International Journal of Security and Its Applications
Vol. 10, No. 4 (2016), pp.331-346
http://dx.doi.org/10.14257/ijsia.2016.10.4.30

Bag of Words Based Surveillance System Using Support Vector Machines

Nadhir Ben Halima[1], Osama Hosam[2*]

[2]*The City for Scientific Research and Technology Applications, IRI, Alexandria, Egypt. ([*]Correspondence)*
[1]*Computer Science Department, Computer Science and Engineering College in Yanbu, Taibah University, Yanbu, KSA*
nbenhalima@taibahu.edu.sa, mohandesosama@yahoo.com

Abstract

Terror attacks are increased worldwide. The early detection of weapons is an important objective for security specialists. In this paper, we proposed an automated surveillance system for detecting fire weapons in cluttered scene. First SIFT features are extracted from the collection of images. Second, K-means clustering is adopted for clustering the SIFT features. Third, a word vocabulary based histogram is implemented by counting occurrences of the extracted clusters in each image. The histogram is the input to Support Vector Machine that will be trained on the collection of images. Finally, the trained SVM is the system classifier that will decide if new image contains a weapon or not. The main contributions of the paper is to adopt the visual words classification scheme in detecting fire weapons. In addition, we used RANSAC to reduce the matching outliers. The system showed high accuracy in detecting fire weapons in images and video surveillance systems.

Keywords*: K-means clustering, SVM, Bag of Words, Fire weapons, Classification*

1. Introduction

Terror attacks have greatly spread worldwide without having a global solution to such a destructive problem. Computer based automatic surveillance algorithms are in great demand to detect the existence of fire weapons in crowded image and video scenes. Police officers can have such surveillance systems to help them in tracking criminals, terrorists and close down any attempted terror attacks. The existence of fire weapons in images and video frames can be detected by using several approaches. One of these approaches is Bag of Visual words (BoW).

BoW representation can be adopted for computer vision classification [1, 2]. VLFeat is an open source library for computer vision [3] the authors proposed to use an algorithm built with the VLFeat library, they proposed a new feature called PHOW which is denser than the SIFT features. They used PHOW as input to the k-means clustering algorithm. A Spatial histogram [4] is calculated by counting the number of occurrences of k-means centers in each image. The histogram is then passed to a non-linear SVM for training. Another BoW based approach is introduced in [5], this algorithm extracts dense SIFT features either from the image or from a spatial pyramid that divides the image into fine sub-regions, then it represents the feature as a presence vector of visual words. The method uses multi-class one-versus-all SVM on multiple kernel learning (MKL). Sivic and Zisserman proposed an inverted file containing quantized local descriptors after indexing them for rapid indexing of video frames [6]. The local descriptors can be represented as visual words by using k-means clustering. It was very easy to reach the same cluster in each video frame by simple comparison of the clustered features. They also

ISSN: 1738-9976 IJSIA

introduced “term frequency-inverse document” tf-id concept to propose weight for each visual word. tf-id will be introduced in section 3. tf-id’s are strong for removing common words which are found in most video frames and also has a proposed stop-list in which ignores the extremely frequent words that appear nearly in each image such as parts of background.

In this paper, we proposed classification techniques based on the Bag of visual words approach to detect fire weapons in cluttered scenes, namely the Bag of Words Surveillance System (BoWSS).

The following are simplified steps of the proposed algorithm:

- Extract the SIFT local feature vectors from the set of the training image. SIFT features are used in this approach because of their strength in classification and immunity to illumination, rotation and scaling.
- Put all extracted local feature vectors into a single set (You do not need to know for which image the local feature is obtained).
- Apply a clustering algorithm (the k-means algorithm is used for its efficiency) over the set of local feature vectors in order to obtain a set of clusters; each cluster represents a single visual word.
- Obtain the spatial histogram (spatial histogram [4] is the global feature vector that counts how many times each visual word occurred in each image).
- Pass the obtained histogram features for each image to the SVM for training.
- Classify new images by using the trained SVM
- Detect the location of the weapon inside the image and remove the matching outliers by using RANSAC. RANSAC is originally proposed in [7].

Contributions in this paper include adopting visual words classification in detecting fire weapons in images. In addition finding the location of the weapon in the image and reducing the matching outliers by using RANSAC.

The remaining of the paper is organized as follow. Section 2 introduces detailed explanation of SIFT features, k-means clustering and SVM, including the way of using them in our algorithm. Section 3 introduces the concept of word indexing, reflects the concept of visual words, and the way to adopt those concepts in the proposed algorithm. Section 4 introduces the results and discussion.

2. Background

Through this research we use SIFT features for their strength in classification and immunity to illumination, rotation and scaling. SIFT features are clustered and grouped using K-means clustering algorithm. The classification will be done using SVM for its high performance and accuracy. In the following sections we introduce SIFT features, K-means clustering, and SVM in detail.

