7 Conclusions and Future Work:

This project started the work on developing an image search engine. The basic framework for an image search engine has been created and all the necessary components to make it scalable, configurable and restartable have been incorporated. Image searching is a complex endeavor and active research is still going on to make it efficient. The results show that text based search engine can have good Precision and Recall with specific queries like names that are not likely to be found in many pages. The results for generic queries and common words are not good because of the assumptions a text-based search engine makes. More algorithms and heuristics can be added to make the search more efficient and get better results. Some of the ideas that can be explored are –

1. To create a search engine that will crawl and index several domains at regular intervals, the crawler may need to be changed to a multi-threaded program that can be run on several servers.
2. The code can be zipped and made into self-installing package in order to be distributed for further enhancement.
3. The queries and the date on which they were made are saved in a database table QUERIES. This can be used in future to cache the most used queries and their results in order to improve performance.
4. A link structure analysis can be done on the pages (Lempel 2001 and Harmandas 1997) for more effective ranking. It must be kept in mind that for an image search engine the results returned are not so high as a text search engine, so ranking assumes less importance than finding relevance. This could increase the precision.
5. More heuristics can be added to eliminate functional images from the result set. Research has to be done to find how they can be identified. Two approaches were used in this project, eliminating images that appear frequently within a page. Eliminate images that are very small (less than 1024 bytes).
6. Another possible direction is to compare the words in the database against the user’s query using all the synonyms for the given word. This can be implemented by using a thesaurus for generating synonyms. This may improve the recall.

7. A “fuzzy match” of the words against the database can be implemented to find similar words. The MATCH() function in MySQL performs a natural language search for a string against a text collection. For each row in the table, MATCH() returns a relevance value, that is, a similarity measure between the search string and the text in that row in the columns named in the MATCH() list. Similarly PERL has a Soundex package that returns relevance based on an input string. Morphological variations of a word can be tested for similarity using stemmers that are available.

8. Although relevance feedback is used in the context of content based searching, it can be studied to see if it can add any value.
8 References:


Appendix A

Developer’s Guide

The crawling is controlled by a config file that looks like this.

```bash
###CONFIGURATION FILE FOR THE CRAWLER###
# database configuration parameters
#-----------------------------------------------#
$driver = "mysql";
$databaseName = q{dbname=xxxxx;host=xxxx};
$dsn = "DBI:$driver:$databaseName";
$user = "xxxxx";
$password = "xxxx";

#---------------------------------------------#
# no of urls to parse, or no of images to get
#---------------------------------------------#
$url_limit = 500;
$img_limit = 5000;
$min_size = 1024;

#---------------------------------------------#
# Number of words surrounding an image that
# will be added as caption words
#---------------------------------------------#
.words_to_get = 10;

#---------------------------------------------#
# File names
#---------------------------------------------#
$logfile = "crawler.log";
$errorlog = "error.log";
$img_dir = "public_html/images";
$indexlog = "indexer.log";
$indexerr = "inderror.log";

#---------------------------------------------#
# Starting URLs
#---------------------------------------------#
$startURL = qq(http://cs.eas.uccs.edu/fac/Faculty.htm);

#---------------------------------------------#
# Domains from which the pages will be processed
#---------------------------------------------#

The programs are crawler.pl, parser.pl, indexer.pl, image.php, search.html.
Following are the DDL statements that can be used to create the Tables.

```sql
create table page
    (page_id int(11) NOT NULL auto_increment,
     page_url varchar(255),
     crawled_date DATE,
     index (page_url),
     PRIMARY KEY (page_id))

create table image_table
    (image_id int(11) NOT NULL auto_increment,
     image_loc varchar(255) NOT NULL, page_id int NOT NULL,
     image_url varchar(255), page_freq int DEFAULT 0,
     PRIMARY KEY (image_id),
     index (image_url),
     FOREIGN KEY (fk_pageId)
     REFERENCES page(page_id))

create table keywords
    (word varchar(100) NOT NULL,
     image_id int NOT NULL,
     location char(1) NOT NULL,
     FOREIGN KEY (fk_imageId)
     REFERENCES image_table(image_id),
     index(image_id))

create table frequency
    (word varchar(100) NOT NULL,
     image_id int NOT NULL,
     title_freq int NOT NULL,
     alt_freq int NOT NULL,
     filename_freq int NOT NULL,
     caption_freq int NOT NULL,
     anchor_freq int NOT NULL,
     page_freq int NOT NULL,
     weight int NOT NULL,
     INDEX (word), index (image_id),
     FOREIGN KEY (fk_id)
     REFERENCES image_table(image_id))
```

In order to port this application to a different machine the following needs to be done. Copy all the PERL programs into your directory. The public_html directory should contain the image directory and the PHP programs. There are two images that are used on the search page, they should also be copied. Finally the img.cfg file needs to be updated to reflect the database parameters. These should also be updated in the image.php program.
Appendix B

User’s Guide

The search query is entered in the main search page http://cs.uccs.edu/~akodavan/search.html. The page looks like this.

The results are returned on another page and clicking an image will take the user to the URL of the page containing the image.
Implementation of an Image Search engine

Image Search

1 Images Found