Content Based Retrieval System for Archaeological Images

Akshay Asthana Ranjan Dutta Anshul Jain Deepak Gupta Sanjay Goel
Department of Computer Science and Information Technology,
Jaypee Institute of Information Technology University (JIITU), Noida, India.
{ akshay.asthana, rd.30330, anshulljain , deepak.jiit } @gmail.com , sanjay.goel@jiit.ac.in

Abstract

Most of the existing image retrieval systems take the textual query from the user and utilize the metadata associated with the database images to retrieve the result. However, the results of these systems are substandard. This paper presents a framework for the matching and retrieval of archaeological images based on highly specific visual features. As a result, the potential users of the system (archaeologists, art historians, curators and students) will have a tool to refine the search for digital images. The system incorporates the identification of distinct visual feature vectors, which are extracted from each image in the database through implementation of high level image feature based algorithms such as Robust Invariant Features (RIF), Scale Invariant Feature Transform (SIFT) and low level image feature based Hierarchical Tree Matching (HTM). The detailed comparison of these approaches, by reviewing their current capabilities, limitations, and potential usefulness in the field of archaeology is presented.

Keywords : *Archaeology, Content Based Image Retrieval (CBIR), Hierarchical Tree Matching (HTM), Robust Invariant Features (RIF), Scale Invariant Feature Transform (SIFT).*

1. Introduction

Since majority of archaeological excavations are destructive in nature and it is possible to excavate a site only once, it is therefore necessary for the Archaeologists to store each and every primary data for further research. As a result, new digital information is constantly being added from active excavation sites. With the accumulation of this digital information in archaeological repositories, large quantities of diverse data have been generated which is managed by various types of traditional information retrieval systems. Presently, much of this archaeological data is spread across various archives, historical societies, libraries and museums. Hence, it is quite a challenge for archaeologists to manage this exponentially increasing data and enable fellow archaeologists, art historians, and other scholars to collect, preserve and expose historical data for research, education and public awareness in an accessible way.

One of the publicly available archaeological digital libraries which have made its mark is "ETANA Digital Library" [GSV06]. It has two core components DigKit and DigBase. DigKit is a tool for collecting and recording data during surveys and excavations, whereas, data from various sources, whether excavation sites or web sites are collected and archived in DigBase. DigBase is an enormous catalog that allows the users to not only search and browse but also query primary records, rate and review artifacts, and receive responses modified to the user's particular interests. ETANA-DL has modeled its archaeological information systems using 5S theory and follows a metadata based approach. Another digital library which is in its initial stage, "The Nautical Archaeology Digital Library" [MPF06] aims to provide a platform for archaeologists to access, manipulate, study and consult a variety of sources from different media, geographical origins, ages, and languages. Its functionality is based on extraction of metadata from large amount of heterogeneous data and structuring, segmenting, categorizing, indexing, and integrating it to form an architecture that supports its goals.

This paper describes a content based image retrieval system which utilizes highly distinctive image features to produce the search result instead of a metadata based approach. We address the sub-problem of "How to distinguish images and more importantly, how to select the distinguishing features for the classification of images?" and "What all are the suitable approaches for the image feature extraction and classification for the image search component in a retrieval system?" The solutions approach proceeds in context to these sub-problems only.

The proposed framework incorporates the identification of distinct visual feature vectors [Low04], that are extracted from each image in the database through implementation of high level image feature based algorithms such as Robust Invariant Features (RIF) [LKK04] , Scale Invariant Feature Transform (SIFT) [Low04], and low level image feature based Hierarchical Tree Structure based image matching [LN03].

For high level image feature generation using RIF and SIFT, we take an image and transforms it into a "large collection of local feature vectors" [SLL02]. The output of these algorithms for a particular image is in a form of set of descriptors. This set of descriptors is invariant to scaling, rotation, and illumination changes [BL02]. The database will contain images along with the set of descriptors. When a new image is added to the database, high level image feature based algorithms would be applied to extract and store the subsequent set of descriptors. The user provides a sample image. The features of the sample image would be compared with the features of all the images in the database. The output image set is ranked according to the number of matched features to yield the final result.

For low level image feature generation The Hierarchical Tree Matching (HTM) [LN03] based approach is used. HTM is faster than the high level image feature based approaches but less accurate in producing the results. The features obtained by these algorithms though are highly distinctive; but the image signature (collection of the image features) obtained is very complex and further processing is required for the practical utilization of the image signature. The whole process is complex in terms of time as well. It has been visualized that in case of the sculptural images, the classification may not

require the image signatures to be complex. Hence, the classification using low level image features is done by the hierarchical tree partitioning of the images and then matching on the basis of the same [EG99]. The process is applied to a preprocessed image for efficient results.

