Chase Howard
2022-02-06
UWB – Professor Chen
CSS 584 Multimedia Database Systems

Assignment 2: CBIR with Relevancy Feedback

How to run?

1) Locate executable on local machine './app/exe/CBIR-RF.exe'
2) Ensure directory containing executable also contains local image database to analyze
3) Double click executable
4) Wait a while
5) Enjoy!

Note: Screenshots with precision values are in the Appendix as well as in the folder './hw2/app/screenshots/'

How to operate?

This interface can be sub-divided into two separate windows. The initial window able to view upon startup of the application will be referred to as the 'Main Window' – upon searching for similar images, there will be a secondary window which becomes visible to the user which will be referred to as the 'Display Window'.

Upon starting the executable, the Main window will be displayed to the user with a collection of image names populating the list box on the left side of the screen. Images names are populated from local storage in a folder entitled 'images' within the same directory as the executable launches. Here a user can click to select and preview an image contained within the local database.

Once a user has identified an image of interest, they then have the option to view a more detailed view of the image by selecting the 'View Details' button which will open the image in the default viewer within the operating system.

If the user wishes to search for similar images, they have three search methods to choose from – 'Color' 'Intensity', or 'Color_Intensity'. By selecting the Color option, the user will be presented with images in the database most like the previewed image based on the calculated similarity. Intensity will return images with similar intensity histograms generated by flattening each RGB pixel to an intensity value based on the following formula:

$$I = 0.299R + 0.587G + 0.114B$$

Selecting 'Color_Intensity' will use a combination of both aforementioned features along with a relevancy feedback system to improve search performance. Upon searching in this mode, the user will be presented with each return image as well as a selectable checkbox labeled 'Relevant'. By selecting the checkbox under a single image, the user will mark that image as relevant. Once all desired images have been selected, the user can select the button displayed at the top labeled 'Refine Search' to update the search parameters. This will adjust the weight of the available features and retrieve relevant items more accurately based upon the updated weights. This process can be repeated indefinitely; however, once a user goes back to the main window and selects the 'Search' button – all history relating to relevant images will be cleared and the weights will be reset.

Once the user has selected the most appropriate method by which to search, they may select the 'Search' button. This will spawn the Display Window which will show the user a preview of the image that was queried along with the search mode that was used. At the bottom of the window will be a grid of twenty images retrieved from the database ranked in order from most to least similar (left to right, then top to bottom). On this page, the user can navigate up or down pages by selecting the 'Prev' or 'Next' buttons at the bottom of the page.

If desired, a user may update the search parameters (either image or mode) on the Main Window prior to closing the Display Window. The Display window will be updated with the new parameter selected by the user. The user may also close the Display window and reload a new search from the Main Window; however, if the Main Window is closed, the user will need to re-launch the executable if any additional searches are needed. The user may close out of all windows to shut down the GUI.

Comparison to Rui98 et. al.

Similarities?

Early in the introduction, the paper outlines a basic 'Computer Centric' System which involves selecting the feature(s) of interest and manually assigning weights to each feature. This is an even further simplified version of the method implemented (mainly due to the fixed, manually input weight of each feature); however, this approach is similar in many ways and many of the drawbacks discussed still persist in the version implemented. One drawback with this method includes the gap generated between higher level human conceptions as low-level features used to describe each image within the database – e.g., the shape and color may not always accurately describe an image of interest, say an apple which can be red / green / yellow and share similar shapes to that of a vase. Additionally, there is variation between each of the human interpretations of a group of images. One human may deem an image relevant when another person may not under the same search criteria – implying the subjective nature of weighting features within images. The main difference in the basic 'Computer Centric' system involves the manual assignment of weights; the weights are calculated based on the variance across features from manually selected 'relevant' images.

