NEAR-DUPLICATE DETECTION USING SYSTEM OF ASSOCIATIVE RELATIONS

A Thesis
Presented to the
Faculty of
San Diego State University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
in
Electrical Engineering

by
Ninada Hebbar
Fall 2011
The Undersigned Faculty Committee Approves the

Thesis of Ninada Hebbar:

Near-Duplicate Detection Using System of Associative Relations

Yusuf Ozurik, Chair
Department of Electrical and Computer Engineering

Sunil Kumar
Department of Electrical and Computer Engineering

Roman Swiniarski
Department of Computer Science

11/18/2011
Approval Date
ABSTRACT OF THE THESIS

Near-Duplicate Detection Using System of Associative Relations
by
Ninada Hebbar
Master of Science in Electrical Engineering
San Diego State University, 2011

With tremendous advancement in the field of communication and the abundant increase in social networking sites, making duplicates of web pages, images and videos has become very simple and easy. With such a high inflow of data, finding a Near-Duplicate at very high speeds becomes a challenging mission. The complexity of this task increases further for a video, as both temporal and spatial information will have to be considered and computations and comparisons should be done at real time.

This thesis addresses the issue of Near-Duplicate Detection, by reducing the computational complexity by the use of pair-wise pixel similarities. System of Associative Relations (SOAR) is used to encode inter-pixel relationships as a sequence of 1’s, 0’s and -1’s which is compared with a huge database of different and Near-Duplicate videos. Instead of considering the pixel value, only the difference is encoded, thus only the gradient change in intensity is considered for computations which proves to be sufficient to uniquely detect a video and greatly reduces the complexity. This algorithm avoids operations like division and other complex mathematical operations and mostly uses exclusive OR’s and additions which can easily be implemented in hardware using simple ‘XOR’ gates and ‘AND’ gates.

The obtained results are compared with the results obtained from a similar method based on spatio-temporal signature, and a better precision and lower false positive rates are seen.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>INTRODUCTION .......................................................... 1</td>
</tr>
<tr>
<td>2</td>
<td>RELATED RESEARCH .................................................. 5</td>
</tr>
<tr>
<td>3</td>
<td>SURF AND CLUSTERING ............................................... 11</td>
</tr>
<tr>
<td></td>
<td>3.1 SURF ........................................................................ 11</td>
</tr>
<tr>
<td></td>
<td>3.1.1 Interest Point Detector ...................................... 12</td>
</tr>
<tr>
<td></td>
<td>3.1.1.1 Integral images ............................................. 13</td>
</tr>
<tr>
<td></td>
<td>3.1.1.2 Box Filters .................................................. 14</td>
</tr>
<tr>
<td></td>
<td>3.1.2 SURF Descriptor .............................................. 15</td>
</tr>
<tr>
<td></td>
<td>3.1.2.1 Orientation Assignment ................................... 15</td>
</tr>
<tr>
<td></td>
<td>3.1.2.2 Descriptor Extraction .................................... 16</td>
</tr>
<tr>
<td></td>
<td>3.2 Clustering .......................................................... 17</td>
</tr>
<tr>
<td></td>
<td>3.2.1 Distance Measure ............................................. 18</td>
</tr>
<tr>
<td></td>
<td>3.2.2 Types of Clustering .......................................... 18</td>
</tr>
<tr>
<td></td>
<td>3.2.3 K-Means Clustering ......................................... 19</td>
</tr>
<tr>
<td>4</td>
<td>SYSTEM OF ASSOCIATIVE RELATIONS .............................. 21</td>
</tr>
<tr>
<td></td>
<td>4.1 System of Associative Relations (SOAR) .................. 21</td>
</tr>
<tr>
<td></td>
<td>4.2 Similarity Measure ............................................. 23</td>
</tr>
<tr>
<td>5</td>
<td>NEAR-DUPLICATE DETECTION USING SYSTEM OF ASSOCIATIVE RELATIONS ............................................. 25</td>
</tr>
<tr>
<td></td>
<td>5.1 Summary .................................................................. 25</td>
</tr>
<tr>
<td></td>
<td>5.2 Obtaining the Signature ....................................... 25</td>
</tr>
<tr>
<td></td>
<td>5.3 Spatial Signature and Quantization ....................... 27</td>
</tr>
<tr>
<td></td>
<td>5.4 Temporal Signature ............................................. 29</td>
</tr>
</tbody>
</table>
6 RESULTS AND DATABASE .............................................................................................. 32
6.1 The Database ........................................................................................................ 32
  6.1.1 Decrease in Brightness .................................................................................. 32
  6.1.2 Dilation .......................................................................................................... 33
  6.1.3 Erosion ............................................................................................................ 34
6.2 Results .................................................................................................................... 35
  6.2.1 Case 1 ............................................................................................................... 37
  6.2.2 Case 2 ............................................................................................................... 37
  6.2.3 Case 3 ............................................................................................................... 38
6.3 Comparisons .......................................................................................................... 39
7 CONCLUSION AND FUTURE WORK ..................................................................... 45

BIBLIOGRAPHY ............................................................................................................. 46
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>(a) Frame of a video partitioned into equal sized blocks, (b) Average values of each block, and (c) Rank matrix.</td>
<td>5</td>
</tr>
<tr>
<td>2.2</td>
<td>Search window.</td>
<td>9</td>
</tr>
<tr>
<td>3.1</td>
<td>(a) Integral Image representation, and (b) Finding the area of region A using integral images.</td>
<td>14</td>
</tr>
<tr>
<td>3.2</td>
<td>Left to right: the (discretised and cropped) Gaussian second order partial derivatives in y-direction and xy-direction, and approximations thereof using box filters. The grey regions are equal to zero.</td>
<td>14</td>
</tr>
<tr>
<td>3.3</td>
<td>Different intensity patterns.</td>
<td>17</td>
</tr>
<tr>
<td>3.4</td>
<td>Clusters.</td>
<td>18</td>
</tr>
<tr>
<td>3.5</td>
<td>K-means clustering.</td>
<td>20</td>
</tr>
<tr>
<td>4.1</td>
<td>Different token patterns.</td>
<td>22</td>
</tr>
<tr>
<td>4.2</td>
<td>Inter pixel signal strength.</td>
<td>23</td>
</tr>
<tr>
<td>5.1</td>
<td>Block diagram of SOAR NDD.</td>
<td>26</td>
</tr>
<tr>
<td>5.2</td>
<td>SURF feature points.</td>
<td>26</td>
</tr>
<tr>
<td>5.3</td>
<td>K-means clustering.</td>
<td>27</td>
</tr>
<tr>
<td>5.4</td>
<td>2 x 2 ordinal measure.</td>
<td>28</td>
</tr>
<tr>
<td>5.5</td>
<td>(a) Miss-America with cluster centre, (b) 7 x 7 window around the cluster centre, (c) 7 x 7 Full SOAR token, and (d) Inter pixel signal strength.</td>
<td>29</td>
</tr>
<tr>
<td>5.6</td>
<td>(a) Frame 1 key frame and (b) Frame 2 showing the search window.</td>
<td>30</td>
</tr>
<tr>
<td>5.7</td>
<td>Best match (a) Frame (n-1) and (b) Frame n.</td>
<td>30</td>
</tr>
<tr>
<td>6.1</td>
<td>Original video.</td>
<td>32</td>
</tr>
<tr>
<td>6.2</td>
<td>Pixel value reduced by 3.</td>
<td>33</td>
</tr>
<tr>
<td>6.3</td>
<td>Miss-America: PV reduced by 5.</td>
<td>33</td>
</tr>
<tr>
<td>6.4</td>
<td>Miss-America: PV reduced by 6.</td>
<td>33</td>
</tr>
<tr>
<td>6.5</td>
<td>Dilation using 1 x 1 structural element.</td>
<td>34</td>
</tr>
<tr>
<td>6.6</td>
<td>Dilation using 3 x 3 structural element.</td>
<td>34</td>
</tr>
<tr>
<td>6.7</td>
<td>Dilation using 5 x 5 structural element.</td>
<td>35</td>
</tr>
<tr>
<td>6.8</td>
<td>Erosion with a 1 x 1 structural element.</td>
<td>36</td>
</tr>
</tbody>
</table>
Figure 6.9. Erosion with a 3 x 3 structural element ............................................................... 36
Figure 6.10. Erosion with a 5 x 5 structural element ............................................................... 36
Figure 6.11. Reduction in PV by 3; erosion = dilation = 1 x 1-1. ............................................ 37
Figure 6.12. Reduction in PV by 3; erosion = dilation = 1 x 1-2. ............................................ 38
Figure 6.13. Reduction in PV by 3; erosion = dilation = 1 x 1-3. ............................................ 38
Figure 6.14. Reduction in PV by 3; erosion = dilation = 1 x 1-4. ............................................ 39
Figure 6.15. Reduction in PV by 5; erosion = dilation = 3 x 3-1. ............................................ 39
Figure 6.16. Reduction in PV by 5; erosion = dilation = 3 x 3-2. ............................................ 40
Figure 6.17. Reduction in PV by 5; erosion = dilation = 3 x 3-3. ............................................ 40
Figure 6.18. Reduction in PV by 5; erosion = dilation = 3 x 3-4. ............................................ 41
Figure 6.19. Reduction in PV by 6; erosion = dilation = 5 x 5-1. ............................................ 41
Figure 6.20. Reduction in PV by 6; erosion = dilation = 5 x 5-2. ............................................ 42
Figure 6.21. Reduction in PV by 6; erosion = dilation = 5 x 5-3. ............................................ 42
Figure 6.22. Reduction in PV by 6; erosion = dilation = 5 x 5-4. ............................................ 43
Figure 6.23. ROC curve: FNR vs FPR. .................................................................................. 43
ACKNOWLEDGEMENTS

