ECE 417 MP6
Content-Based Image Retrieval & Relevance Feedback

ECE 417 – Multimedia Signal Processing

University of Illinois at Urbana-Champaign

Spring 2014

Instructor: Professor Hasegawa-Johnson

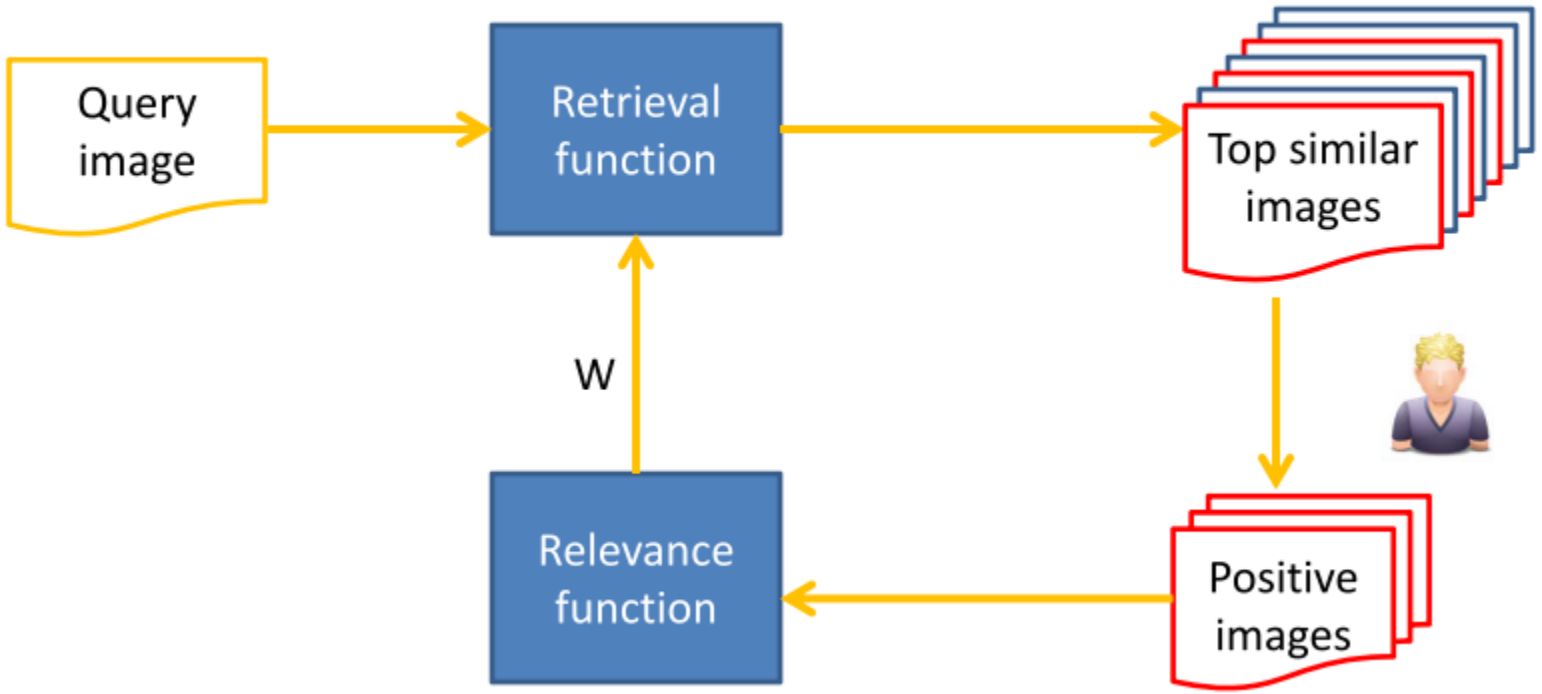
TA's: Daniel Soberal, Po-Sen Huang

Prepared by Po-Sen Huang

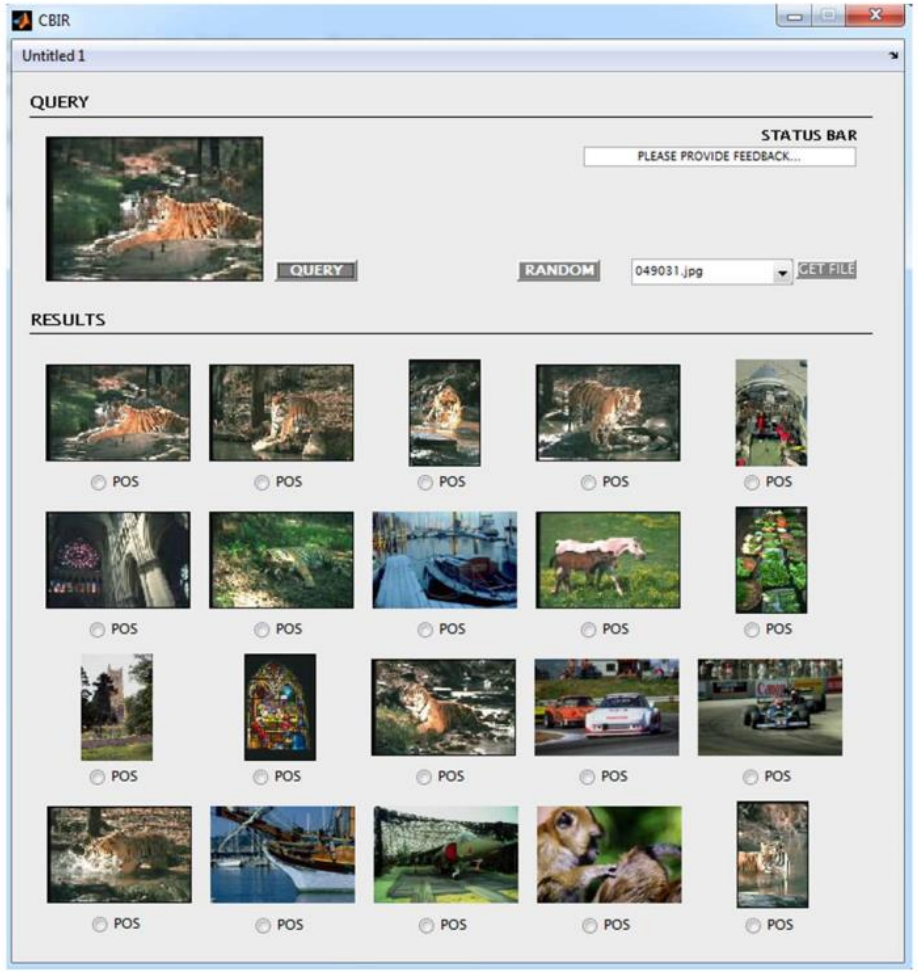
Overview

- Data
  - You are given 1400 images with their 47-dim visual feature (so you do not need to worry about feature extraction)
  - These image features are stored in [handles.META_DATA]
- Tasks
  - Implement the retrieval function for a set of query images (3.1)
  - Implement the relevance feedback function (3.2)

