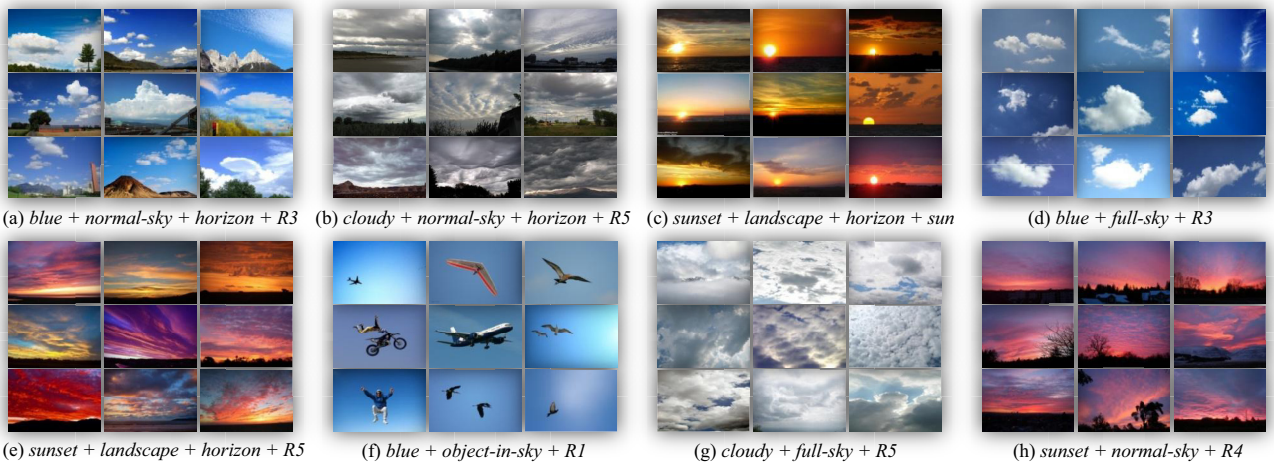
SkyFinder: Attribute-based Sky Image Search

Litian Tao
Beihang University

Lu Yuan
Hong Kong University of Science and Technology

Jian Sun
Microsoft Research Asia



(a) *blue + normal-sky + horizon + R3* (b) *cloudy + normal-sky + horizon + R5* (c) *sunset + landscape + horizon + sun* (d) *blue + full-sky + R3*

(e) *sunset + landscape + horizon + R5* (f) *blue + object-in-sky + R1* (g) *cloudy + full-sky + R5* (h) *sunset + normal-sky + R4*

Figure 1: *Sky search results by specifying a set of semantic attributes (category (blue-sky/cloudy-sky/sunset) + layout (landscape/normal-sky/full-sky/object-in-sky/others) + horizon height + sun position + richness (R1 ∼ R5)) in our search system.*

Abstract. In this paper, we present SkyFinder, an interactive search system of over a half million sky images downloaded from the Internet. Using a set of automatically extracted, semantic *sky attributes* (category, layout, richness, horizon, etc.), the user can find a desired sky image, such as "a landscape with rich clouds at sunset" or "a whole blue sky with white clouds". The system is fully automatic and scalable. It computes all sky attributes offline, then provides an interactive online search engine. Moreover, we build a sky graph based on the sky attributes, so that the user can smoothly explore and find a path within the space of skies. We also show how our system can be used for controllable sky replacement.

1 Introduction

Using a beautiful sky image as the background, or replacing the sky in an existing image, is very common in 2D design, film production, and image editing. The main reason is that many interesting foregrounds or events were taken under a boring (featureless or colorless) sky. Another technical reason is that the high dynamic range of the scene often results in an over-exposed sky.

However, searching for a desired sky image is often a frustrating process. Today's commercial image search engines are based only on text that surrounds an image, which may be inaccurate. The retrieved images using keywords are often noisy, low quality, and disorganized. Page-by-page browsing in photography forums is also an ineffective approach when the number of images is very large. Many content based image retrieval (CBIR) systems can find similar images to a query image. However, providing a good query image is also a search problem – and a difficult one for the user.