2. Related Research

This section discusses some of current content based image retrieval methods and systems [VT01]. It compares specific systems, rather than general architectures. Most of the systems analyzed are web based applications with interfaces designed in accordance to the client side requirements and specifications. For each of the systems, the conventional parameters of the algorithmic approach, namely the input, processing, and output are analyzed. The analysis clearly shows that most of these systems use low level image features, and few use high level semantically meaningful features for retrieval process.

a) Advanced Multimedia Oriented Retrieval Engine (AMORE)
Input: The category of the image, keyword or a sample image or the image URL, the level of importance (eg. 'very important', 'not important') to the shape and color for the visual similarity, and the image database.
Processing: Amore follows the a region-based approach to find images visually similar to the query image. The input image is segmented into blocks (24 x 24), and then matched with the database images like template matching in form of blocks. The color similarity is analyzed on the basis of the HLS distance between the images.
Output: The retrieved images are presented in a form of Thumbnails and uses a Query Result Visualization Environment to organize the search results.
URL: http://www.ccrl.com/amore/.

b) Alexandria Digital Library
Input: A digital globe map with dynamic functionalities (powered by Google), a catalog of images, alphanumeric query parameters.
Processing: Searching and matching the data on the basis of the supplied parameters. The image texture features and the keywords are utilized for the transformation of the input into output.
Output: The output contains the matching sections of the map and are presented in a form of thumbnails.
URL: http://www.alexandria.ucsb.edu/adl.html

c) AltaVista Search Engine
Input: A textual query and the image database.
Processing: The low level image features and the metadata is used for the matching of the query image with the database images for the retrieval of output image set.
Output: The retrieved images are presented in a form of thumbnails.
URL: http://image.altavista.com

d) Berkeley Digital Library Project (BDLP)
Input: The user is provided a set of 13 colors (red, orange, yellow, green, blue-green, light blue, blue, purple, pink, brown, white, gray and black). The magnitude and the size of region for each color is selected by the user.
Processing: For each image in the database, its hue, saturation and value (HSV) is mapped in 13 color channels and are stored as a text string. The text substring matching is used for retrieval purpose.
Output: The retrieved images are presented in a form of Thumbnails and the captions contain the metadata associated with them (eg. Name of the collection table, unique ID).
URL: http://elib.cs.berkeley.edu/

e) Blobworld
Input: The user submits an image in order to see its segmented representation into blobs, that are described by the color distribution and mean texture descriptors. Then user selects the relevant blobs to match, and specifies the relative importance of the blob features. More than one regions can be used for querying.
Processing: The matching of the images is carried on the basis of the color histogram based approach. The similarity between the matching images is calculated on the basis of the Euclidean distances between the two image descriptors.
Output: The retrieved images are ranked and are shown together with the image displaying the regions.
URL: http://elib.cs.berkeley.edu/photos/blobworld/

f) Comparison Algorithm for Navigating Digital Image Databases (CANDID)
Input: The input contains a query image and the database.
Processing: Several low level image features like color, shape, and texture are computed at every pixel in the image and then a histogram of feature vector is created. Then a continuous probability density function over this multi-dimensional feature space is computed. The comparison between two images is done on the basis of the difference between these distributions, by taking an infinite integral over the entire multi-dimensional feature vector.
Output: Matched results are returned back to the system. No specific GUI or output formula.
URL: http://public.lanl.gov/kelly/CANDID/index.html

g) DrawSearch
Input: The input query can be supplied either on the basis of the color and shape of the image or texture based.
Processing: The system extracts image features such color, shape and texture. In the retrieval for color and shape-based sub-system, two cosine similarity functions (one for color and shape each) are estimated and are merged linearly to form a final similarity vector. Whereas, in the retrieval for texture based sub-system, only one similarity function is estimated which is based on the normalized Euclidean distance between the texture feature vectors of query image and image in the database.

Output: The retrieved images are presented in a form of thumbnails which are ranked.
URL: http://deecom03.poliba.it/DrawSearch/DrawSearch.html

h) DWR Picture Retrieval System (Chabot / Cypress)
Input: User selects one of more fields from the list (CD Disk Number, CD Image number, DWR ID, Category, Subject, Organization, Job Request, Keywords, Photographer, Film Format, Shoot Date, Perspective, OID, Entry date, Indexers, Comments, Location, Colors and Concept) and provide their descriptions to proceed with his search.
Processing: The basic matching features are the low level image features (color histogram) and the metadata associated with the image. Keywords are used to refine the search focus.
Output: The retrieved images are ambiguously ranked.
URL: http://http.cs.berkeley.edu/~ginger/chabot.html.

i) Fast Object Color Based Query System (FOCUS)
Input: Image sub-region marked by the user in any image from the database, which contains the object to be searched.
Processing: The image is segmented into regions of dimensions 100 x 100. Quantized HSV color histogram value is calculated for each region. A frequency table for each color in the HSV space is created. On the basis of the peaks in the frequency table and those obtained in the query image, the comparison is performed for detecting a match. The number of peak values matches decides the selection of an output result image.
Output: The retrieved images are presented in a form of thumbnails at the bottom of the webpage and are ranked..
URL: http://wagga.cs.umass.edu/~mdas/color_proj.html

j) Flexible Image Database System (FIDS)
Input: The user can choose an input query image and specific features; further the refined queries can be reused as input again.
Processing: The matching is done on the basis of the weighted average of the histogram distances and the wavelet coefficients.
Output: The retrieved images are ranked and presented in a form of thumbnails.
URL: http://www.cs.washington.edu/research/imagedatabase