The heart of the article focuses on the idea of adaptive / automatically adjusting weights based on user feedback, extremely like the system implemented in assignment 2. This method still requires the assumption that lower-level features can accurately represent higher level concepts (which isn't always the case); however, it is an improvement as it no longer requires manual setting of weights by the user – removing the need for the user to understand the lower-level features being captures by the CBIR system as well as the level of subjectivity intertwined in that. All the above have been included in the CBIR Relevancy Feedback system implemented in Assignment 2; however, the implementation details differ significantly, especially with respect to the complexity arising from multilevel application of weighting and similarities when analyzing the variance among the features.

Many of the general steps taken during the CBIR system are similar, such as the gaussian normalization of features and weight updating; however, the paper outlines a much more complex and effective method of CBIR through methods described below.

Differences?

The article touches upon previously completed research on more advanced human interaction mechanisms which have been used in the past such as 'interactive region segmentation', 'interactive image database annotation', 'supervised learning techniques', as well as 'interactive integration of keywords and higher-level concepts' – all of which aimed to improve the retrieval system performance through human interaction. The following discussion pertains specifically to the article of interest and will not discuss any of the differing techniques just mentioned.

One early difference in the CBIR system presented the paper by Rui et. al, includes the various different weights which are used. These weights exist at multiple levels, allowing for adaptive control over which weights are emphasized and which features the weights apply to. There is a W_i, which applies to a single feature, W_{ij} which applies to a representation of a feature, or W_{ijk} which applies to components describing a set of representations. Each of these weights are adaptively modified as user feedback is gathered to improve search parameters through multiple levels (bottom up: similarity -> representation -> feature). Additionally, each of these weights are linear combinations of each other, allowing for analysis of higher-level weights based on linear combinations of lower-level weights. This allows for consistent mathematical operations across multiple levels of differentiation between features, representations, and components. This multi-level system is modular and adaptive to capture more relevant user inputs and related them to the retrieval algorithm.

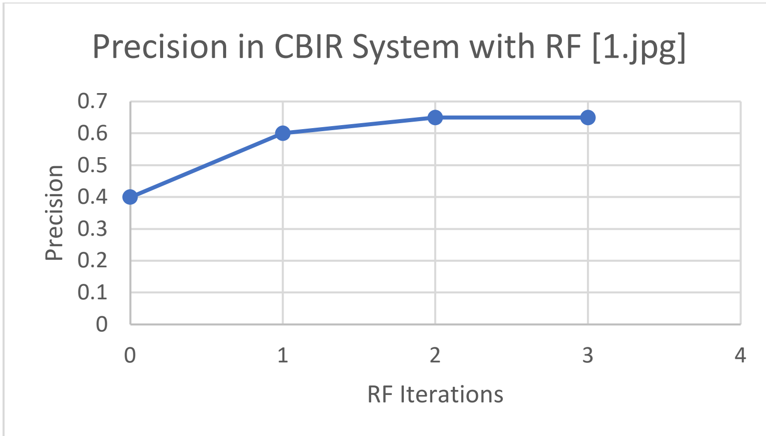
With multiple layers of weights available to be applied, there arises a need for both intra and inter-normalization to retain properties of linear combination of the vector components. Intra-normalization puts equal emphasis on different components within a representation vector while Inter-Normalization applies equal emphases on each similarity value. This allows for different similarity metrics to be combined and compared, allowing for additional flexibility in calculation of similarity. This paper takes an additional step in the normalization process to place 'all' (99%) of values between the range of [0,1] instead of the result of a Gaussian normalization which ranges from [-1, 1].

Additionally, this paper presents a method by which to apply varying similarity or distance metrics (e.g. Euclidean, Histogram Intersection, Manhattan, etc). The variable in the similarity metrics allows for more appropriate calculations of similarity based on the feature, representation, or component being compared by the user. By normalizing each of the similarity metrics, they can be compared against each other. This is an added layer of complexity which wasn't included in the CBIR system developed in Assignment 2.

