PAPER

Query-by-Sketch Based Image Synthesis

David GAVILAN^{†a)}, *Nonmember*, Suguru SAITO[†], and Masayuki NAKAJIMA[†], *Members*

SUMMARY Using query-by-sketch we propose an application to efficiently create collages with some user interaction. Using rough color strokes that represent the target collage, images are automatically retrieved and segmented to create a seamless collage. The database is indexed using simple geometrical and color features for each region, and histograms that represent these features for each image. The image collection is then queried by means of a simple paint tool. The individual segments retrieved are added to the collage using Poisson image editing or alpha matting. The user is able to modify the default segmentations interactively, as well as the position, scale, and blending options for each object. The resulting collage can then be used as an input query to find other relevant images from the database.

key words: query-by-sketch, image synthesis, image compositing, collage, relevance feedback

1. Introduction

With the development of digital media and the Internet, many users have moved from being passive media consumers to become active content creators. Many of these creators browse the Web looking for images to use in their works. While the final goal of the user is to create a new image, retrieval systems just provide ways of representing intermediate results. Let us call *intention gap* the gap between the user's ultimate goal and the output given by the system. The *representation gap* is then the gap between the user's goal and the ways the system let us represent such goal. In this paper, we unify the process of search and creation in a single application, called *sketch-to-collage*, to bridge the intention gap. In order to bridge the representation gap, we use sketches as input, which are a human-made abstraction of the image that the user wants to obtain. In this paper, the final image can be an existing photograph retrieved from a database, a newly synthesized photograph, or a more general synthesized collage.

A collage is "a form of art in which various materials such as photographs and pieces of paper or fabric are arranged and stuck to a backing". The act of creation of a collage supports creative expression and allows the user to express feelings, thoughts, and emotions [1]. However, collage has gained attention as a good visual summarization technique [2], [3], while relegating the artistic expression to a secondary place.

Visual image summarization techniques focus on providing a visual summary in the form of a collage, although in general a collage does not need to be a (good) visual summary [4]. Since browsing image collections can be a tiresome task because of the large amount of visual information the user needs to process, automatic summarization techniques need to define what is important in pictures and summarize them accordingly. Content-based image summarization techniques provide high-level descriptions of images, by means of clustering [5], [6] or categorization [7], [8]. On the other hand, collages used as a visual summarization technique try to create a result that is appealing to the user. This appeal is defined with the selection of the images appearing in the collage and the way they are displayed. Therefore, while known to be a subjective matter that may change from user to user, for automatic collages appealingness depends only on the problem formulation, like in AutoCollage [2].

This paper addresses the problem of interactively creating an image collage. An interactive technique should let the user select what should appear in the collage, how it is displayed, and also the purpose of the collage. With the help of image retrieval techniques the user can easily select photographs from an image collection and focus on the artistic part of the collage. If the purpose of the collage is to offer a visual summary, for instance, users will select photographs that are relevant to them, and then arrange these pictures in their own creative way.

Our system makes use of this image retrieval based interaction for creating a seamless collage. Figure 1 is an example of such an interaction process. The main idea is that the user directly paints in a canvas a rough sketch that represents the desired composition result. The system finds possible image matches for the different parts of the sketch, and the user selects from these matches to obtain the final composite image.

The process of collage creation can be thought of a generalized relevance feedback method. In image retrieval, typical relevance feedback methods involves the user just picking a relevant image from the displayed results [9]. In our case, the user selects parts from images, selecting what is relevant from different sources. The created collage is, thus, the most relevant match. This collage can then be used to query the image database to find other relevant images.

Manuscript received January 31, 2008.

Manuscript revised May 5, 2008.

[†]The authors are with the Graduate School of Information Science & Engineering, Tokyo Institute of Technology, Tokyo, 152-8552 Japan.

a) E-mail: david@img.cs.titech.ac.jp

DOI: 10.1093/ietisy/e91-d.9.2341

*Oxford American Dictionary.

Fig. 1 Creation process of a collage for a trip summary.

2. Related Work

Several tools have been introduced to aid in the creation of collages. Systems like Picture Collage [3], Digital Tapestry [10], or AutoCollage [2], formulate the problem as a global optimization procedure in order to automatically create collages. They address the problem of packing images in a target canvas, while setting restrictions such as overlap minimization or energy minimization along boundaries. AutoCollage also adds object recognition, particularly sky and faces, for a better search of regions of interest inside images. In Collage Authoring [11], the layout is decided by means of the metadata contained in the pictures. The aesthetic of the collage depends strictly on the problem formulation and in the input set. In this work, we want the user to decide what is to appear in the collage.

In Semantic Photo Synthesis [12], the user decides what is relevant in the composition by describing the “collage” semantically, that is, using keywords and positioning them in a canvas. A labeled image database is used to grab segments and synthesize a new photograph (not a collage) using graph-cut optimization. In our system, labels are not necessary, and the hues of the sketch are directly related to the hues of the resulting collage. Moreover, the user can use several sketches as input, and all the parts of the collage can be modified interactively. Images and segments are retrieved in real time, and not passively synthesized in a unique photograph.

On the other hand, the main focus of the Kandinsky system [4] is to create aesthetically interesting collages. Given some keywords, relevant images are selected and placed in a collage that follows the template given by the artist. In our approach, the template is directly created by the user in the form of a rough color sketch, and at the same

time, the sketch is used to retrieve relevant pictures from the image collection. Therefore, the core of both the image selection and collage layout is the query-by-sketch system. In a sense, this is similar to Collection Understanding [13]: users may have no prior knowledge of the metadata or the photographs found in the image collection. With the help of query by example, we can easily browse an image database and look for objects that match the aesthetic representation conveyed in our input sketch.

