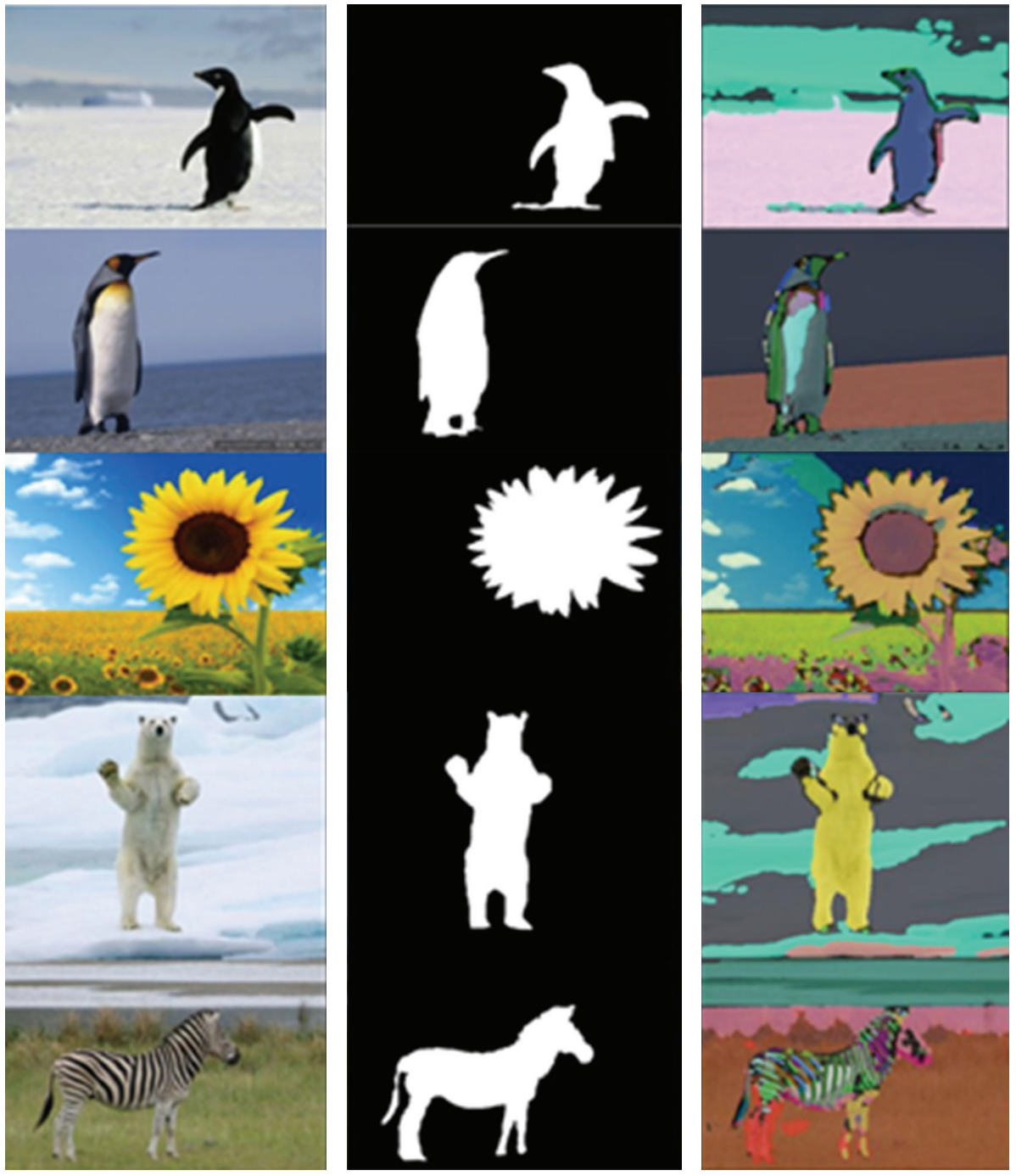
For a given image, we first use the algorithm in Section 3.2 to get the saliency map and its boundary; the user will input the sketch and then we use a cascade way to finish later works. For each metric, we sort the similar shapes in a descending way according to the user sketches and retain the ones with higher proportion as candidate image. For example, we may retain images that keep $T_C = 80\%$, $T_S = 80\%$, and $T_F = 70\%$ according to the circularity [35], the solidity [36], and the Fourier descriptor [37], respectively. The corresponding values of these descriptors are 1, 1, and 15. The Euclidean distance with corresponding features of the user sketch is used to compare these descriptors. At last, we will sort these images by shape context [38] and choose the 20 leading images.