I would like to sincerely express my thanks to Dr. Yusuf Ozturk for giving me an opportunity to work with him. Over the last two years he has offered me guidance and helped me choose the right path. I thank him for being patient with me; believing in me and helping me gain knowledge in the field of image processing.

I would also like to express my thanks to Dr. Sunil Kumar for his help and guidance. The challenging in-class projects given by him helped me in understanding various image processing concepts and paved a way for my research work. I would also like to thank Dr. Roman Swiniarski for having faith in me and accepting to be my committee member.

I would like to thank my family for being there for me and offering their support when it mattered.

I would also like to thank Deepa, Priyanka, Sanmathi and Arun for bearing with me and believing in me.
CHAPTER 1

INTRODUCTION

The tremendous advancement and popularity of the internet during the 1970s gave way to easy data availability and information accessibility worldwide. With this sixth greatest invention of all times, also came a setback, duplications of original documents and web pages became common which led to huge problems in web-crawling and plagiarism.

The research on copyright detection dates back to a few decades and it all started with text documents and web pages. With so much information available on the World Wide Web and thousands of web pages being created every day, presence of duplicates and Near-Duplicates became a common problem. To access all this abundant information, a need for aid from automatic tools became necessary which led to the discovery of search engines. In order to give efficient search results in the first page of the result, the most important task for designers is to eliminate Near-Duplicate web pages which drastically reduce unwanted searches, since a survey shows that about 1.7%-7% of the search results are duplicates or Near-Duplicates. Near-Duplicates are defined as modifications inflicted on original documents and hence detecting Near-Duplicates is slightly trickier than detecting duplicates. Near-Duplicates will not be bit wise similar but will differ only in aspects like the advertisements displayed, time stamps or some other form of minor changes.

Near-Duplicate Detection (NDD) is challenging because of scalability and speed constraints. Any search engine searches billions of web pages every day and to store all of these searches multi terabytes of storage space is required. In order to reduce indexing of high amount of database Near-Duplicates should be detected at very high speeds. Hence if a solution to NDD is found, not only large amounts of redundant information can be eliminated but also efficient search engines can be designed and copyright issues can be addressed with greater speeds and accuracy.

However the above mentioned challenges are not easy to deal with and researchers have come up with various methods to address the issue of NDD. One of the earliest techniques proposed by Broder et al. [1] in 1997 is based on tokenizing documents into a list
of words that are similar. It uses a syntactic approach, more popularly known as shingling where all word sequences of adjacent words are extracted and if two documents contain the same set of shingles or similar set of shingles to a large extent they are termed as duplicates and Near-Duplicates respectively. Di Lucca et al. [2] proposed a technique to detect duplicate web pages using HTML and ASP scripts. Analysis of a number of web pages and web application was carried out for this technique. Bar Youssef et al. [3] presented a novel method called DUST (Different URLs and Similar Text) for detecting duplicate URLs. Server logs were used to come up with rules that resulted in transforming URLs to other URLs of similar content. Cho et al. [4] used clustering techniques to separate out web pages of similar content. Using this technique they reported that not only web-crawling was reduced by almost 40% but also the search results were arranged in better way for end users. Chowdery et al. [5] used collection statistics to come up with a novel method for NDD. Using a vast database of documents with different lengths, different degrees of near duplication they reported higher efficiency and reduction in computational speeds by one fifth.

Researchers all over the world also came up with tools which would rapidly detect Near-Duplicate web pages. One such tool was introduced by Cooper et al. [6]. A rapid phase recognizer system was used to come up with the most important words and phrases in a document which was then compared with other documents in the database by running a simple database query. If the number of words that is not present in both the documents is below a pre-determined threshold then the documents are termed similar. To avoid plagiarism Finkel et al. [7] proposed a web accessible tool which uses signature extraction. A unique signature for each web document is found and stored which can then be compared with a database of documents to identify similar documents. Jalbert et al. [8] proposed an automatic tool which would report duplicate bugs using surface features, textual semantics, and graph clustering. There are many more tools which aim at increasing the accuracy and efficiency of detecting Near-Duplicates on the web. New and faster solutions are being found everyday to cater to this problem.

After having made considerable improvement in detecting web pages, the next challenge was to detect Near-Duplicates in images and videos.
With growing popularity of the World Wide Web, multimedia content has become easily available and widely used and with live televisions, news, radio stations and music available at finger tips, the problem of duplication also spreads to images and videos.

The challenges faced in detecting Near-Duplicate Images and videos are slightly different from that of web pages. In order to understand the challenges involved in images and videos we first need to dig deeper into image and video analysis.

Most state of the art methods are content based i.e., depending on the content of the image important features are extracted. Human visual system plays an important role in recognizing important features in an image. Some of the important characteristics of a human eye are:

1. Human eye is sensitive to edges (changes in frequency).
2. Human eye is not very sensitive to high frequency components and hence high frequency components can be neglected with very little loss of data.
3. Human eye is less sensitive to changes in color. This allows us to give more importance to luminance components over chrominance components.

With this information a general deduction can be made as to which part of the image is more important. Keeping these characteristics in mind many algorithms were proposed for feature point detection and object recognition in images. Video in most cases is imagined to be a series of images put together and the algorithms developed for images are extended for videos. In a video feature points are detected for a key frame and these features are tracked in consecutive frames to ensure the robustness of the points. The biggest challenge in videos is to consider temporal (within the frame) and spatial (between consecutive frames) information of the video and come up with a signature which is unique, robust and simple.

There are different kinds of videos available on the internet today, from slow moving videos which has very little change in information from one frame to another to very fast moving videos with too many scene changes from one frame to another. A technique which can cover a database of such dissimilar videos and simultaneously cater to variety of applications is said to be more efficient. For instance, video conferencing or live broadcast of events will have very strict time constraints, in which case quality is compromised to achieve high speeds. Hence algorithms with very low computational complexities are used, whereas
for stored videos, more importance is given to accuracy and precision even if it takes a longer time.

The process of Near-Duplicate Detection (NDD) in videos can broadly be divided into three main steps;

1. Key-Frame Detection: An important frame in a video sequence is first detected.
2. Feature Extraction: Important feature points depending on the type of video and the application are extracted.
3. Tracking: These feature points are tracked in subsequent frames to retain robust points throughout the video.