Sketches have been used for image retrieval in several occasions. Jacobs et al. [14] proposed a fast system based on wavelets. Their approach has been implemented in several existing applications, like Retrivr [15] or ImgSeek [16]. However, wavelets are very sensitive to details such as the position of the wavelet coefficients in the image, and the precise value of these coefficients. Images that have similar color distributions to a human observer can become harder to retrieve. This is especially true for sketches, where parts like the sky are usually painted uniformly, and with a blue hue not existing in photographs. In our approach, we take two countermeasures to this. The first one is to limit the color input to a gamut of colors appearing in photographs. The second one is to use a region-based approach for retrieval. By using whole regions instead of wavelets coefficients, we can allow for more variation in displacements or colors.

There are some query-by-sketch systems that use regions for indexing images. Fonseca et al. [17] built a topology tree to represent the sketch, and used it to index clipart databases. Similarly, Di Sciascio et al. [18] used a topology graph to index photographs. In their approach, images need to be properly segmented, and then several geometric and color features are computed for every region. These are used as the nodes of the graph, and the problem becomes a graph match search. To avoid the computational complexity of using graphs, we propose the use of region histograms, that is, histograms representing each of the features for each region. Our method is not only fast, but also allows loose matches, that is, visually similar images with different topologies can be retrieved. Therefore, it allows some noise in the segmentation of the images in the database, as well as using very rough sketches as input.

3. Framework

The proposed system has three main subsystems: the segmentation module, the retrieval system, and the compositing module. Photographs are segmented in a few regions using *K*-means clustering, color categorization, and statistics order filtering. Then, images are stored in a database, storing separate properties for each region, which are used for retrieval. The user draws a sketch with simple strokes that are used to retrieve particular segments that are passed to the compositing module. The retrieval and compositing are interactive, so the user can select different images and create several arrangements in the composite.

Figure 2 shows the application that implements this

Fig. 2 Query interface.

framework. The top left window is where the user “paints queries”. The middle window is the object query, while the bottom-left window is the image query, both explained in the next section. The windows on the right are part of the compositing module explained in Sects. 5 and 6; the bottom-right window is the clipping window, and the top-right one is the collage result. Please check the application home-page [19] for manuals and tutorials.

4. Query by Sketch

In this section we describe the interactive process of querying the image database to select pictures for the collage.

4.1 Query Formulation

The process of creating a new collage starts by drawing in a canvas the rough arrangements of objects to appear in the collage. The top left window in Fig. 2 shows the painting tool, which consists of just a painting canvas, a slider to change the diameter of the rounded brush, and a color palette. Every stroke drawn in the canvas queries the image database for similar segments. The individual properties used for retrieval are described in detail in the next section.

Based on previous experience with sketch interfaces [20], the color palette is accessed just using fourteen color buttons. They represent the minimal set of eleven universal color categories [21] –black, gray, white, brown, pink, orange, purple, blue, yellow, green, and red–, and are complemented with light brown, light green, and azure. Azure and light brown were added to better represent the presence of sky and skin regions, respectively, in photograph databases [22]. Light green was suggested by users of our previous experiments [20] to represent the contrast in different vegetations.

Because we tend to remember colors more vivid than they really are [23], users would generally select colors outside the natural range in photographs, resulting in a decrease

Fig. 3 2D projected palette using SOM on RGB and its discrete version, arranged by color category.

of query performance. Therefore, for better color query results, we want to somehow limit the color gamut to that of real photographs. To do so, RGB samples of each color category were taken from a sample database of photographs, and then projected in a two-dimensional grid using Self-Organizing Maps (SOM). Following standard interface designs, we restricted the button size to 50×50 pixels. Each single pixel of the button is an RGB color vector that corresponds to a single neuron in a SOM of 50×50 neurons. When the user clicks on a color button, the interface displays a continuous color palette generated from the correspondent SOM of that color category.

When asked, users said it is difficult to perceive the subtle variations of each SOM-generated palette, and even more to remember the hues used. This is possibly related to the fact that we can barely retain four colors in the visual working memory [24]. Thus, we discretized each palette in four clusters using K-means, except for achromatic categories –black, gray, and white. Because there may be photographs with a gamut outside the observed norm, these color bins are completed with nine points along the boundary of the RGB cube: blue (0, 0, 1), cyan (0, 1, 1), clover (0, 0.5, 0), green (0, 1, 0), yellow (1, 1, 0), orange (1, 0.5, 0), red (1, 0, 0), magenta (1, 0, 1), purple (0.5, 0, 1). In total, we obtain the 52 bins shown in Fig. 3. The user can switch between this discrete palette or the one described earlier. The neural network for color categorization, the ten SOMs, and the 52 bins can be found online [25].

4.2 Object Retrieval

As pointed by Ullman [26], the most informative features in an image are typically of intermediate complexity. This may be the reason of the increasing popularity of region-based image retrieval methods [9]. In this paper we use our previous work on mobile image retrieval [27], tuned for query-by-sketch.

Briefly, every photograph in the database is partitioned in a few segments. For every segment we compute the following features: center of gravity $\mathbf{p}_0$; a vector $\mathbf{v} = (v_1, v_2)$ containing the volume v_1 normalized by the area of the whole image and the volume v_2 with respect its bounding box; color, represented by a Gaussian distribution on each band in the $L^*a^*b^*$ color space; and shape, represented by an *elongation* measure $\mathcal{E}$, and its orientation β . Every stroke of the sketch drawn by the user is also repre-

sented by the same features, and the matching regions from the database are the ones with closer distance d^{ij} between two objects i and j , $d^{ij} = w_p d_p^{ij} + w_v d_v^{ij} + w_c d_c^{ij} + w_o d_o^{ij}$, where $d_p^{ij} = \langle \mathbf{p}_0^i, \mathbf{p}_0^j \rangle$, $d_v^{ij} = \langle \mathbf{v}^i, \mathbf{v}^j \rangle$, $d_c^{ij} = \langle \theta^i, \theta^j \rangle$, and d_o are the position, volume, color, and orientation distance, respectively; $\langle \cdot, \cdot \rangle$ is the euclidean distance; w_p , w_v , w_c , and w_o are weights that can be modified by the user. The combined orientation-elongation distance d_o is defined as $d_o^{ij} = \frac{1}{2} (\mathcal{E}^i \mathcal{E}^j (1 - \cos(\beta^i - \beta^j)) + |\mathcal{E}^i - \mathcal{E}^j|)$.