In this paper, we present SkyFinder, an attribute-based sky image search system, with over a half million sky images downloaded from the Internet. In an offline indexing process, a set of semantic *sky attributes* (e.g., category, layout, richness, horizon, sun position) are automatically extracted from each image. Then in an online search, the user can *interactively* search sky images based on any combination of preferred sky attributes shown in Figure 1. For example, the query may be "a landscape at sunset with the sun on the bottom left" (Figure 1(c)), "a sky covered with black clouds, the horizon at the very bottom" (Figure 1(b)), or "a clear blue sky with a flying object" (Figure 1(f)). Furthermore, based on the attribute-based search, we build a sky graph to let the user smoothly explore and find transitional "paths" within the space of skies.

A key contribution of our work is that we pose a difficult, content based image search problem as a simple attribute based "text" search problem. We present a complete system (automatic offline + interactive online) to allow the user to perform the sky image search at the semantic level. Our system includes three *novel* building blocks: a set of effective sky attributes design and automatic extraction techniques, an intuitive user interface for the attribute based search, and an effective path finding algorithm in the sky space. Because the searching is based on a set of discrete attributes, the whole system is also easy-to-scale. In this paper, we also demonstrate how to aid the user to effortlessly replace the sky in an existing image using our system.

1.1 Related works

Unlike many CBIR approaches which require a query image (or a hand drawing [Jacobs et al. 1995]), the user can directly start from abstract attributes in our system. For a comprehensive literature review of CBIR, please refer to [Datta et al. 2008]. Here, we review related works on large image collections in computer graphics.

Leveraging a large image collection has been demonstrated as a powerful way to address many difficult problems. For example, Scene Completion [Hays and Efros 2007] fills holes in a photograph using elements taken from semantically similar scenes; Face Swapping [Bitouk et al. 2008] replaces a human face by similar faces

Figure 2: *Training examples.*

ranked in terms of pose, color, lighting and blending cost. Photo Tourism [Snavely et al. 2006] enables the user to explore photo collections at the same physical location using 3D information. An extended system [Snavely et al. 2008] finds optimal paths between views so that the user has a fluid, 6DOF navigation capability. In [Matusik et al. 2005], the user is allowed to continuously navigate in a collection of texture images.

Two works most related to ours are Semantic Photo Synthesis [Johnson et al. 2006] and Photo Clip Art [Lalonde et al. 2008a]. Semantic Photo Synthesis synthesizes new images by stitching several retrieved images. In their retrieval system, each image region has only a single category label (e. g. sky, water, and road). Thus, the user can only find a number of sky images without any controls. On the contrary, our system extracts multiple semantic attributes (e. g. richness, layout, and horizon) from a sky region. The user has a greater flexibility on controlling the retrieval results by combining semantic attributes. Photo Clip Art inserts foreground objects into a photo by searching through a large image object collection. Their search part adopts an example-based search system using color-histogram. In contrast, the user can use our system to directly search a desired sky image without providing a query example. Moreover, the color-histogram based retrieval on a half million images is impractical for an online system; our attribute-based search system is much more efficient and can provide interactive experience on a very large image collection.

Our work is inspired from a very recent face search system [Kumar et al. 2008] in which face attributes are pre-computed. In our system, we automatically extract the attributes of sky images. As an important and attractive element, sky images have also been studied in other fields such as capturing and simulation [Stumpfel et al. 2004; Lalonde et al. 2008b].

2 Sky Attributes

In this section, we describe how we create a large sky image collection and extract sky attributes.

2.1 Data collection

We collect sky images from Flickr.com. We first search for user groups (http://www.flickr.com/groups/) using five keywords "sky", "cloud", "sunset","sunrise", and "storm". Then, we download the images from a number of large user groups (each group contains more than $2,000$ images). It took several days to obtain 1.3 million images from 95 user groups such as, "colourskies", "southfloridasky", "red-sky-at-night", and so on.