Every algorithm differs in the kind of method used to perform these steps. Although the challenges are many, applications are plenty too. Here are some of the major applications of finding Near-Duplicate and duplicate videos on the web;

- Illegal use of images/videos: With ever increasing social networking sites and other websites, image and video traffic also seems to be increasing at linearally. This has led to use of videos and images illegally without adhering to the terms of the copyright. This major issue can be avoided with NDD which would be highly beneficial to free video sharing websites like you tube.

- Media tracking: Media tracking is the idea of knowing when and where a piece of information is being used. Monitoring a TV commercial is a very important type of media tracking. Competitor companies would like to keep track of when and by whom a particular commercial is being watched so that their marketing strategies can be changed accordingly.

- Efficient search results for search engines: It is seen that more than 40% of the web traffic is because of duplicates and Near-Duplicates. Elimination of Near-Duplicates will eliminate almost 7% of the searches giving the user more efficient search results.

Having seen the potential research opportunities in the area of NDD, this thesis mainly focuses on detecting Near-Duplicates in a video. The detection algorithm is based on System of Associative Relations (SOAR) and aims at reducing the computational complexity thereby increasing the speed at which Near-Duplicates are detected. A detailed literature review of the state of the art NDD methods for video is given in Chapter 2, with feature point detection explained in Chapter 3, SOAR is described in Chapter 4 and its extension to Near-Duplicate detection is described in Chapter 5 and all the experimental results are shown in Chapter 6.
CHAPTER 2

RELATED RESEARCH

Content based copy detection has been a widely researched area for the past decade. One of the very first methods used color histogram based matching for its simplicity, but the efficiency of color histogram is not very good as it does not exploit the spatial information in a video. The state of the art methods vary from being as simple as using just an ordinal measure to as complicated as using a probabilistic approach.

Changick Kim et al. [9] exploited both the spatial and temporal information in a video to come up with an efficient algorithm using ordinal measure, which was used earlier for stereo matching. Here, an image is divided into blocks of size m x n. An average value of all the pixels in this m x n window is found as shown in Figure 2.1(b) and this average value is converted to a rank matrix, wherein the lowest average value is given a rank of 1 and the highest average value is given a rank of ‘N’, where N is the number of m x n blocks in Figure 2.1(a). Thus in Figure 2.1(c) 37 would get a value of 1, 44 would get a value of 2 and so on.

The rank matrix obtained in Figure 2.1(c) is converted to a one dimensional array to obtain a 1 x N rank matrix for N blocks. If this rank matrix is for the \(i^{th}\) frame of the original video, then a rank matrix of the same dimension is obtained for the \(i^{th}\) frame of a query video and a spatial distance measure is now obtained between these two rank matrices by taking the difference between them given by,
where, \(d\) is the normalized distance and \(C\) is the worst case scenario distance between the two rank matrices, which is when both the rank matrices are the exact complements of each other. \(\pi_{q,i}\) is the \(i^{th}\) frame of the query video and \(\pi_{t,p+i}\) is the \(i^{th}\) frame of the target video.

After averaging the distance in equation (1) over all \(N\) frames of the video, a temporal distance \(D_t\) is computed by dividing each frame into four quadrants and comparing the pixel value at each of these quadrants from one frame to another. The normalized temporal distance is given as,

\[
D_t(\pi_{q,i}, \pi_{t,p+i}) = \frac{1}{C} \sum_{i=1}^{m} |\pi_{q,i}^j - \pi_{t,p+i}^j|
\]  

**Equation (1)**

In equation (2) \(V_q\) represents the query video clip, \([V_t[p:p+N-1]\) is a sub video of a target video used to measure the dissimilarity, \(d_t(\delta_q, \delta_t)\) is the normalized temporal distance and is defined by equation (3).

\[
d_t(\delta_q, \delta_t) = \frac{1}{4(N-1)} \sum_{i=1}^{4} \left( \sum_{i=1}^{N-1} \|f(\delta_q^i) - \delta_t^i\| \right)
\]  

**Equation (2)**

where,

\[
f(x) = \frac{|x|}{2}
\]  

**Equation (4)**

and \(\delta_q^i\) is defined as,

\[
\delta_q^i = \begin{cases} 
1 & \text{if } V_q^i > V_{i-1}^i \\
0 & \text{if } V_q^i = V_{i-1}^i \\
-1 & \text{if } V_q^i < V_{i-1}^i 
\end{cases}
\]  

**Equation (5)**

The variable \(j\) in equations (3) and (5) runs from 1 to the number of partitions in the video which in this algorithm is assumed to be 4 and the variable \(i\) runs from 1 to \(N-1\), where \(N\) is the number of frames in the video. Both the spatial and temporal distance is combined to obtain the final distance measure, the spatial distance is multiplied by a factor of \(\alpha\) and the temporal distance by a factor of \((1-\alpha)\) to observe the contribution of the two distances. Experimental results of this algorithm shows that the best results were obtained for an \(\alpha\) value of 0.5. The drawback of this method is huge computations which can be reduced by using a more robust algorithm for the detection of the initial feature points as described in detail in Chapter 5.
In [10] feature points are detected using SIFT as described by David Lowe [11]. A 3 x 3 image patch is used to describe each of the feature points and sum of differences are computed between each pairs of these patches. Once the sum of differences is obtained, Average Difference Per Pixel is obtained (ADPP) by dividing the sum of differences by total number of pixels in the grid. Use of integral images makes this process 10 times faster. If there are nine different patches, a 36 dimension Local-Difference Pattern (LDP) feature descriptor is obtained and when this is extended to a video, a comparison is made between consecutive frames and hence 18 image patches will be present instead of 9 which will result in a 144 dimension descriptor instead of a 36 dimension descriptor.

Each of these 144 descriptors is further quantized by using two bits for each descriptor. Two bits are used to increase the uniqueness of the signature. If the feature value is between a threshold T and 255 it will be encoded as ‘11’, if it is in the range (0, T) it will be encoded as ‘10’, (0, -T) as ‘01’ and (-T, -255) as ‘00’ respectively. This results in a 288 bit descriptor between consecutive frames of a video.

In [12], a method is proposed for Near-Duplicate detection on the fly. The video sequence is represented by a high dimensional feature vector \( \mathbf{f}_i \) using a color histogram as shown in equation (6). A Video Distance Trajectory (VDT) which reduces the N-dimensional vector into one dimensional vector is defined as shown in equation (7).

\[
V = \{f_1, f_2, f_3, \ldots , f_N\}
\]

\[
VDT = \{d(f_1, o), d(f_2, o), \ldots , d(f_w, o)\}
\]

The ‘o’ in equation (7) represents the reference point and d is the Euclidean distance measure between the feature point and the reference point. In order to find an optimum reference point, Principal Component Analysis (PCA) is used as it is seen that a reference point which can maximally differentiate frames, always lies in the direction determined by the first PCA. A Linear Smoothing Function (LSF) was described to represent the VDT, line segments in a VDT represent shots (basic unit in a video), sudden changes in a VDT correspond to boundaries of a video, fade-in and fade-out in a video can also be determined by observing the shape of the VDT. LSF encodes VDT in such a way that important information in a video is preserved i.e., LSF is defined as the triplet \((\alpha, \beta, l)\) where, \(\alpha\) and \(\beta\) form a line which fits the VDT segment with minimum SSE and \(l\) represents the length of the
line segment. With this method a better precision was reported when compared to other existing similar method.