The retrieved region is used as a simple mask representing the region of interest and it is automatically selected for a composite. The user can select other areas of interest from the related image. Figure 4 shows example strokes and the segmented objects retrieved from the CCT dataset described in the appendix. The default parameters are set to $w_p = 0.12$, $w_v = 0.25$, $w_c = 0.25$, and $w_o = 0.25$.

4.3 Image Retrieval

As the user adds more strokes to the sketch, the system retrieves new regions from the database. Moreover, it also retrieves images that match the current sketch. This image query is quickly performed using region histograms [27]. A *region histogram* is a vector-based sorted visual signature.

Fig. 4 Example object queries, sorted by rank.

The elements of each vector are sorted regarding the geometry of each region, for instance, from smaller to bigger when measuring the size. While using graphs can be used for perfect matches, graphs would also require perfect segmentations. Moreover, the complexity of graph matching is of order $O(n^p)$, p being the number of segments in the picture. On the other hand, fully independent features or bags of features can be matched in linear time, but no geometry can be imposed on the spatial relationship. Region histograms can also be matched in real time, but they contain inherent geometrical constraints because of the imposed ordering. These constraints, though, are much more permissive than matching graphs, thus, more appropriate for the kind of loose matches desired in sketch-based input interfaces.

Once the region histograms are computed, the distance between two images A and B is defined as $d_r^{AB} = w_p \langle \mathbf{h}_p^A, \mathbf{h}_p^B \rangle + w_s \langle \mathbf{h}_s^A, \mathbf{h}_s^B \rangle + w_c \langle \mathbf{h}_u^A, \mathbf{h}_u^B \rangle \langle \mathbf{h}_c^A, \mathbf{h}_c^B \rangle + w_v \langle \mathbf{h}_v^A, \mathbf{h}_v^B \rangle$, where $\mathbf{h}_p^A$ is the position histogram of image A , $\mathbf{h}_s^A$ its shape histogram, $\mathbf{h}_u^A$ and $\mathbf{h}_c^A$ its color histograms, $\mathbf{h}_v^A$ its size histogram, and w_p , w_s , w_c , and w_v are weights for each feature.

Because for some creations the user may want to limit the results of the query with higher level semantics, keywords can also be used to filter queries. This requires a weakly labeled image database.

5. Object Substraction

Our system uses sketches to retrieve segments from a collection of images, as described in previous section. As explained in Sect. 4, given an input stroke, the system retrieves a region Ω_j from some image in the database. This region is scaled up from a hard-segmented thumbnail version of the original image by quantizing this image with the color means of its thumbnail. Figure 5 is an example of this process.

In order to extract just big relevant objects, we reduce the internal scale by reducing the image size. By computing

Fig. 5 Scaling up segmented thumbnails. Top-left: linearly scaled-up thumbnail; top-right: dilated selection quantized with thumbnail palette; bottom-left: original image with the computed boundary; bottom-right: subtracted object.

Fig. 6 Interactively refining the object boundary.

these thumbnails, we also reduce the processing time and the storage requirements. The thumbnail is segmented using K -means on the CIELAB color space, where K is estimated by counting the static number of color categories [28]. We apply statistics order filtering to the clustered image to eliminate smaller regions and close holes. The statistics order filter is a median-like operator that sets every pixel to the most frequent value within a neighborhood. For our database, we resize images to a thumbnail of 60×80 pixels and filter them with a 3×3 filter.

By this method, we quickly obtain a partition of the image into a set of regions $\{\Omega_k\}$. Since these regions are non-intersecting, that is, $\Omega_i \cap \Omega_j = \emptyset, i \neq j$, fast refinements of the initial segmentation can be easily obtained by grouping segments. It is possible to refine this initial segmentation by specifying a region Ω_0 with a lasso around the object of interest in the same image. A first approximation of the object is then obtained directly from the preprocessed segmentation; it is defined as all the regions strictly contained in Ω_0 , that is, $\Omega_{obj} = \bigcup_{\Omega_k \subset \Omega_0} \Omega_k$.

The user can see this result immediately. However, this result may be very rough and, in general, it contains discontinuities. In Fig. 6, region Ω_a and Ω_b are separated by background pixels that may actually be part of the object. Therefore, the contour is further optimized by computing the shortest path inside Ω_0 that does not intersect any of the inner regions. This path is computed using 2D dynamic programming, as described in the Intelligent Scissors segmentation [29], and the gradient of the image is used to compute the local costs. As discussed in Sect. 6, a target gradient can be added at this point to minimize the energy of our target composite, as proposed by Jia et al. [30]. Like in their work, we compute the closed path $\partial\Omega$ using a cut C , the shortest between the contour $\partial\Omega_0$ and Ω_{obj} .

For separated regions like the one in the figure, most probably a path will exist that discards one of the regions. To avoid such paths, we simply connect all the regions inside Ω_0 with a line L that goes from one region to another, selecting the closest one at each step. For each region, the point in its contour closest to its center of gravity is used as reference. This line converts the region $\Omega_0 \setminus \Omega_{obj}$ into a region of genus-1 type. As described in Drag-and-Drop pasting [30], the cut C then converts this region into genus-0 type, and the problem of finding the closed curve is converted into finding

Fig. 7 Interactive object subtraction example.