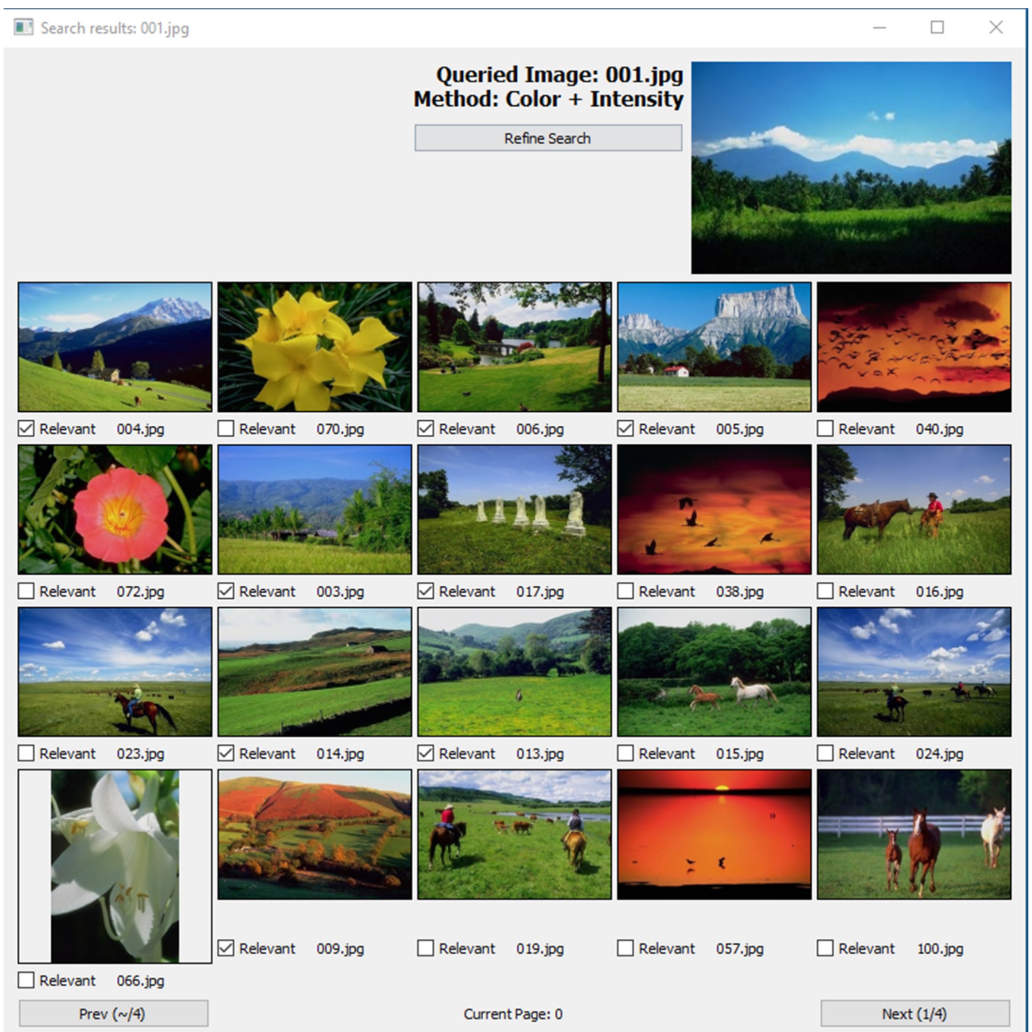
Finally, the user themselves also has the option to rate images relevance with varying levels of relevancy (5 levels). This further refines the adaptive weights applied to each of the images based on a semi-arbitrary score assigned to each of the 5 levels. There is a tradeoff here, as an increased number of levels provides a better feedback system; however, decreases the usability of the system from a user's perspective. These adaptive levels only apply to the images contained within the retrieved set (N_{RT}) and all other images within the database are given a score of zero as to not affect the results.

