Online Submission ID: 0262

Sketch and Match: Scene Montage Using a Huge Image Collection

Yiming Liu[1,2*†] Dong Xu[1†] Jiaping Wang[2‡] Chi-Keung Tang[3§] Haoda Huang[2‡] Xin Tong[2‡] Baining Guo[2‡]
[1]Nanyang Technological University [2]Microsoft Research Asia [3]The Hong Kong University of Science and Technology

Abstract

We present a sketch-based approach to find matching source images for seamless image composition, by leveraging a large amount of image corpus collected from the Internet. Given a target image where the user draws a rough sketch to indicate the desired object fill-in, our system automatically searches a large image database, and returns a sparse set of matching images. These matching images contain salient regions semantically similar to the user-supplied sketch. Once the user has selected the preferred source region, it will be seamlessly pasted onto the target image where the sketch is drawn.

1 Introduction

At times we all wish to control how the scene would look before pressing the camera's shutter-release button. Some of us would like a rail track be seen in a remote village, or wish a landscape photo be filled with blossoming sunflowers to make the scene more lively. *Scene montage* provides us with a practical, photo-editing capability to inject our subjective expectation on the scene, allowing us to create a new photograph by filling in new objects on a target image.

Scene montage has two goals: First, the newly-inserted objects should match the user's intention, which can be very diverse. Second, the image composite should be seamless while preserving the semantical validity of the target photograph after filling in new objects. Roughly speaking, the output photograph should look natural.

Without considering image semantics, image montage can be regarded as a special application of image completion, also known as image inpainting or hole filling. Various image completion techniques were proposed to create a new image by filling in missing areas or erasing unwanted regions. The well-known image completion methods used example-based approaches [Efros and Leung 1999; Efros and Freeman 2001; Drori et al. 2003; Criminisi et al. 2003; Sun et al. 2005] to hallucinate missing contents by employing texture synthesis or related techniques. However, image completion is inherently an under-constrained problem: without any user's intention on the desired fill-in, a large number of completion results are equally plausible.

Observing that the space of differentiable scenes is in a manageable size, Hays and Efros [2007] proposed an image completion method by leveraging a huge image database containing about 2.3 million photographs collected from the photo sharing website *Flickr*. Given a target image to be completed, they first searched for semantically similar images from the database. The best matching image is blended with the target using Poisson-blending, along the seams which are given by graph-cuts. While good completion results were obtained, the user had no direct control on the content being filled in. A possible approach, which was demonstrated in [Johnson et al. 2006], is to provide an interface for the user to indicate the expected fill-in by using textual description. An image patch can also be used in their system in lieu of text for specifying the desired fill-in. While textual description can effectively limit the search space, a noun such as "fence" can produce many irrelevant fence images while none of them can meet the user's expected *appearance* of the fence he or she desire. Another example consisting of different types of clouds is shown in Figure 10. In such case, it is difficult for the user to indicate the search intention by submitting a text query as in [Johnson et al. 2006].

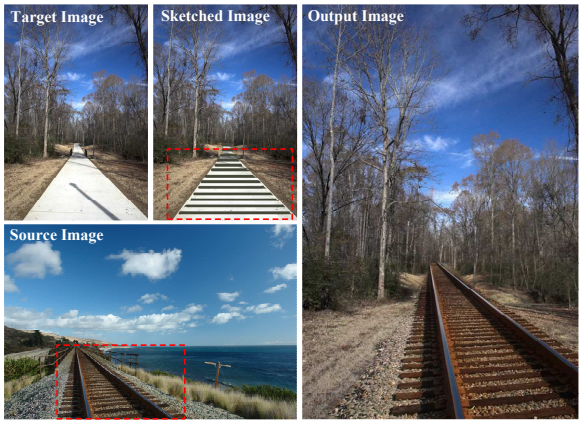


Figure 1: Given a target image where the user provides a rough sketch of the desired fill-in (e.g., a rail track), matching regions of the source images are automatically searched from a huge image collection. Note the correspondence between the sketch in the target and the rail track in the matching source image.

In this paper, we propose a "sketch-and-match" approach for finding matching images for seamless compositing: after the user has drawn on the target image a rough sketch of the object, the system will automatically return a sparse set of relevant source images that are immediately ready of compositing. As shown in the seminal work *Vision* [Marr 1982], a sketch plays a significant role in image understanding. For example, the descriptive power of a sketch was demonstrated in [Guo et al. 2003] where the *same* input can be faithfully re-synthesized using only a *primal sketch* which constitutes of simple 2D primitives such as points, lines, and junctions. Our system, in addition, allows the user to draw color strokes. Specifically, the user can draw a "disk" filled with color, rather of a primal sketch of a "circle," to better convey the desired fill-in. A clone brush can be also used in our system to copy regions into the sketched areas.

While novel in its sketch-and-match approach, our method shares with recent successful data-driven approaches which leverage a huge image database containing millions of images, such as image completion [Hays and Efros 2007] and other computer vision tasks (*e.g.*, object and scene recognition [Torralba et al. 2008], and image annotation [Wang et al. 2008]).

On the other hand, to efficiently search for the desired object fill-in from a huge image database, we propose a two-step coarse-to-fine approach in this paper: the first step retrieves *images* with regions locally similar to the target sketch; the second step re-ranks the matching *regions* of the top-ranked candidate images based on semantical validity. Specifically, local salient regions are detected from the sketched targeted image in the first step, where local descriptors are extracted. These descriptors are encoded into *visual*

*Yiming was a visiting student at Microsoft Research Asia.
†e-mail: {ymliu, dongxu}@ntu.edu.sg.
‡e-mail: {jiapw, hahuang, xtong, bainguo}@microsoft.com.
§e-mail: cktang@cs.ust.hk.

words [Sivic and Zisserman 2003], thereby allowing the problem to be translated into one of text search where efficient text-based retrieval method can be applied to 'mine' relevant source images from the huge image database. In this work, we use the inverted file method, which is able to filter out about 99% irrelevant source images that do not share any visual words in common with the target image. To rank the remaining relevant images, we propose an improved similarity measure for matching images based on visual words. To provide efficient feedback, in the second step, only the matching regions of the top 10,000 ranked candidate images are re-ranked by their semantical validity, which is measured using gist features [Oliva and Torralba 2006]. Finally, the user selects one image region from the top 40 re-ranked image regions, which will be seamlessly pasted onto the target image.

Our extensive experiments indicate that a sufficient number of semantically valid image regions are ranked in the top positions, which can be used to satisfy the user's intended fill-in. When compared with [Hays and Efros 2007], our system allows the user to have direct control over the search process. When compared with [Johnson et al. 2006], our system is novel in its use of a sketch to drive the search process.

Over the past decades a large number of Content Based Image Retrieval (CBIR) systems have been developed for retrieving images from image databases (see the two comprehensive surveys [Datta et al. 2008; Smeulders et al. 2000]). However, many CBIR systems used global features (e.g., color, texture, shape, etc.) which fail to retrieve images with local regions that are semantically valid, because the detailed properties of local sketched areas cannot be adequately represented by global features. Therefore, we argue that the existing CBIR algorithms cannot be directly deployed in our task. In contrast, the local feature based ranking method is employed in our system to rank the relevant images with regions locally similar to the target sketch, followed by the semantic re-ranking to obtain the semantically valid image regions.