Another real time video clip detection method is described in [13]. UQLIPS proposes two new methods, Bounded Co-ordinate System (BCS) and FRAme Symbolization (FRAS). BCS concentrates on the spatial information and comes up with a compact, global signature exploiting the dominant content of the video and FRAS assigns symbols to each clip in order to take the temporal information into consideration. BCS is used to represent the N-dimensional feature vector, obtained from any feature extraction method, using just a reference point and line segments. If the video sequence is represented by $M = \{m_1, m_2, \ldots, m_n\}$, where $m_i$ is a D-dimensional feature vector, then the Bounded Co-ordinate System of this video sequence is as defined in equation (8).

$$\text{BCS}(M) = (O, \phi_1, \phi_2, \ldots, \phi_d)$$

Where, $O$ represents the reference point, which is the mean of all the D-dimensional feature vectors, $\phi$ represents the orientation. Each point has a reference and d directional line segments to represent a clip. The video similarity between two sequences can be obtained by comparing their BCSs which are compared by performing operations like translation, rotation and scale invariance.

FRAS performs clustering over a whole frame to come up with a dictionary of symbols, each of these dictionary elements will represent a small cluster of the frame, whose radius is lesser than a predetermined threshold. These sub frame clusters is represented as \{cluster Id, cluster centre, radius and number of frames\}. For a given video sequence, $M = \{m_1, m_2, \ldots, m_n\}$, the mapping from frame $x_i$ to the corresponding symbols $s_i$, $S = \{s_1, s_2, \ldots, s_n\}$ can be done easily and compared using a Probability Edit Distance (PED). In PED the similarity between two symbols is measured by computing the probability between the two symbols.

In [14] fast signatures using color histograms to describe the global features and computationally slow but more efficient local signatures are used to eliminate Near-Duplicate and duplicate videos in web searches, thereby making search engines more productive and efficient. Using this algorithm, the desired search data can be made available to the user in the first page within the first few search results itself. Using color histogram, a global signature, $H_i = \{h_1, h_2, \ldots, h_m\}$ with 18 bins for hue, 3 for saturation and 3 for value in
the HSV color model is described with the value of the subscript \( m \) adding up to 24. Thus the video signature (VS) becomes an \( m \)-dimensional vector given by equation (9).

\[
VS = (s_1, s_2, \ldots, s_m)
\]  

(9)

where,

\[
s_i = \frac{1}{n} \sum_{j=1}^{n} h_{ij}
\]

(10)

Given \( n \) key frames, \( h_{ij} \) in equation (10) refers to the histogram of the \( i^{th} \) bin in the \( j^{th} \) key frame. The computational complexity in this algorithm is reduced by eliminating extremely dissimilar videos just by comparing the global signatures using a Euclidean distance measure. Local signatures are computed only for those videos whose distance measure is lesser than a predetermined threshold indicating that the video is quite similar to the query video. Since more than about 20% of the videos in a web search are duplicates of each other and a lot more are Near-Duplicates with very little deviation from the original content, use of the threshold helps eliminate most of the videos leading to a faster and more efficient algorithm.

In order to detect a Near-Duplicate key frame, a Hessian Affine matrix is used to find the feature points and these feature points are described using PCA SIFT which results in a 36 dimensional feature vector. A video is classified as similar if the number of matching feature points within a search window in two videos exceeds a predetermined threshold value. However defining a search window around the key point will not increase the false positives as similar points are mostly seen around the same region in subsequent frames of a video, it will only lead to reduction in unwanted comparisons. Given two video sequences the search window is defined as shown in Figure (2.2).

\[
\text{max}(1, i - df - w) \quad \text{min}(1 + df + w, n)
\]

Figure 2.2. Search window.
If the length of the video sequence is $n$, then the range of the search window is defined as, 

$$Range = [\max(1, i - df - w), \min(1 + df + w, n)]$$

where, ‘$w$’ is the window size and ‘$df$’ is the difference between the lengths of the two video sequences.
CHAPTER 3

SURF AND CLUSTERING

3.1 SURF

All parts of an image is not of importance and being able to efficiently extract parts of a video which are of importance becomes the first step in any image processing algorithm. Feature extraction mainly involves finding interest points which are usually the corners, blobs or other significant parts in an image. These interest points should have a very high repeatability rate i.e., the same interest points should be detected at different angles of rotation or at different resolutions or in different camera views in order to classify it as a stable feature point. After detecting the interest points the neighboring points are observed and a unique feature vector is assigned to each of the interest points.

Speeded Up Robust Feature (SURF) proposed by Herbert Bay et al. [15] takes all the important aspects of SIFT [11] and increases the speed of computation with the use of integral images, thus giving a fast, robust, scale and rotation invariant feature detector and descriptor with lesser feature points. Only scale invariance and rotation invariance was considered as it was found that this covered most of the common variants that can be done on an image and this seemed to be a good compromise between achieving robustness and reducing computational complexity. In most cases it was seen that even rotation invariance is not required and in such cases a scale invariant form of SURF can be used which is called ‘Upright’ SURF (U-SURF) [15].

There are various algorithms that have been proposed for feature detection and description. Harris corner detector [16] is the most commonly used detector for its simplicity but it is not scale invariant. An improved version of Harris corner detector, the Hessian-Laplace detector which makes use of the determinant of Hessian matrix to select the location and Laplacian to select the scale was implemented by Mikolajczyk et al. [17]. David Lowe [11] was the one who made drastic improvements in the speed of computation by describing the Laplacian of Gaussian function using a Difference of Gaussian (DOG) filter. Lowe’s Scale Invariant Feature Transform (SIFT) [11] has outperformed the other existing
methods by using a locally oriented gradient around the interest point to describe the interest points and stores the bins in a 128-dimensional vector. One of the improvements suggested by Ke et al. [18] was to apply Principal Component Analysis on the 128 dimensional vector which reduced the size of the vector to 36 dimensions, but it was seen that the distinctiveness was reduced when compared to SIFT and the complexity in feature extraction nullified the reduction in vector length. After studying various published comparisons it can be concluded that Hessian based detectors are more stable and repeatable than Harris based detectors and approximations like DoG can bring speed at a very low cost.

3.1.1 Interest Point Detector

SURF uses Hessian based detector for interest point detection but uses integral images and a very basic approximation which makes it computationally faster and hence the name fast Hessian Detector.

Hessian detector uses Hessian matrix to find interest points which are scale invariant in multiple scales. Equation (11) defines a 2 x 2 Hessian matrix at scale $\sigma$.

$$H(x, \sigma) = \begin{bmatrix} D_{xx}(x, \sigma) & D_{xy}(x, \sigma) \\ D_{xy}(x, \sigma) & D_{yy}(x, \sigma) \end{bmatrix}$$ (11)

In equation (11), $D_{xx}(x, \sigma)$ and $D_{yy}(x, \sigma)$ are the second order partial derivatives in the x and y directions respectively and $D_{xy}(x, \sigma)$ is the second order partial derivative in the x-y direction. Each of these derivatives are computed by convolving a Gaussian second order derivative $\frac{\partial^2}{\partial x^2} g(\sigma)$ with the image $I$ in the current iteration scale and derivatives must then be scaled appropriately to account for multiple scales.

Interest points at each scale are determined by computing the determinant and trace of the Hessian matrix using the relations given in equations (12) and (13).

$$\text{Det}(H) = D_{xx}D_{yy} - (D_{xy})^2$$ (12)

$$\text{Tr}(H) = D_{xx} + D_{yy}$$ (13)

The local extrema of both the determinant and trace is considered as the interest point in each scale. Hessian detector is spatially localized using Laplacian of Gaussian (LoG) and hence the name Hessian-Laplace detector. But in Fast-Hessian detector that is used in SURF the hessian matrix is used to compute both the scale and location.
Gaussians are widely used in scale space analysis because of the ease of computation and the robustness that it offers, but Gaussian functions will have to be cropped and discretised in order to avoid aliasing. However even if Gaussian filters are used it is seen that aliasing still persists when the image is sub-sampled. Also, Gaussian functions have a property that no new structures can be created when the resolution of the image is lowered, but this seems to work only for 1-D images and the same cannot be successfully replicated for a 2-D image. Thus considering these drawbacks of Gaussian and knowing that Gaussian functions are anyway non-ideal, in SURF the authors approximate Gaussian functions, combining the LoG used by Lowe in [11] with box filters to further increase the speed of computation.