Training data. We randomly pick a number of images from the downloaded data as training examples for the later categorization tasks. For each selected image, we manually label it as one of three categories: ***blue-sky*** (blue sky with white clouds), ***cloudy-sky*** (non-blue sky with grey clouds), and ***sunset*** (red or reddish-yellow sky with dark foreground). Images that cannot be assigned with confidence are disregarded. Figure 2 shows typical examples. For each labeled sky image, we use an interactive image cutout tool [Li et al. 2004] to separate sky and non-sky regions. The obtained *sky region map* will be used to train an automatic sky region segmentation method (described below). The final training data consists of 500 blue-sky images, 700 cloudy-sky images, 800 sunset images, associated with sky region maps.

Figure 3: *Layout. From left to right: full-sky, object-in-sky, landscape, normal-sky, and others.*

2.2 Extraction of sky attributes

To characterize a sky image, we extract five kinds of sky attributes. In the attributes extraction stage, all images are resized such that the width and height do not exceed 400 pixels.

1. Category. We train three classifiers to determine the degree of membership of an image to the three categories we previously defined. We represent each image as a "bag-of-words" - a collection of evenly sampled 16x16 patches (sampled at 8-pixel intervals), each assigned to the nearest codeword in a visual codebook. The patch is represented by the concatenation of its SIFT descriptor [Lowe 2004] and mean HSV color. A codebook with 2,500 codewords is learned by performing Randomized Forests algorithm [Moosmann et al. 2006] on 250,000 patches which are randomly sampled from all training images. Please refer to [Dance et al. 2004] for additional details on the bag-of-words model.

Then, three SVM [Vapnik 1995] classifiers are trained for the three categories. For example, the blue-sky classifier is trained using the blue-sky training images as positive examples and the other training images as negative examples. For each image, we apply the three classifiers and then use the three obtained SVM scores as its category attribute.

2. Layout. A sky region map provides rich layout information of a sky image. In the following, we describe how we automatically obtain the sky region from an image and extract the layout attributes.

We first extract the sky region map. Using the training images (with the sky region map), we individually train a sky/non-sky pixel classifier for each of the three categories. Each classifier uses the same visual descriptor we introduced above as the feature, and the patches from the sky/non-sky region within the category as the positive/negative examples. We again choose Randomized Forests [Moosmann et al. 2006] as our classifier due to its efficiency and capability of outputting a soft label ([0,1]) for each pixel. After the pixel level classification, the obtained soft labels are used as the data terms in a graph cut based segmentation [Boykov and Jolly 2001] to produce a binary sky region map.

After the segmentation, we remove the images that have a small sky region ($< 30\%$ of the image area). Our final sky image database contains about 0.5 million images.

Next, we extract the layout attributes. Given the sky region map, we first estimate the line of the horizon, by moving a horizontal line upwards from the bottom of the image, until the number of sky pixels below the line is greater than $0.05A$, where A is the number of sky pixels in the whole image. Then, we categorize a sky image into one of five types: ***full-sky***, in which the sky region covers the whole image; ***object-in-sky***, in which the sky region covers the whole image except for one or more holes that may be due to a flying object such as a bird; ***landscape***, for which 95-100% of the pixels above the horizon are sky pixels; ***normal-sky***, where 75-95% of the pixels above the horizon are sky pixels; and ***others*** for those images that cannot be categorized into the previous four types. Typical examples with different layouts are shown in Figure 3.

3. Horizon height. We discretize the height of the horizon into

eight levels. This attribute gives the user greater control over the layout. It is also useful for automatic sky replacement.

4. Sun existence/position. We detect the existence and position of the sun for sunset and cloudy-sky category but not for blue-sky, since we found that the number of blue-sky images containing the sun is relatively small (less than 1% of blue-sky images) on Flickr.com because people usually avoid capturing and uploading this kind of images which are often saturated or over-exposured. Moreover, we observed that the intensity difference between sun and cloud is larger in CMYK color space for sunset and cloudy-sky than other color space. In the sunset category, we detect the largest connected component whose brightness is greater than a threshold (245 by default) in the magenta (M) channel. In the cloudy-sky category, we perform the same detection in the black (K) channel. If the aspect ratio of the detection region is within the range [0.4, 2.5] and the ratio of region's area to the area of region's bounding box is greater than 0.5 (an empirical description to the shape of visible sun), a sun is detected.