Overall, the paper presents a novel sketch-based scene montage system. Our technical contributions are two-fold: 1) a novel two-step ranking approach for efficiently searching semantically valid source image regions; these regions can be used to fulfill user's requirement that can be expressed easily via sketching; 2) the novel adaptation of text-based retrieval techniques in scene montage.

2 Related Work

Seamless Image Composition Conventional object cut-and-paste methods [Pérez et al. 2003; Agarwala et al. 2004; Jia et al. 2006] in general consists of two steps: 1) objects of interest are extracted from source images either manually or by automatic algorithms; 2) the extracted objects are pasted onto the target image.

By solving Poisson equations, Poisson-blending [Pérez et al. 2003] seamlessly blends an image object with a target image. Unsightly color smearing will however be resulted if the user does not carefully specify a boundary enclosing the object. In interactive digital photomontage [Agarwala et al. 2004], different areas from multiple source images are combined to create a new target image. The user first draws a number of strokes. Then graph-cuts is applied to determine the seams between the combined regions. Poisson-blending is used to reduce the remaining visible artifacts. Drag-and-Drop pasting [Jia et al. 2006] uses dynamic programming to optimize the seam so that the user does not need to carefully specify the locus of the boundary curve. These methods have demonstrated success in real applications, given that the expected objects are present in the source images collection.

However, It is a non-trivial task for the user to provide a suitable source image that is consistent with the target image, where the expected objects with the desired appearance are present. At times, the user may even be unwilling to look for such suitable source images, especially in a huge image database such as the world wide web. Our proposed technique can be directly deployed as an orthogonal solution to complement existing object cut-and-paste methods, by automatically searching for the desired source images to make seamless image editing an easier task.

Image Completion Without considering the user's intention, various image completion techniques were proposed [Efros and Leung 1999; Efros and Freeman 2001; Drori et al. 2003; Criminisi et al. 2003; Sun et al. 2005] to create a new image by filling in the missing areas, or erasing unwanted regions using existing image data. The well-known methods utilize texture synthesis techniques and non-parametric methods for image completion. While a variety of objects or scenes can be used to fill in the hole seamlessly with a reasonable semantic interpretation, it is possible that none of the results meet the user's requirement.

Methods Based on a Large Image Collection The huge collection of images available in the Internet opened up new research opportunities for computer graphics [Snavely et al. 2006; Lalonde et al. 2007; Liu et al. 2008; Snavely et al. 2008]. Hays and Efros [2007] proposed a scene completion system using a large image collection. Their system successfully completes a wide range of scene images based on the global scene descriptor [Oliva and Torralba 2006]. Johnson et al [2006] proposed to use textural description to retrieve the desirable images from an annotated image database for image composition. An image patch can also be used in their system in lieu of text for specifying the desired fill-in.

The most related works to ours are [Hays and Efros 2007] and [Johnson et al. 2006]. We note that the global scene descriptor used in [Hays and Efros 2007] cannot be used to distinguish similar images with only local appearance variations (*e.g.*, the sketches drawn by users). Our system on the other hand uses local descriptors extracted from salient regions to represent local characteristics of images at all levels. When compared with [Johnson et al. 2006], our system is a more intuitive alternative to the user for conveying the desired appearance of the fill-in object. We believe our proposed approach is complementary to [Johnson et al. 2006], when it is difficult to use text to express the expected objects with the desirable appearance (*e.g.*, different types of clouds in Figure 10). Note that the work in [Johnson et al. 2006] classify the database images into a small number of semantic classes. If the query supplied by the user is out of the vocabulary of semantic classes, the user has to provide image patches to convey the search intention.

3 System Overview

Figure 2 summarizes our overall approach. The user draws sketches in a local region of the target image to convey his or her intention (shown in Figure 2(a)). Then, the sketched region along with its neighboring area which is to be removed (referred to as *working area*) is completed with the matching image patch, which is searched from millions of sources images collected from the Internet. To efficiently search the desired source images, we propose a two-step ranking approach by using local SIFT feature [Lowe 2004] in the first step and global gist feature [Oliva and Torralba 2006] in the second step.

Specifically, in the first step, we extract a set of local SIFT descriptors from the sketched target image [Lowe 2004] to represent local properties of the target image (The extracted SIFT features from the sketched area are shown in Figure 2(b)). Similar to [Sivic and

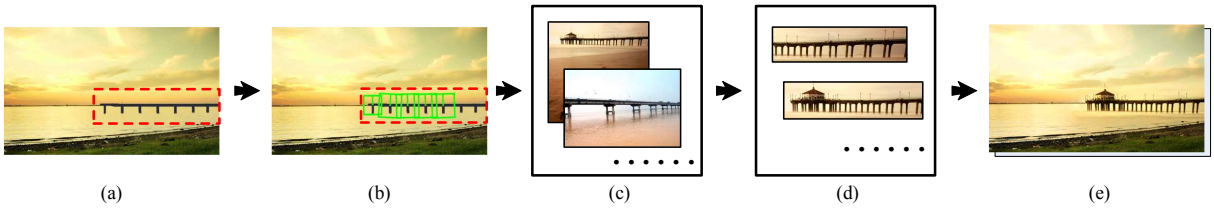


Figure 2: System overview. (a) A sketched target image. (b) The extracted SIFT descriptors from the sketched area (highlighted with green boxes). (c) Top ranked candidate images after the first-step ranking. (d) Top re-ranked candidate regions after the second-step re-ranking. (e) Composited results.

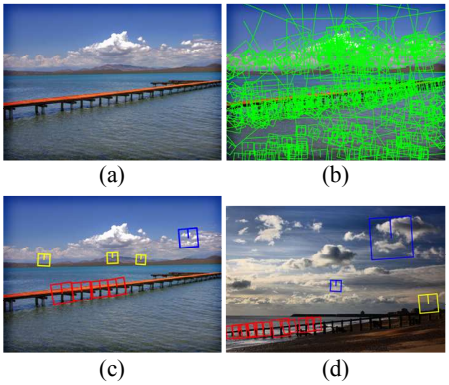


Figure 3: An illustration of the relationship between SIFT features and visual words. (a) A database image. (b) The extracted SIFT descriptors (highlighted with green boxes). (c)–(d) SIFT descriptors detected from two database images that are quantized into three visual words (highlighted in red, yellow and blue respectively).

Zisserman 2003], we quantize the SIFT descriptors of the target and source images into visual words, using the vector quantization method described in [Philbin et al. 2007] for efficient image retrieval. We then use the inverted file method to obtain a small subset of relevant images (referred to as *candidate source images*), which have at least one visual word in common with the target image. We rank the candidate source images according to an improved similarity measure based on the extracted visual word features. We observe that the top-ranked candidate images contain a sufficient number of locally similar patches to fulfill the user's intention. Note that searching based on SIFT features can only retrieve images with regions locally similar to the target sketch (shown in Figure 2(c)). In the second step, we further re-rank the matching regions of the top-ranked candidate images by similarity based on local semantical validity (shown in Figure 2(d)). The similarity is measured by comparing the gist features extracted from the expanded working area of the target image with those detected from the corresponding regions in candidate source images.