k) Formula Image Retrieval (FIR)
Input: The query is an input example image and the image database.
Processing: Images of any size and format are transformed into thumbnail images of fixed size (128X128). The colors of the thumbnails are then converted into a color space that approximates color differences as perceived by humans (from RGB to LUV). Then, thumbnails are transformed into wavelet space by a non-standard Haar analyzer and coefficients having trivial information are removed. The resulting highly compact feature vector is finally stored. The similarity between two images is calculated on the basis of the weighted Euclidean distance between their feature vectors.
Output: The output contains the top 20 images along with their weighted Euclidean distances from the feature vectors of the query image.
URL: http://www.igd.fhg.de/igd-7/projects/formula

l) ImageFinder
Input : The user provides a query image and the image database.
Processing: The matching of the images is done on a special kind of a artificial neural network called the Boltzman Machine. It has a simple learning algorithm that learns complex patterns occurring in the training data.
Output: The output contains the set of link to the relevant images found.
URL: http://attrasoft.com/abm3_4.html

m) ImageRover
Input: The user first provides a set of keywords for the image search and afterwards goes for choosing the images out of the set as sample images for more refined search.
Processing: The color and texture orientations in the image are utilized for the purpose of matching. The matching is prominently on the basis of the color and texture orientations along with the application of relevance feedback techniques.
Output: The retrieved images are presented in a form of thumbnails and are ranked. User can also select the number of images to be displayed.
URL: http://www.cs.bu.edu/groups/ivc/ImageRover/

n) Netra
Input: Query image is selected by the user from the indexed database and is then used as input.
Processing: The system is based on the low level image feature matching. Each database image is segmented into regions of matching color. Color, texture and shape feature vectors are extracted from these regions. The similarity between the two color feature vectors are calculated on the basis of weighted euclidean distance. The similarity between the two texture feature vectors are calculated on the basis of L_1 – distance between them. The similarity between the two shape feature vectors are calculated on the basis of euclidean distance. Color, texture, and shape are indexed separately. The first feature the user selects is considered for primary search and ranking criterion. Then other possible features together are used to further order the retrieval result.
Output: The retrieved images are presented in a form of thumbnails and are ranked.
URL: http://maya.ece.ucsb.edu/Netra

o) VIR Image Engine
Input: The engine provides the user with the options for the formulation of the GUI and the input and output parameters. The variability of the tool can be utilized in a number of ways according to the user requirement.
Processing: The processing again is flexible depending upon the input supplied by the user to the system. The dynamism in the system allows the usage of the low level as well as the high

level image features for the classification and matching of the images.
Output: Variable and user oriented output.
URL: http://www.virage.com/products/vir-irw.html

3. Low level feature based retrieval using Hierarchical Tree Matching (HTM)

The idea is to represent the coherent image region into a hierarchical tree structure followed by the creation of an image feature matrix [LN03]. The process is as follows –

3.1. The Hierarchical Tree Formation

Each image in the database is represented with the help of a hierarchical tree structure with three levels of distribution. A tree [Figure – 1] is created for each image that contains three basic levels. The three level non-overlapping regions are used for accumulation of the image features for comparison.

Level 1: The whole image of size M x N.
Level 2: The 9 overlapping regions each of (M/2) x (N/2).
Level 3: The 36 non-overlapping sub-regions, each of size (M/4) x (N/4), created by dividing each of the 9 regions on level 2 into four parts. Some of these regions are not unique.

3.2. The Feature Extraction Process

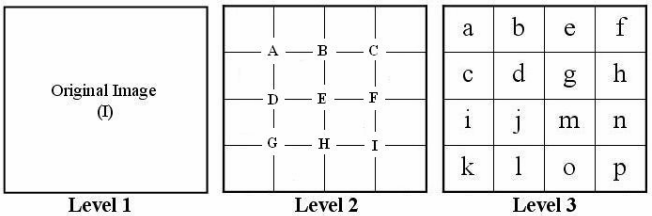
The average color [DMK01] of each of the sixteen non-overlapping image regions [Figure – 1] in the RGB color space is computed for image indexing. The average color for an image region 'a' covered by the pixel p(xα,yα) to p(xβ,yβ), is given by –

$$\left\{ U_i(a) = \frac{\left(\sum_{X=x_\alpha, Y=y_\alpha}^{x_\beta, y_\beta} p_i(X,Y)\right)}{N} \right\}_{i \in (R,G,B)}$$

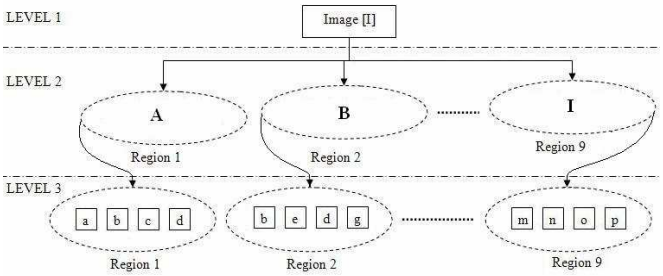
where N is the total number of pixels in the image region 'a'. Finally a 3-D global color feature vector for each image region is obtained and is represented vector 'V' as:

$$\vec{V}(a) = \begin{bmatrix} U_R(a) \\ U_G(a) \\ U_B(a) \end{bmatrix}$$

This feature vector of the image is used for comparison.