5. Richness. The richness of the sky or clouds can be roughly characterized by the amount of image edges. We use an adaptive linear combination of the edges numbers detected by a Sobel detector and a Canny detector in the sky region, since the Canny detector is good at detecting small scale edges while the Sobel detector is more suitable for middle and large scale edges. Let n_s and n_c be the detected edge numbers by the Sobel detector and the Canny detector. The edge number n of the image is: $n = \kappa \cdot n_s \cdot s(-\frac{n_s-1000}{100}) + n_c \cdot s(\frac{n_s-1000}{100})$, where $s(x) = \frac{1}{1+\exp(-x)}$ is a sigmoid function and κ is a constant parameter to make the edge number of Sobel and Canny comparable (empirically set to 8). The equation indicates that if the edge number by the Sobel detector is small, a more weight is given to the Canny detector and vice versa. Then, we quantize the edge number into five intervals so that the set of images in the database are evenly divided. Finally, each image is assigned to one of the five degrees of richness.

2.3 Quantitative evaluations

To evaluate our attributes extraction algorithms, we performed a study on nearly 6,000 test sky images that are randomly chosen from the downloaded images (different from training dataset). For each image, we manually label it as one of three categories, separate sky and non-sky regions, and identify the sun existence/position. After the labeling, we obtained 2,969 blue-sky images, 969 cloudy-sky images and 1,800 sunset images, in which the percentages of images with suns are respectively $0.9\%, 10.0\%, 25.6\%$.

Table 1 shows the precision/recall (P/R) of the category classification, sky region segmentation, and sun detection. The overall performance is high: the precision/recall are over 85% in most cases. Note that the recall of sun detection in sunset is low because the color and brightness of sun has a very large variance in sunset. The precision of the sky segmentation in cloudy-sky is lower than other categories since the appearance differences of sky/non-sky region (both are gloomy) are smaller than those in blue-sky and sunset.

Table 1: *Quantitative evaluations of our attributes extraction method. The precision/recall (P/R) is measured at the pixel level for the sky segmentation.*

	blue-sky (P/R)	cloudy-sky (P/R)	sunset (P/R)
category classification	99.2% / 96.7%	88.8% / 94.7%	97.3% / 98.2%
sky segmentation	95.2% / 93.6%	88.9% / 96.6%	92.2% / 95.4%
sun detection	- / -	80.8% / 82.5%	91.3% / 72.6%

3 Attribute-Based Search

Using a set of sky attributes makes the search system simple, efficient and scalable. In this section, we present the user interface and search examples of the system.

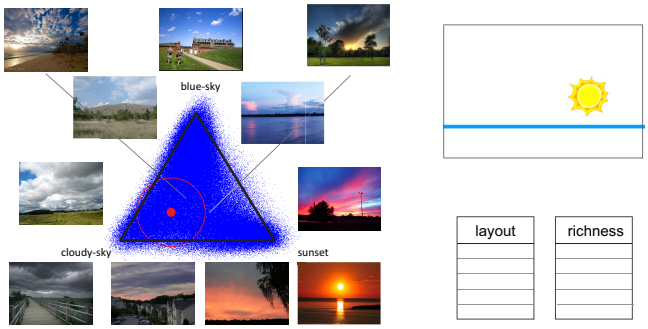


Figure 4: *User interface. Left: (2D) category triangle. The red solid dot is the reference point, and the red circle is the search radius. Right: horizon and sun canvas, and layout and richness controls.*

3.1 User interface

Category triangle. Since each image has three category (SVM) scores, this triple of scores can be viewed as a point in a 3D space. We observed that the set of points from all the images lie approximately on a flat "triangle" in 3D. To provide a simpler interface, we project these points into 2D using principal components analysis [Duda 2001]. Then, we find a minimal-area 2D triangle that contains 90% points using an exhaustive search. The triangle is transformed into an equilateral triangle.

As shown in Figure 4, the equilateral triangle provides a semantic organization of the sky images. When we move from the blue-sky vertex to the sunset vertex, the images *gradually* change from blue sky with white clouds in daytime, to sky with red clouds and a dark foreground at sunset. The images in between tend to have skies before sunset. Similarly, clouds gradually change from white to grey when we move from the blue-sky vertex to the cloudy-sky vertex. The images in the center are usually a mixture of the three categories. We call this equilateral triangle the "category triangle".