Finally, image composition is performed after the user selects one of the top re-ranked candidate regions of the source images (shown in Figure 2(e)).

It should be noted that the global features (e.g., gist) cannot be used in the first step because they cannot represent the detailed characteristics of local areas at different scales: to match local regions using global features, all the possible sub-windows of the images in the database need to be compared with those of the sketched regions of the target image, in a manner similar to non-parametric texture synthesis [Efros and Freeman 2001; Kwatra et al. 2003]. Given the sheer size of our image database, this is computationally intractable.

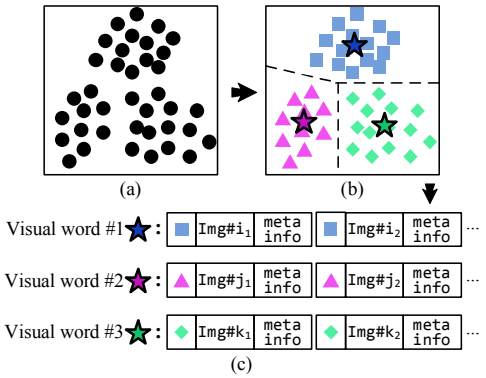


Figure 4: An illustration of local salient region matching. (a) A set of n_s randomly sampled descriptors from source images. (b) We group these descriptors into K clusters. The entire set of cluster centroids (indicated by stars) is referred to as the vocabulary of the *visual words*. (c) In the inverted file, for each visual word, we record the information of all the descriptors in the cluster, including the images where descriptors are detected and the descriptors' meta-information (such as position, size and orientation).

4 Local Feature-Based Ranking

We built our image corpus by downloading photographs from Flickr.com based on group and keyword search, which was also done in [Hays and Efros 2007]. In total, we downloaded about 2.5×10^6 images and the image sizes are larger than 800×600. We also observed that the top ranked images are quite disappointing, when the size of the image database is only ten thousand. A significant performance leap is achieved by increasing the image corpus to more than 2 million.

In our system, the user needs only to draw rough sketches to convey his or her intention. The frequently used sketches include: (a) color strokes, where the user draws rough structure of their intended fill-in content, (b) a clone brush, where the user copies content from other areas into the sketched area. Clone brush is often used to indicate the location where the original content should be removed. Careful application of clone brush is unnecessary, because the sketch will not be involved in the image composition stage.

4.1 Local Feature Extraction

Our system automatically searches for matching images in the database that contain regions semantically similar to the user-supplied sketches. Specifically, we extract a set of SIFT descriptors from salient regions, which are detected by Difference-of-Gaussian (DoG) interest point detector [Lowe 2004]. The SIFT feature detection is applied to both the target and the source image. The characteristics of local regions, including the sketches provided by the users, are encoded in the SIFT descriptors. Note that other local features or interest point detectors could potentially be used, and sev-

eral interest point detectors and local features are respectively compared in [Mikolajczyk et al. 2005] and [Mikolajczyk and Schmid 2005]. The recent study [Zhang et al. 2007] also demonstrated that combination of multiple interest point detectors and local descriptors can generally improve image matching and recognition performance, but the computation cost will also increase.

To speed up the matching process, we further eliminate detected salient areas whose sizes are less than a threshold. If the region is too small, a SIFT feature is not sufficiently robust against noise. Moreover, it is possible that small region contains only uninteresting and common image structures. Even if two small regions have similar SIFT descriptors, it is still possible that the surrounding regions are irrelevant. In our implementation, each image contains about 600 SIFT descriptors on average. Figure 3(b) shows the extracted SIFT features from a database image shown in Figure 3(a).

4.2 Local Salient Region Matching

Following [Sivic and Zisserman 2003], we quantize the SIFT descriptors into visual words using the vector quantization or clustering. Figure 4 illustrates the clustering process, where the cluster centers are referred to as *visual words*. Figure 3 shows a real example where a small subset of SIFT descriptors respectively extracted in (c) and (d) are quantized into the three visual words (highlighted in three colors). Note that SIFT descriptors extracted from similar local regions are quantized into the same visual words. For example, the red, yellow and blue SIFT descriptors extracted from the three regions correspond respectively to the jetty, the area between the cloud and the horizon, and the cloud.

Considering that we have millions of SIFT descriptors sampled from the collected source images, traditional clustering method such as K-Means fails to scale up to this size. Approximate K-Means (AKM) [Philbin et al. 2007] and Hierarchical K-Means (HKM) [Nistér and Stewénius 2006] were proposed for scalable clustering. In this work, we adopt AKM because of its promising performance for large-scale image retrieval [Philbin et al. 2007]. To alleviate the time-consuming computation in finding nearest neighbors between the points and the cluster centers in K-Means, a group of randomized k-d trees is used in AKM to approximate the nearest neighbor search.

Each SIFT descriptor is then labeled by a visual word by finding its closest cluster center. Each target or source image can therefore be represented as a 'bag' of unordered visual words. As shown in Figure 4(c), we build the inverted file, which has an entry for each visual word in the vocabulary, followed by a list of all source images (and other meta-information, such as positions) in which the visual word occurs. Similar to text mining where the inverted file method is used to efficiently find all text documents where a given word occurs, the same method can be used to efficiently find a small set of source images (referred to as candidate source images) that shares at least one common visual word with the target image.

We need to determine the number of clusters in AKM. The recent work [Philbin et al. 2007] used large vocabularies containing hundreds of thousands of clusters, and their work also showed that the image retrieval performance is quite flat when using moderately larger vocabularies. We therefore randomly choose about $n_s = 2.4 \times 10^7$ SIFT descriptors, which are grouped into $K = 1.2 \times 10^5$ clusters using AKM. Using two machines where each has two dual core CPUs, it takes about two weeks to perform the clustering, the quantization of the SIFT features detected in the source images into visual words, and the construction of the inverted file. Note that the above time-consuming process is performed off-line. In the query stage, the inverted file method can quickly filter out about 99% of the source images that do not share any visual word in common with the target image. Only a small set of candidate source images are retrieved for the subsequent process.

4.3 Feature-Based Ranking Using tf-idf

We rank the candidate images by similarity to the target image based on visual words, and then choose the top ranked 10,000 candidate images for semantic re-ranking (section 5). The similarity measure plays a crucial role in our system for finding the most locally similar images. In this work, we adopt the **tf-idf** feature (term frequency-inverse document frequency) [Jones 1972; Salton and Yang 1973], which was originally used in text mining.

Assume that the j-th image d_j has n_j SIFT descriptors, in which $n_{i,j}$ SIFT descriptors are quantized into the i-th visual word, $i = 1,\ldots,K$. Then the image d_j can be represented by a K-dimension tf-idf vector $(tfidf_{1,j},\ldots,tfidf_{i,j},\ldots,tfidf_{K,j})$. Term frequency (tf) weighs the occurrence of a word in a given document. We define the frequency of the i-th word occurring in the image d_j as:

$$\mathrm{tf}_{i,j} = \frac{n_{i,j}}{n_j}. \tag{1}$$

In Eq. (1), the total number of descriptors n_j is used to normalize the word frequency. Without the normalization term, the term frequency of large-size image with large area of texture content will be higher than other images.