Fig. 8 Alpha matting vs. Poisson blending.

the shortest path.

Notice that the definition of Ω_{obj} leaves Ω_d out, in Fig. 6. There may be some cases where the user wants to actually grab half of an object, so any hard segmentation will fail. For these cases, we let the user select *band optimization*, that consists of applying the same closed path boundary optimization but just inside a belt around the boundary $\partial\Omega_0$, regardless Ω_{obj} . Using this belt, we could compute the vertical seams in the first step in Fig. 1. We use the thick brush shown in Fig. 7 to make the interactive selections. In lasso mode, Ω_{obj} is extracted from the strictly contained regions of the selection. In band mode, Ω_{obj} is the hole inside the selection. Thanks to this band, it is possible to cut just a piece of the apple.

6. Seamless Image Compositing

The collage is initially a blank canvas, and as explained in previous sections, we add objects to it from a database of images by the process of retrieval and selection. However, every *source* piece that we add may be very different from the *target* collage. We use Poisson image editing [31] to seamlessly blend the colors of the source and the target. To prevent decolorization, we propose using the following vector field, $\mathbf{v}(p) = \beta \nabla g(p) + \max(0, 1 - \beta) \nabla f^*(p)$. By setting β to a value greater than 1, we can emphasize the gradient of the source to avoid some of the not always desirable color changes associated with Poisson image editing. In other cases, the user can select a simple alpha matte. Figure 8 shows the difference between the alpha matte and the color blending. Notice that the bird looks almost transparent with $\beta = 1$, that is, the default gradient import from the original paper. With our function we can easily emphasize the source

object.

7. Results and Evaluation

Our system can be used in several ways to create different kinds of collages. The average retrieval speed is 10.3 msec (on a MacBook CoreDuo 1.83 GHz, JavaVM 1.5) for images, and 39.2 msec for segments. Our extended drag-and-drop pasting implementation, including the Poisson solver, takes an average of 2 seconds for objects no bigger than 320×320 pixels. Boundaries and colors are updated in separate threads, so the user can see the retrieved object in the collage canvas in real time.

7.1 Weight Tuning

As described in the appendix, we can optimize the weights of our metric given a set of examples. We need two different sets of weights for two different types of inputs, sketches and photographs. Collages are synthesized photographs, so when using them as input, we will use the set of weights for photographs. For computing the weights of sketch input, we used the accompanying set of 268 sketches of the CCT dataset. For computing the weights of photograph input, we randomly selected 300 photographs from the CCT dataset.

For region histograms, the obtained weights are $w_p = 0.84$, $w_v = 0.44$, $w_s = 0.016$, $w_c = 1.13$, when the input is a sketch. For photographs, the weights are $w_p = 0.86$, $w_v = 0.40$, $w_s = 0.03$, and $w_c = 1.15$. As reference, we use another region-based metric, the fuzzy query used in Blob-world [6]. This metric selects one region from each image and their distance is used to represent the distance between the two images. The selected region is the one that maximizes the distance between the best matching region pairs. If we assume that the most relevant object is always centered, we could simply compute the distance between the objects closest to the image center instead. For the fuzzy query, the optimum weights are $w_p = 1.33$, $w_v = 1.21$, $w_o = 1.36$, $w_c = 1.04$ for sketches, and $w_p = 1.76$, $w_v = 2.15$, $w_o = 2.9$, $w_c = 1.4$ for photographs. For the middle region, the optimum weights are 1.73, 0.78, 1.35, 0.53 for sketches, and 2.95, 1.68, 3.69, 1.78 for photographs.

In the same way, we can compute the optimum parameters for other type of features and metrics. For the wavelet-based method proposed by Jacobs et al. [14], the optimum weights for sketches and photographs are shown in Tables 1 and 2, respectively.

7.2 Query-by-Synthesis

As explained in the introduction, this system can be thought as a generalized method for relevance feedback, where the synthesized picture is the feedback that the user gives to the system to explain to it what is a relevant match. If we create collages that represent images from the database, we can evaluate the performance of the system by checking the retrieval efficiency when using the collage or synthesized pho-

Table 1 Wavelet weights for sketch input.

b	0	1	2	3	4	5
$w^Y[b]$	0.616	0.014	0.175	0.302	0.490	0.529
$w^I[b]$	3.354	0.041	0.090	0.122	0.472	0.116
$w^Q[b]$	12.141	0.008	0.102	0.175	0.426	0.239

Table 2 Wavelet weights for photograph input.

b	0	1	2	3	4	5
$w^Y[b]$	3.042	0.034	0.206	0.300	0.815	0.538
$w^I[b]$	8.163	0.064	0.166	0.208	0.394	0.406
$w^Q[b]$	38.118	0.046	0.177	0.194	0.567	0.377

Fig. 9 Images for the testing. Images are grouped in three, starting from the input sketch, the synthesized image, and the target photograph, respectively.

tograph as input. For this purpose, we selected 30 sketches from the CCT dataset, and quickly generated collages using each sketch as input. Figure 9 shows 30 triplets of images, each triplet containing the input sketch, the synthesized image, and the target photograph, respectively.

We compare the retrieval performance of using the sketch as input, using typical relevance feedback, and using our collage as input. Relevance feedback is simulated by randomly selecting a relevant image from the 50 first retrieved images of the sketch query. If the target image is present among these, it is selected instead. Then, the weights are changed for photograph input and the system is queried again.

Following the notation of the appendix, a relevant match is any with a label r^{ij} greater than zero. We use the following performance measures, based on the definitions by Müller et al. [32],

- $rel_0, rel_1, \widetilde{Rank}$: rank at which the original photograph

Table 3 Average performance of the different methods.