Appendix A: Supplemental Material

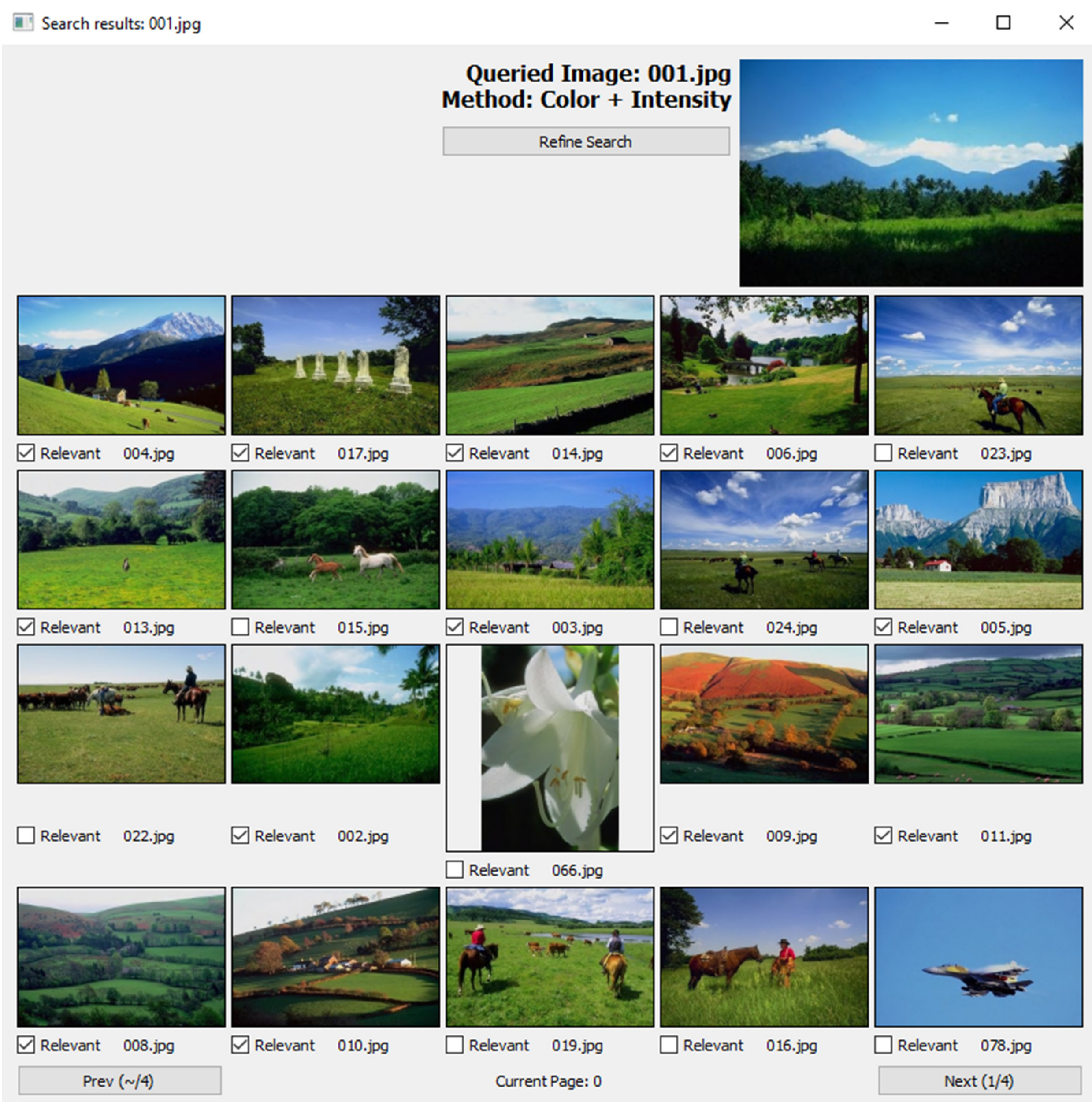
Precision: Note, precision does not count the retrieval of the same image as a True Positive