In text mining, researchers have observed that frequently appeared words (*e.g.*, "the") are not useful. Inverse document frequency (idf) is then used to down-weigh the visual words (*e.g.*, common local structures of images) that appear too often in the source images. Assume the total number of images containing the i-th word is m_i, and the total number of images is $|D|$. We define the inverse document frequency as:

$$\mathrm{idf}_i = \log \frac{|D|}{m_i}. \tag{2}$$

The tf-idf value of the i-th word in the image d_j is:

$$\mathrm{tfidf}_{i,j} = \mathrm{tf}_{i,j} \cdot \mathrm{idf}_i. \tag{3}$$

The tf-idf value can be directly used to measure the similarity s_j between the target image d_q and a candidate source image d_j:

$$s_j = \sum_{i=1}^{K} (\mathrm{tfidf}_{i,j})(\mathrm{tfidf}_{i,q}) = \sum_{i=1}^{K} (n_{i,j} n_{i,q}) \frac{1}{n_j n_q} \log \frac{|D|}{m_i} \log \frac{|D|}{m_q}. \tag{4}$$

This standard similarity measure, however, has two limitations. First, even two SIFT features are significantly different in terms of size and orientation, the two features may be still considered matched. Second, it is difficult to control the weights of descriptors on the user's sketch using this measure. In the following, we define an improved similarity measure s'_j.

4.4 An improved similarity measure

To emphasize the descriptors on the user's sketch, a possible solution is to independently consider each SIFT descriptor in the target image d_q when we compute tf-idf. We define S_i as the set of descriptors quantized into the i-th word in the target image d_q. Note that the size of the set S_i is equal to $n_{i,q}$. We can therefore independently assign a tf value $\frac{1}{n_q}$ to each descriptor in the set S_i, namely:

$$\mathrm{tf}_{i,q} = \frac{n_{i,q}}{n_q} = \sum_{k \in S_i} \frac{1}{n_q}. \tag{5}$$

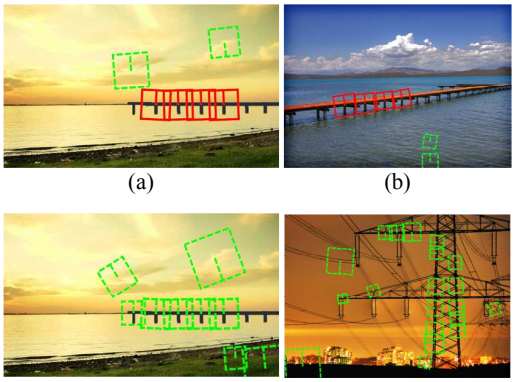


Figure 5: An illustration of our improved similarity measure. (a,b) and (c,d) are two matched image pair, where (a) and (c) are the sketched target image (also shown in Figure 2), (b) and (d) are two source images in the database. For each image pair, the matched SIFT descriptors used in the standard similarity measure are shown with red solid rectangles and green dashed rectangles. The mismatched SIFT descriptors (shown in green dashed rectangles) are not counted when computing our improved similarity measure.

Suppose the k-th SIFT descriptor of the target image are well matched to $h_{k,j}$ SIFT descriptors of the source image d_j. Note the salient regions detected by the DoG interest point detector are represented by four parameters: the position (x,y), the scale and the orientation [Lowe 2004]. In this work, the two SIFT descriptors are considered as well matched when their scale ratio and orientation difference are within the range of $[0.8,2.0]$ and $\pm 30°$respectively. In addition, we define a weight w_k for each SIFT descriptor of the target image. We set the weights for the SIFT descriptors in the sketched areas and other areas as $w_k = 10.0$ and $w_k = 1.0$ respectively to emphasize the SIFT descriptors in the sketch area. We then define a new similarity as

$$s'_j = \sum_{i=1}^{K} \left(\sum_{k \in S_i} w_k h_{k,j} \right) \frac{1}{n_q n_j} \log \frac{|D|}{m_i} \times \log \frac{|D|}{m_q}. \qquad (6)$$

When compared with Eq. (4), $n_{i,j} n_{i,q}$ is replaced by $\sum_{k \in S_i} w_k h_{k,j}$. Suppose the target image has two SIFT descriptors A and B, and the source image has three SIFT descriptors X, Y and Z. We also assume all the five descriptors are quantized into the i-th visual word. Then we have $n_{i,j} n_{i,q} = 6$ from Eq. (4). Suppose the descriptor A from the sketched area is matched only to the descriptors X and Y, and the descriptor B is matched only to the descriptors Z. From Eq. (6), we have $\sum_{k \in S_i} w_k h_{k,j} = 2 \times 10 + 1 = 21$. Therefore the SIFT descriptors in the sketch area are emphasized, while the mismatched descriptors are not counted. Figure 5 compares the standard similarity measure and our improved similarity measure. The rank of the image in Figure 5 (b) is promoted from 233 to 23, and the rank of the image in Figure 5 (d) is demoted from 11 to a rank beyond 10,000 after using our improved similarity measure, because all falsely matched SIFT descriptors with inconsistent scales or orientations are not counted.

We also note that the scale and orientation information of SIFT descriptors were recently used in [Jegou et al. 2008] to verify the geometrical consistency for image retrieval. In [Jegou et al. 2008], the rank of the database image is up if the SIFT descriptors of the database image after transformation with an optimal 'global' angle and scale can be better matched to the SIFT descriptors of the query image. Otherwise, the rank value of the database image will be down. In contrast to [Jegou et al. 2008], we independently remove the mismatched pair of SIFT descriptors and emphasize the

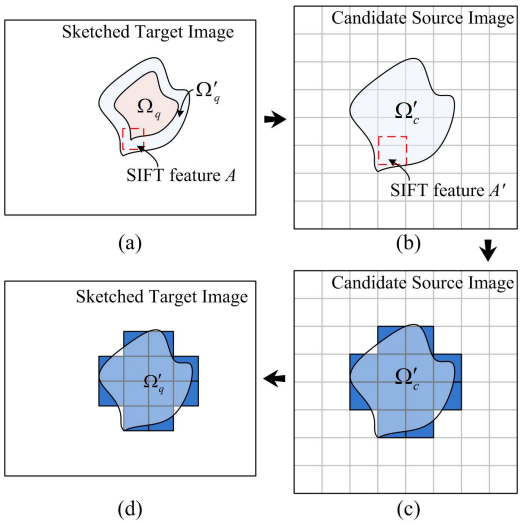


Figure 6: An illustration for gist-based local difference computation. (a) Sketched area in the target image. SIFT feature A is highlighted using a red dashed box. (b) The corresponding area in the candidate source image. The SIFT feature A' which is corresponding to A is also highlighted with a red dashed box. (c) Rasterized area in the candidate source image. (d) Rasterized sketched area in the target image.

SIFT descriptors in the sketched area using Eq. (6), instead of employing a single optimal global transform for all the SIFT descriptors. Our method is more suitable for searching images with areas locally similar to the sketch region, because the sketch region may only occupy a small portion of the target image. Thus, the global transformation may not be affected at all by such small sketch region.

5 Semantic Re-ranking

The top ranked candidate source images returned from local feature-based ranking contain sufficient local patches similar to the target regions. We further re-rank the matching regions of the 10,000 top ranked candidate source images and select 40 semantically valid candidate image regions.