	HSV	mm	mid	RH	W	W-RF1	W-C
rel_0	1488.8	1094.1	1135	683.7	530.7	2535.4	407.9
rel_1	100.1	7.27	19.43	5.3	12.67	55.17	10
$\widehat{Rank}$	0.39	0.35	0.33	0.32	0.33	0.45	0.29
$P(20)$	0.18	0.25	0.28	0.34	0.31	0.21	0.42
$R_P(.5)$	0.05	0.03	0.07	0.11	0.05	0.06	0.14
π_f^0	0.07	0.13	0.07	0.17	0.37	0.04	0.27
π_f	0.63	0.87	0.77	0.97	0.90	0.53	0.83

is retrieved, the first relevant image is retrieved, and normalized average rank of relevant images, respectively. The smaller the average rank, the better. For random retrieval, it tends to 0.5.

- $P(20)$: precision after 20 images are retrieved;
- $R_P(.5)$: recall at precision .5;
- π_f^0, π_f : the probability of quickly finding the original picture, or any relevant image, respectively. We compute this by counting the number of queries such that $rel_0 \leq 20$, and $rel_1 \leq 20$, respectively, divided by the total number of queries;
- *PR-graph*: precision vs. recall graph;

Table 3 shows the average performance for all the queries. Using the sketches as input, region histograms (RH) are compared with the HSV color histogram (HSV) used in VisualSeek [33], with the fuzzy query (mm) of Blobworld [6], with the reference region at the middle of the picture (mid), and with the wavelet coefficients (W) of Jacobs et al. metric [14]. When using photographs as input, wavelets perform slightly better than other methods, so we selected wavelets to compare typical relevance feedback (RF1) with the use of synthesized photographs (C) as feedback.

When using sketches as input, with our region histograms it is highly probable (97%) to find something relevant in all cases. The precision is also higher, as shown in Fig. 10. The relevancy of each feature in the region histogram is directly proportional to the value of the obtained weights when tuning the metric, that is, from more to less relevant: color, position, size, and shape. As mentioned in Sect. 4.3, region histograms are more appropriate for query-by-sketch because they loosely try to keep the geometric relations of the segments in the image. The sketch drawn by the user can be thought as a properly segmented image, and the query tries to find similarly segmented photographs. When the input is changed to a photograph or a collage, the segmentation of the input may fail in some parts of the picture, and the errors are propagated to the match. Therefore, region histograms do not perform so well with photograph input.

The initial result using the sketch input and region histograms is even better than using simple relevance feedback (RF). This apparently unexpected behavior of RF is normal if we think that the sketch is a human abstraction that contains the most important elements to be retrieved. When we click on a relevant image, on the other hand, we are including everything that is in the picture, whether relevant or not.

Fig. 10 PR-Graphs using sketch inputs, for region histograms (RH), HSV color histograms (HSV), wavelets (W), fuzzy query (mm), and middle region (mid). **Fig. 11** PR-Graphs using sketches, relevance feedback (RF1), and collages as input queries.

Using collages as input doubles the performance of RF, as shown in the graph of Fig. 11. While the use of collages does not increase the probability of quickly finding a relevant picture, it increases the probability of finding the original target image and highly increases precision. This shows that collages are an important aid towards bridging the intention gap discussed in the introduction.

7.3 Collage Taxonomy by Example

In previous section, we showed examples of synthesized photographs. These look real enough to be matched to something existing in the database. But our system can be used to create completely new examples as well. For the examples below, synthesis becomes the only solution to match user's intention. These examples can be used as a taxonomy of the different kind of images that can be created with sketch-to-collage. For this experiment, we added 3928 images from a private collection to the CCT dataset.

Cornucopia. Figure 12 contains two examples of collages made from a wide variety of objects. The collage is created by drawing a sketch that expresses a certain color composition or design. The system continuously retrieves

Fig. 12 Cornucopia.

Fig. 13 Tribute to Arcimboldo.

images with every stroke. After drawing, the user just needs to click on the retrieved objects, and they are drawn in the canvas in a similar position. In Fig. 13, the user sets w_p to zero, and freely positions the retrieved parts. Several sketches are used to retrieve each object. Poisson image editing is used to modify the colors of certain objects. For example, to make the red objects on the frame darker, or to blend the redness of the fish with the color of the face.

Trip summary. While in previous example, we mainly used the object retrieval –middle window in Fig. 2– to generate the composite, Figure 1 (1) illustrates an example of using image retrieval –bottom window in Fig. 2–. A single sketch is used to retrieve similar images, that are composed together using band optimization, to form a *landscape ring*. To the resulting collage, several kinds of food from the private image collection are added, as shown in the following steps of the figure. The collage can be used as a summarization of a trip, for instance.

Zoetrope. After drawing a landscape ring, we can clear the canvas and draw more sketches to find particular images,

Fig. 14 Zoetrope.

Fig. 15 “Maria meets Amaterasu.”

if we are familiar with the image collection. This is shown in Fig. 14. In this example, the same image is repeated several times in the background, and only the foreground object changes. We use the lasso –marked as red dots– to select the region of interest. For the portrait on the right, Poisson with gradient import has been used, and the one in the middle is set to $\beta = 1.5$.

Progressive portrait. In the example of Fig. 15, we use a single sketch but drawn in a particular order. First, we draw a background that slightly resembles a beach, to retrieve images of beaches. We select one, and then paint someone in the beach, to retrieve images of beaches with people in the center. We add the girl to the night beach using Poisson blending. Then, we add the foreground objects by adding the additional elements and retrieving objects. The boundary of some objects are adjusted with the dotted red lasso, to contain additional segments.

