Integrating Color, Texture, and Geometry for Image Retrieval

Nicholas R. Howe and Daniel P. Huttenlocher
Department of Computer Science, Cornell University

Abstract

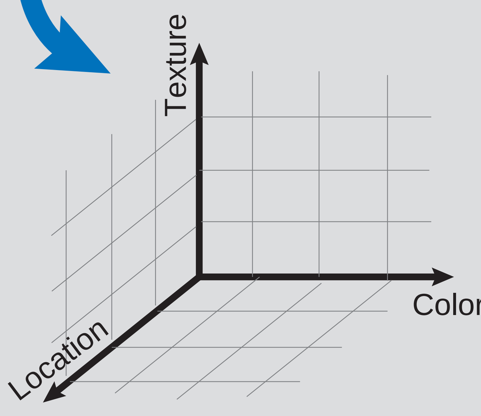
This paper examines the problem of image retrieval from large, heterogeneous image databases. We present a technique that fulfills several needs identifiedby surveying recent research in the field. This technique fairly integratesa diverse and expandable set of image properties (for example, color, texture,and location) in a retrieval framework, and allows end-users substantial controlover their use. We propose a novel set of evaluation methods in addition toapplying established tests for image retrieval; our technique proves competitivewith state-of-the-art methods in these tests and does better on certain tasks. Furthermore, it improves on many standard image retrieval algorithms bysupporting queries based on subsections of images. For certain queries this capability significantly increases the relevance of the images retrieved, andfurther expands the user's control over the retrieval process.

The STAIRS* Engine

A series of transformations converts a raw image into a vector that captures the spatial layout of color and texture in the image. The image is described in terms of its component *image tokens*, or small patches described by their color, texture and location. The joint histogram of these token values forms the representation of the image, which is compared with other images using a modified cosine metric. Using a similarity matrix **S** in the distance equation allows the user to influence the importance of each feature in the final distance metric. By tuning simple parameters, the user can produce an **S** matrix with the desired weighting of color, texture, and location.

Original image scaled to 128x192 pixels

Fast segmentation into ~500 patches.

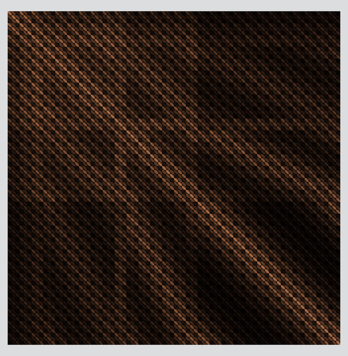


Each piece is binned by color, local texture, and location in image.

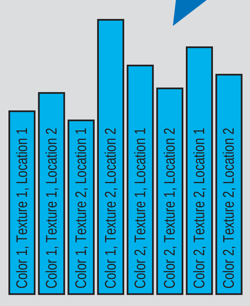
Color similarity

Texture similarity

Location similarity



Overall similarity
(Light = more similar)



Final image representation is joint histogram of segment features, represented as a vector.

Simple parameters allow the user to specify how much credit should be given for imperfectly matching color, texture, and location. Independent similarity matrices for each feature are combined via the Kronecker product to produce the full matrix **S**.

$$\mathbf{h} = \langle h_{111}, h_{112}, h_{113}, \ldots, h_{lmn} \rangle$$

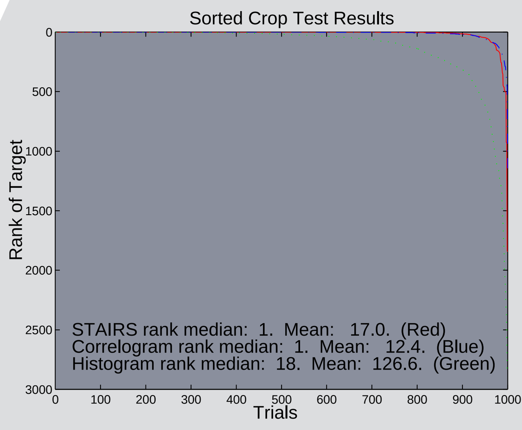
$$D(h_1, h_2) = \cos^{-1}\left(\frac{h_1^T \boldsymbol{S} h_2}{\sqrt{(h_1^T \boldsymbol{S} h_1)(h_1^T \boldsymbol{S} h_2)}} \right)$$

Compare two images based on their histogram vectors, using a modified cosine metric. Matching is speeded by caching of terms in denominator and by using a Kronecker decomposition of S. Further pruning can be obtained by calculating the cosine in a projected space.