4.2. Contextual Subgraph Matching. What we did in this part benefit from "contextual subgraph matching" which is proposed by Hu et al. [26]. We denoted the input image by "a" and we can obtain its PatchNet Ψ_a according to Section 3.2, and its salient region is Ω_a. We express this region as a new node N_a and the nodes will be denoted by $N_{a,i}^s$ $(i = 1, 2, \ldots, L)$ that are in the same level with N_a, where L is the total number of nodes that are at the same level with N_a, and we defined these nodes as the contextual environment of node N_a. In this paper, we will match "a" with images that retrieved from our database one by one before we get an image whose scene is consistent with "a" or the images we retrieved from database have been matched all. Denote the image of database by "b", its PatchNet by Ψ_b, the salient region which is similar to Ω_a is Ω_b, and its corresponding node by N_b. Also, we defined the nodes as $N_{b,j}^s$ $(j = 1, 2, \ldots, K, K$ may not be equal to $L)$.

Defining the distance $D(N_{a,i}^s, N_{b,j}^s)$ reasonably is critical to scene consistency verification and in the definition of the distance, there are two kinds of very important similarities: one is the appearance similarity between $P(N_{a,i}^s)$ and $P(N_{b,j}^s)$, it means that the two nodes have the same representative patches; the other is the location similarity between $(N_a, N_{a,i}^s)$ and $(N_b, N_{b,j}^s)$. If there may be some overlap between the contextual maps $M(N_a, N_{a,i}^s)$ and $M(N_b, N_{b,j}^s)$, which means that if N_a is something aligned with N_b spatially, then there may be similarities between $N_{a,i}^s$ and $N_{b,j}^s$. This is because different images have different structure, even though they may be under the same scene. For instance, in an image the sky may be at the upper left of a house, but in another image the sky is at the upper right of the house. For the two images, although they could match very well, strict similarity measure may give a low matching score. Avoiding this, a more flexible contextual overlap is proposed as follows:

$$O\left(N_{a,i}^s, N_{b,j}^s\right) = \sum \left(M\left(N_a, N_{a,i}^s\right) \cdot M\left(N_b, N_{b,j}^s\right)\right). \quad (7)$$

This is the two maps' dot product sum. When the two maps share common high probability regions, the overlap will be very high and this makes it more flexible to match those images whose structure is slightly different.



(a) (b) (c)

Figure 6: Saliency and PatchNet appearance map: original image (a), mask (b), and PatchNet appearance map (c).

Definition of the overall distance between the two nodes is as follows:

$$D\left(N_{a,i}^{s}, N_{b,j}^{s}\right) = \begin{cases} d_c\left(P\left(N_{a,i}^{s}\right), P\left(N_{b,j}^{s}\right)\right), & O\left(N,N\right) \geq T_o \\ \infty, & \text{else,} \end{cases} \tag{8}$$

where $d_c(P(N_{a,i}^{s}), P(N_{b,j}^{s}))$ is the difference of patch appearance, unless the contextual overlap between the two nodes is no bigger than the threshold value T_o (here it is set to 10), the two nodes are treated incompatible, and we set the distance to infinity, else their similarities are the distance of the appearance distance between patches. If there is an image that can make the appearance distance of two nodes less than the threshold (in this paper we set it to 900), then we think that the image scene is consistent, else not consistent.

Figure 7: Matching results.

Figure 8: Results of scene consistency verification.