### 3.1.1.1 INTEGRAL IMAGES

Using box filters to generate approximations of second order Gaussian functions, generate comparable results to that of the conventional discretisation and cropping and proves to be much faster by the use of integral images.

Two dimensional image features can be computed at a very fast rate using integral images. Integral image can be computed using equation (14).

\[
\sum_{i=0}^{x} \sum_{j=0}^{y} \text{I}(i, j)
\]

where, \( I(x) \) is the Integral Image and \( I(i,j) \) is the Input Image. Thus, integral image is calculated by the sum of the values between the point and the origin. Once the integral image is computed, calculation of the area of any rectangle will just be four operations.

One pass over the entire image is enough to compute the integral image through an iterative method shown by equations (15) and (16) where \( m(x, y) \) in equation (15) represents the cumulative row sum.

\[
m(x, y) = m(x, y - 1) + I(x, y) \quad (15)
\]

\[
\sum_{i=0}^{x} \sum_{j=0}^{y} \text{I}(i, j)
\]

Let us consider an integral image as shown in Figure 3.1(a), to understand the computation of integral images better.

Figure 3.1(b) shows a region ‘A’ engulfed in a rectangle EFGH. The area of A can be computed using just four reference images as given by equation (17).
Equation (17) holds good for any size of the rectangle and hence integral images prove to be very useful when computing large areas.

### 3.1.1.2 BOX FILTERS

Box filters give a fairly good approximation of the Gaussian function and the ease of computation makes it a very attractive solution. Figure 3.2 shows a 9 x 9 box filter and also the discretized and cropped Gaussian function. It can clearly be seen that comparable results are generated between the conventional discretisation and cropping and box filters.

In order to achieve scale invariance, most scale spaces have a pyramidal structure where a smoothening Gaussian filter is applied at each scale and subsequently sub-sampled to get to a higher level of the pyramid. In SURF the use of integral images and box filters avoids applying the same filter to the output of the previous layer, instead the filter size of the box filters can be up scaled and applied at the same speed to the original image to obtain the
different layers of the pyramid. In [15] a scale of 1.2 is considered the base layer which corresponds to the 9 x 9 box filter shown in Figure 3.2. The subsequent higher layers of the pyramid are obtained by filtering the image with masks of sizes 15 x 15, 21 x 21, 27 x 27. Since the ratio of the filter sizes remain constant after scaling, even the Gaussian derivates scale accordingly, i.e., a 21 x 21 filter would correspond to a scale of $\sigma = 5/3 \times 1.2 = 2$. It should be noted that the difference between the sizes of the filter from one scale to another gets doubled for every new octave (6 to 12 to 24 and so on).

3.1.2 SURF Descriptor

As discussed earlier SIFT [11] descriptor has been the most popular descriptor, combining localized and gradient features to achieve a scale invariant distinct feature detector. However SURF reduces the dimensions of the feature descriptor from 128 dimensions in SIFT to just 36 dimensions with little or no compromise in scale invariance and uniqueness of the feature descriptor. With the help of information obtained from a circular region around the interest point, an orientation is found such that it is repeatable at different scales and camera views. A square region is constructed around the obtained orientation from which the descriptor is finally derived.

3.1.2.1 ORIENTATION ASSIGNMENT

In order to understand orientation assignment we need to know a little about wavelet transform. Wavelet transform is defined as using a certain orthonormal series to represent a square integral function. Haar wavelets are the simplest of the wavelet transforms which cannot be differentiated as it is discrete. A mother Haar wavelet is defined as shown in equation (18).

$$\psi(t) = \begin{cases} 1 & 0 \leq t < \frac{1}{2} \\ -1 & \frac{1}{2} \leq t < 1 \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

Haar wavelets are used for finding a reproducible orientation which is computed in the x and y direction in a circular region of radius six times that of the current scale of the image. It should also be noted that sampling is done at a sample interval equal to s, the current scale of the image, along with finding the wavelets at each scale. Hence very big
Haar wavelets are obtained for higher scales and so integral images is used here as well to make computations fast. Use of integral images reduces the number of operations to just six for computing Haar wavelets in x and y direction in any scale. After computing the Haar responses they are weighted with a Gaussian centered around the interest point and the most dominant direction is found by summing up all the responses within a sliding window, covering an angle of $\pi/3$. The horizontal and vertical responses are then added to obtain a new vector length and the longest vector length will be the dominant direction of the interest point and the one that eventually represents the interest point. The size of the sliding window is the choice of the user and can be chosen depending upon the image being tested. If the size of the window is chosen to be too small, then only a single wavelet response will be recognized which may not be the actual dominant orientation and if the size of the window is too big, then the maxima may be detected at a orientation which may not be the dominant orientation for the current interest point. Hence a window size can be determined experimentally. U-SURF or Upright-SURF mentioned earlier skips this step and does not provide rotational invariance, but speeds up the process and is highly helpful in images where the camera is still or just moves horizontally.

### 3.1.2.2 Descriptor Extraction

The extraction of a feature vector or a descriptor involves taking a square region around the interest point, placed at an orientation obtained from the orientation assignment step discussed earlier. This transformation will not be necessary for U-SURF as the orientation assignment step is skipped in it. In [15] the radius of this square window is chosen to be 20 times the scale. This square region is then divided into sub regions of dimensions 4 x 4 and a few simple features are extracted in each of these sub regions. Haar wavelet responses are found in the horizontal and vertical direction defined with respect to the dominant orientation. The horizontal and vertical responses, $d_x$ and $d_y$ are convolved with a Gaussian function to increase the robustness and reduce localization errors as was done previously.

All of the obtained Haar responses are summed up over each sub region and a sum of the absolute differences of these are found out to account for polarity changes in the intensity. Thus this will result in a four dimensional feature vector containing $\sum d_x, \sum d_y$. 
\[ \sum |dx| \text{ and } \sum |dy| \] and the size of the descriptor vector for all of the sub regions will be 64.

Let us consider different images with very different intensities as shown in Figure 3.3 and see how the distinctiveness is achieved.

![Figure 3.3. Different intensity patterns.](image)

In case of the Figure 3.3(a), where there is no gradient change in intensity all the four values, namely, \( \sum dx \), \( \sum dy \), \( \sum |dx| \) and \( \sum |dy| \) are low. Figure 3.3(b) has frequency changes only in the x direction and hence the value of \( \sum |dx| \) is high and all the other values are low. If there is a gradient change in the intensity in the x direction, like in Figure 3.3(c), both \( \sum dx \) and \( \sum |dx| \) are high. When different combinations of such intensity variations combine in a large image, it can easily be distinguished by looking at the descriptor. Herbert Bay et al. [15] conducted various experiments with higher order wavelets, Principal Component Analysis, median values, different window and sub-region sizes etc. to come with the most optimal solution. A faster method SURF-36 was proposed which uses 3 x 3 sub regions instead of 4 x 4 sub regions which results in a 64 length vector. The 3 x 3 sub region worsens the performance w.r.t SURF-64 but gives comparable results to other state of the art methods and enables very fast computation speeds for matching.

### 3.2 Clustering

The feature extraction algorithm, SURF, described in Section 3.1 has a very high efficiency of detecting very stable and unique points, but this exhaustive detail comes with a cost, a huge number of feature points! On an average SURF comes up with about 70 feature points which results in huge amount of data. Such large scale data not only results in heavy computations but also needs a lot of memory. In order to reduce such complexities, clustering is done to eliminate all points which are very close to each other.

Cluster analysis is a process of assigning a set of observations into subsets such that all the observations in each subset are similar in some sense. Figure 3.4 illustrates cluster
analysis by separating clusters based on a certain similarity. Each color indicates a different cluster [19].

3.2.1 Distance Measure

The first step of any clustering method is to choose a distance measure which determines the similarity between different observations. The following distance measure is usually used by most researchers.

- **Euclidean Distance**: computed by taking the nth root of the sum of squares between co-ordinates in an n-dimensional space.