(a) *Image divided into regions at different levels*



(b) *The three level tree created for each Image*

Figure 1 : *The Hierarchical Partition of the image and the resultant tree structure*

3.3. The Distance Measure

The similarity of the two image regions is calculated on the basis of the weighted L1-Norm for the two:

$$\|V(a) - V(b)\| = \sum_{i \in (R,G,B)} \left(\frac{|V_i(a) - V_i(b)|}{\beta_{V_i}} \right)$$

where β_{Vi} is the normalizing component (the grid pixel count, used in experimentation). For the comparison of the sub-image the above formula can be modified to take into account the contribution of the sub-image accordingly. The distance measure between two images is calculated on the basis of the sectional norms distances with the help of the formula:

$$D(Q, I_D) = \sum_{j=0}^{n-1} \left(\frac{\|V(Q_j) - V(I_{D_j})\|}{n} \right)$$

where Q is the input query image, I_D is the image from the database and n is the number of unique non-overlapping sub-regions. The cumulative distance between the images gives the distance measure. The distance measures between the input query image and each image in the database are then sorted in ascending order and the corresponding images are displayed.

3.4 Image retrieval using HTM

In order to retrieve the relevant images from the database, we start with comparing the full tree structures of each database image with the query image. Then, to compare the sub-image regions of the database image at different hierarchical levels, we pass the query image's tree structure within that of the database image. This iterative process continues till all the sub-trees have been processed. The minimum distance computed between any sub-region of the database image being processed and the query image is considered as the distance between this database image and the query image. Finally, this distance is utilized for ranking the matching results.

4. High level feature based approach

High level image features refer to the content of the image which defines image as a whole and not as a discrete pixel values. It is computationally expensive but highly robust to the changes in scaling, rotation, illumination and affine transformation. Robust Invariant Features (RIF) and Scale Invariant Feature Transform (SIFT) are analyzed and implemented to identify the high-level features.

4.1. Scale Invariant Feature Transform (SIFT)

Each image can have large number of visual features, therefore it is not possible select these features manually. So the Scale-Invariant Feature Transform (SIFT) algorithm [Low04] is used to generate the high level image features automatically.

4.1.1. SIFT feature extraction

SIFT is an efficient Image Feature extraction algorithm being invariant to rotation, scale change, affine transformation and partially invariant to changes in illumination [MS05]. SIFT is applied to all the images in the database and SIFT features generated are stored in the database for matching purpose. SIFT [Low04], have four main steps–

Step - 1 Scale-space extrema detection
The first stage of keypoint detection is to identify locations and scales that can be repeatably assigned under differing views of the same object. Detecting locations that are invariant to scale change of the image can be accomplished by searching for stable features across all possible scales, using a continuous function of scale known as scale space.

A sample image I (x , y) is given as input. The Difference of Gaussian function (DOG) between the two nearby scales separated by a constant multiplicative factor 'k' is to be calculated.

Step - 2 Keypoint localization
After the DOG has been calculated, the local extrema has to be detected. Here, the comparison between the value of D(x,y,σ) for each pixel with the D(x,y,σ) of 8 neighbors in the same scale and 9 neighbors in the scale above and below it is made. If the D(x,y,σ) of a pixels is the minimum or maximum among its 26 neighbors, it is marked as a candidate keypoint .

Step - 3 Orientation assignment
Rotation Invariance is achieved by computing the Descriptor relative to keypoint's orientation. So, to find the orientation –

- Select the Gaussian Smoothed Image L at the keypoint's scale.

$$L(x, y) = G(x, y, \sigma) * I(x, y)$$

- Calculate the Gradient magnitude m(x,y) and orientation θ(x,y) for each and every pixel.

$$m(x, y) = \sqrt{L(dx, y)^2 + L(x, dy)^2}$$

$$\theta(x, y) = \tan^{-1}\left(\frac{L(x, dy)}{L(dx, y)}\right)$$

- To assign the orientation to each keypoint, we compute the dominant orientation in the image patch of 3X3 centered at the keypoint location.
- Orientations are divided into 36 bins (10 degrees for each bin)
- Pixel's contribution to the bin is equal to its Gradient magnitude **m**, if the Gradient orientation **θ** lies in the range of the bin.
- Total bin's weight is equal to the sum of contributions from all pixels.
- Assigned Orientation to the keypoint is **θ**, which is represented by the highest peak in the Gradient Histogram.

Step - 4 Keypoint descriptor calculation.
The local gradient data, used above, is also used to create keypoint descriptors. The gradient information is rotated to line up with the orientation of the keypoint and then weighted by a Gaussian with variance of 1.5 * keypoint scale. This data is then used to create a set of histograms over a window centered on the keypoint. Keypoint descriptors typically use a set of 16 histograms, aligned in a 4x4 grid, each with 8 orientation bins. This result in a feature vector containing 128 elements.

4.1.3. Extending SIFT to PCA-SIFT

From the SIFT Descriptor point of view, which are already highly distinct in nature, the Principal components analysis (PCA) is used to reduce multidimensional data sets to lower dimensions for analysis [KS04]. PCA involves the computation of the Eigen value decomposition of a data set, usually after mean centering the data for each attribute. PCA-SIFT algorithm [HH05] for local descriptors accepts the same input as the standard SIFT descriptor. The inputs are the sub-pixel location, scale, and dominant orientations of the keypoint.