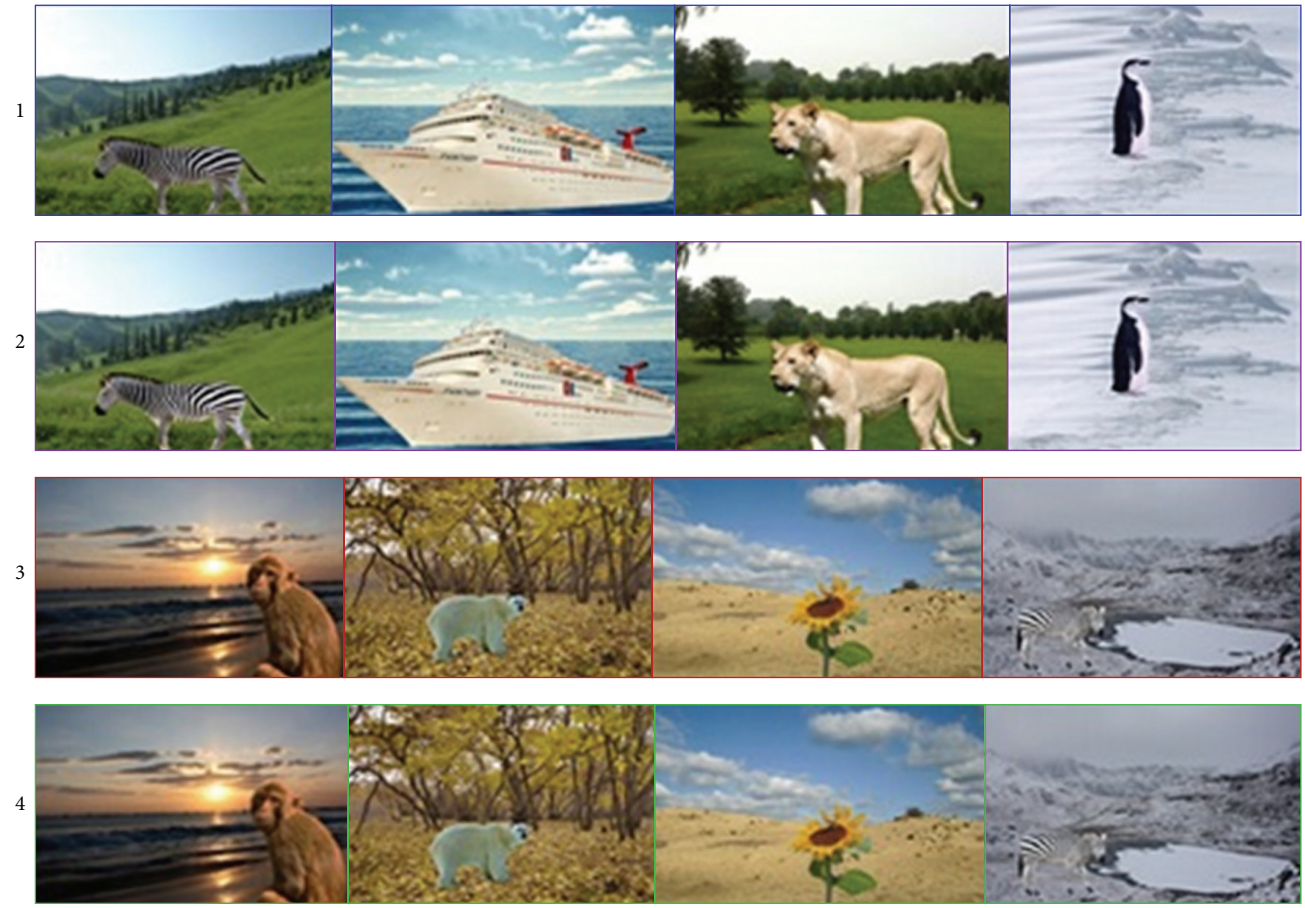


Figure 9: Results of scene consistency verification. Row 1 and row 3 are results of our experiments and row 2 and row 4 are results of literature [5]. The border color indicates the verification result for each image (blue and green: scene is consistent; red and purple: scene is inconsistent).

