PhotoSketch: A Sketch Based Image Query and Compositing System

Mathias Eitz[1], Kristian Hildebrand[1], Tamy Boubekeur[2] and Marc Alexa[1]
[1]TU Berlin, [2] Telecom ParisTech & LTCI CNRS

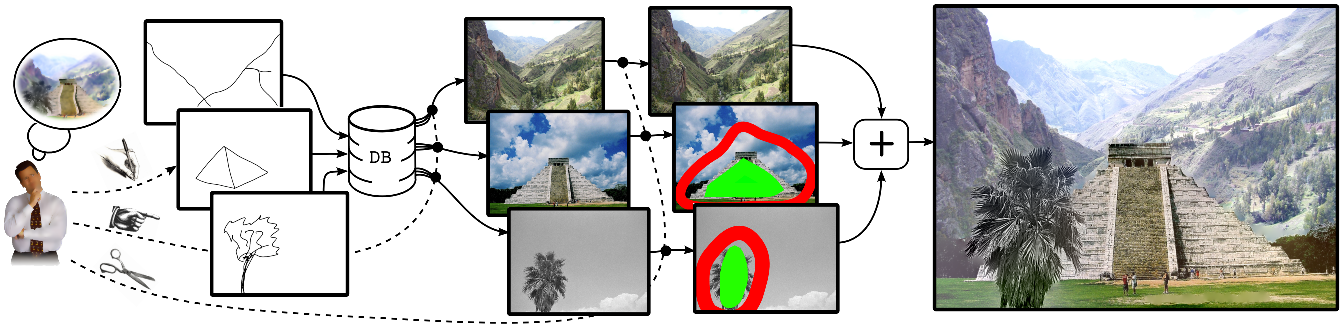

Figure 1: *PhotoSketch: realistic images from user sketches using large image database queries and variational image compositing.*

1 Introduction

We introduce a system for progressively creating images through a simple sketching and compositing interface. A large database of over 1.5 million images is searched for matches to a user's binary outline sketch; the results of this search can be combined interactively to synthesize the desired image. We introduce image descriptors for the task of estimating the difference between images and binary outline sketches. The compositing part is based on graph cut and Poisson blending. We demonstrate that the resulting system allows generating complex images in an intuitive way.

2 Overview, workflow, interface

The aim of PhotoSketch is to provide the user with an intuitive interface for composing new images from parts of existing images. The fundamental principle is a loop, in which the user queries a large database of images for matches to his outline sketch, chooses a query result, extracts a region from the chosen image, and pastes it into the current image (see Figure 1).

A main question is how to query the database. It turns out that sketching binary feature lines is a natural way to quickly communicate the mental concept of a shape [Hirata and Kato 1992]. Clearly, sketching only feature lines allows matching images with different colors: this has the benefit of generating more matches but will also include undesired ones. We compensate the latter problem by clustering the query result so that each cluster represents similarly colored images. In particular, the following steps are performed in one loop of the PhotoSketch image generation process.

Binary outlines are sketched by the user to define the desired *shape* of the content and used to query the large image database. The result of the query for shapes is a small collection of pictures with similar structure but spanning a potentially large range of hues. In our system we typically query for approximately 50 to 100 images. In order to provide the user with a mechanism for quickly finding the correctly colored image in the result set, we cluster the search results according to a color histogram descriptor into a small number of clusters (typically in the order of five to ten). Then the user can select the cluster with the desired color and choose from this cluster the image best matching the shape outlined in the sketch. Once a query result is chosen, the image is rendered half-transparently over the current composition. The user can then transform the new image until the desired part is in the right position. We provide the user with a stroke based interface for quickly marking the desired region and seamlessly merging the selection with the target image [Agarwala et al. 2004]. Using GraphCut, we optimize the selection such that both a seamless composition with the existing background image and an exact selection of the desired image region is achieved. We further reduce possible seams by performing the merging step in the gradient domain. Once the selection is merged into one image, the user continues with sketching the next desired object or part. This loop repeats until the user is satisfied with the result. The workflow is visualized in Figure 1.

3 Sketch-vs-Image Descriptors

Clearly, the main type of information in the user input is the direction of the stroke relative to its position. This information relates best to the direction of gradients. We have designed image descriptors based on cell feature descriptors to address this problem [Eitz et al. 2009]. These descriptors capture the main directions in each part of the image and are computed for all images in the database in an offline process. During interaction, the user sketch provides direction information for each spatial region in the sketch. The descriptor generated from it is then compared against all descriptors in the database and the images corresponding to the best matching descriptors are displayed to the user.