In our user interface, we allow the user to place and move a 2D reference point in the triangle. The images are retrieved and ranked in terms of their 2D distance to the reference point. The user can also specify a radius to limit the number of retrieved images.

Horizon and sun canvas. The user can intuitively draw the positions of the horizon and the sun as he likes, as shown in Figure 4. Removing the horizon or the sun from the canvas removes that attribute from the search.

Layout and richness. The user can select layout (five types) and richness (five levels) attributes through two drop-down lists.

3.2 Attribute-based search: an example

Figure 5 gives step-by-step search results by incrementally adding sky attributes: 1) a number of randomly sampled images (Figure 5 (a)); 2) the user places the reference point near the blue-sky vertex. The search results (Figure 5 (b)) are all from the blue-sky category; 3) by selecting a moderate richness, all returned images (Figure 5 (c)) contain moderately rich cloud content; 4) adding the landscape attribute filters out non-landscape images (Figure 5 (d)); 5) setting the horizon height results in a set of more consistent landscapes (Figure 5 (e)).

3.3 Color based re-ranking

Color is an important characteristic in a sky image search. In our system, we represent the color of each sky image as a color signature $s = \{w_k, c_k\}_{k=1}^{K}$, where w_k is a weight, c_k is a color in LAB space, and $K(= 3)$ is the number of color components. The

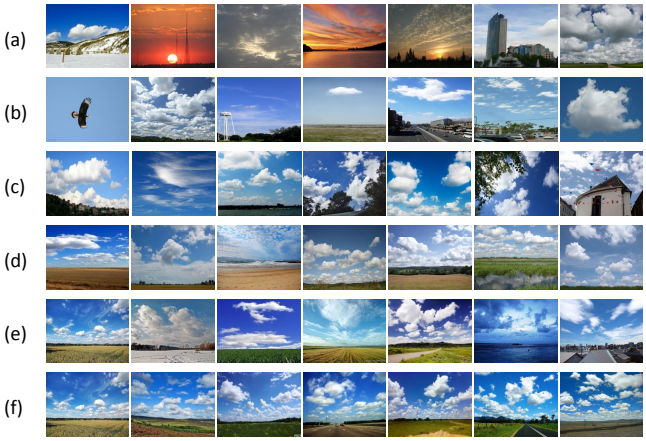


Figure 5: *A search example. (a) randomly retrieved images. (b) blue-sky. (c) blue-sky + richness. (d) blue-sky + richness + landscape. (e) blue-sky + richness + landscape + horizon. (f) color based re-ranking results (using the first image as the query).*

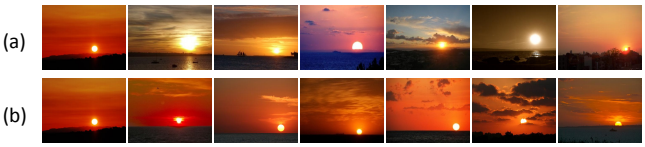


Figure 6: *Another search example. (a) sunset + landscape + horizon + sun position. (b) color based re-ranking results.*

color signature is obtained offline by clustering all pixels in the sky region using the K-means algorithm.

At any time during the search, the user can select an image of interest and find more similar results in terms of the color. The similarity between two signatures is measured by the Earth Mover's Distance (EMD) [Rubner et al. 1998]. The results are ranked based on the EMD distance, as shown in Figure 1(h) and Figure 5(f). The EMD distance computation is very efficient when K is small. In our system, matching one image to 10,000 images takes only 0.5s and the color based re-ranking is usually performed in the last search step on a limited number of candidates. Figure 6 shows another search and re-ranked results using a combination of sunset, landscape, horizon and sun position.

4 Path Search

The user might be able to find a number of sky images that are similar to the wanted result, but do not quite match. In cases such as this, our system identifies a number of intermediate images between two retrieved images to help the user explore the sky space in a continuous manner.