2.1 SIFT Descriptors

The Scale Invariant Feature Transform (SIFT) was originally introduced by Lowe [8]. SIFT features are the most popular features used in computer vision. The SIFT extraction algorithm detects salient image regions called key points and extracts discriminative descriptors of their appearance called descriptors. SIFT descriptors are robust to lighting variations, in addition SIFT key points are invariant to view point changes such as translation, rotation and rescaling of the image. SIFT feature extraction algorithm is done in four stages, namely:

- Scale-Space Extrema Detection
- Key point Localization
- Orientation Assignment

- Key point descriptor.

Scale-Space Extrema Detection: This step mainly aims at identifying locations and defines scales that are identifiable for an object when it is seen from different points of view. This is done by using the "Scale space" function, with reasonable assumptions; scale space is based on the Gaussian function. The scale space function is given by

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (1)$$

Where * is the convolution operator, $G(x, y, \sigma)$ is Gaussian with variable scales and $I(x, y)$ is the input image. x, y are the vertical and horizontal image pixel locations respectively. To detect stable key point locations in the scale space, Difference of Gaussian (*DoG*) is used. *DoG* locates the scale space extrema *i.e.* $D(x, y, \sigma)$, by computing the difference between two images in different scales with a factor of k. $D(x, y, \sigma)$ is given by:

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (2)$$

To detect the local maxima and minima of $D(x, y, \sigma)$ each point is compared with its 8 neighbor points in the same scale, and its 9 neighbors up and down one scale. If this point is the minimum or maximum of all of its neighbor points, then this point is the extrema.

Key point Localization: In this stage, points are eliminated from the list of key points having low contrast or poorly localized on an edge. This is done by calculating the Laplacian for each key point found in "Scale-Space Extrema Detection" stage. The location of the extrema z is given by:

$$z = \frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial x} \quad (3)$$

where D is Taylor expansion of the *DoG* scale-space function. D is calculated at the candidate key point and x is the offset from that point. The point will be excluded if the function of its value at z is below threshold value. Thus we succeeded to remove extrema's with low contrast. For poor localized extrema's, there is a large principle curvature across the edge, but small curvature in the perpendicular direction in the *DoG* function. If *DoG* is below the largest to smallest eigenvector ratio at the location and scale of the key point, the key point is rejected.

Orientation Assignment: This step assigns orientation to each key point based on local or spatial image properties. The key point descriptor will be explained later, it will be described by using the orientation. The approach is to; first, use the key points scale to select the Gaussian smoothed image L from the previous phase.

Second, compute the gradient magnitude m and the orientation θ as follows:

$$m(x, y) = \sqrt{((L(x+1, y) - L(x-1, y))^2 + ((L(x, y+1) - L(x, y-1))^2} \quad (4)$$

$$\theta(x, y) = tan^{-1} \frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \quad (5)$$

where L is the Gaussian blurred image. Third, form an orientation histogram from the gradient orientation of sample points. Fourth, locate the highest peaks (at most 3 peaks) in the Histogram to create a key point with that orientation. Finally, fit a parabola to the highest peaks in the histogram by interpolating the peak positions.

Key point Descriptor: The data obtained from the above phase is used to create key point descriptors. Key point descriptors are shown Figure. 1. The gradient information is rotated to line up with the orientation of the key point.

Then it is weighted by Gaussian with variance of (1.5 * key point scale), this data is used to create histograms over a window centered on the key point. Key point descriptors use

16 histograms aligned in a 4x4 grid, each with 8 orientation bins, overall feature vector is then 16x8=128 elements.

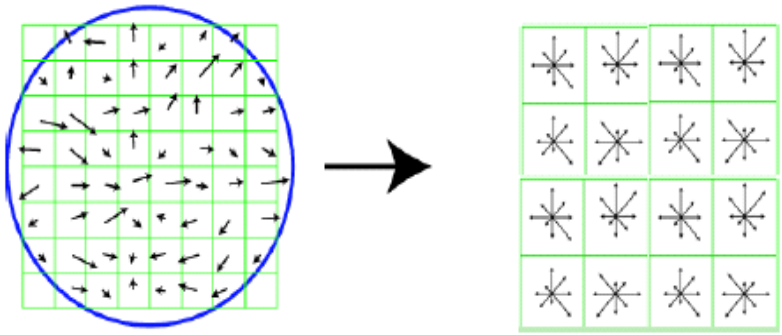


Figure 1. Key Point Descriptor, (left) the Gradient Map (right) with a 4x4 Location Grid and 8 Orientations (128 Dimensions)

The resulting vectors are the SIFT descriptors. Figure 2. Shows the SIFT descriptors presented on weapon image. SIFT descriptors are used mainly in feature matching and object identification. Due to the large number of SIFT features of an image, (for example 512x512 pixel image may generate 3000 features), substantial levels of occlusion are possible. RANSAC can be used to overcome possible outliers.

2.2 K-Means Clustering

Clustering can be defined as method of grouping together comparable points, feature vectors images, among other factors. On the other hand, segmentation refers to the division of the image into various significant regions. Notably, K-means method of clustering shown in Figure 3 is a commonly applied algorithm in various data extraction applications[9], while SIFT manifests several clustering applications[6]. Evidently, it has a direct implementation plan and is quite easy to comprehend. In fact, it is often used in grouping together same points and representing them using a single token. Using the typical k-means algorithm is initially presented in various ways [10].