The resulting light weight descriptors (9 Kb per image) can all be loaded into a linear array in main memory. As a consequence, the image queries are fast enough (0.4 to 3.5 seconds on a Apple MacPro, Xeon 2.8 GHz QuadCore, 32 GB Ram) to allow users to *interactively* compose realistic images from millions of pictures.

Compared to existing approaches, e.g. [Gavilan et al. 2007], PhotoSketch enables interactive image creation from *large* databases, avoids the need for presegmented images and provides an intuitive way to define a region of interest on the fly.

References

AGARWALA, A., DONTCHEVA, M., AGRAWALA, M., DRUCKER, S., COLBURN, A., CURLESS, B., SALESIN, D., AND COHEN, M. 2004. Interactive digital photomontage. *ACM Transactions on Graphics (TOG) 23*, 3, 294–302.

EITZ, M., HILDEBRAND, K., BOUBEKEUR, T., AND ALEXA, M. 2009. A descriptor for large scale image retrieval based on sketched feature lines. In *Sketch-Based Interfaces and Modeling, to appear*.

GAVILAN, D., SAITO, S., AND NAKAJIMA, M. 2007. Sketch-to-collage. In *ACM SIGGRAPH 2007 posters*, 35.

HIRATA, K., AND KATO, T. 1992. Query by visual example-content based image retrieval. In *Advances in Database Technology*, Springer, 56–71.

PhotoSketch: A sketch based image query and compositing system
Additional Materials