Step – 1 Computation of Projection Matrix
For each keypoint -

- Extract an image patch around it with size 21 x 21 pixels
- Calculate horizontal and vertical gradients, resulting in a vector of size 20 x 20 x 2 = 800
- From Figure 2 we calculate the vertical and horizontal gradients.

$$Vertical\ gradient\ V = [m + m' \cos(\theta' - \theta)].\sin\theta$$

$$Horizontal\ gradient\ H = [m + m' \cos(\theta' - \theta)].\cos\theta$$

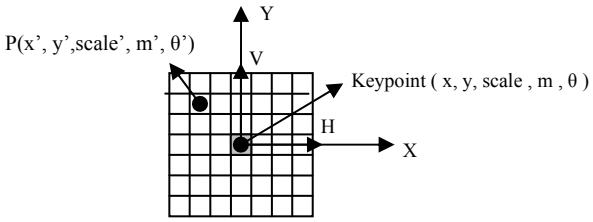


Figure 2 : *Horizontal and Vertical gradients*

We perform the above operation for each and every keypoint. The resultant is a Matrix M (Number of Keypoints , 800). Next, we compute the Covariance Matrix for the Matrix M. The resultant Covariance Matrix Cov(A) is of dimensions 800x800. The Eigen Values and Eigen Vectors are calculated for the Cov(A) matrix. Out of the 800 Eigen Values, first 20 non-zero Eigen Values are selected and a Modal matrix is computed from these Eigen Values. This Modal matrix is known as the Projection Matrix. The Project matrix is of dimensions 800 x 20. Projection matrix is computed once and saved for future use.

Step – 2 Project the image patch around the Keypoint
To project the image patch around the Keypoint, the feature vector is multiplied with the Projection Matrix computed earlier. Let N be the number of SIFT Keypoints detected. Hence, the dimensionality of each SIFT descriptor is reduced to 20 and the whole Image feature vector is reduced to N x 20 dimensions.

4.1.4. Image retrieval using SIFT and PCA-SIFT

It is done to find the best candidate match for each keypoint which is its nearest neighbor in the database of keypoints from training images. The nearest neighbor is defined as the keypoint with minimum Euclidean distance for the invariant descriptor vector.

There will be a number of SIFT descriptors which will not have any correct match in the database because they arise from background clutter or were not detected [Low04]. Therefore, this threshold of Euclidean Distance to the maximum of 1.0 will help us to discard features that do not have any good match to the database. A more effective measure is obtained by comparing the distance of the closest neighbor to that of the second-closest neighbor. This measure performs well because correct matches need to be highly distinct and significantly closer than the closest incorrect match to achieve reliable matching [Low04]. Let the Euclidean Distance between the nearest neighbor and the second nearest neighbor be ED_1 and ED_2 respectively. If, ED_1 and ED_2 satisfies the condition -

$$\frac{ED_1}{ED_2} < K \qquad (K = 0.85 \text{ is used for experiments})$$

Then, we can safely say the closest descriptor match, with Euclidean distance ED1, is highly distinct and qualifies to be a correct match. The drawback of exhaustive matching is the high computational complexity that cannot be ignored.

4.2. Robust Invariant Features (RIF)

4.2.1. RIF feature extraction

The five step process for extracting Robust Invariant Features (RIF) [LKK04] is as follows –

Step - 1 Interest point detection.
Harris corner detection algorithm is used to identify the interest points. It is a popular interest point detector due to its strong invariance to rotation, scale, illumination variation and image noise [MS04]. A corner can be defined as the intersection of two edges or as a point for which there is two dominant and different edge directions in a local neighborhood of the point.

The algorithm tests each pixel in the image to see if a corner is present. It considering how similar a patch centered on the pixel is to nearby, largely overlapping patches. The similarity is measured by taking the sum of squared differences (SSD) between the two patches. A lower number indicated more similarity. The lowest SSD becomes the *corner strength*.

If the pixel is in a region of uniform intensity, then the nearby patches would look similar. If the pixel is on an edge, then nearby patches would look quite different. If the pixel is on a feature with variation in all directions, then none of the nearby patches would look similar.

Step - 2 Tracking of interest points.
After a set of multi-scale interest points is extracted from the input image, tracking and grouping is performed in order to derive points that are more stable in scale space. The idea is to cluster those multi-scale interest points corresponding to the same local structure. The linking process is propagating from the highest scale level to the lower scales based on the principle of nearest neighbor searching within the uncertainty regions. The tracking of interest points continues until no corresponding links within the allowed uncertainty ranges. The uncertainty range propagation is very important here, as it highly affects the grouping results.

Step - 3 Grouping of tracked interest points.
The algorithm for Grouping [LKK04] is as follows:

(i) Start with the initial scale level, set the current scale level $l = L$ and current number of groups $k = 0$.