Similarly as in [Hays and Efros 2007], in this step we extract gist descriptors from six oriented edge responses at four scales aggregated to 8×8 blocks. Each block is represented as a 24-dimensional gist feature. For re-ranking efficiency, we store the gist descriptors of all source images into the gist feature database. Suppose that a SIFT feature A in the sketched region and its corresponding SIFT feature A' in the source image are respectively extracted at positions (x,y) and (x',y'), and that their respective scales are s and s'. We can then calculate the displacement (dx,dy) and scale variation ds. The sketched region may contain multiple SIFT features. If these SIFT features are matched to the SIFT features of different regions of one source image, all the matching regions are considered for re-ranking.

Let Ω_q be the sketched region on the target image. As shown in Figure 6(a), we grow Ω_q by $0.4 \times |\Omega_q|$ and denote the expanded area as Ω'_q. Based on (dx,dy,ds), we can find the corresponding areas Ω_c and Ω'_c in the source image. The set of blocks where at least half of the area is covered by Ω'_c is found. Suppose that the number of such blocks is n_c. For each of the n_c blocks (shown in Figure 6(c)), we directly retrieve the extracted gist features from the gist feature database.

Then, for all the covered blocks by Ω'_c, we can inversely find the

(a) (b) (c)

Figure 7: Illustration of semantic re-ranking. (a,b) and (a,c) are two matched image pairs, where (a) is the sketched target (also shown in Figure 2) and (b) (c) are two source images in the database.

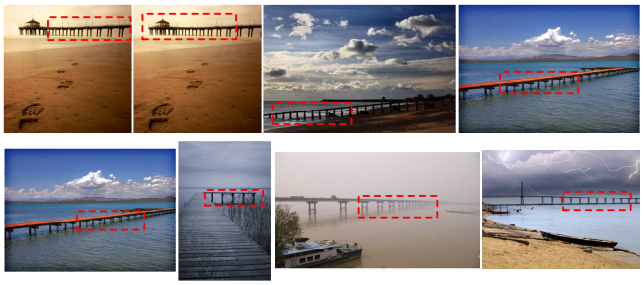


Figure 8: The eight selected top-ranked candidate image regions on the corresponding source images for the "jetty" example shown in Figure 2.

corresponding n_c blocks of the sketched region (shown in Figure 6(d)). The gist feature is similarly extracted by aggregating the corresponding pixel-wise responses to n_c blocks with summed area table [Crow 1984] within $O(n_c)$-time cost. For both the source image and target image, the gist features extracted from n_c blocks are put together as a long vector. Finally, we pick 40 candidate image regions based on the fused similarities from SIFT feature and gist feature with equal weight.

In our implementation, we normalize the similarities of SIFT feature into $[0,1]$ by dividing them using the maximum similarity. The Gaussian kernel is used to calculate the similarity based on gist feature. Note that the similarity from SIFT feature is image-level measure. If one source image contains multiple regions for consideration, we use the same image-level similarities for all the regions.

Figure 7 demonstrates the effect of gist based re-ranking for retrieving semantically valid image regions to the fulfill the user's intended requirement. Without accounting for the gist-based similarity, the rank of the image in Figure 7(b) and (c) are 32 and 16 respectively. When both SIFT and gist features are considered in ranking and re-ranking, the final rank of the image region in Figure 7(b) is within top 20, and the rank of the image region in Figure 7(c) is 337. It is also worth mentioning that we grow Ω_q to the expanded area Ω'_q, therefore, the gist feature also encodes the semantics of the surrounding areas. In this example, the contextual correlation between "jetty" and "river" is also helpful to obtain semantically valid image regions. For this case, after semantic re-ranking, our system successfully retrieves ten candidate image regions in the top 40 retrieved image regions. Eight selected candidate image regions are shown in Figure 8.

6 Seamless Composition

To seamlessly merge the selected candidate image region into the target image, we first align the candidate image region with the sketched area on the target image. Then we use graph-cut algorithm to find an optimal seam. Finally, we use Poisson blending to fuse the candidate region into the working area.

6.1 Local Alignment Optimization

For each candidate source patch Ω'_c, we optimize its position on the target image. We build a multi-level image pyramid for Ω'_c and Ω'_q. The local alignment is performed from the coarsest to the finest level. At each level, we set the displacement within $(\pm 2, \pm 2)$, and the optimal displacement is determined according to the highest normalized cross correlation between the target and candidate source patches. We do not optimize the orientation of candidate source patches because most photographs are captured in a similar orientation.

6.2 Finding Optimal Seam

To seamlessly paste a candidate source patch into the target image, we need to find an optimal seam $\partial\Omega^*$ within $\Omega'_q \backslash \Omega_q$. Pixels in the image area inside Ω^* will be obtained from the candidate source patch, and those outside of the area will be from target image.

We define $L(x) \in \{0,1\}$ where $L(x) = 1$ means that we use the existing target image data and $L(x) = 0$ means that we use the candidate source image data. We minimize a similar function as in [Hays and Efros 2007] to find the optimal seam:

$$C(L) = \sum_x C_d(x, L(x)) + \sum_{x,y} C_i(x, y, L(x), L(y)) \tag{7}$$

The resulting region should not contain too many pixels whose colors are significantly different from the colors of the user's sketch. The data penalty $C_d(x, L(x))$ is therefore defined as:

$$\begin{cases} C_d(x,1) = \lambda_d |I_e(x) - I_s(x)| \\ C_d(x,0) = \lambda_d |I_p(x) - I_s(x)| \end{cases} \tag{8}$$

where $I_e(x)$, $I_p(x)$ and $I_s(x)$ are the luminance of pixel x in the target image, the candidate source image and the sketched region respectively. In Eq. (8), a higher penalty $\lambda_d |I_e(x) - I_s(x)|$ for pixel x is used, if $L(x) = 1$ and the luminance difference between pixel x in the existing target image and the sketched region is high. Similarly, a higher penalty $\lambda_d |I_p(x) - I_s(x)|$ for pixel x is used, if $L(x) = 0$ and the luminance difference between pixel x in the candidate source image and the sketch region is high.

We use the same method as in [Hays and Efros 2007] to define the smoothness term $C_i(x, y, L(x), L(y))$ and also use the min-cut algorithm described in [Boykov and Kolmogorov 2004] to solve the $L(x)$ in Eq. (7). In our system, we also allow the user to mask the regions in which the seam cannot pass through.

6.3 Gradient-based Composition

We solve Poisson equations on the entire image area to fuse target image and candidate patch seamlessly. We choose the Poisson solver described in [Agarwala 2007], which can drastically improve the speed of blending without apparent loss of quality.

7 Results and Discussion

Using our interactive scene montage system, the user can actively control the search process by supplying a rough sketch to constrain the rough appearance of the desired objects. Our system will automatically find semantically-valid content to fill in, provided that the sketch is reasonably drawn rather than meaningless scribbles. We use two machines to search the desirable image regions in parallel, where the image data including the raw data, gist features, SIFT features and meta-information as well as the inverted file and the vocabulary of visual words are distributed on two machines, each

running with a dual core CPU (2.33GHz), 4GB RAM and two hard disks. The whole search process takes about 1.5 minutes only, including the communication time among machines.