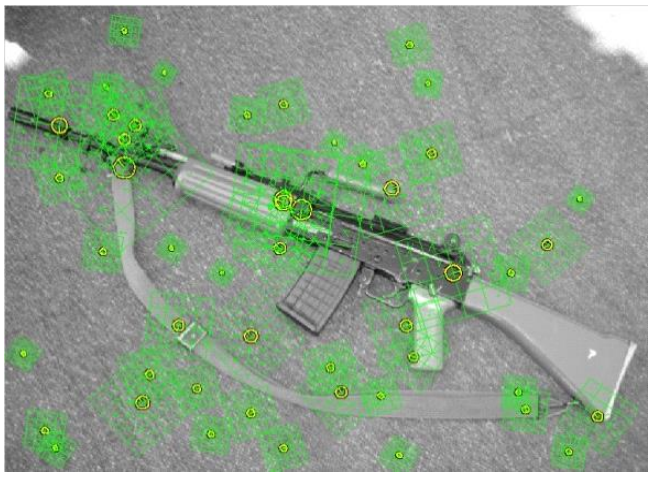


Figure 2. Showing a Selected 50 out of 17251 SIFT Descriptors Displayed on a Fire Weapon

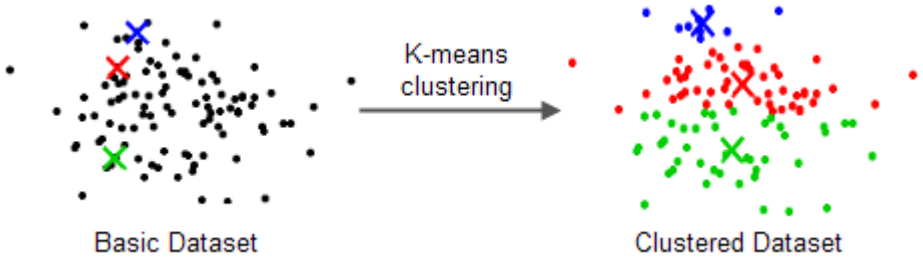


Figure 3. K-means Clustering. (left) the Dataset without Clustering, Cross Marks show the Means (centers) that are Selected in the First Iteration, (right) Data after Clustering and Assigning each Value to the Nearest Center

Mathematically, the k-means algorithm [11] is represented as follow, let $X=[x_i, i=1, 2, 3,.., N]$ stand for n-dimensional positions that need to be clustered into a number of K-Clusters $C=\{C_i, i=1,2,3, ... K\}$. Notably, K-means algorithm gets dividers making the error existing between the Cluster C_i mean (μ_k) and the Point x_i minimized. Take up μ_k be the cluster C_k mean, the squared inaccuracy between μ_k and the positions x_i in C_k is referred to as

$$E(C_k) = \sum_{x_i \in C_k} \|x_i - \mu_k\| \quad (6)$$

The main objective of K-means minimizing the Sum of Squared Error (SSE) across all the K clusters shown as:

$$min_{\mu_1 \quad \ldots \quad \mu_k} E(C_k) = \sum_{k=1}^{n} \sum_{x_i \in C_k} \|x_i - \mu_k\|^2 \quad (7)$$

in case $\|x_i - \mu_k\|^2$ implies Euclidean distance correspondence measure function.

2.3 Support Vector Machines

SVM refers to a dominant classifier that is utilized to carry out machine learning as well as classification. Notably, SVM also means a special aspect of Kernel methods [12]. Assuming that there exist a dataset that is described using typically two dimensions, for instance, a house is labeled by the use of number of rooms available (x_1) and its area(x_2). The dataset often contain a group of houses having a decision of either buying (class1) or not buying (class2). Moreover, SVM implies qualified (supervised learning) that uses the prevailing dataset; a consideration is often given to every dimension in identifying its importance. In cases a new house is realized, SVM has the sole decision on whether or not to buy it.

Based on the house illustration, SVM classifier is characterized as a direct classifier to create a distinction of the two main classes shown in Figure 4(a). The major objective of using SVM is finding the line separating the two classes using maximum margin. Besides, Support vectors make up the closest sample positions to the line as shown in Figure 4(b). Furthermore, the data used in such cases is known as directly distinguishable, which is an ideal case. Naturally, dataset is non-linearly divisible as shown in Figure 4 (c). There are two solutions for the non-linearly distinguishable data:

- *Applying Slack Variables*: Let the dataset remain intact but study the errors applying various slack variables. Often, Slack variables are utilized to cumulate the errors from the positions found in the erroneous side[13]. Figure 4 (d)
- *Applying Kernel*: Map the main vector nonlinearly to a great dimensional section then apply linear classifier to the new region (feature space).