Flow Chart

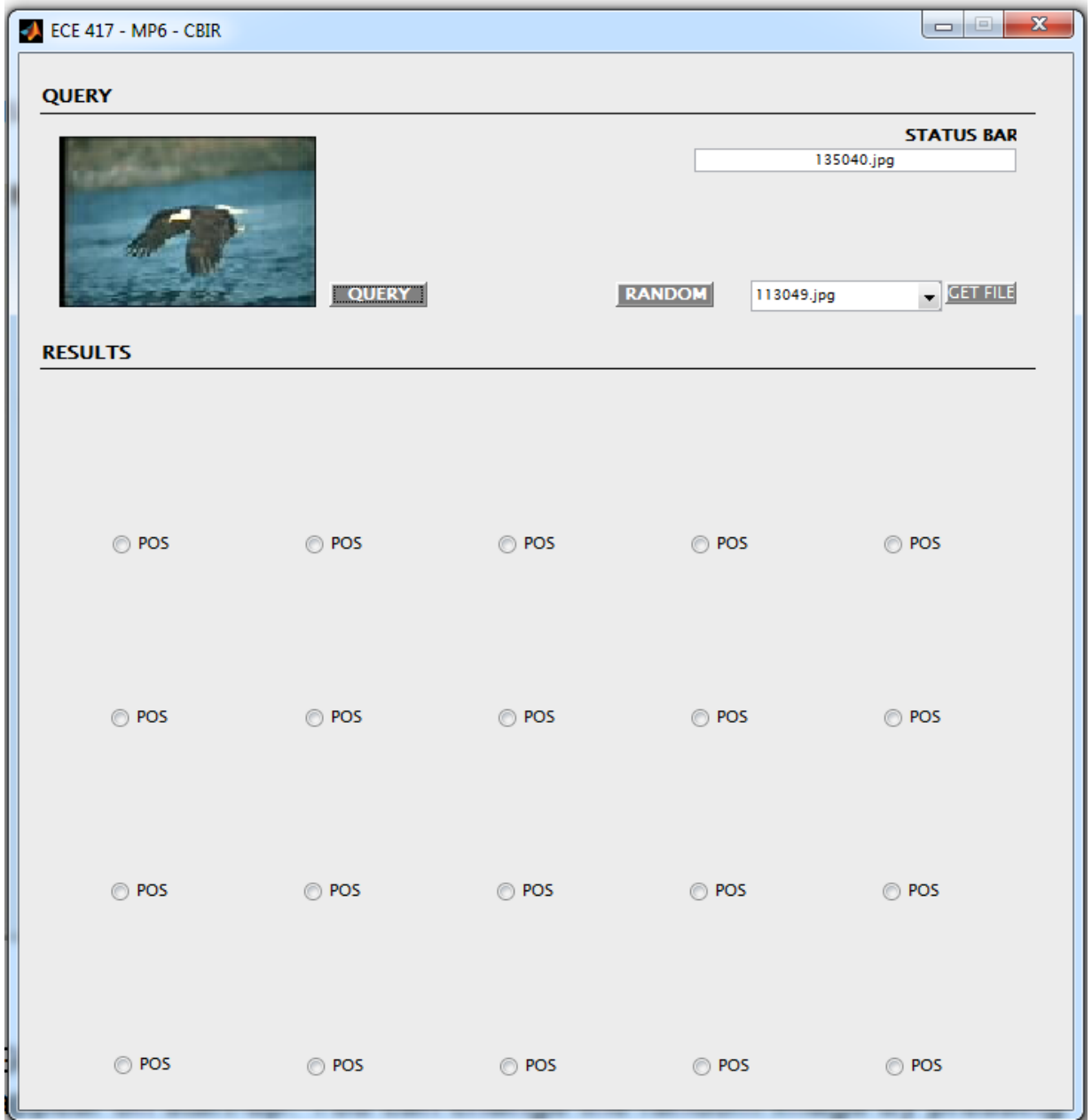

Query image
Retrieval function
Top similar images
W
Relevance function
Positive images

Graphical User Interface (GUI)



Initially

- Run cbirMP
- Buttons:
  - Random
  - Get File
  - QUERY
  - POS



GUI Shell

- Each interactive component in the GUI (radio buttons, push-buttons, etc.) is controlled by a *call-back function*
- Every time a user-event is detected (mouse-click, hover, etc.) the appropriate call-back function is called
- For example, once the '**QUERY**' button is pushed, the functionality in the query-button's call-back function will be called
- Variables within a call-back function are **local**
- MATLAB uses the `**handles**' structure to share data globally

handles

- Initialize to hold all the components in the GUI
- To get the positive indices, type:

  >>handles.posInds;
- Add a local variable K to be available for all the call-back functions, type:

  >>handles.K = K;

  >>guidata(hObject,handles); % guidata function saves the '**handles**' structure globally

Important Global Variables

name	type	description
handles.META_DATA	[47 x 1400 double]	47-dimensional image feature for each of 1400 images in the database.
handles.W	[47 x 47 double]	Relevance feedback weighting matrix
handles.posInds	[1 x (variable) double]	Indices of images in the database that have thus far been labeled as relevant
handles.currentTopInds	[1 x (variable) double]	Indices of the Top 20 returned images given a particular query