(ii) For the current scale level l,

a) Assign a new group G_k for every interest point without a link to the upper scale level and set $k \leftarrow k+1$.

b) For each interest point P_i^l in this level, search the nearest neighbor point link in the lower level $l - 1$. If (the link exists and within the range of uncertainty) Then, assign this link to the group corresponding to

P^l_i. Else terminate the corresponding linking at current level *l*.

iii) Go to the next scale level $l \leftarrow l - 1$ and iterate ii ~ iii, until all interest points are assigned to a group or $l = 0$.

iv) Store the feature groups $G_1, G_2, \ldots, G_k$.

Step - 4 Localization of the scale space.

In each group, Laplacian of Gaussian (LoG) of each point is calculated. It then selects the points for which the LoG attains a maximum over scale. In [MS04], several differential operators were compared and experiments showed that LoG finds the highest percentage of correct characteristic scales to be found.

$$|LOG\ (x, \sigma_l)| = \sigma_l^2 |L_{xx}(x, \sigma_l) + L_{yy}(x, \sigma_l)|$$

The LoG is well adapted to blob detection due to its circular symmetry, but it also provides a good estimation of the characteristic scale for other local structures such as corners, edges, ridges and multi-junctions.

Step – 5 Computation of Zernike Moments.

A symmetrical patch is identified around an interest point which is represented by Zernike moments. Zernike moment descriptors are rotational invariant. Zernike moment descriptors can be computed in different ways. The authors in reference [SLL02] give various approaches to compute Zernike moments. A q-recursive method is proposed for fast computation of Zernike moments. Experimentally it was shown that proposed method takes least amount of time for computation of Zernike moment of any order.

Further in reference [KKK04] author has tried to overcome difficulties in object recognition such as illumination change, background clutter, and occlusion. Applying the scale space theory to image model, which enhances the corner extraction process, solves scale problem. Illumination invariance is achieved by normalizing the moment with zero order Zernike moment. In calculation of Zernike moments image patches are mapped to unit circles. If the image patches are mapped to ellipse, then corresponding Zernike descriptors become affine invariant too without harming the rotational invariance [AKB07].

There exists another variant of Zernike, named pseudo Zernike moments [Che03]. Pseudo Zernike moments (PZMs) offer more feature vectors than Zernike moments. They can be directly computed from any scaled image without prior knowledge of the normalization parameters, or assistance of geometric or radial moments. They are also more robust to image noise than original Zernike moments. However, the computation of PZM takes longer time than original Zernike moments. The time complexity can be further controlled by altering the order of the moment. The results in this paper are derived from PZM of order five wherein each interest point is denoted by twenty-one values.

4.2.3. Image retrieval using RIF

After extracting image features, similarity between the images are compared through the Euclidean distance between the feature vector of query image and each image in the database. Features with distance below a threshold are considered to be matched. To avoid domination of subgroups, normalization is done.

$$Z_i = \frac{(Z_i - Z_{mean})}{\sigma_z}$$

Here, each value in Zernike moment, Z_i, is subtracted by its mean, Z_{mean}, and divided by its standard deviation, σ_z.

5. Result and Discussion

The proposed approaches were successfully implemented as software for content based retrieval of archaeological images. The software was rigorously tested on a database of 150 images. The results obtained were found to be invariant to Brightness [+/- 50%], Gray Scaled image, Hue [+/- 50%], Rotation [Clockwise and Anti- Clockwise] [5 and 10 degrees], Scaling [25%], Skew [25% Horizontal and Vertical]).

The performance evaluation starts with the comparison of the space and time complexity of the four feature extraction processes namely – HTM, SIFT, PCA-SIFT and RIF. From [Tables – 1, 2 and 3] HTM has the most efficient space and time complexity. This supports the fact that most of the image retrieval prefers low-level features over high-level features. Among the high-level counterparts, SIFT is a fast feature extractor than PCA-SIFT and RIF but has very high space complexity. PCA-SIFT and RIF, on the other hand, are considerably slower but have less space complexity. For example, a 160 X 213 image has approximately 350 SIFT (or PCA-SIFT) features and 25 RIF features. Then the SIFT feature vector would contain *350 x 128 = 44,800* values, PCA-SIFT feature vector would contain 350 x 20 = 7000 values and RIF feature vector would contain *25 x 21 = 525* values.

Table 1 : *Dimensions of each feature*

Feature Extraction Approaches	Dimension of each feature
HTM	1
SIFT	128
PCA-SIFT	20
RIF	21

The time of feature extraction is of utmost importance when the user wishes to add a new image to the database. The efficiency, distinctiveness and suitability of algorithms are put

Table 2 : *Size of feature vector*

Size of Feature Vector	81 X 109	160 X 213	240 X 319	450 X 446
HTM	16	16	16	16
SIFT and PCA SIFT	50-70	300-500	550-700	>700
RIF	2-10	20-30	35-45	>50

Table 3 : *Time taken to extract the feature vector*

Time taken (sec/image)	81 X 109	160 X 213	240 X 319	450 X 446
HTM	0.02	0.05	.07	0.2
SIFT	0.5	0.75	1.2	3
PCA-SIFT	25	40	50	65
RIF	3	12	26	52

to test when this query image is processed and matching is performed. However if the query image is picked from the database itself, feature extraction loses its significance and now it's the retrieval actions that derive the efficiency of the system. It is worth noting here that the computation time for retrieval of similar images is a function of the size of the feature vector of each image in the database. Table – 4 summarizes the performance of all the retrieval techniques over some basic image transformations (denoted as *Parameters*).