Mathematically [14], the divider in 2-dimensional region is always in line with:
$w_1x_1+w_2x_2+b=0 \quad or \ \mathbf{w} \bullet \mathbf{x} + b=0$ (8)

$\boldsymbol{w} \bullet \boldsymbol{x} = 0$ is signified using two vectors found at the source with the dot product =0, implying that they are fully orthogonal. Therefore, w vector implies the vertical direction taken to the line. $\boldsymbol{w} \bullet \boldsymbol{x} + b = 0$ shows an equation used for the line lifted b distance on the upward or downward directions from the source, Fig.. 4(b).
The main margin lines show:

$$\boldsymbol{w} \bullet \boldsymbol{x} + b = +1 \quad (9)$$
$$\boldsymbol{w} \bullet \boldsymbol{x} + b = -1$$

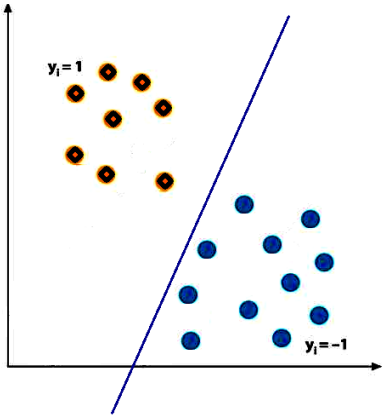
The two equations above are combinable into the following single equation:

$$y_i(\boldsymbol{w} \bullet \boldsymbol{x} + b) >= 1 \quad (10)$$

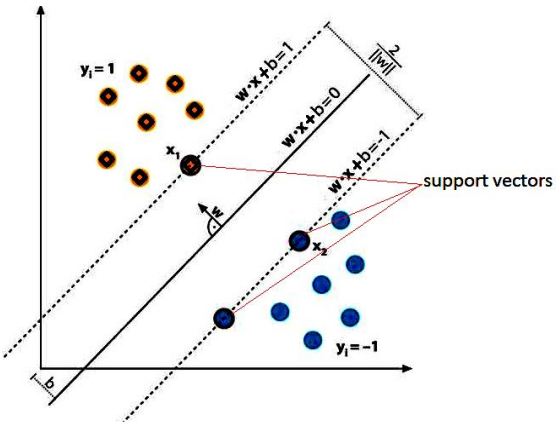
Remember the difference from point (x_0,y_0) to line: $Ax+By+c = 0$ is: $|Ax_0 + By_0 + c|/\text{sqrt}(A^2+B^2)$, therefore, The distance existing between $\mathbf{w} \bullet \mathbf{x} + b = +1$ and $\mathbf{w} \bullet \mathbf{x} + b = -1$ can be $|\mathbf{w} \bullet \mathbf{x} + b|/||w|| = 1/||w||$, therefore, the aggregate distance found at the point between $\boldsymbol{w} \bullet \boldsymbol{x} + b = +1$ and $\boldsymbol{w} \bullet \boldsymbol{x} + b = -1$ is finally: $2/||\boldsymbol{w}||$. For maximization of the margin, it is advisable to minimize $||\mathbf{w}||$, or consistently reduce $||\mathbf{w}||^2 = \mathbf{w'} \bullet \mathbf{w}$ or $\frac{1}{2}\ \mathbf{w'} \bullet \mathbf{w}$, in that "'" shows the matrix transfer. Generally, we want a hyperplane able to solve the optimization problem below:

Minimize $\frac{1}{2}\ \mathbf{w'} \bullet \mathbf{w}$

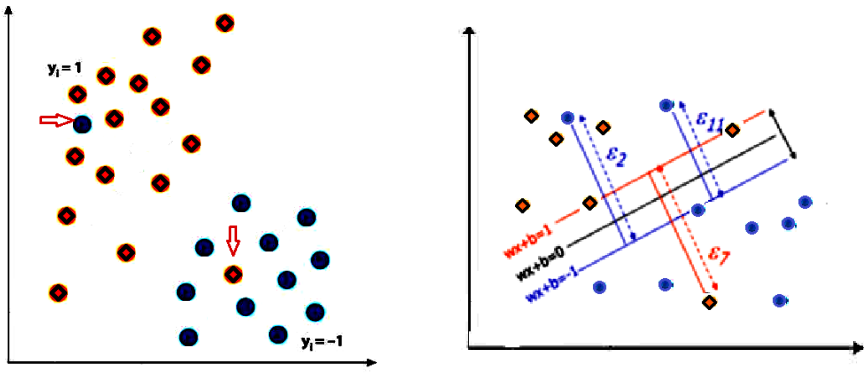
$$\textit{Subject to}\ y_i(\mathbf{w} \bullet \mathbf{x} + b) >= 1, \textit{ for all } i \quad (11)$$



(a) The separating hyperline, there are infinite number of lines, only one is shown



(b) there are only one line which maximizes the margin, the figure shows that line with support vectors