5. Experimental Results

In order to demonstrate the feasibility and effectiveness of the algorithm in this paper, we implement the experiment on a PC with an Intel Corei3 CPU and 4 GB RAM, running 64 bit Windows 7. For the simplicity of computation, the length and the width of images in our database are normalized to no more than 800 pixels. Now we will detail the experiment results as follows.

5.1. Results of Saliency Detection and Their Corresponding PatchNet Appearance Map. Since there are so many images in our database, we will only show part of the results. See Figure 6; the left column is the original image; the middle column is the mask of saliency map which also can be seen as the sketch of object; and the right column is the PatchNet appearance map. We can see the background and salient object from Figure 6.

5.2. Results of Sketch Matching. By using the method of sketch based image retrieval, we try to get the similar sketches from the database according to input mask and then get the images whose masks are similar to the input mask. In Figure 7, the first and the third rows with red border are our input image mask; the rest is the searching results. The second and the fourth rows are the original images that correspond to the masks of first and the third rows. We can see part of the matching results in Figure 7.

5.3. Results of Scene Consistency Verification. After obtaining the matching results, we will complete the contextual map and then judge whether the scene is consistent. Looking at Figure 8, the images with red borders are that scene is inconsistent, and the ones with blue borders are that scene is consistent. The other images are part of the ones that are from our database who have been matched with the input image. Experimental results demonstrate the feasibility and effectiveness of our algorithm.

In order to show the effectiveness of the method in our paper, we will compare the method in our paper with the methods of literature [5] (classifying composite images into realistic versus nonrealistic) and literature [39] (scene identification), respectively (just look at Figures 9 and 10). The method of literature [5] is just using the color compatibility for assessing image realism. The method of literature [39] proposed a discriminative measure to rank image patterns sampled from target scene classes for scene identification. The

Figure 10: Discriminative patterns and PatchNet map.

Table 1: The accuracy of our method and the method of literature [5].

	Penguin	Ship	Skating	Panda	Sunflower	Monkey
Inconsistent	5	4	5	6	4	5
Irreal but consistent	4	4	3	5	6	5
Recall of ours	0.92	0.88	0.91	0.88	0.96	0.92
Recall of [5]	0.71	0.67	0.74	0.68	0.73	0.72

advantage of our algorithm is that we use the contextual information of the image.

In Figure 9, the images with blue borders are consistent scene and the ones with red borders are inconsistent in our experiments. And the images with green borders are those whose scenes are consistent and the ones with purple borders are inconsistent in literature [5]. From Figure 9 we can find that some scenes are consistent (e.g., the images in row 2), but it may be judged as unrealistic images in literature [5]. Also, some scenes are obviously inconsistent (e.g., the images in row 4) but literature [5] judged them as realistic images since the color of the objects in those images is something like the background. Certainly we may get the wrong result (e.g., the images in row 1, but the scenes are really consistent).

In Figure 10, row 1 is input images; row 2 is the results of literature [39]; and row 3 is PatchNet appearance map. From Figure 10, we can see that the objects (pandas and dogs) have similar discriminative patterns (see row 2 in Figure 10) and will be judged as the same class. So naturally the scenes are identified the same as in literature [39]. Obviously pandas and dogs are of neither the same class nor the scenes. Seen from the PatchNet appearance map (row 3 in Figure 10), although the discriminative patterns of the pandas and the dogs are very similar, the semantic of representative patch is different.

To compare the accuracy of our method with literature [5] (Table 1), we selected 6 classes from our database, and each class has 20 images. Also we add some inconsistent images and some irreal but consistent images for each class. For class penguin, there are 24 consistent scene images and only 20 real images. The recall rate of the two methods is compared. Some irreal but consistent images are judged as inconsistent scene, so the recall rate for literature [5] is low.

6. Conclusions

In this paper, we propose a new method to judge the scene consistency. We construct a consistent scene database with

minimal manual intervention. By downloading, saliency detecting, segmentation, matting, and analyzing amounts of different scene images from the Internet, we use semantic information of PatchNet to determine whether or not the scene is consistent. PatchNet summarizes image appearance in terms of a small number of representative patches for image regions, linking them in a hierarchical graph model to describe image structure. Fast scene semantic matching can be achieved by graph matching. For existing scene image in our semantic-based scene database, it is accurate to judge scene consistency. For nonexistent scene image, using our pipeline, it is easy to construct a new class of scene image. We achieved a good result in a certain extent but still have a long way to go for wide applications. In the future we still have a lot to do, including the following.