To test the performance of our system, we use a total 31 examples with diverse sketch appearance, including nine examples shown in Figures 9 and 10, twelve examples shown in Figures 1, 11, 12, 13, and 14, and ten examples shown in the supplementary material. We carefully check the semantical validity between the returned top-40 image regions and the user-supplied sketches. For all the examples, about 4.7 image regions in average are obtained in the top-40 image regions. While the retrieval performance is about 12% only, at least one image region can be seamlessly merged into the target image for all the examples, which is enough for the real application. As the first sketch-based scene montage system, we believe the retrieval performance of our system can be further improved by employing additional information, such as text, in the future.

7.1 Robustness and Benefits of Sketch

We evaluate the robustness of our sketch-based montage system in this section. Shown in Figure 9 is a "cloud" sketch whose shape is progressively changed by the user. As shown in Figure 9(a)–(c), while the shapes of "cloud" are different, our system can effectively retrieve relevant source images that contain the specific kinds of clouds desired by the user, which are seamlessly pasted onto the respective target images. This experiment demonstrates the robustness of our system in tolerating slight variation of sketches. Even if the user-supplied sketch becomes ambiguous as shown in Figure 9(d), our system can still retrieve image regions resembling to the sketch, thanks to the effectiveness of our two-step ranking method for retrieving semantically valid image regions from the huge image collection. In this example, our system retrieves images containing the "cloud" or the "moon", and the user can select the desirable one for composition. However, if the user draws a totally different sketch not resembling to a cloud any more, our system will no longer retrieve any cloud images (see Figure 9(e)).

In Figure 10, we show the benefits of sketch for representing the diverse shapes of the object fill-in. In this example, the user draws five types of "cloud" sketch on a target image. Our system successfully retrieves the desirable source image regions to fulfill the user's search intention. All resulting composites look seamless.

7.2 Results

Our system can handle a variety of sketch scenarios:

Inserting small objects into an area with simple structure Since small objects can hardly change the rough appearance of an image, the use of global gist descriptors fail to express the user's intention. On the other hand, our algorithm is geared to capture local salient regions of the user-drawn sketch, and emphasizes them with higher weighting. Therefore, our algorithm can find source images that better meet the user's requirement. Two examples are shown in Figure 11, in which the user inserts in the respective target images a person and a bird flock.

Merging large scenes into the target image In Figure 12, we show four example results where the user inserts into the respective target images the Golden Gate Bridge, a mountain, a sunflower field, and a lake. Here, the user's sketch changes considerably the global appearance of the target image. Thus, the gist descriptor is capable of reflecting the user's intention, and therefore is effective in returning a reasonable set of source images. However, notice that it is in fact unnecessary to search for a source image that matches the global structure with the input. Refer to Figure 12: in the third row, although the global appearance of the input and source image are quite different, the matching region of the source image can be composited producing a new image with a valid semantic interpretation. This demonstrates that semantic validity does not require the entire scene to be roughly matched. Rather, it suffices to ensure that the working area be well matched to generate a semantically valid composition result.

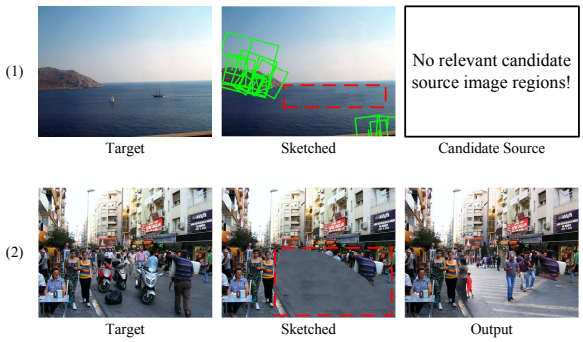


Figure 15: Two failure cases.

Embedding objects into scenes with complex structure Figure 1 and Figure 13 show a total of five examples. In each case, the user roughly draws on the respective original images a train track, a building roof, a jetty, a few steps and a fence. Eight selected candidate image regions for the "jetty" example are also shown in Figure 8. In particular, note that in Figure 13, although the matching source image and the sketched target image have significantly different global appearance, the resulting composite is still reasonable.

Merging multiple objects into scenes Our system can also insert multiple objects into the same target image. Figure 14 shows an example, in which a boat, a duck and a group of buildings are seamlessly pasted into the target image where the sketches are drawn. The resulting composite is seamless and has a natural look.

7.3 Limitations

There are two limitations in our algorithm. First, due to the large scale of the database and the retrieval performance consideration, we do not extract SIFT descriptors from a dense grid on the image. Instead, we extract SIFT descriptors from salient regions, which are detected by DoG interest point detector [Lowe 2004]. As shown in Figure 6, our system decides the set of candidate source image regions by using the matched SIFT descriptors of the expanded sketched area (see more details in Section 5). As a result, our approach does not work well in situations where no SIFT descriptors are extracted in the sketch area, such as constant-color image regions (e.g., a calm sea). The top of Figure 15 shows an example: the user wants to remove the sailing boat from the sea. But no local salient regions are present in the sketch area of the target image (SIFT descriptors extracted from the target image are highlighted in the second column). Therefore, no relevant candidate source image regions are retrieved. However, note that existing texture synthesis methods can be used to fill in the area. Second, if the user draws a sketch in areas with very intricate structures and complex semantical information, the graph-cut algorithm discussed in Section 6.2 may not be able find a reasonable seam. The bottom of Figure 15 shows one such example. While our system can successfully retrieve the semantically valid image region, half of the man's body was left in this example, because it is difficult to find a reasonable seam in this complex case.

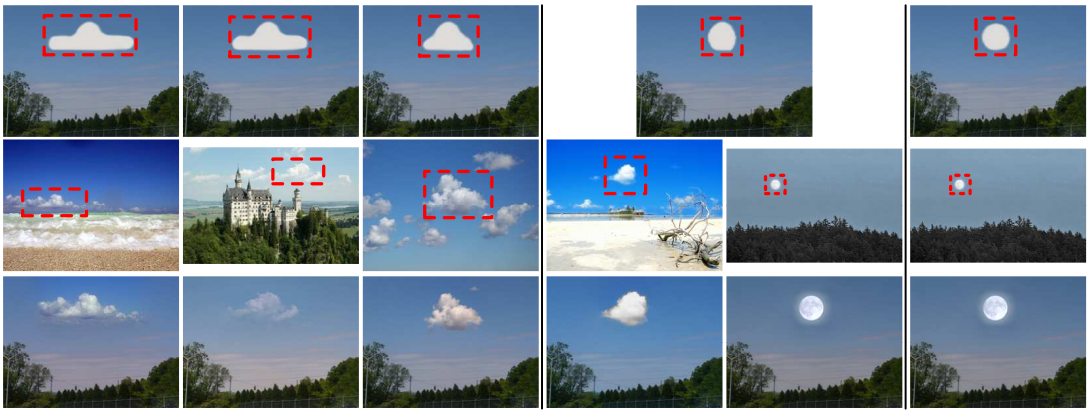


Figure 9: Illustration of robustness of sketch. The shape of the "cloud" sketch is progressively changed by the user. Top: five sketched images. Middle: The selected image regions on their respective source images. Bottom: Composited results.