(c) Some of sample points may incur errors as shown by the arrows

(d) considering the error of the outliers in SVM calculations

Figure 4. Support Vector Machines

Lagrange multipliers often used in simplification of the optimization problem above. For function *f(x)* as well as constraint function *g(x)*, Lagrange multipliers used include $L=f(x) - \alpha g(x)$. Applying this formula, equation (11) creates

$$L= \tfrac{1}{2}\, \mathbf{w'\cdot w} - \sum \alpha_i [y_i(\mathbf{w} \cdot \mathbf{x} + b) - 1].$$
$$\text{Subject to } \alpha_i >= 0, \text{ for all } i \tag{12}$$

The restraint $\alpha_i >= 0$ comes as a result of L's inequality constraint. Seeking the gradient of L to disappear in respect with *w, b, α*, differentiation is done with respect to all the three variables and a zero differentiation. The results are:

$$W=\sum \alpha_i y_i x_i,\ \sum \alpha_i y_i = 0. \tag{13}$$

Returning them to the Lagrangian L, the below quadratic programming problem comes up

$$\begin{aligned} &\text{maximiz} \quad L_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j Y_i Y_j X_i^t X_j \\ &\text{constrained to} \quad \alpha_i \geq 0 \ \forall i \text{ and } \textstyle\sum_{i=1}^{n} \alpha_i Y_i = 0 \end{aligned} \tag{14}$$

Prior to solving the optimization problem considering the slack variables case plus the kernel case is key.

Slack Variables: Using nonlinearly divisible data, the key knowledge of applying slack variable is allowing some unwanted points are in the erroneous side and solve the problem continually, Fig. 4(d). Errors of such positions are often reflected in equation (10), and will be

$y_i(\mathbf{w} \cdot \mathbf{x} + b) >= 1-\varepsilon_i$ changing the optimization problem to

$$\text{Minimize } \|\mathbf{w}\|^2 + C\sum \varepsilon_i$$
$$\text{Subject to } y_i(\mathbf{w} \cdot \mathbf{x} + b) >= 1-\varepsilon_i,$$
$$\text{for all } i,\ \varepsilon_i >= 0 \text{ for all } i \tag{15}$$

C means the Trade-off limit, it is user-modified. Since having *w* that has *N* positions and errors in *N* points, the optimization error is solvable using *2N* constraints. Being a quadratic programming issue, SVM is soft-margin support vector machine.

Kernels: Evidently, Cover's theorem asserted that "pattern-classification issue cast in a great lengthy region non-linearly and has high likelihood of being linearly separable compared to a low-dimensional space".

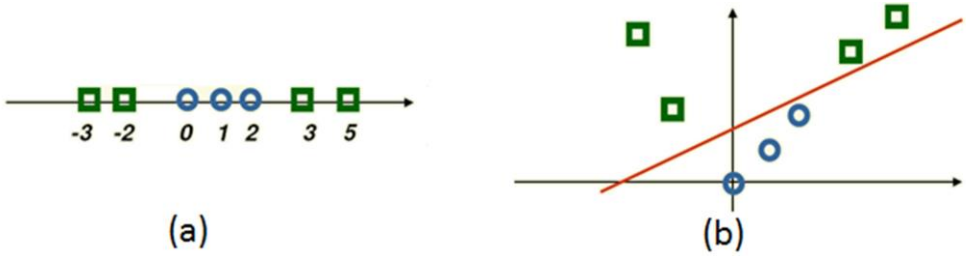
From Figure. 5, the data shows non-linearly separable features in 1D. However, moving it to the higher feature region makes it linearly separable.

Scientifically, Kernels are a group of transformation uses applied for the changing of the data to greater spaces. For instance, transformation function $Ø(x)=(x,x^2)$ changes the linear region to a quadratic space.

From Fig 5, Mapping 1D to 2D space gets linearly divisible data (a) initial 1D space (b) factor space using $Ø(x)=(x,x^2)$

3. Visual Vocabulary Classification Algorithm

Retrieval of texts generally consists of the following steps [15]: First, the main document needs to be read in terms of a collection of many words. Second, representing the words by their stem, for instance, "walking" and "walks" get symbolized using "walk". Third, the prevalent words such as "an", "a", "and",



(a) Original 1D Space (b) Feature Space with $Ø(x)=(x,x^2)$

Figure 5. Mapping from 1D to 2D Space (Feature Space) for Getting Linearly Separable Data

"The" and "is" are deleted from the list while common words are omitted since they cannot distinguish the document. The rest of the words get assigned a special identifier. Each document gets symbolized as a vector obtained using the frequencies of each word found in the document. The constituents are then assigned weights using ordinary weights called "term frequency, inverse document frequency." Thereafter, tf-idf; [6] tf-idf gets calculated as follows:

Assuming there exist a vocabulary consisting of k words, each document gets represented using a K-vector $V_d=(t_1,....,t_i,.....,t_k)^T$ showing weighted word incidences that has components

$$t_i = \frac{n_{id}}{n_d} log \frac{N}{n_i} \quad (16)$$

in that n_{id} shows the instances of frequencies of the word i in the doc d, n_d shows the aggregate words found in the entire document d, n_i shows the occurrences of the term i in the entire databank and N shows all the documents found in the entire databank. The weighting means a product from two terms:

- The *word frequency* n_{id}/n_d,
- The *inverse document frequency* $\log N/n_i$.