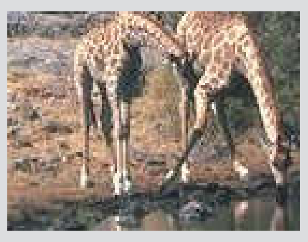
Evaluation

We evaluate STAIRS on two complementary tasks (Classification and Altered-Image queries), in comparison with two alternate algorithms (autocorrelograms and color histograms). The results show better performance than the baseline (histogram) algorithm, and competitive performance with the autocorrelograms. The three algorithms differ somewhat in their areas of strength and weakness.

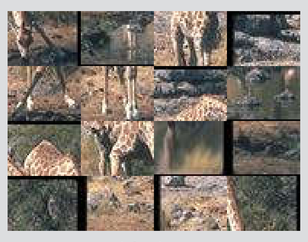
Altered-Image Tests



Original Image

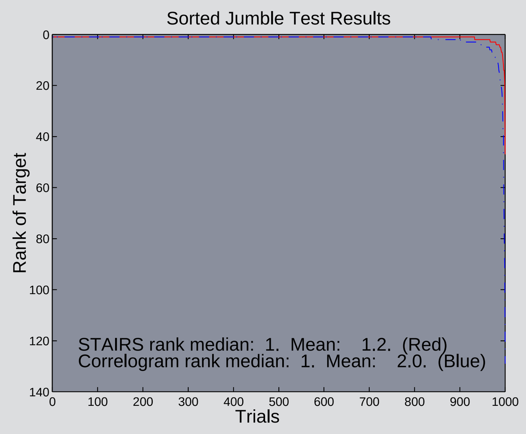


Cropped Image

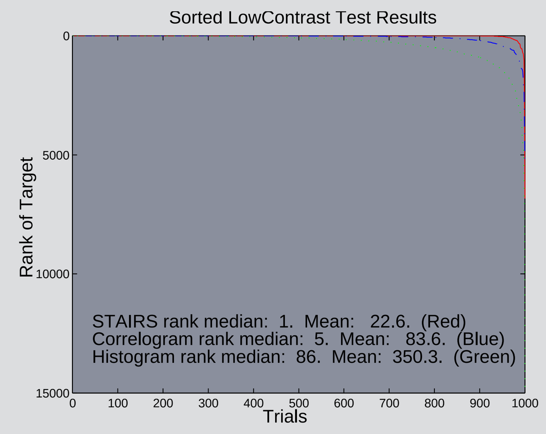


Jumbled Image

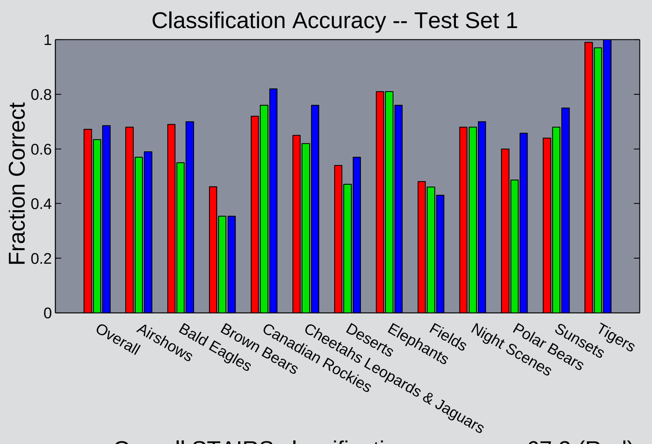