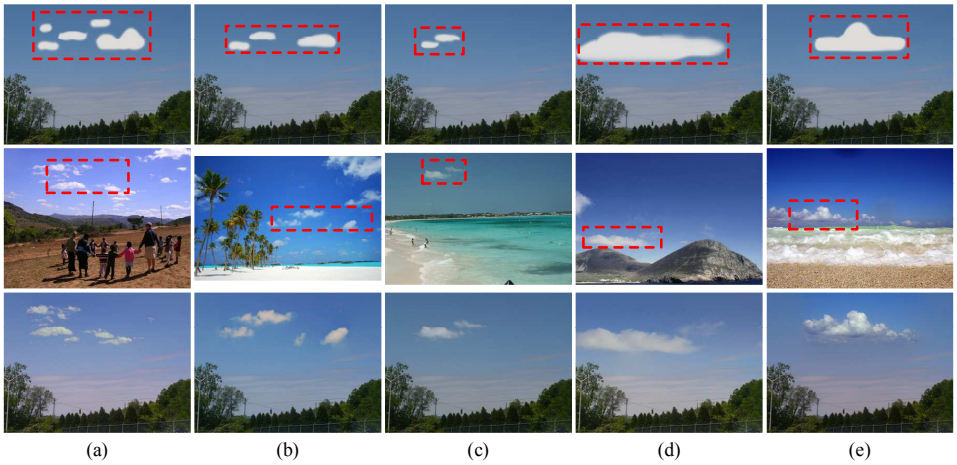


Figure 10: Illustration of benefits of sketch. The user draws five types of sketches for representing the diverse shapes of the "cloud" sketch. Top: the sketched images. Middle: the selected image regions on their respective source images. Bottom: Composited results.

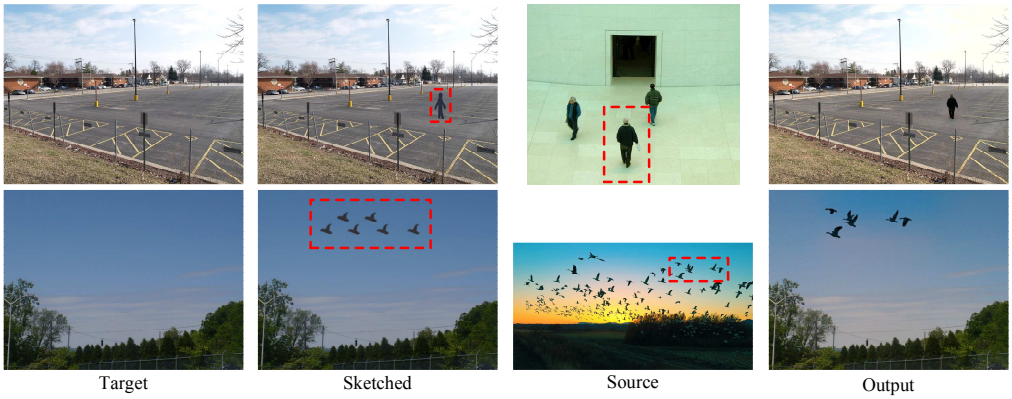


Figure 11: Inserting small objects into areas with simple structure. User-supplied sketches (column 2) and the selected candidate image regions (column 3) are highlighted by red dashed boxes.

8 Concluding Remarks

While previous algorithms [Pérez et al. 2003; Agarwala et al. 2004; Jia et al. 2006] require the input of one or more source photographs, in this paper, we propose a system that only requires rough user-supplied sketches on the target image to indicate the user's intention. Our algorithm will then automatically retrieve suitable source image regions in a huge image database, and seamlessly composite the pertinent region into the target image.

The search for suitable source image regions in a large online image database, such as the world wide web, can be a frustrating experience. Our contribution lies in the automatic translation of a user's rough input sketch into a set of candidate source images ranked by similarity based on visual words, which are then re-ranked by se-

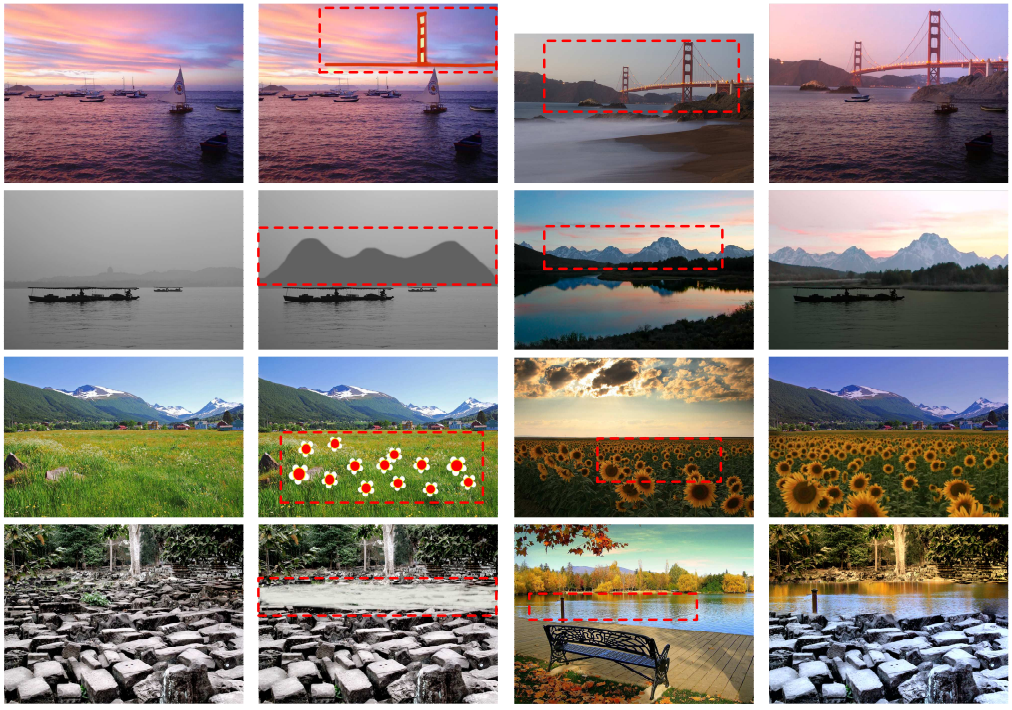
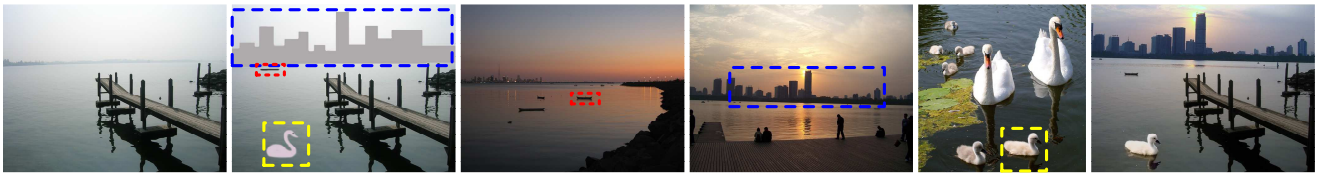


Figure 12: Merging large scenes into target images. User-supplied sketches (column 2) and the selected candidate image regions (column 3) are highlighted by red dashed boxes.

Figure 13: Embedding objects into areas with complex structure or texture. User-supplied sketches (column 2) and the selected candidate image regions (column 3) are highlighted by red dashed boxes.