Ostensibly, the word frequency weights words that occur more often in a given document, while the inverse document frequency weights the words appearing repeatedly in the whole database.

How is it possible to use text document processing in aiding visual search? The image becomes analogous to the document and can be represented by using a group of visual

factors like SIFT features. Besides, the problem comes up since document words often become discrete "tokens" as the image elements become high degree real value positions. Therefore, how can discrete "visual words" be attained?

To achieve this application of quantization procedure on the group of the image elements, we proposed BoWss algorithm. There are two main phases that are included in the BoWSS algorithm: supervised training and testing. The supervised training phase contains the following steps:

Image Set Labeling: image labeling involves the examination of a series of images which will be selected for extraction, and labeled with the identification "containing weapon" or "not containing weapon".

Feature Extraction: Extract the SIFT features from the entire collection of images, this will be large set of features for the entire dataset.

Feature Clustering: In order to cluster all these features –obtained from previous step - into a collection of centers, which are represented by one word each, K-means clustering will be applied.

Spatial Histogram Calculation: Count how many times each word appears in each image, so that a Spatial Histogram can be created. To obtain a faster count, it is recommended to use the kd-tree algorithm.

SVM training: Run the calculated spatial histogram through SVM and to use the RANSAC for removing all the matching outliers.

In Figure. 6, we depicted the training phase of the proposed visual vocabulary classification algorithm (BoWss). Here we describe the detailed stages of the training phase of BoWss.

3.1 Image Set Labeling

Image labeling is done with assigning +ve or –ve label to each image in the training set. The images with weapons will be assigned +ve label and the image without weapon will be assigned –ve label. In BoWSS, supervised SVM training will be used. In which the training is a pair consisting of input object and the desired output. Actually the labels will not be added to the images in this phase, instead in the subsequent algorithm stages, each extracted word will be assigned a label (-1 or +1).

3.2 Feature Extraction

This phase is similar to reading all documents and extracting all words from all documents which is used in text retrieval algorithm. SIFT features are extracted as they are extremely important when rotated weapons, different resolutions and illuminations are needed to be discovered. That's why SIFT features are more suitable compared to other features for using in BoWSS. There will be a high number of features that will be obtained, and all these will be put together, thus creating one set. The set of features obtained in this stage will be assigned an identifier; the identifier defines the feature's source image.

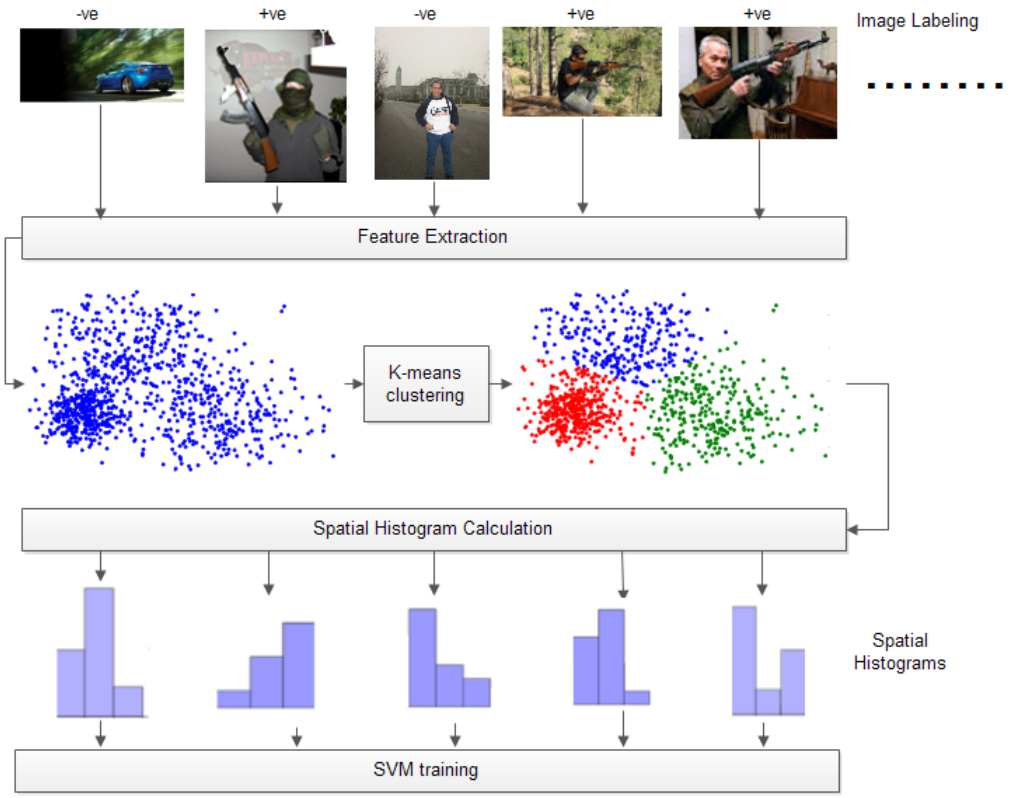


Figure 6. The Proposed System Architecture, The Training Phase of BoWSS, an Example of 3 Visual Words