Table 4 : *Robustness towards Image transformations*

S.N	Parameters	HTM	SIFT	PCA-SIFT	RIF
1	Brightness	Good	Good	Good	Good
2	GrayScaling	Good	Good	Good	Good
3	Hue	Good	Good	Good	Good
4	Scaling	Ordinary	Good	Ordinary	Ordinary
5	Skewing	Poor	Good	Poor	Ordinary
6	Rotation	Poor	Good	Good	Good

Here *good*, *ordinary* and *poor* refers to quality (in decreasing order) of invariance achieved over the corresponding parameter. The dominance of high-level features over the low-level features becomes evident during retrieval. HTM shows *poor* invariant qualities over two parameters whereas SIFT, PCA-SIFT and RIF, as expected, depict better performance.

Along with the comparison of retrieval time, the comparison of the accuracy of all the approaches by identifying the number of correct matches (True positives) displayed is shown in Table – 5.

Table 5 : *Comparison of retrieval time and robustness*

Performance Evaluation	**Number of True positives**			
Variation Parameters	**HTM**	**SIFT**	**PCA-SIFT**	**RIF**
Brightness (+/- 50%)	19/20	20/20	18/20	20/20
Graying	20/20	20/20	17/20	20/20
Hue (+/- 50%)	17/20	20/20	17/20	20/20
Scaling (+/- 25%)	11/20	17/20	11/20	12/20
Skew (25% Horizontal and Vertical)	8/20	19/20	7/20	14/20
Rotation (5 and 10 degrees)	7/20	19/20	18/20	20/20
Retrieval Time (Sec)	**≈ 0**	**≈ 650**	**≈ 100**	**≈ 1**

The table lists the results for a random set of 20 images taken from the database. Number of correct matches (True positives) in the table refers to the instances when the relevant image was among the top three displayed results by the system for a particular query image.

Here HTM scores over the time factor, but losses the race when it comes to accuracy of results particularly while dealing with a variation in rotation, scaling or skewing of an image. SIFT, despite being highly accurate, is too slow to be deployed as a retrieval system. The application of the Principal Components Analysis (PCA) on reduces the retrieval time by almost six times, it is still not quick enough to compete with HTM and RIF. On the other hand, despite of its slow performance during feature extraction, RIF emerges as the most efficient approach for retrieving archaeological images as it enlightens both retrieval speed and accuracy.

6. Conclusion

From the results it is evident that high-level features are more accurate and RIF also provides a better retrieval speed. Each approach presented in the paper has its own pros and cons. Its applicability depends upon the requirement of the client. Figure 3 shows the retrieval results via HTM, SIFT and RIF for an input query image. As a future work, we aim to couple SIFT and RIF with Boosting process [TV01]. This algorithm is a modified version of the AdaBoost algorithm and will improve the efficiency of the matching process. Although, a "Super-fast" Content- based image retrieval system is still a distant dream, the work presented here is a small step in that direction.

References

[AKB07] Gholamreza Amayeh, Shohreh Kasaei, , George Bebis, Alireza Tavakkoli, Konstantinos Veropoulos: Improvement of Zernike Moment descriptors on affine transformed shapes. www.cse.unr.edu/~bebis/ISSPA07.pdf

[BL02] Matthew Brown, David G. Lowe : Invariant features from interest point groups. In British Machine Vision Conference, BMVC 2002, Cardiff, Wales, pp. 656-665.

[Che03] Qing Chen : Evaluation of OCR Algorithms for Images with Different Spatial Resolutions and Noises. A thesis submitted to the School of Graduate Studies and Research 2003, Ottawa-Carleton Institute for Electrical Engineering.

[DMK01] Y. Deng, B. S. Manjunath, C. Kenney, M. S. Moore, H. Shin : An Efficient Color Representation for Image Retrieval,IEEE Trans. Image Processing 2001,10(1):140–147.

[EG99] John Ekins, Margaret Graham : Content-based Image Retrieval. Technical Report 1999, University of Northumbria at Newcastle, JISC Technology Applications.

[GSV06] Douglas Gorton, Rao Shen, Naga Srinivas Vemuri, Weiguo Fan, Edward A. Fox : ETANA-GIS: GIS for archaeological digital libraries. In Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries.

[HH05] Andreas Haug, Stefan Hinterstoisser, Description, detection and matching of image features – SIFT and Nearest