(a) (b) (c) (d) (e) (f)

Figure 14: Inserting multiple objects into the same target image. (a) Target image. (b) Sketched image (user-supplied sketches are highlighted by three colored dashed boxes). (c)–(e) Three selected candidate image regions. (f) Composited result.

mantical validity based on gist feature. In our novel adaptation of text-based retrieval techniques in scene montage, our large photo collection is analogous to a large text database, while the desired source image is analogous to the text to be mined. In this paper, we also propose an improved similarity measure to rank relevant source images based on visual words for our application.

In computational photography, a lot of attention has been paid on "how to generate realistic images given a source and a target image", while the question "how to find good source images?" has received surprisingly less attention. With the ever increasing size of online photo collection in the Internet, the latter question has become highly relevant. We believe this paper has made a fruitful and significant first pass in addressing some of the key issues in answering the latter question. We also believe that our sketch-based scene montage method can be integrated with text-query based methods to achieve better results, which will be investigated in the future.

References

AGARWALA, A., DONTCHEVA, M., AGRAWALA, M., DRUCKER, S., COLBURN, A., CURLESS, B., SALESIN, D., AND COHEN, M. 2004. Interactive digital photomontage. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, vol. 23, 294–302.

AGARWALA, A. 2007. Efficient gradient-domain compositing using quadtrees. In *SIGGRAPH '07: ACM SIGGRAPH 2007 Papers*, vol. 26, 94.

BOYKOV, Y., AND KOLMOGOROV, V. 2004. An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. *IEEE TPAMI 26*, 9 (Sept.), 1124–1137.

CRIMINISI, A., PEREZ, P., AND TOYAMA, K. 2003. Object removal by exemplar-based inpainting. *IEEE CVPR'03 2* (June), II–721–II–728 vol.2.

CROW, F. C. 1984. Summed-area tables for texture mapping. In *SIGGRAPH '84: ACM SIGGRAPH 1984 Papers*, ACM, New York, NY, USA, 207–212.

DATTA, R., JOSHI, D., LI, J., AND WANG, J. Z. 2008. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys 40*, 2, 1–60.

DRORI, I., COHEN-OR, D., AND YESHURUN, H. 2003. Fragment-based image completion. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, vol. 22, 303–312.

EFROS, A. A., AND FREEMAN, W. T. 2001. Image quilting for texture synthesis and transfer. In *SIGGRAPH '01*, 341–346.

EFROS, A. A., AND LEUNG, T. K. 1999. Texture synthesis by non-parametric sampling. In *IEEE ICCV '99*, 1033.

GUO, C., ZHU, S., AND WU, Y. 2003. Towards a mathematical theory of primal sketch and sketchability. In *ICCV03*, 1228–1235.

HAYS, J., AND EFROS, A. A. 2007. Scene completion using millions of photographs. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, 4.

JEGOU, H., DOUZE, M., AND SCHMID, C. 2008. Hamming embedding and weak geometric consistency for large scale image search. In *European Conference on Computer Vision*, Springer, A. Z. David Forsyth, Philip Torr, Ed., vol. I of *LNCS*, 304–317.

JIA, J., SUN, J., TANG, C.-K., AND SHUM, H.-Y. 2006. Drag-and-drop pasting. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, 631–637.

JOHNSON, M., BROSTOW, G., SHOTTON, J., ARANDJELOVIC, O., KWATRA, V., AND CIPOLLA, R. 2006. Semantic photo synthesis. 407–413.

JONES, K. S. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*.

KWATRA, V., SCHÖDL, A., ESSA, I., TURK, G., AND BOBICK, A. 2003. Graphcut textures: image and video synthesis using graph cuts. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, 277–286.

LALONDE, J.-F., HOIEM, D., EFROS, A. A., ROTHER, C., WINN, J., AND CRIMINISI, A. 2007. Photo clip art. In *SIGGRAPH '07: ACM SIGGRAPH 2007 Papers*, ACM, New York, NY, USA, 3.

LIU, X., WAN, L., QU, Y., WONG, T.-T., LIN, S., LEUNG, C.-S., AND HENG, P.-A. 2008. Intrinsic colorization. *ACM Trans. Graph. 27*, 5, 1–9.

LOWE, D. G. 2004. Distinctive image features from scale-invariant keypoints. *IJCV 60*, 2, 91–110.

MARR, D. 1982. Vision: A computational investigation into the human representation and processing of visual information. In *W.H. Freeman*.

MIKOLAJCZYK, K., AND SCHMID, C. 2005. A performance evaluation of local descriptors. *IEEE TPAMI 27*, 10, 1615–1630.

MIKOLAJCZYK, K., TUYTELAARS, T., SCHMID, C., ZISSERMAN, A., MATAS, J., SCHAFFALITZKY, F., KADIR, T., AND GOOL, L. V. 2005. A comparison of affine region detectors. *IJCV 65*, 1/2, 43–72.

NISTÉR, D., AND STEWÉNIUS, H. 2006. Scalable recognition with a vocabulary tree. In *IEEE CVPR*, IEEE Computer Society, New York, NY, USA, 2161–2168.

OLIVA, A., AND TORRALBA, A. 2006. Building the gist of a scene: the role of global image features in recognition. *Progress in brain research 155*, 23–36.

PÉREZ, P., GANGNET, M., AND BLAKE, A. 2003. Poisson image editing. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, vol. 22, 313–318.

PHILBIN, J., CHUM, O., ISARD, M., SIVIC, J., AND ZISSERMAN, A. 2007. Object retrieval with large vocabularies and fast spatial matching. In *IEEE CVPR'07*, IEEE Computer Society, Los Alamitos, CA, USA, vol. 0, 1–8.

SALTON, G., AND YANG, C. 1973. On the specification of term values in automatic indexing. *Journal of Documentation 29*.

SIVIC, J., AND ZISSERMAN, A. 2003. Video google: A text retrieval approach to object matching in videos. In *IEEE ICCV '03*, 1470.

SMEULDERS, A., WORRING, M., SANTINI, S., AND GUPTA, A. 2000. Content-based image retrieval at the end of the early years. *T-PAMI 22*, 12, 1349–1380.

SNAVELY, N., SEITZ, S. M., AND SZELISKI, R. 2006. Photo tourism: exploring photo collections in 3d. *ACM Trans. Graph. 25*, 3, 835–846.

SNAVELY, N., GARG, R., SEITZ, S. M., AND SZELISKI, R. 2008. Finding paths through the world's photos. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2008) 27*, 3, 11–21.

SUN, J., YUAN, L., JIA, J., AND SHUM, H.-Y. 2005. Image completion with structure propagation. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, 861–868.

TORRALBA, A., FERGUS, R., AND FREEMAN, W. 2008. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE TPAMI 30*, 11 (Nov.), 1958–1970.

WANG, X.-J., ZHANG, L., LI, X., AND MA, W.-Y. 2008. Annotating images by mining image search results. *IEEE TPAMI 30*, 11 (Nov.), 1919–1932.

ZHANG, J., MARSZAŁEK, M., LAZEBNIK, S., AND SCHMID, C. 2007. Local features and kernels for classification of texture and object categories: a comprehensive study. *IJCV 73*, 2 (jun), 213–238.