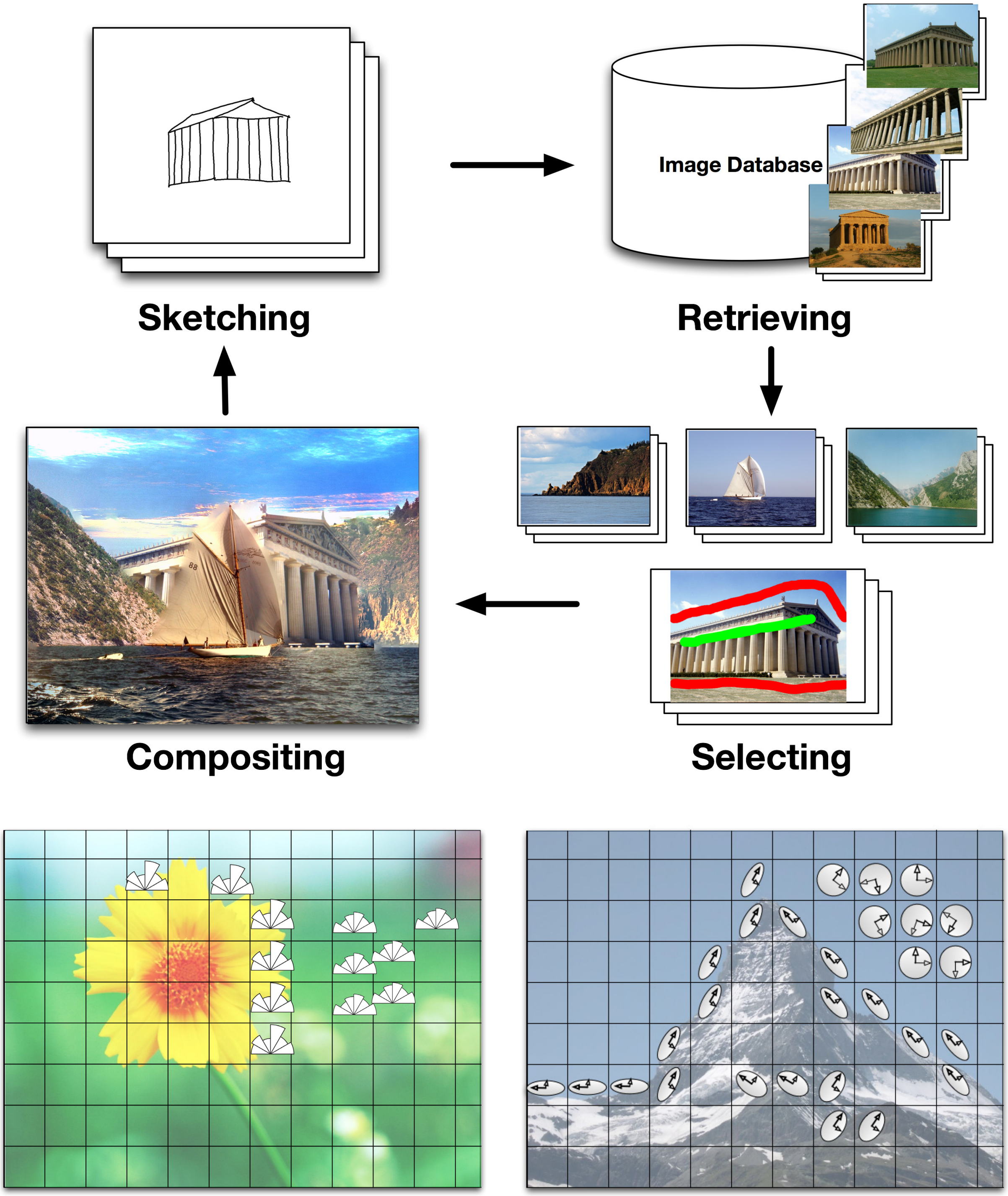

Figure 1: *Top: PhotoSketch workflow. The user starts by sketching a part of the desired content, the system retrieves best matches from a large database of images and presents those to the user. The user selects the best match and uses interactive compositing to create an updated version of the desired image. This process is iterated until the user is satisfied with the composite image. Bottom: Image Descriptors. The edge histogram descriptor stores for each cell the sum of squared gradient magnitudes falling into one of six discrete orientation bins (left). The tensor descriptor subdivides the image into rectangular tiles. For each tile a structure tensor is computed, depicted by the ellipses (right).*

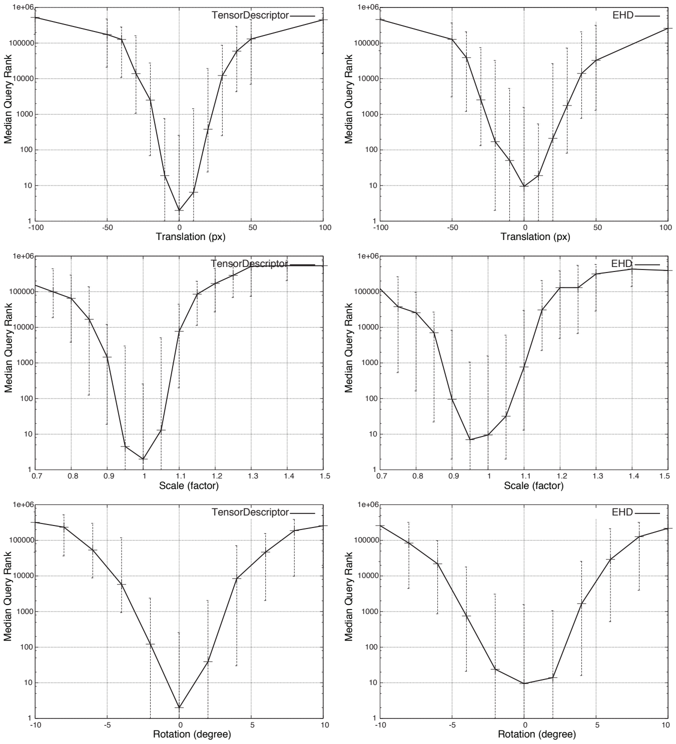

Figure 2: *The results of an objective evaluation of the image descriptors. For 27 images contained in the database of 1.5 million images we have generated queries based on hand drawn sketches. The graphs show the median rank and the upper and lower quartile of the original image over translations, rotations, and scales of the input.*