3.3 Feature Clustering

The key features are made discrete by applying one of the key clustering processes; K-means algorithm becomes the most suitable for the entire clustering stage. Every feature gets assigned a particular mean or center. Additionally, the image gets separated visually to make a group of centers also known as “Visual words” which is an analogous word used in document processing. The visual words here represent weapons, trees, cars, humans, etc. Similarity between features is measured with adaptive descriptor, which is used because weapons have different shapes, sizes and rotation angles. This is similar to adding the stems of the word such as "Walking", "Walks" to the original word which is "Walk". In this stage also the common words such as "an", "is" will be deleted, in images the features represent background, sky, and fixed areas in a movie frame will be discarded, because these features will not discriminate different images. A label will be added to each visual word to identify if the feature is weapon feature (+1) or non-weapon feature (-1).

3.4 Spatial Histogram Calculation

The three-dimensional histogram is created through the calculation of the instances with which visual words are found to occur in every image. Each image is represented as a vector obtained using the frequencies of each visual word in the document. Each image is represented with K-vector as described previously in text retrieval systems $V_d=(t_1,....,t_i,.....,t_k)^T$. where k is the number of visual words. And t_i is the weight of each visual word in the image; t_i is calculated by using equation (16). Each item in V_d is represented by column in the resulting spatial histogram. In this step, randomized KD-Tree is adopted in our experiments [17]. Randomized KD-tree is an extended version of

the basic KD- tree proposed in [21]. Randomized KD-tree is adopted to create faster histogram counting. The basic KD-tree is shown in Figure. 7.
KD-tree is similar to the decision tree, except that in KD-tree we split according the median along the dimension having the highest variance. Each node stores single separator with single data point and leaves are empty. The separator defines the location of the median visual word that separates the features to two sets. And recursively divides each new set. Histogram calculations depicted in equation (6) will be faster since the feature set is now divided into regions containing approximately similar features. The basic KD-tree splits the data into two half according to the greatest variance. Compared with basic KD-tree, the randomized KD-tree split dimension randomly from the dimensions having data with the highest variance. When the tree is used for searching, single priority queue is used for the entire random tree for feasible and easy search.

3.5 SVM Training

The main objective of SVM is to learn the mapping : X → Y, where $xi \in X$ is some object and $yi \in Y$ is a class label. In this paper we use two class SVM classifier, so $xi \in$ Rn, $yi \in \{\pm 1\}$. Suppose there are 100 images some of them contain weapons and some others are clear from weapons. SVM will be trained on the 100 images for classifying the two classes. So SVM will decide for new test image if it contains weapons or not.
The second phase is SVM classification; we deal with the simplest classifier which is binary classifier. Given the training dataset which is a collection of images with their labels (x_i, y_i) for $i = 1 \ldots N$, N is the number of images, with $x_i \in R^n$ and $y_i \in \{-1, 1\}$, learn a classifier f(x) such that

$$f(x_i) = \begin{cases} \geq 0 & y_i = +1, \\ < 0 & y_i = -1 \end{cases}$$

In other words $y_i f(x_i) > 0$ for an optimal classification of the images. Linearly separable samples will be dealt directly; non-linearly separable samples will be solved with kernels.

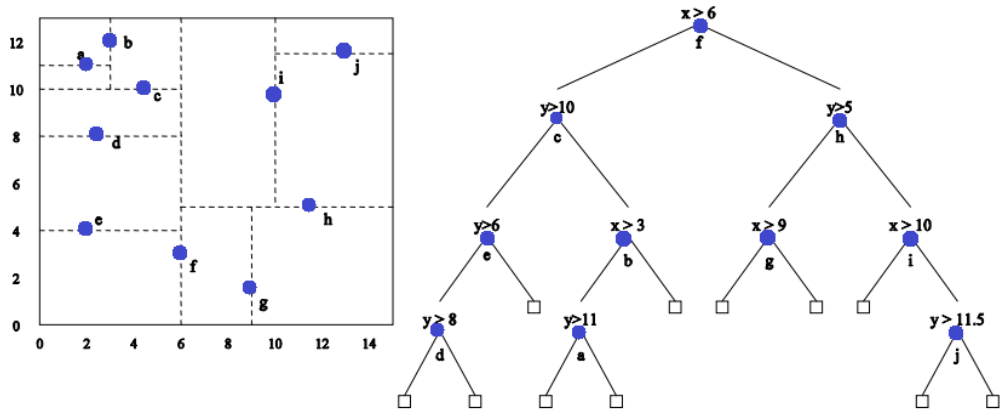


Figure 7. Basic KD-Tree, each Circle Represents a Feature, and Each Area is Used to Limit the Histogram Calculations, (Left) The Division Cluster Areas (Right) the KD-Tree

4 Final Results and Discussions

We have decided to include the VLFeat Library in the implementation of the BoWSS algorithm, as it is primarily used for computer vision algorithms. One of its main features is represented by the inclusion of common building blocks for computer vision, with components such as clustering, feature detection and extraction, etc [3]. A great option is

represented by the fact that it can easily be integrated into MATLAB. Here are the algorithms that we have decided to include in the VLFeat:

- K-Means clustering
- Dense SIFT (PHOW)
- SIFT feature detector and descriptor
- Randomized kd-tree

Our experimental test also includes the collection of a wide variety of images from the internet, in addition to those gathered from the digital Camera, so we can obtain a large dataset. We have used a variety of pictures, all combined into collections, each containing 200 images, including a set of pistol images, gun images, rifle images, a combination of images, and a final set that doesn't contain any type of weapons, as this is simply used as a training for SVM.