(1) The algorithm of saliency detection needs to be improved in order to reduce the computing time consumption. A good saliency detection algorithm is beneficial to improve the accuracy of scene classification.

(2) In this paper we use static images as research object, but with the development of science and technology, composite video may be coming into people's daily life. For example, some films have a lot of postprocessing. If the scene is not consistent, it will bring unexpected visual to the audience. So the scene consistency detection applied to video field may be a significant research direction.

(3) The coverage of the database in this paper is still limited, and our experiment only proved the effectiveness of part images. So next, we need to enrich our database and expand the application.

Conflict of Interests

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

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant nos. 61173173, 61272430, and 61373079 and the Provincial Natural Science Foundation of Shandong under Grant no. ZR2013FM015.

References

[1] A. Vailaya, A. Jain, and H. Zhang, "On image classification: city vs. landscapes," *Pattern Recognition*, vol. 31, no. 12, pp. 1921–1935, 1988.

[2] S. T. Wang, S. M. Hu, and J. G. Sun, "Image retrieval based on color spatial feature," *Journal of Software*, vol. 13, no. 10, pp. 2031–2036, 2002.

[3] M. C. Potter, "Short-term conceptual memory for pictures," *Journal of Experimental Psychology: Human Learning and Memory*, vol. 2, no. 5, pp. 509–522, 1976.

[4] A. Oliva, "Chapter 41—Gist of the Scene," *Neurobiology of Attention*, pp. 251–256, 2005.

[5] J. F. Lalonde and A. A. Efros, "Using color compatibility for assessing image realism," in *Proceedings of the IEEE 11th International Conference on Computer Vision (ICCV '07)*, pp. 1–8, Rio de Janeiro, Brazil, October 2007.

[6] S. Goferman, Z. M. Lihi, and A. Tal, "Context-aware saliency detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 10, pp. 1915–1926, 2012.

[7] A. Oliva and A. Torralba, "The role of context in object recognition," *Trends in Cognitive Sciences*, vol. 11, no. 12, pp. 520–527, 2007.

[8] A. Torralba, A. Oliva, M. S. Castelhano, and J. M. Henderson, "Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search," *Psychological Review*, vol. 113, no. 4, pp. 766–786, 2006.

[9] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[10] L. Itti and C. Koch, "A saliency-based search mechanism for overt and covert shifts of visual attention," *Vision Research*, vol. 40, no. 10–12, pp. 1489–1506, 2000.

[11] K. Y. Shi, K. Z. Wang, J. B. Lu, and L. Lin, "PISA: pixelwise image saliency by aggregating complementary appearance contrast measures with spatial priors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2115–2122, Portland, Ore, USA, June 2013.

[12] J. Han, K. N. Ngan, M. Li, and H.-J. Zhang, "Unsupervised extraction of visual attention objects in color images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 1, pp. 141–145, 2006.

[13] X. D. Hou and L. Q. Zhang, "Saliency detection: a spectral residual approach," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '07)*, pp. 1–8, June 2007.

[14] X. H. Li, H. C. Lu, L. H. Zhang, X. Ruan, and M. H. Yang, "Saliency detection via dense and sparse reconstruction," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2976–2983, December 2013.

[15] L. Zhu, D. A. Klein, S. Frintrop, Z. G. Cao, and A. B. Cremers, "Multi-scale region-based saliency detection using W2 distance on N-dimensional normal distributions," in *Proceedings of the International Conference on Image Processing*, pp. 176–180, September 2013.

[16] M. M. Cheng, N. J. Mitra, X. L. Huang, P. H. S. Torr, and S. M. Hu, "Salient object detection and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 409–416, June 2011.