Night & Day. Figure 16 is an example of semantic contrast: we can easily mix night and day. Automatic techniques try to minimize semantic contrast by retrieving similar images, but the user may just want the opposite, as in the example. The collage in Fig. 17 has been created using the same basic idea, but restricting queries with the keyword ‘animal’, using the images of MSRCORID, to place “tourists” in that famous spot.

More examples and videos showing how to interact with the system can be found online [19].

Fig. 16 Tribute to René Magritte.

Fig. 17 Setting poppy.

7.4 Usefulness & Usability Survey

As opposed to the example in “query-by-synthesis”, for the collages created in the previous section, we do not have a direct measure of relevance. Because of this, we performed a survey [19] in order to evaluate the usefulness and usability of our system. Users consider the application useful, easy to use, and an aid to creativity. Most people considered the application responsive, especially the retrieval part. Query-by-sketch is regarded as the best feature by professionals, while regular users usually selected segmentation. The results generated with this system were in general preferred to those of automatic collaging systems. While systems like Digital Tapestry [10] or AutoCollage [2] are powerful tools for summarization, there are several non-trivial tasks that are not possible without interaction. The most obvious one is the selection of favorite pictures. When the collection is small, is easy to browse and select them, but for bigger collections, query-by-example becomes a powerful tool. Other non-trivial tasks include relocation of photographs, duplication of objects, selection of secondary objects, layering objects such as windows, or symmetrization. Figure 18 can be used to summarize these problems. The output for automatic systems is always fixed, while with our system the possibilities are endless. Notice also the appearance of undesired artifacts: in Digital Tapestry, the cut back of a horse, a piece of sky in the middle, a big empty window with no context; in Autocollage, a window showing the sky on the bottom of the collage, a big wall in the middle. With a few clicks, in our examples we can easily get rid of that wall,

Fig. 18 From left to right, up-down: Photopile, Digital Tapestry, Auto-collage, and three collages created interactively with Sketch-To-Collage.

duplicate objects such as the shield, make sure that all the objects are complete, or layer objects such as the windows. Note that our collages are at a lower resolution, since they have been created with thumbnail versions (160×120) of the original images, due to the lack of the original sources.

8. Conclusions and Future Work

In this paper, we have introduced an interactive way of making collages based on a query-by-sketch interface. Our tool works with rough color strokes that represent the idea of the final collage. By using image retrieval, the user does not need to explore an entire image collection but just a subset of relevant objects. Moreover, creativity is added to the process of making a collage, something that was not possible in collages made with automatic summarization techniques. A sketch is a form of human-made abstraction that helps us bridge the representation gap in retrieval systems, while sketch-to-collage helps us bridge the intention gap. We have shown that the synthesized image is more relevant than any of the ones found with common relevance feedback methods.

The query-by-sketch method can be extended in several ways. The painting tool could be improved by using textured brushes. By painting with not only colors but also textures, higher retrieval precision can be expected for certain types of objects, such as grass or animal fur. Since every segment in the database is represented by a color Gaussian, the color variations introduced by textures can be encoded in the PDF of a segment without the need of additional texture features. However, for recognizing the orientation of these textures, we should add additional texture coefficients to the database, such as the result of multi-band pass filters.

The retrieval process can also be improved by studying

the relations of strokes in time and space. When drawing a person, for instance, the program could be aware that one stroke (e.g. an arm) connected to another (e.g. the body), was drawn just after the other, so these two regions should be related, and images containing these kinds of relations ranked higher.

Boundary optimization is still far from perfect. For strongly textured backgrounds, the computed boundary may be optimum for Poisson color blending, but we import undesired textures into the target collage, as the graffiti surrounding the hair of the girl in the right-most portrait in Fig. 14. A possible solution may be to flatten the textures between the selection and the computed object boundary.

Finally, the sketch-retrieve-composite idea can be extended to other media for contents creation. For instance, music creation: the user mumbles pieces of melodies, then similar music samples are retrieved from a music database, and the bits of melodies are merged together seamlessly in a new melody. The same could be applied for videos or other types of contents.