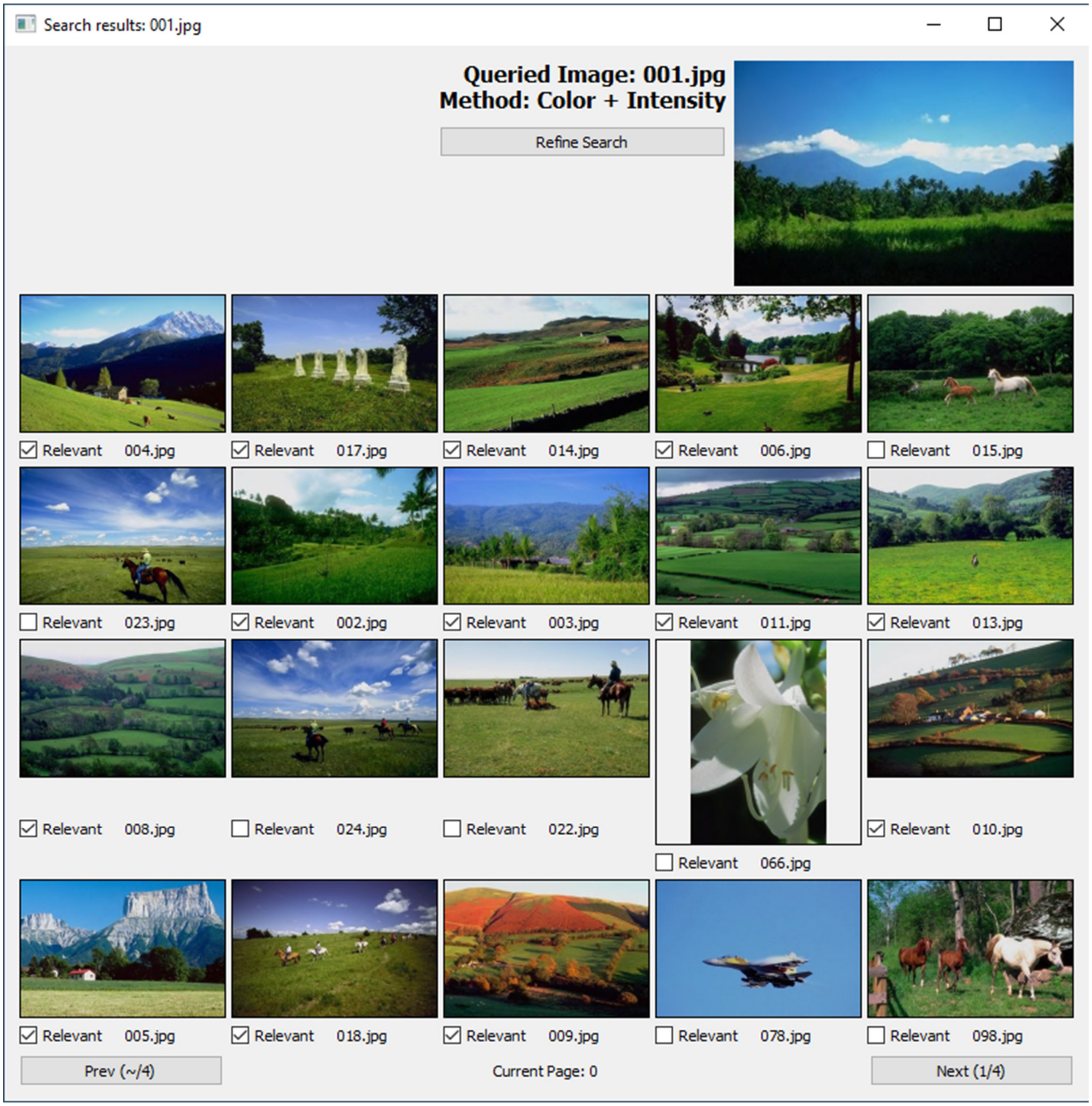
- Image: 1.jpg – Initial Query for Color + Intensity
- Precision [8 / 20] = 0.4



- Image: 1.jpg – 1st round of Relevant Feedback
- Precision [12 / 20] = 0.6



- Image: 1.jpg – 2nd round of Relevant Feedback
- Precision [13 / 20] = 0.65



- Image: 1.jpg – 3rd round of Relevant Feedback
- Precision [13 / 20] = 0.65