Neighbor Search. Lecture Tutorial 2005, Department of Computer Science | Technische Universität München.
[KKK04] Sung-Ho Kim , So Kweon, Cheol Kim : Probabilistic Model-based Object Recognition using Local Zernike Moments. Technical Report 2004, Department of Electrical Eng. & Computer Science, KAIST.
[KS04] Y. Ke, R. Sukthankar : PCA-SIFT A More Distinctive Representation for Local Image Descriptors. In Proceedings of Computer Vision and Pattern Recognition 2004.
[LKK04] Zhe Lin, Sungho Kim, So Kweon : Robust Invariant Features for Object Recognition and Mobile Robot Navigation. Technical Report 2004, Department of EECS, KAIST.
[Low04] D. G. Lowe : Distinctive image features from scale-invariant keypoints. In International Journal of Computer Vision 2004, 60(2), pp. 91-110.
[LN03] Jie Luo, Mario A. Nascimento : Content Based Sub-Image Retrieval via Hierarchical Tree Matching. Proceedings of the 1st ACM international workshop on Multimedia databases, New Orleans, LA, USA, 2003.
[MPF06] Monroy C, Parks N, Furuta R, Castro F : The Nautical Archaeology Digital Library, 10th European Conference on Research and Advanced Technology for Digital Libraries ECDL 2006, Alicante, Spain.
[MS04] K. Mikolajczyk, C. SCHMID : Scale & Affine Invariant Interest Point Detectors. In International Journal of Computer Vision 2004, 60(1), 63–86.
[MS05] Krystian Mikolajczyk, Cordelia Schmid : A performance evaluation of local descriptors. In IEEE Transactions on Pattern Analysis & Machine Intelligence 2005, Volume 27.
[SLL02] Stephen Se, David G. Lowe Jim Little : Global localization using distinctive visual features. In International Conference on Intelligent Robots and Systems, IROS 2002, Lausanne, Switzerland, pp. 226-231.
[TV01] Kinh Tieu and Paul Viola, "Boosting Image Retrieval", IEEE Conference on Computer Vision and Pattern Recognition, 2000
[VT01] R. Veltkamp and M. Tanase. Content-based image retrieval systems: a survey. Technical Report UU-CS-2000-34, Utrecht University, 2000.

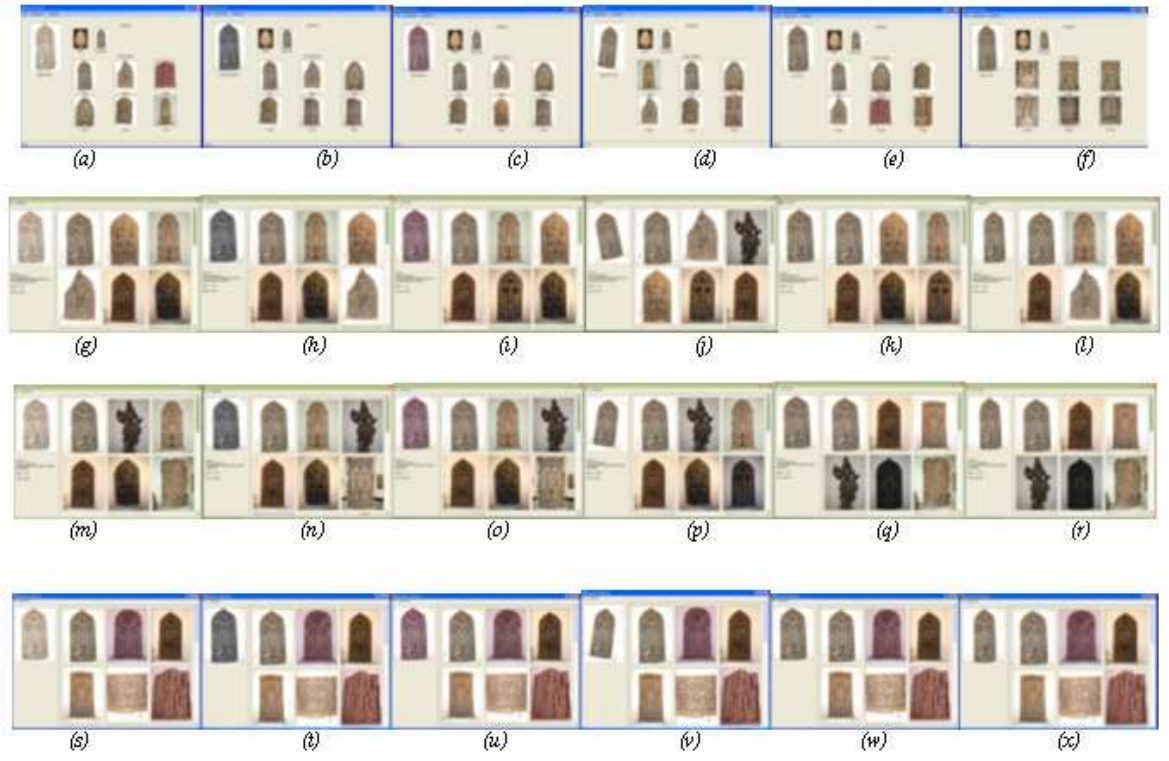


Figure 3 : *Retrieval results for a Query Image via HTM, SIFT, PCA-SIFT and RIF respectively.*

(a)-(f) *shows the retrieval results for variation in Brightness, Graying, Hue, Rotation, Size and Skewing respectively for HTM.*

(g)–(l) *shows the retrieval results for variation in Brightness, Graying, Hue, Rotation, Size and Skewing respectively for SIFT.*

(m)–(r) *shows the retrieval results for variation in Brightness,Graying,Hue,Rotation,Size,Skewing respectively for PCA SIFT.*

(s) – (x) *shows the retrieval results for variation in Brightness, Graying, Hue, Rotation, Size and Skewing respectively for RIF.*