[17] C. Rother, V. Kolmogorov, and A. Blake, "GrabCut—interactive foreground extraction using iterated graph cuts," *ACM Transaction on Graphics*, vol. 23, no. 3, pp. 309–314, 2004.

[18] P. Mehrani and O. Veksler, "Saliency segmentation based on learning and graph cut refinement," in *Proceedings of the British Machine Vision Conference*, pp. 110.1–110.12, September 2010.

[19] Y. Fu, J. Cheng, Z. Li, and H. Lu, "Saliency Cuts: an automatic approach to object segmentation," in *Proceedings of the 19th International Conference on Pattern Recognition (ICPR '08)*, pp. 1–4, Tampa, Fla, USA, December 2008.

[20] S. Bagon, O. Boiman, and M. Irani, "What is a good image segment? A unified approach to segment extraction," in *Proceedings of the European Conference on Computer Vision*, pp. 30–44, October 2008.

[21] R. Achanta and S. Süsstrunk, "Saliency detection using maximum symmetric surround," in *Proceedings of the 17th IEEE International Conference on Image Processing (ICIP '10)*, pp. 2653–2656, September 2010.

[22] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*, pp. 341–346, Los Angeles, Calif, USA, 2001.

[23] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "PatchMatch: a randomized correspondence algorithm for structural image editing," *ACM Transactions on Graphics*, vol. 28, no. 3, article 24, 2009.

[24] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani, "Summarizing visual data using bidirectional similarity," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR' 08)*, June 2008.

[25] T. S. Cho, S. Avidan, and W. T. Freeman, "The patch transform and its applications to image editing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 8, pp. 1489–1500, 2010.

[26] S. M. Hu, F. L. Zhang, M. Wang, R. R. Martin, and J. Wang, "PatchNet: a patch-based image representation for interactive library-driven image editing," *ACM Transactions on Graphics*, vol. 32, no. 6, pp. 196–206, 2013.

[27] W. Zhang, X. G. Wang, and X. O. Tang, "Lighting and pose robust face sketch synthesis," in *Computer Vision–ECCV*, pp. 420–433, 2010.

[28] B. Klare and A. K. Jain, "Sketch to photo matching: a feature-based approach," in *Proceedings of the SPIE 7667 Biometric Technology for Human Identification VII*, Orlando, Fla, USA, April 2010.

[29] Y. Cao, C. H. Wang, L. Q. Zhang, and I. Zhang, "Edgel index for large-scale sketch-based image search," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 761–768, June 2011.

[30] M. M. Cheng, G. X. Zhang, N. J. Mitra, X. Huang, and S. Hu, "Global contrast based salient region detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 409–416, Providence, RI, USA, June 2011.

[31] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[32] M. M. Cheng, N. J. Mitra, X. Huang, and S. M. Hu, "SalientShape: group saliency in image collections," *The Visual Computer*, vol. 30, no. 4, pp. 443–453, 2014.

[33] L. Lin, P. Luo, X. W. Chen, and K. Zeng, "Representing and recognizing objects with massive local image patches," *Pattern Recognition*, vol. 45, no. 1, pp. 231–240, 2012.

[34] K. Hirata and T. Kato, "Query by visual example-content based image retrieval," in *Proceedings of the 3rd International Conference on Extending (EDBT '92)*, 1992, pp. 56–71.

[35] D. Zhang and G. Lu, "Review of shape representation and description techniques," *Pattern Recognition*, vol. 37, no. 1, pp. 1–19, 2004.

[36] M. Flickner, H. Sawhney, W. Niblack et al., "Query by image and video content: the QBIC system," *Computer*, vol. 28, no. 9, pp. 23–32, 1995.

[37] D. Zhang and G. Lu, "Shape-based image retrieval using generic Fourier descriptor," *Signal Processing: Image Communication*, vol. 17, no. 10, pp. 825–848, 2002.

[38] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 509–522, 2002.

[39] J. H. Lim, J. P. Chevallet, and S. Gao, "Scene identification using discriminative patterns," in *Proceedings of the 18th International Conference on Pattern Recognition (ICPR '06)*, vol. 2, pp. 642–645, Hong Kong, August 2006.