Low-Contrast Image



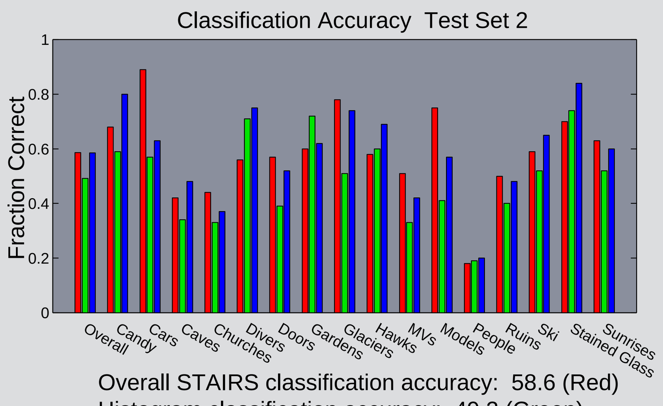
In altered-image tests, the an image is altered algorithmically and used as the query. The goal is to retrieve the original, unaltered image. Plots show the rank at which the target was retrieved in 1000 trials, sorted by performance.



Classification Test



Overall STAIRS classification accuracy: 67.2 (Red)
Histogram classification accuracy: 63.4 (Green)
Correlogram classification accuracy: 68.5 (Blue)



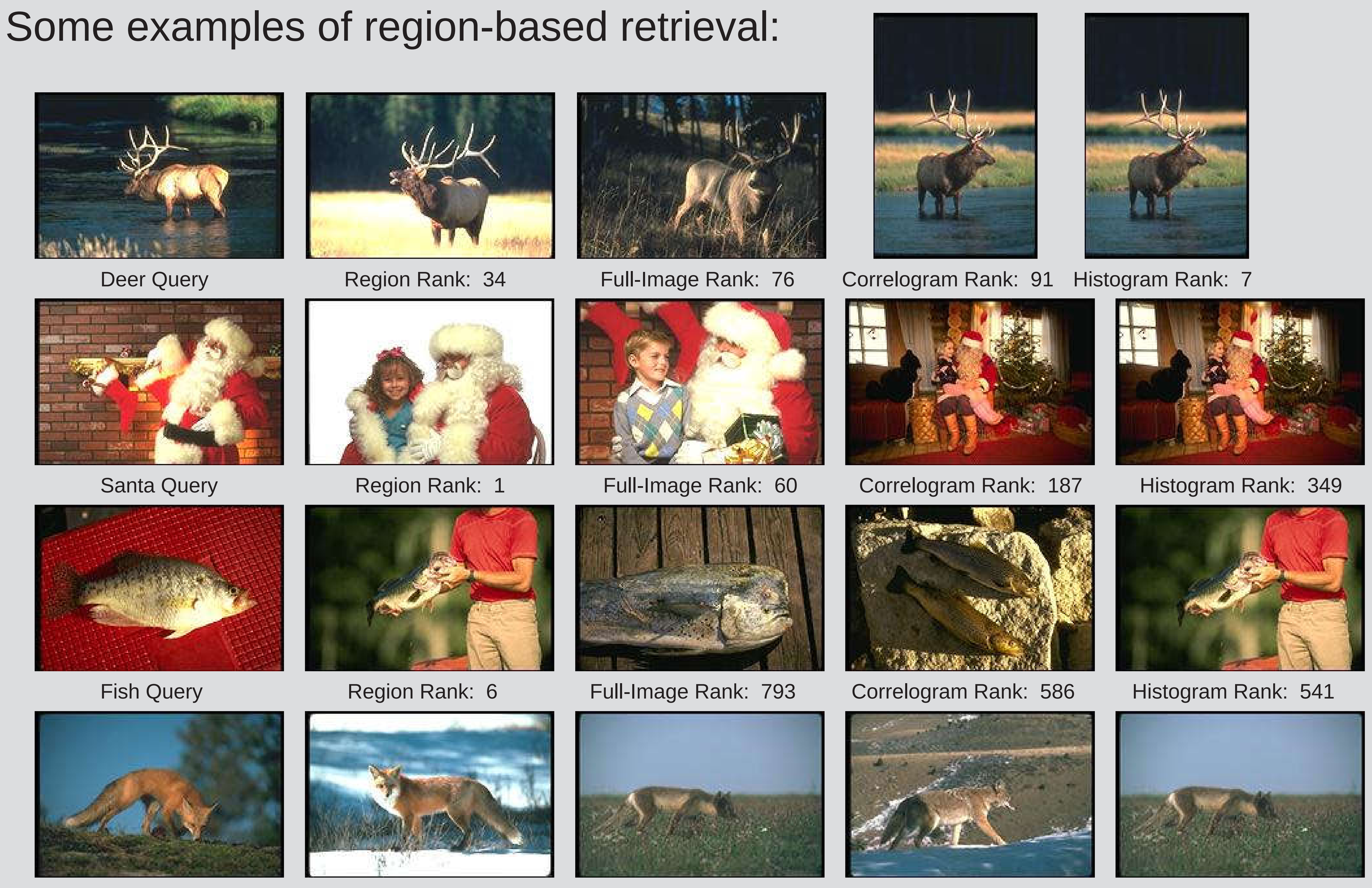
Overall STAIRS classification accuracy: 58.6 (Red)
Histogram classification accuracy: 49.2 (Green)
Correlogram classification accuracy: 58.5 (Blue)