References

- [1] D. McKay, S.J. Cunningham, and K. Thomson, "Exploring the user experience through collage," Proc. 7th Int. Conf., NZ Chapter of the ACM's Special Interest Group on Human-Computer Interaction (SIGCHI-NZ), pp.109–115, 2006.
- [2] C. Rother, L. Bordeaux, Y. Hamadi, and A. Blake, "Autocollage," ACM Trans. Graph., vol.25, no.3, pp.847–852, 2006.
- [3] J. Wang, L. Quan, J. Sun, X. Tang, and H.Y. Shum, "Picture collage," CVPR (1), pp.347–354, IEEE Computer Society, 2006.
- [4] J. Fogarty, J. Forlizzi, and S.E. Hudson, "Aesthetic information collages: Generating decorative displays that contain information," UIST '01: Proc. 14th ACM Symp. on User Interf. Soft. and Tech., pp.141–150, New York, NY, USA, 2001.
- [5] K. Barnard, P. Duygulu, and D.A. Forsyth, "Clustering art," CVPR (2), pp.434–441, IEEE Computer Society, 2001.
- [6] C. Carson, M. Thomas, S. Belongie, J.M. Hellerstein, and J. Malik, "Blobworld: Image segmentation using expectation-maximization and its application to image querying," IEEE Trans. Pattern Anal. Mach. Intell., vol.24, no.8, pp.1026–1038, Aug. 2002.
- [7] A.B. Benitez and S.F. Chang, "Image classification using multimedia knowledge networks," Proc. IEEE Int'l Conf. on Img. Processing, Barcelona, Spain, Sept. 2003.
- [8] A. Mojsilović and B. Rogowitz, "Semantic metric for image library exploration," IEEE Trans. Multimed., vol.6, no.6, pp.828–838, Dec. 2004.
- [9] R. Datta, D. Joshi, J. Li, and J.Z. Wang, "Image retrieval: Ideas, influences, and trends of the new age," ACM Comput. Surv., vol.40, no.2, article 5, p.60, 2008.
- [10] C. Rother, S. Kumar, V. Kolmogorov, and A. Blake, "Digital tapestry," Comput. Vis. Pattern Recognit., vol.1, no.1, pp.589–596, June 2005.
- [11] N. Diakopoulos and I. Essa, "Mediating photo collage authoring," UIST '05: Proc. 18th ACM Symp. on User Interf. Soft. and Tech., pp.183–186, New York, NY, USA, 2005.
- [12] M. Johnson, G. Brostow, J. Shotton, O. Arandjelovic, V. Kwatra, and R. Cipolla, "Semantic photo synthesis," Computer Graphics Forum (EG'06), vol.25, no.3, pp.407–413, 2006.
- [13] M. Chang, J.J. Leggett, R. Furuta, A. Kerne, J.P. Williams, S.A. Burns, and R.G. Bias, "Collection understanding," JCDL '04: Proc. 4th ACM/IEEE-CS Conf. on Digital Libs., pp.334–342, New York, NY, USA, 2004.
- [14] C.E. Jacobs, A. Finkelstein, and D. Salesin, "Fast multiresolution image querying," Proc. ACM SIGGRAPH, pp.277–286, 1995.
- [15] C. Langreitters, "Retriever," 2006. <http://labs.systemone.at/retrievr/>
- [16] R.N. Cabral, "Imgseek," 2005. <http://www.imgseek.net/>
- [17] M. Fonseca, B. Barroso, P. Ribeiro, and J. Jorge, "Sketch-based retrieval of clipart drawings," AVI '04: Proc. Working Conf. on Adv. Visual Interf., pp.429–432, New York, NY, USA, 2004.
- [18] E.D. Sciascio, F.M. Donini, and M. Mongiello, "Spatial layout representation for query-by-sketch content-based image retrieval," Pattern Recognit. Lett., vol.23, no.13, pp.1599–1612, 2002.
- [19] D. Gavilan, "Sketch-to-collage homepage," 2007. <http://www.img.cs.titech.ac.jp/~david/sketch2collage>
- [20] D. Gavilan, H. Takahashi, S. Saito, and M. Nakajima, "Sketch-based evaluation of image segmentations methods," IEICE Trans. Inf. & Syst., vol.E90-D, no.1, pp.156–164, Jan. 2007.
- [21] B. Berlin and P. Kay, Basic Color Terms: Their Universality and Evolution, The David Hume Series of Philosophy and Cognitive Science Reissues, CSLI Publications, 1999.
- [22] D. Gavilan, H. Takahashi, and M. Nakajima, "Color blobs-based image retrieval in broad domains," Proc. IWAIT'05, pp.115–120, Jeju, Korea, Jan. 2005.
- [23] R.D. Zakia, Perception & Imaging, 2nd ed., Focal Press, 2002.
- [24] E. Vogel, G. Woodman, and S. Luck, "Storage of features, conjunctions, and objects in visual working memory," J. Experimental Psychology: Human Perception & Performance, vol.27, no.1, pp.92–114, Feb. 2001.
- [25] D. Gavilan, "A-sketch-db homepage," 2007. <http://www.img.cs.titech.ac.jp/~david/a-sketch-DB>
- [26] S. Ullman, "Object recognition and segmentation by a fragment-based hierarchy," Trends in Cognitive Sciences, vol.11, no.2, pp.58–64, 2006.
- [27] D. Gavilan, H. Takahashi, S. Saito, and M. Nakajima, "Mobile image retrieval using morphological color segmentation," 3rd Int'l Conf. on Mobile and Ubiquitous Computing, ed. IPSJ, pp.51–58, Oct. 2006.
- [28] D. Gavilan, H. Takahashi, and M. Nakajima, "Image categorization using color blobs in a mobile environment," Computer Graphics Forum (EG 2003), vol.22, no.3, pp.427–432, Sept. 2003.
- [29] E.N. Mortensen and W.A. Barrett, "Intelligent scissors for image composition," SIGGRAPH, pp.191–198, 1995.
- [30] J. Jia, J. Sun, C.K. Tang, and H.Y. Shum, "Drag-and-drop pasting," ACM Trans. Graph., vol.25, no.3, pp.631–637, 2006.
- [31] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," ACM Trans. Graph., vol.22, no.3, pp.313–318, 2003.
- [32] H. Müller, W. Müller, D.M. Squire, S. Marchand-Maillet, and T. Pun, "Performance evaluation in content-based image retrieval: Overview and proposals," Pattern Recogn. Lett., vol.22, no.5, pp.593–601, 2001.
- [33] J.R. Smith and S.F. Chang, "Visualseek: A fully automated content-based image query system," Proc. fourth ACM international conference on Multimedia, ed. ACM, pp.87–98, Boston, Massachusetts, United States, 1997.
- [34] Microsoft Research Cambridge, "Object recognition image database 1.0," July 2005. <http://research.microsoft.com/vision/cambridge/recognition/>
- [35] C.V. Group, "Caltech datasets," <http://www.vision.caltech.edu/html-files/archive.html>
- [36] J.S. Long, Regression Models for Categorical and Limited Dependent Variables, Sage Publications, 1997.
- [37] G. Smyth, "Statbox 4.2," 2003. <http://www.statsci.org/matlab/statbox.html>