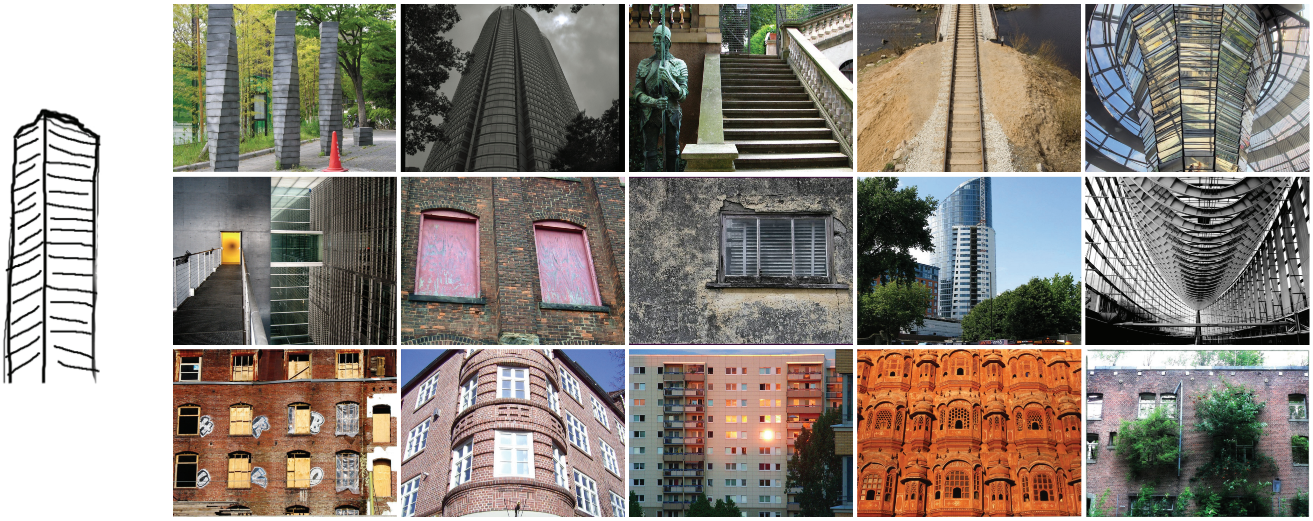

Figure 3: *The first 15 matches (left to right, top to bottom) for the query on the left. The query sketch has been generated from an image in the database, which has been ranked 5474. Note that the first matches provide a very reasonable answer to the user query, meaning that a low rank of the image used to generate the sketch does not imply the descriptor failed.*

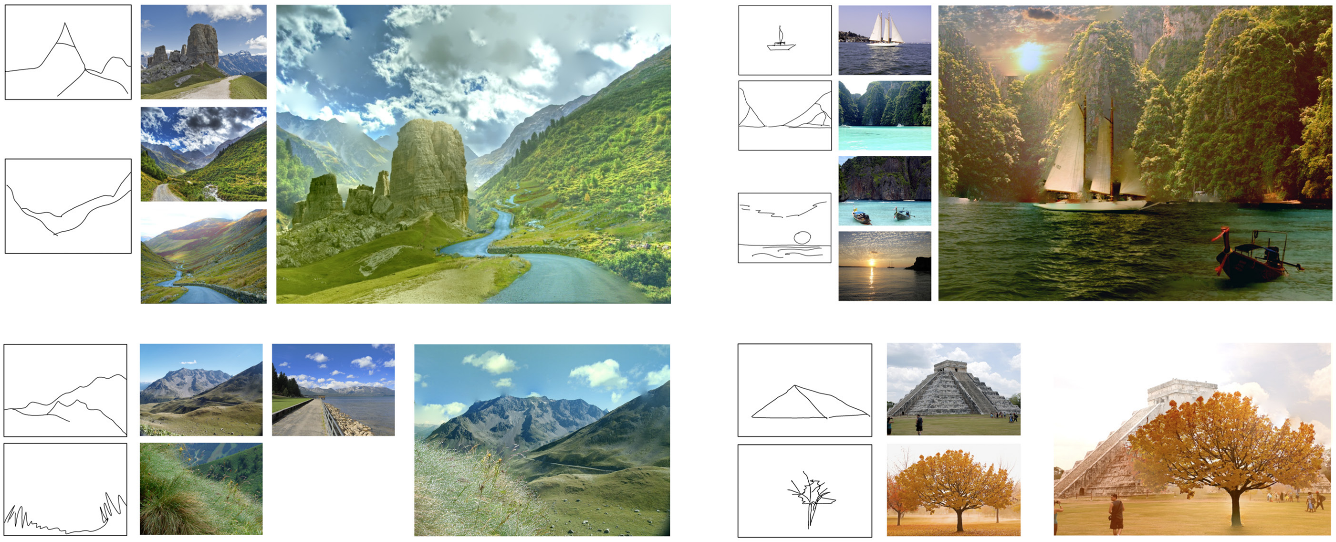

Figure 4: *Some final compositing results created with the PhotoSketch system. The user's input sketches are shown on the left, chosen search results in the middle and the final composite example on the right.*