- **Hamming Distance**: gives the measure of the number of substitutes required to change one form to another. This is usually used for binary distances. The number of bit changes are measured which helps in determining the distance between two vectors.

3.2.2 Types of Clustering

Depending on the way clusters are created, clustering can be classified into various types. They are:

- **Hierarchical clustering**: Clustering based on this method use previously generated clusters to find new clusters. Agglomerative clustering is one such method where in each data set is combined to form clusters iteratively. Figure 3.4 elaborates this approach, iteratively each of the points is grouped into three different clusters and the corresponding color was assigned to make a distinction.

- **Partitional Clustering**: Clustering based on this algorithm determines all clusters at once unlike Hierarchical clustering. A lot of clustering algorithms in use today come under this category. Some of the examples are K-means, Fuzzy-c logic, QT(Quality Threshold), Locality Sensitive Hashing etc.

- **Subspace Clustering**: Clustering based on this algorithm look at clusters that are only visible in a particular sub space. Correlation based clustering falls under this category.
3.2.3 K-Means Clustering

K-means clustering is a very widely used clustering algorithm both for its simplicity and speed. Since speed is one of the main criterion for our algorithm, K-means clustering is a perfect choice. K-means clustering is a learning algorithm which starts off with random clusters, and then builds on it with each iteration. It is important to know that the cluster centers in K-means clustering is computed by finding the average of all the points that belong to that cluster. K-means clustering algorithm follows the steps mentioned below;

1. Number of clusters(N) required is chosen
2. Randomly N clusters are generated and the centers of these generated clusters are found. Random points can be chosen to act as cluster centers as well.
3. Each point is now associated with the nearest cluster centre. The nearest measure is obtained by choosing any of the Distance Measures discussed in Section 3.2.1.
4. The new Cluster centers are computed.

Steps 3 and 4 are repeated until the cluster centres converge, i.e., until the cluster centre from the previous iteration is retained.

Figure 3.5 illustrates the K-means clustering algorithm with two cluster centers m and n. The path taken by m and n to reach the cluster centre is shown by black and gray cross marks respectively. From Figure 3.5 it can also be observed that after four iterations the cluster centers are localized and do not change. The final boundary between the two clusters is shown as a solid black line. This is a fairly simple case with just two cluster centers, however in computer vision algorithms the number of clusters are many. Although K-means clustering offers simplicity and speed of calculation, randomly chosen initial clusters generate different cluster centers each time the clustering algorithm is run. This is one of the biggest disadvantages and it can sometimes be overcome by re-running the algorithm a few times before a decision is taken on the final cluster centers.
Figure 3.5. K-means clustering.
CHAPTER 4

SYSTEM OF ASSOCIATIVE RELATIONS

4.1 SYSTEM OF ASSOCIATIVE RELATIONS (SOAR)

SOAR bases its algorithm on associative memory. Associative memory is a content based memory storage technique which is used widely in neural networks. A Random Access Memory (RAM) as we all know returns a data when a memory location is specified, where as a Content Addressable Memory (CAM) or Associative memory takes the data as an input and searches for it in the entire memory space, it then returns multiple address locations where the data was found. In one search operation CAM searches the whole memory, hence it is a lot faster than RAM, but this speed comes with a cost, each memory location will have to have its own comparators to compare the stored data with the input data, this increases the size and in turn the manufacturing cost. However in computer vision applications and neural network applications CAM gives far better results and is frequently used.

SOAR has proposed a new similarity measure that can be used for texture analysis and classification. SOAR is an extension of a technique called Multi Valued Recursive Network, which is a non linear network based on associative memory and it gives comparable results to that of Mean Square Error (MSE). The biggest advantage of SOAR is that it uses a similarity measure which uses simple comparisons instead of the sum of square errors used in MSE or Mean Absolute Difference (MAD). In this thesis we have tried to extend the applications of SOAR to NDD and motion estimation. Use of SOAR considerably speeds up the process of comparisons and reduces the hardware complexity as it does not use any complex operations like multiplications and divisions.

In order to compute the pair wise pixel interaction, SOAR defines non uniform masks called tokens. The size and shape of the token can be varied by the user to suit his or her needs. A token can vary from being a simple 3 x 3 window to a more complex circular window. Figure 4.1 shows examples of a couple of possible token shapes.

Upon deciding the size of the token, each token is associated with a pixel in the image and this in turn is linked to a processing node, which decides the relations between pixels.
Figure 4.1. Different token patterns.

Thus for the token shape in Figure 4.1(a) 5 processing nodes are required and for that in Figure 4.1(b) 9 processing nodes are required. If we assume that each of these nodes are uni-directionally connected to one another, then for the token in Figure 4.1(b) 72 links are required which can be reduced to just 36 links for very obvious reasons that is discussed later. Each of these nine processing nodes are connected to one another by signal strengths called Inter Pixel connection strength represented as \( T(i, j; k, l) \), and defined as shown in equation (19).

\[
T(i, j; k, l) = \text{sgn}(V_{ij}, V_{kl}) \text{ whr } l \leq i, j, k, l \leq q
\]

\[
\text{except: } i, j \neq q
\]

(19)

\{V_{ij}; l = 1, 2, \ldots, q\} is the pixel intensity and the signum function, sgn is defined in equation (20).

\[
\text{sgn}(x) = \begin{cases} 
1 & x > 0 \\
0 & x = 0 \\
-1 & x < 0 
\end{cases}
\]

(20)

The degree of association between pixels can be determined by larger/smaller pixel values within the window, i.e., if the value of the centre pixel (the pixel whose relationship with the other pixels is being determined) is greater than the pixel with which the comparison is being made, a ‘1’ is used for representing the signal strength, if it smaller a ‘-1’ is encoded else if they are equal a ‘0’ is encoded, which by and large is represented mathematically in equation (20). When the connection strengths behave in such a way so that \( T(k, l; i, j) = -T(i, j; k, l) \), then the connection strengths are said to be anti-symmetric, i.e., if one signal is increasing, the other signal will be decreasing by the same amount. Figure 4.2 shows the signal strength for the token shape in Figure 4.1(b).
Equation (19) defines the signal strength for a single block, but this can be extended to M number of blocks just by taking the summation over all blocks as represented in equation (21) [20].

\[
T_p(i, j; k, l) = \sum_{p=1}^{M} sgn(V_{ij}^p, V_{kl}^p) \text{ whr } l \leq i, j, k, l \leq q \\
\text{except: } i, j \neq q
\]

(21)

**4.2 Similarity Measure**

After having established the connection strengths between the processing nodes in a block, a similarity measure can be used to compare this signal strength with the signal strength of other blocks. This comparative measure is denoted by an Energy function as shown in equation (22).

\[
E = \sum_i \sum_j \sum_k \sum_l T(i, j; k, l) * sgn(V_{ij} - V_{kl})
\]

(22)

\(T(i, j; k, l)\) in equation (22) gives the strength measure at the first location and \(sgn(V_{ij} - V_{kl})\) gives the signal strength at the second location. The signal strengths are multiplied and summed to cover the whole region of the two tokens. On substituting for \(T(i, j; k, l)\) in equation (21), we obtain equation (23) which helps in understanding and deducing few basic properties of this energy measure.

\[
E = \sum_i \sum_j \sum_k \sum_l sgn(V_{ij}^p - V_{kl}^p) * sgn(V_{ij} - V_{kl})
\]

(23)
If equation (23) is observed closely it can be deduced that $E$ attains a maximum value when the two regions are similar which corresponds to a correlation coefficient of 1, a minimum value when the patterns are similar but in the opposite direction which corresponds to a correlation coefficient of -1 and a value of 0 or close to zero when the two patterns are dissimilar. This similarity measure is a correlation indication function and is very similar to the correlation in single pattern functions. In [20] it is reported that a comparative study between MSE and SOAR yielded a high degree of correlation for an 8 x 8 SOAR token as compared to a 4 x 4 SOAR token because an 8 x 8 token contains more neighboring pixel information than a 4 x 4 token. Another important aspect of SOAR is that it is not dependent on the mean because the ordering between pixels is considered and not the pixel value itself. Thus the amount of difference between the pixel values don’t matter, the only thing that matters is whether the pixel value is greater than, lesser than or equal to the interest point.
CHAPTER 5
NEAR-DUPLICATE DETECTION USING SYSTEM OF ASSOCIATIVE RELATIONS

5.1 SUMMARY
SOAR NDD aims at extending the application of SOAR to Near-Duplicate Detection, implicitly also establishing its usage in motion estimation and motion tracking. Duplicate detection involves a lot of comparisons and inter-pixel relations, hence SOAR with its ability to establish inter pixel strengths at high speeds becomes the ideal choice. The energy function of SOAR, defined in equation (23) is used periodically for comparisons and wherever possible SOAR is used in place of Mean Square Error because of the high correlation between the two.