Task 1: Retrieval function (I)

- Given several query images (including those from user feedback), return top 20 images that are visually similar to the query images
- INPUT:
  - Image features for all images (handles.META_DATA),
    Indices of the query images (handles.posInds)
- OUTPUT:
  - Rank-sorted list of top 20 indices (handles.currentTopInds)

Task 1: Retrieval function (II)

- 3 Subtasks:
  1) Compute the mean image-feature vector q_c for the query images (given as the indices in **handles.posInds**)
  2) Compute the Mahalanobis distance between the mean query images vector to each of the images in the database

     $$D(j) = (q_c - x_j)^T W (q_c - x_j)$$

     where q_c is the mean query-images feature, x_j is the j-th database image feature vector
  3) Sort distances in $D(j)$ to get the indices of the top 20 similar images and return the indices to **handles.currentTopInds**

- Note: To complete the 2nd subtask, you need the weighting matrix from task 2. You can first implement Task 2 by just returning a 47x47 identity matrix from the relevance feedback function

Task 2: Relevance feedback function (I)

- Given a set of similar images (from user's feedback), find a diagonal weighting matrix such that the features with smaller variances have larger weights and vice versa
- INPUT:
  - Image features for all images (handles.META_DATA)
  - Indices of the query images (handles.posInds)
- OUTPUT:
  - A diagonal weighting matrix (W)
- NOTE: if there is no image from user feedback, you have to return a 47x47 identity matrix

Task 2: Relevance feedback function (II)

Retrieval function calls **RF(handles)** to analyze the features for images already labeled as positive in order to compute a weighting matrix W

1) Determine whether the feedback is round 0 (initial seed image)

2) If round 0, return a 47x47 identity matrix

3) If not, for each feature dimension (47 in total)
   - Obtain the value of this feature for each of the positively-labeled images (**handles.posInds**) The end result should be a list of *K* values, where *K* is the length of **handles.posInds**
   - Compute the variance of this sample and add **0.0222** (for regularization)
   - Form and return a diagonal weighting matrix **W** such that:

$$W(i,i) = \frac{1}{\sigma_i^2 + 0.0222}$$

Extra credit: Full weighting matrix

- Can you ‘find’ a full weighting matrix?
  - Inverse of covariance matrix
    - Any problem?
    - Regularized inverse of covariance matrix: $inv(XX^T + aI)$
  - PCA
    - Given a set of images, find the projection matrix P such that principal components (those directions with high variance) are removed
    - $W = P^T P$ (P is a d-by-47 matrix)

Extra credit 2: Using negative feedbacks

- Can you exploit negative feedbacks as well?
  - Implicit negative feedback: those images not being selected as positive
    - Assuming all the positive images are selected as positive
- How?
- LDA:
  - Given a few of positive and negative images, find the projection matrix P that best separate the two groups
  - Again, $W = P^T P$ (P is a d-by-47 matrix)

Experiments

- Conduct five interactive retrieval sessions (with different seed images as provided in the drop-down menu) to explore the effect of relevance feedback on retrieval performance
- Retrieval accuracy will be quantified by ***Precision***
- Precision is defined as the proportion of relevant images returned to total number of images returned (in this case 20). For each of the seed images, you will provide a plot showing precision as a function of feedback round for 3 rounds of search (1 initial + 2 relevance feedback). You may put all plots on the same figure.

Deliverables

- MATLAB code for a fully-functional CBIR GUI. Both empty functions should be complete, and the program should work properly
- Quantitative results in plot form for your interactive retrieval experiments (Precision versus feedback round)
- Qualitative comments about your results. Specifically:
  - What is the general trend of precision versus interaction round? (Or, what should it be?)
  - Why do you think certain plots are different than others? (from an image representation perspective)

MATLAB Hint

- Keep one GUI open at a time
- Use break-points to trace the codes, since the order-of-execution is more complex in GUI