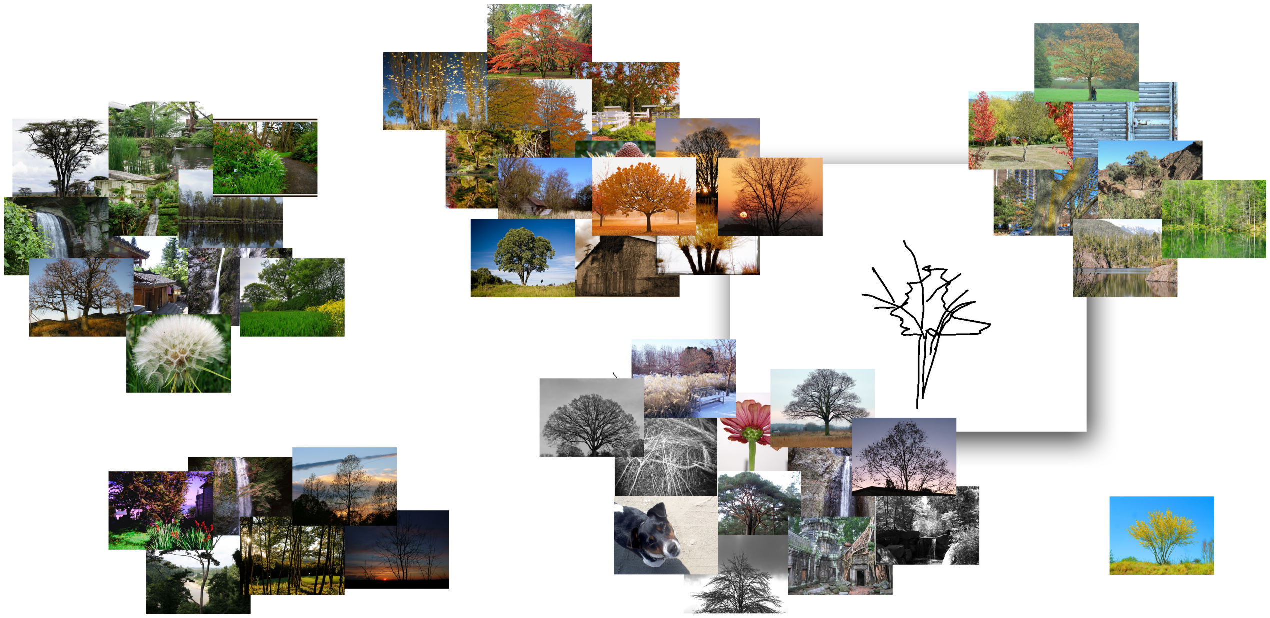

Figure 5: *Answer of the PhotoSketch system to the sketch presented in the middle. The first 50 matches are clustered into 6 clusters. Note that the result set contains a very high percentage of trees as probably desired by the user.*

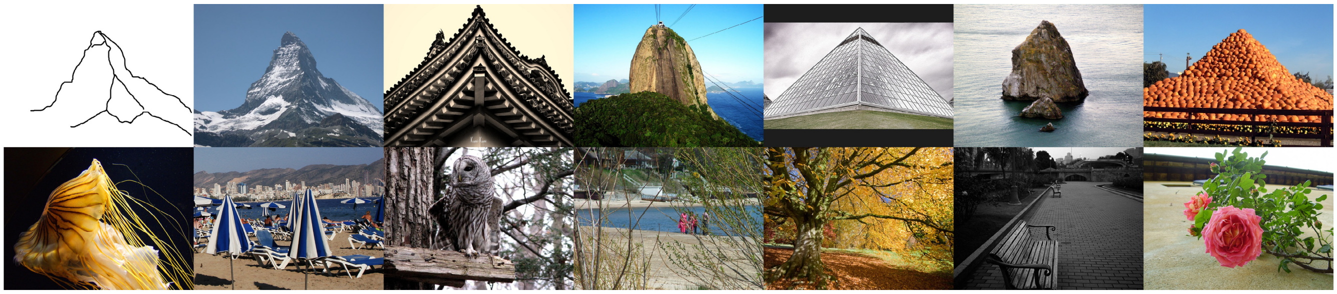

Figure 6: *Shown is a hand-picked subset of the results when querying the database for 100 images matching the sketch on the top left. In the top row we show "expected" results, and rather "unexpected" results in the bottom row.*