Results based upon leave-one-out cross-validation accuracy.

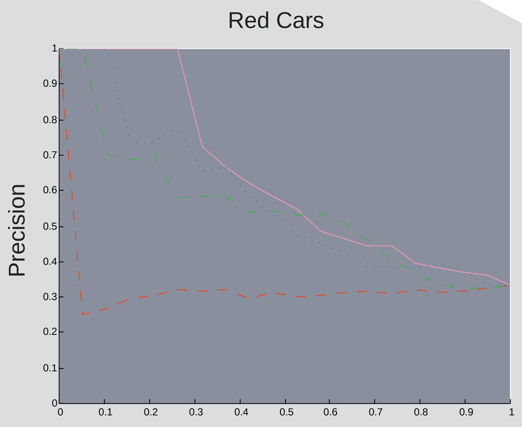
Region-Based Retrieval

Often users are interested in only a portion of an image, perhaps a particular object in a scene containing many others. In suchcases, a query based on the full image will return many false hits due tospurious matches with irrelevant areas of the scene. To solve this problem, an image retrieval system must retrieve images based upon a match of some region in the targetimage with a specified region of the query image. Stairs supports a form of region matching as a special case of a more generalcapability: matching some image tokens more or less strictly than others. Inthis framework, a region query is formed by requiring a close match in the region of interest, while allowing the rest of the image to match anything. Because the image has already been segmented into tokens, any areas that potentially match the target can be identified easily.

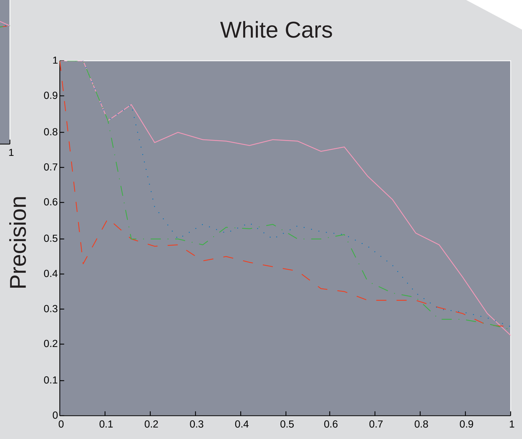
Some examples of region-based retrieval:



Deer Query	Region Rank: 34	Full-Image Rank: 76	Correlogram Rank: 91	Histogram Rank: 7
Santa Query	Region Rank: 1	Full-Image Rank: 60	Correlogram Rank: 187	Histogram Rank: 349
Fish Query	Region Rank: 6	Full-Image Rank: 793	Correlogram Rank: 586	Histogram Rank: 541
Fox Query	Region Rank: 5	Full-Image Rank: 74	Correlogram Rank: 371	Histogram Rank: 728



Recall/Precision curves for a small (200 image) retrieval task show that region-based retrieval can be highly effective (pink curve). Shown for comparison are full-image Stairs (red), Histograms (green), and Correlograms (blue).



Region matching is carried out in the standard Stairs feamework by using a different **S** matrix.

$$S(i,j) = \begin{cases} S_{match}(i,j) & \text{if i and j are tokens in target region,} \\ S_{no\text{-}match}(i,j) & \text{if neither i nor j are tokens in target region,} \\ 0 & \text{otherwise.} \end{cases}$$

*Semantic-Token-Based Automatic Image Retrieval System