Sky graph. To achieve this goal, we build a sky image graph and find a smooth path between any two nodes. Note that it is intractable to construct a fully connected graph on a half million nodes. Instead, we use the attribute based search to obtain a sparse graph. Specifically, for each image in the database, we first combine the category triangle and the richness to obtain the top $2,000$ matched images. Then, we re-rank these images using color and keep the top 200 images as the neighbors of the selected image. Specifically, we establish an edge between two nodes if they are similar on category, richness, and color. We use the color similarity (based on EMD distance) as the weight of the edge. Using attribute based search makes graph building both effective and efficient. It only took eight hours to build the whole graph on a quad core machine.

Finding a path. To find a smooth path, we compute a min-max cost shortest path between two nodes. Let $\mathbf{p}(\epsilon) = \{e_0, e_1, ..., e_s, ...\}$ be

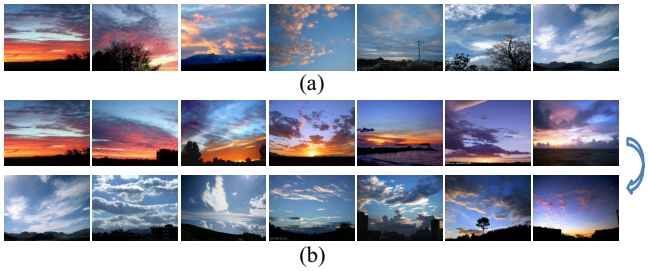


Figure 7: *Path search. (a) results by the shortest path. (b) results by the min-max cost shortest path.*

Figure 8: *Sky replacement results. The top-left in each group is the input image. The user can either find a set of diverse skies by attributes, or find a number of similar skies (in dashed box).*

the shortest path whose max-transition-cost $\max\{e_0, e_1, ..., e_s, ...\}$ is not greater than a value ϵ, where e_s is the edge weight on the path. Our goal is to find the shortest path with minimal max-transition-cost:

$$\mathbf{p}^* = \arg \min_{\mathbf{p}(\epsilon) \in P} \max_{s \in \mathbf{p}(\epsilon)} e_s, \quad (1)$$

where $P = \{p(\epsilon) | \epsilon > 0\}$ contains all shortest paths for various values of ϵ. Because the range of edge weights is limited in our problem, we discretize the value ϵ into 16 levels within the range $[0, 10]$ and perform a binary search to obtain a good approximate solution.

We use the shortest path with minimal max-transition-cost, because we have found in practice that the standard shortest path often does not work well. Without limiting the max-transition-cost, the resulting path, though usually shorter, will contain large jumps. Figure 7 shows a comparison.

5 Sky Replacement

Since the user can effectively search for desired images, we only enforce a weak geometric constraint during the search. Suppose Q and R are regions above the horizon in the query image and the retrieved image. We require that the overlap ratio $\frac{Q \cap R}{Q \cup R}$ be not less than a certain value (typically 0.75). To replace the sky, we simply replace the sky region of the query image with the sky region of the

Figure 9: *Replacement results using a searched path.*

Figure 10: *Inaccurate sky segmentation. Dark-red is the segmentation mask and light-red is the ground-truth.*

retrieved image, by aligning their horizons.

To obtain visually plausible results, we may need to adapt the brightness and color of the foreground to the new sky. Some recoloring techniques [Reinhard et al. 2005] and [Lalonde and Efros 2007] are used to directly transfer the colors from a source image/region to a target image/region to make it appear more realistic. Our sky replacement method adopts a different way to adapt the color of two regions. In our problem, we observed that the correlation of lighting between the sky region and the non-sky region is high in the blue-sky and cloudy-sky categories, and low in the sunset category. Therefore, we apply a category-specific color transfer in HSV space. If the retrieved image belongs to the blue-sky or cloudy-sky category, we compute the color transfer variables (shift and variance multiplier) [Reinhard et al. 2005] between two sky regions and then apply the variables to the non-sky region; if the retrieved image belongs to the sunset category, we directly transfer the color of the source non-sky region in the retrieved image to the target non-sky region.

Figure 8 shows two sky replacement results. By combining different attributes, the user can either find highly diverse results or narrow the search down to a set of similar results. In Figure 9, multiple replacement results are obtained using a searched path.