As described earlier, the basic principle behind any NDD algorithm is to detect a unique signature of the video and then compare this signature vector with other signatures in the database. This signature should be compact and robust to changes in brightness, frame rate, resolution etc. We propose to use SOAR to come up with this unique signature, taking the temporal distance into consideration which is nothing but the similarity that can be extracted from one frame to another.

5.2 OBTAINING THE SIGNATURE
The block diagram in Figure 5.1 gives a pictorial representation of the algorithm. The key frame of both the query and the non-query video follow the same path, whereas, a non key frame will bypass both the SURF and clustering stages. SURF as described in Chapter 3, is used to obtain feature points in the first frame of the video. The blue circles shown in Figure 5.2 represent the feature points that are obtained. An open SURF algorithm as described in [21] was used to arrive at the SURF points.

As can be seen these feature points describe the most important parts of the video which is both scale and rotation invariant, but the density of the feature points is too high and
reasoning says that all of these feature points are not needed in obtaining the signature. This will only lead to unwanted computations resulting in a very slow algorithm. Hence after obtaining the SURF feature points, the video is passed through a K-means Clustering block. K-means clustering uses the technique as described in Section 3.2 to reduce the high density
of feature points seen in Figure 5.2 to just five points, shown as red ‘X’s in Figure 5.3. The distance measure used for K-means clustering is L1 distance which is obtained by finding the modulus of the difference between the two data points, mathematically represented as given in equation 24:

$$D(x^i, y^i) = \sum_{i=1}^{N} |x^i - y^i|$$

where $x$ and $y$ are N dimensional vectors as described in equations 25 and 26:

$$x^i = (x^1, x^2, \ldots, x^N)$$

$$y^i = (y^1, y^2, \ldots, y^N)$$

Figure 5.3. K-means clustering.

After applying K-means clustering, only the cluster centers which is a representation of each of the five clusters is used for further processing. For instance only the points with co-ordinates (88, 56), (114, 57), (137, 100), (83, 109) and (65, 86) from Figure 5.4 are used to compute the SOAR signature and also for tracking along subsequent frames of the video.

**5.3 Spatial Signature and Quantization**

SOAR as described in Chapter 4, gives the texture difference between two points in an image. If we consider only ordinal measure to come up with the signature we could end up classifying two completely different images as the same. Let us consider two video sequences M and N, shown in Figure 5.4.

Figure 5.4(a) corresponds to a video sequence M and Figure 5.4(b) corresponds to a video sequence N. An ordinal measure would return the same signature for both of these
sequences as the gradient change in intensity in these two sequences is the same although
they are two completely un-correlated video sequences.

In order to overcome this drawback and make the signature more robust, in this
algorithm we consider the temporal information as well. SOAR is used to establish the
spatial relation between points in an image and the energy function in SOAR as described in
equation (23) is used to measure the temporal distance and track the cluster points in
subsequent frames.

For obtaining the spatial correlation, a 7 x 7 window is selected around the cluster
points as shown in Figure 5.5(a) and Figure 5.5(b). Using equations (19) and (20) for a
7 x 7 full token, as shown in Figure 5.5 (c), yields the Spatial Signature(SS) which is a string
of 1’s, 0’s and -1’s as seen in Figure 5.5 (d). The signature obtained from Figure 5.5(d) is
converted to a 1 x 49 length vector to make representation and comparisons easier.

It can be seen that in Figure 5.5(b), a value of 102 which is 73 pixel values lesser than
the centre pixel value, is assigned a value of ‘-1’ and a value of 170 which is just 5 pixel
value lesser that the centre pixel value is also assigned a value ‘-1’. By doing this a lot of
important information is lost which will eventually result in classifying two completely
different data points as the same.

In order to get more distinction we quantize the values based on a certain threshold,
‘T’. If the range of the pixel difference is (1, T) a “10” is encoded, if it is (T, 255) a “11” is
encoded, (-T, -255) encodes a “-1-1” and (0, -T) encodes a “-10” respectively. Thus a

![Figure 5.4. 2 x 2 ordinal measure.](image)
49 length vector is converted to a 98 length vector after quantization. Although quantization results in a signature of twice the original length, it adds more precision and helps in distinguishing closely correlated textures. This procedure is repeated for each of the five cluster centres shown in Figure 5.5(a).

### 5.4 Temporal Signature

In a video it is observed that redundancy is more from one frame to another, rather than in a single frame, hence temporal measure becomes very vital in a video. After having established the spatial relation between the cluster centres within each frame we come up with a temporal signature TS. To determine the temporal signature we adopt a method similar to that used in [9]. In [9] the temporal signature is directly determined by comparing the pixel values in consecutive frames, but here the cluster centre obtained in the key frame are first tracked in subsequent frames to see if it persists long enough to be considered stable.
Let us consider a video sequence $V_m$, assuming that the first frame of $V_m$ is the key frame, then a search window of size $16 \times 16$ is taken around the point of interest in the next frame which is frame 2 as shown in Figure 5.6. SOAR energy function is used as a similarity metric to find the best match for the cluster centre in Frame 2. Observing equation (23) closely, it can be deduced that the point which drives the SOAR energy function to a value very close to ‘1’ (high correlation) is considered the best match. This process is repeated for all other cluster centers to obtain the corresponding points in the next frame as depicted by Figure 5.7.

![Figure 5.6. (a) Frame 1 key frame and (b) Frame 2 showing the search window.](image)

![Figure 5.7. Best match (a) Frame (n-1) and (b) Frame n.](image)

After finding the best match, a temporal signature $TS$ can be defined for the video sequence $V_m$ as used in [9].
Thus the temporal signature will be one less than the number of frames in the video.

It is this temporal signature that is stored in the database and compared with other similar video signatures to determine the Near-Duplicates.
CHAPTER 6

RESULTS AND DATABASE

6.1 THE DATABASE

To assess the stability of the Near-Duplicate Detection estimation, a database of 12 original videos with three Near-Duplicates of each is considered, resulting in a total of 48 videos. The Near-Duplicates are defined as follows;

1. Decrease in Brightness.
2. Dilation of the video.
3. Erosion of the video.

Considering the ‘Miss-America’ video from the database, a detail description of ways to arrive at the near duplicates is given below. See Figure 6.1 shows the original Miss-America video which is used as the query clip for our experiments.

Figure 6.1. Original video.

6.1.1 Decrease in Brightness

To obtain a near duplicate video with decrease in brightness each pixel value is subtracted by a certain amount. In order to test the algorithm for different levels of error, the Pixel Values (PV) are first reduced by 3 which results in Figure 6.2 and yields a Peak Signal to Noise Ratio (PSNR) of 30.33 db.
The error is increased by reducing the PV by 5 and then by 6, to get an estimate of the False Positive Rate and False Negative Rate of the algorithm. Reduction in PV by 5 gives a PSNR of 30.15 db and reduction in PV by 6 gives a PSNR value of 28.71 db which results in a slightly erroneous videos as shown in Figures 6.3 and 6.4 respectively.