Appendix A: The CCT Dataset

In the CCT dataset [25], images are distributed in fold-

ers containing a common label, and subfolders that represent the category. We have extended the MSRCORID dataset [34] with photographs of people, and removed the label “miscellaneous”, because of its ambiguity for defining relevancy. For the category “people”, we added three subcategories: single, multiple, and face. “Single” contains pictures with only one person; “multiple”, several persons; and “face”, frontal portraits from the CALTECH frontal face dataset [35]. “Single” and “multiple” examples are extracted from a personal collection. In total, the dataset contains 4665 photographs. The dataset is accompanied by 268 sketches drawn by 12 different users, that represent some of the photographs. Each sketch in our dataset represents a single photograph. In many cases, however, we do not want to retrieve a particular photograph, but any photograph that resembles what we draw. To sort things out, we can define an ordered relevancy scheme like this one,

$$r^{ij} = \begin{cases} 0 & \text{non-relevant} \\ 1 & \text{if } i, j \text{ have the same root group} \\ 2 & \text{if } i, j \text{ are in the same group} \\ 3 & \text{if } i = j \end{cases}$$

Given a sketch, here i represents the photograph that was used as reference for that sketch, and j , any other photograph in the database. A “root group” is any of the folders in the root directory of the CCT dataset. Two images belong to the same group if they are located exactly in the same folder. With these scheme, we avoid arbitrary relevancy levels even for arbitrarily deep trees.

Appendix B: Weight Tuning

Given a set of example pairs –in our case, sketches and photographs–, we can tune the weights of a given metric so they best fit the categorization ability of the metric. For a 2-class classification problem –here, relevant or non-relevant–, we will also have a single threshold for that metric. Given m observable features, we can use a model of the form,

$$\xi^{ij} = \alpha + \sum_{k=1}^m w_k d_k^{ij} + \varepsilon^{ij}, \quad (\text{A.1})$$

where d_k^{ij} is the distance between image (or sketch) i and image j of their k -th feature, w_k is the weight for that feature, α a constant value, and ε^{ij} an error not directly observable. Let us think for a moment that ξ^{ij} is directly the relevancy r^{ij} defined earlier. For a 2-class problem, that would be either zero or one. We could apply linear regression on all the data and find the weights, but then the result would be biased towards mismatches, since for every single match, the rest of the database is a mismatch. Even if we do not use all the data, linear regression will tend to select just the best feature and discard the rest, if the weights are assumed to be positive.

Instead, let us use a statistics approach. We use an ordered logit model [36], where ξ^{ij} is a latent variable that can

not be directly observed. Instead, we observe the relevancy variable r^{ij} ,

$$r^{ij} = \begin{cases} 0 & \text{if } \xi^{ij} \leq \theta_0 \\ 1 & \text{if } \theta_0 < \xi^{ij} \leq \theta_1 \\ \dots & \dots \\ s-2 & \text{if } \theta_{s-3} < \xi^{ij} \leq \theta_{s-2} \\ s-1 & \text{if } \theta_{s-2} < \xi^{ij} \end{cases}$$

s is the number of relevancy levels, ordered from non-relevant at all, to perfect match. θ_l is the threshold to put a sample in the category of level l or in the next level. Then,

$$\begin{aligned} P(r^{ij} \leq l) &= P(\xi^{ij} \leq \theta_l) = P\left(\alpha + \sum_{k=1}^m w_k d_k^{ij} + \varepsilon^{ij} \leq \theta_l\right) \\ &= P\left(\varepsilon^{ij} \leq \theta_l - \alpha - \sum_{k=1}^m w_k d_k^{ij}\right) \end{aligned} \quad (\text{A.2})$$

Assuming that the errors ε^{ij} are independently distributed according a logistic distribution (similar to the unit-normal), we get the logit model,

$$\text{logit}[P(r^{ij} \leq l)] = \ln \frac{P(r^{ij} \leq l)}{P(r^{ij} > l)} = \theta_l - \alpha - \sum_{k=1}^m w_k d_k^{ij} \quad (\text{A.3})$$

By using the maximum likelihood estimation, we can find the weights $\mathbf{w}_{\text{MLE}}$, given a matrix of distances D and a vector of observations $\mathbf{r}$. Each row of D is a pair (sketch $_i$ × image $_j$); and every column, the distance between each individual feature.

For solving the ordered logit we use a statistics toolbox [37] running under GNU Octave. For a matrix of 1.2 M rows (268 sketches × 4665 photographs) and four weights, assuming that you can hold all the matrix in memory, the optimization process takes 3 seconds on a Dual Core AMD Opteron CPU at 2.4 GHz.

David Gavilan was born in Barcelona, Spain, on November 14, 1978. He received the Higher Eng. degree of Computer Science from the Autonomous University of Barcelona (UAB), Spain in 2001. He was a research student at the Computer Science department, the Graduate School of Information Science and Engineering, Tokyo Institute of Technology, Japan, from April 2002 until March 2003, where he received the Master degree of Computer Science in 2005. He is now candidate for a PhD

degree of Computer Science at the same department. His fields of interest include computer vision, image processing, human perception and image retrieval.

Suguru Saito received a B.Eng. degree in 1994 and a D.Eng. degree in 1999 from Tokyo Institute of Technology, Japan. Since 2004, he has been an associate professor at the department of computer science and engineering, Tokyo Institute of Technology, Japan. His fields of interest are image processing, image reconstruction, computer graphics, color science and human perception.

Masayuki Nakajima received the B.E.E. degree and Dr.Eng. degree from the Tokyo Institute of Technology, Japan, in 1969 and 1975, respectively. Since 1975 he had been with the department of imaging science and engineering, Tokyo Institute of Technology, Yokohama, Japan. He is now a professor at the department of computer science, the faculty of graduate school of information science and engineering, Tokyo Institute of Technology, Tokyo, Japan. His fields of interest are computer graphics,

pattern recognition, image processing and virtual reality.