6 Conclusions

In this paper, we have presented SkyFinder, a sky image search system which allows the user to find preferred sky images using a set of semantic attributes from a half million sky images. The system is fully automatic and easy to scale. It is also very efficient so that the user can interactively search the results.

In the future, we would like to address the several issues: a) improve the sky/non-sky segmentation. Some failure cases of the segmentation are shown in Figure 10. Using higher resolution features may help; b) reduce the errors in horizon estimation. A more sophisticated model [Hoiem et al. 2005] for horizon estimation may improve accuracy; c) add cloud attributes such as cirrus, altocumulus, and cumulus. Texture classification could be used to distinguish these different cloud types.

Acknowledgments We thank the anonymous reviewers for helping us to improve this paper, Stephen Lin for his help in proofreading. We also like to thank the Flickr users who placed their photo under Creative Commons License. This work was performed when Litian Tao and Lu Yuan visited Microsoft Research Asia.

References

BITOUK, D., KUMAR, N., DHILLON, S., BELHUMEUR, P., AND NAYAR, S. K. 2008. Face swapping: automatically replacing faces in photographs. *ACM Trans. Graph. 27*, 3, 1–8.

BOYKOV, Y., AND JOLLY, M. P. 2001. Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. *ICCV I*, 105–112.

DANCE, C., WILLAMOWSKI, J., FAN, L., BRAY, C., AND CSURKA, G. 2004. Visual categorization with bags of keypoint. In *ECCV Workshop on Statistical Learning in Computer Vision*.

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.

DUDA, R., H. P. S. D. 2001. *Pattern Classification (2nd)*. Wiley Press.

HAYS, J., AND EFROS, A. A. 2007. Scene completion using millions of photographs. *ACM Trans. Graph. 26*, 3, 4.

HOIEM, D., EFROS, A. A., AND HEBERT, M. 2005. Automatic photo pop-up. *ACM Trans. Graph. 24*, 3, 577–584.

JACOBS, C. E., FINKELSTEIN, A., AND SALESIN, D. H. 1995. Fast multiresolution image querying. In *SIGGRAPH*, 277–286.

JOHNSON, M., BROSTOW, G., SHOTTON, J., ARANDJELOVIC, O., KWATRA, V., AND CIPOLLA, R. 2006. Semantic photo synthesis. *Computer Graphics Forum*.

KUMAR, N., BELHUMEUR, P., AND NAYAR, S. 2008. Facetracer: A search engine for large collections of images with faces. IV: 340–353.

LALONDE, J.-F., AND EFROS, A. A. 2007. Using color compatibility for assessing image realism. In *ICCV*.

LALONDE, J.-F., HOIEM, D., EFROS, A. A., ROTHER, C., WINN, J., AND CRIMINISI, A. 2008. Photo clip art. *ACM Trans. Graph. 26*, 3, 3.

LALONDE, J.-F., SRINIVASA, NARASIMHAN, G., AND EFROS, A. A. 2008. What does the sky tell us about the camera? In *ECCV*.

LI, Y., SUN, J., TANG, C., AND SHUM, H. 2004. Lazy snapping. *Proceedings of ACM SIGGRAPH*, 303–308.

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

MATUSIK, W., ZWICKER, M., AND DURAND, F. 2005. Texture design using a simplicial complex of morphable textures. *ACM Trans. Graph. 24*, 3, 787–794.

MOOSMANN, F., TRIGGS, B., AND JURIE., F. 2006. Fast discriminative visual codebooks using randomized clustering forests. In *NIPS*.

REINHARD, E., ASHIKHMIN, M., GOOCH, B., AND SHIRLEY, P. 2005. Color transfer between images. *IEEE Computer Graphics and Applications 21*, 5, 34–41.

RUBNER, Y., TOMASI, C., AND GUIBAS, L. 1998. A metric for distributions with applications to image databases. *ICCV*, 59–66.

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 Trans. Graph. 27*, 5, 1–11.

STUMPFEL, J., JONES, A., WENGER, A., TCHOU, C., HAWKINS, T., AND DEBEVEC, P. 2004. Direct hdr capture of the sun and sky. In *AFRIGRAPH*.

VAPNIK, V. 1995. *The Nature of Statistical Learning Theory*. Springer-Verlag.