6.1.2 Dilation

Dilation is a morphological process of enlargement of the image using a structural element which defines the neighborhood around which the video has to be dilated. Dilation is
generally used to fill small holes like pepper noise mostly in gray scale images but here, dilation is used to generate erroneous near duplicates from the original videos. A structural element can be of any shape and size. If $A$ is the input sequence and $B$ a structuring element, mathematically dilation is defined as in equation 24.

$$A \oplus B = \bigcup_{b \in B} A_b$$

This structural element is run across the boundary of the image like a window, by placing the centre of the structural element on the pixel location that needs to be dilated. In case of binary images, even if one of the neighbouring pixels is one all the pixel values in the neighbourhood is replaced by one where as in case of a color or a grayscale image even if one of the pixel in the neighbourhood is a foreground pixel, then all of the pixels are replaced by the value of the foreground pixel, but if all of them are background pixels, the pixels values are retained. In our experiment we have chosen a square structural element of dimensions $1 \times 1$, $3 \times 3$ and $5 \times 5$. Figures 6.5 through 6.7 show dilated Miss-America images with structural elements of $1 \times 1$, $3 \times 3$ and $5 \times 5$ respectively and the corresponding PSNRs for these images are 31.05 db, 28.05 db and 23.45 db.

Figure 6.5. Dilation using $1 \times 1$ structural element.
Figure 6.6. Dilation using $3 \times 3$ structural element.

6.1.3 Erosion

Erosion is the dual of dilation. Dilating the foreground pixels is equivalent to eroding the background pixels. Mathematical definition of erosion is given by equation (25):

$$A \ominus B = \bigcap_{b \in B} A_b$$
Figure 6.7. Dilation using 5 x 5 structural element.

A ⊖ B = \bigcap_{b \in B} A_b

(25)

where A is the input video sequence and B the structural element.

A square structural element of dimensions 1 x 1, 3 x 3 and 5 x 5 are run along the boundary of the original Miss-America video, with the centre of the structural element coinciding with the pixel value of interest in the input image. Each pixel value is replaced by the minimum value in the neighborhood which is defined by the structural element that is used. In case of a binary image even if one value in the neighborhood is zero, the pixel value of interest is reset to zero. In case of gray scale and colored images the pixel value is reset to the minimum pixel value among all the neighbors. Figures 6.8 through 6.10 show the effect of erosion using 1 x 1, 3 x 3 and 5 x 5 structuring elements respectively and the corresponding PSNRs are found to be 31.05 db, 28.34 db and 23.93 db.

Experiments were conducted on Miss-America and Foreman video. The algorithm is first run for the query video, the obtained signature is then given as an input to the database. This stored signature of the query video is used for comparisons with videos in the database.

6.2 RESULTS

The graphs below, show the obtained results by running the proposed algorithm against the Miss-America video.
Figure 6.8. Erosion with a 1 x 1 structural element.

Figure 6.9. Erosion with a 3 x 3 structural element.

Figure 6.10. Erosion with a 5 x 5 structural element.

The x-axis for all the results represents the Near-Duplicates as mentioned below:

- 1 represents the original video
- 2 represents the dilated video
- 3 represents the eroded video
- 4 represents the original video with decrease in Pixel Value (PV)
The y-axis for all the cases represents the final distance obtained after taking the difference between the Temporal Signatures of the target video and the query video represented as in equation (26).

\[ D = \frac{1}{N} \sum_{i=1}^{N} |TS^i_q - TS^i_t| \]  

(26)

\( TS^i_q \) in equation (26) represents the temporal signature of the \( i^{th} \) frame of the query video, \( TS^i_t \) represents a similar temporal signature of the \( i^{th} \) frame but of the target video and \( N \) is the number of frames. From equation (26) it can also be deduced that a value of \( D \) closer to zero indicates that the two video sequences are similar.

### 6.2.1 Case 1

In Figures 6.11 through 6.14 the Near-Duplicates are obtained by reducing the PV by 3 and by eroding and dilating the original video by using a square structural element of dimension 1 x 1. It should be noted that a value of \( D \) close to zero is a good match. As can be seen from these graphs, each time Miss-America is detected with 100% accuracy.

![Figure 6.11. Reduction in PV by 3; erosion = dilation = 1 x 1-1.](image)

### 6.2.2 Case 2

The next step is to increase the level of errors introduced in the video to observe the level of robustness of the algorithm. The PV is reduced by 5 and the video is eroded and dilated using a square structuring element of 3 x 3. Figures 6.15 through 6.18 shows the
obtained results in a graph. It is again seen that Miss-America is detected with 100% accuracy by setting the threshold at 1.

### 6.2.3 Case 3

The error introduced in the Near-Duplicates was increased up a notch, the PV was decreased by 6 and a 5 x 5 structuring element is used for both erosion and dilation as described in Section 6.1. Figures 6.19 through 6.22 shows the graphs of the results obtained.
and it is clearly seen that the algorithm fails when such large amounts of error is introduced. By keeping the threshold at 4.2, the near duplicates can still be detected with a higher False Positive Rate and False Negative Rate when compared to Case 1 and Case 2.

### 6.3 COMPARISONS

A comparison is drawn between the proposed algorithm and the algorithm from [9] which is described in Chapter 2. In order to compare the two algorithms, variables like False
Positive Rate (FPR) and False Negative Rate (FNR) are used which are defined in the following sections.

Let \( M_{nc} \) be the total number of non copy clips and \( M_c \) be the total number of copy clips, then False Negatives (FN) is defined as the number of copy clips that are not detected and False Positives (FP) are the number of non-copy clips that are detected. FPR and FNR can then be defined as in equations (27) and (28).
Figure 6.18. Reduction in PV by 5; erosion = dilation = 3 x 3-4.

Figure 6.19. Reduction in PV by 6; erosion = dilation = 5 x 5-1.

\[
FNR(\tau) = \frac{FN}{M_c} \quad (27)
\]

\[
FPR(\tau) = \frac{FP}{M_{nc}} \quad (28)
\]

where, \( \tau \) is the set threshold for which equations 27 and 28 are computed.

Experiments were conducted and FPR and FNR were calculated for both the proposed algorithm and the algorithm from [9]. A Receiver Operating Curve (ROC) curve which is a plot of FPR Vs FNR for the two algorithms is as shown in Figure 6.23.
FPR and FNR are computed for three cases with different error levels as described below.

Case 1. The error introduced to obtain the Near-Duplicates in this case is:

- Reduction in the pixel value by 3
- Dilation using a square structural element of dimension 1 x 1
- Erosion using a square structural element of dimension 1 x 1
Case 2. The error introduced to obtain the Near-Duplicates in this case is:
- Reduction in the pixel value by 5
- Dilation using a square structural element of dimension 3 x 3
- Erosion using a square structural element of dimension 3 x 3

Case 3. The error introduced to obtain the Near-Duplicates in this case is:
- Reduction in the pixel value by 6
• Dilation using a square structural element of dimension 5 x 5
• Erosion using a square structural element of dimension 5 x 5

A threshold for the distance measure is fixed at 4.2. With this threshold it can be clearly seen from the FNR Vs FPR graph in figure 39 that the proposed algorithm has a better False Positive Rate although the False Negative Rates are similar. This shows that the proposed algorithm has a better precision in detecting the copy clips when compared to the algorithm in [9].
CHAPTER 7

CONCLUSION AND FUTURE WORK

In this thesis we have explored various ways to do Near-Duplicate Detection and come up with a novel way to use SOAR in NDD. Using SOAR has proved to reduce computational complexity and make the implementation of the architecture in a hardware simple and straightforward. Various experiments were conducted to observe the robustness of the unique signature of a video and performance metrics like FPR and FNR were computed and compared. Use of methods involving more of binary additions and multiplications, which translate to XOR and AND gates in hardware results in higher speeds.

This algorithm can be explored further to make it robust even for fast moving videos. An algorithm to find a key frame descriptor can also be explored and implemented along with the proposed algorithm to provide better results. SOAR was also implicitly extended to do tracking which can be explored further to do motion estimation.
BIBLIOGRAPHY