This document includes two different sets of images, the ones containing weapons that are cluttered and are referred to as mixed images, that the ones that don not contain any type of weapon, and are known as clear images. We've collected random images only for testing purposes.

Next, we are showing the SIFT features that have been extracted, and also the means in which they are gathered and clustered so they can create a visual word. In addition to this, we are also presenting the accurate classification of our selected algorithm. The final step will be to demonstrate how we can detect the location of each weapon.

4.1 Visual Words and Extracted Features

The images used for training are used to extract the SIFT features. Table 1 shows the high number of descriptors that can be extracted (image 2 – more than 100.000 descriptors). When it comes to the processing delay, this can be observed when the SIFT features are extracted and this extraction also reveals he fact that the is a high processing time.

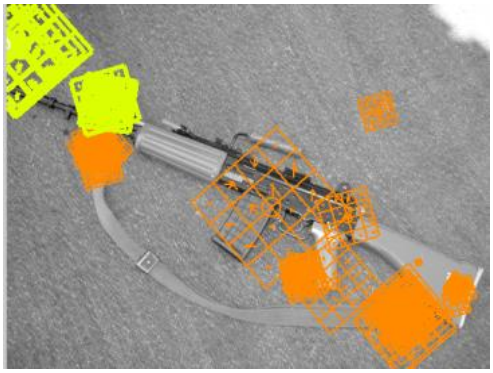
Our decision was to save the SIFT features on our Hard Disk, as soon as they were extracted, so we can obtain faster access for future tests and processing. The fact that we are directly using the trained SVM, there will not be any processing problems during the classification, even though there may be some problems while video processing.

Figure. 8 shows the SIFT descriptors, and represents 2 images from Table 1 (3 and 4). As a result, we have 45938 descriptors in the first image and just 27940 in the second one, but we have decided to work with just 20 descriptors so we can obtain a much better representation. The ones that have the same size and orientation are shown as an identical sub-image pattern (Figure. 8 b – background pattern – up-left corner). If there are different sizes, it simply means that higher levels of the descriptors are found, each having higher resolutions when it comes to the Gaussian pyramids.

Table 1. The Representation of the SIFT Feature of just 14 Images (Mixed and Clear)

Serial	Image size	Type	Type of features	# descriptors
1	480 x 640	Mixed	Dense SIFT	85094
2	301 x 251	Mixed	Dense SIFT	121238
3	307 x 230	Mixed	Dense SIFT	45938
4	670 x 456	Mixed	Dense SIFT	27940
5	1355 x 725	Mixed	Dense SIFT	94130
6	410 x 435	Mixed	Dense SIFT	58990
7	500 x 232	Mixed	Dense SIFT	44068
8	450 x 650	Clear	Dense SIFT	96138

9	194 x 259	Clear	Dense SIFT	42424
10	224 x 225	Clear	Dense SIFT	17746
11	183 x 275	Clear	Dense SIFT	17694
12	334 x 151	Clear	Dense SIFT	17540
13	259 x 194	Clear	Dense SIFT	17600
14	950 x 634	Clear	Dense SIFT	17746



(a)

(a) Descriptors that represent the weapon are presented in Orange

(b)

(b) Descriptors that represent the weapon are presented in Yellow. (Orange descriptors are thinner than Yellow descriptors in case of gray scale printing)

Figure 8. SIFT Descriptors shown on Mixed Images for Different Clusters after K-means Clustering

Figure. 8 represents the SIFT descriptors after the k-means clustering was applied on various mixed images

(a) Orange represents the descriptors marked for weapons
(b) Yellow represents the descriptors marked for weapons

In case the printing is of a gray scale, the Yellow descriptors are much thicker than the Orange ones.

We gather all the descriptors in just one set, once the k-mean clustering was applied, resulting in more than a few hundreds of millions of descriptors from just 600 images. To solve this problem, we decided to choose, with no patterns in mind, a subset containing 10.000 images, to which we applied the k-means clustering feature. For a faster result with the k-means clustering, we decided to go with the Elkan [16] algorithm and not the Approximate Nearest Neighbor (ANN) as it offers much faster results. The final result obtained was a collection of visual words, similar to the SIFT descriptors. Fig. 9 shows a set of images including: Figure. 9 (a) represented by image 5 from Table 1, Figure. 9 (b) represented by image 4 from Table 1. This pair is represented with only 3 clusters, and the same color that appears in both images represent the same visual word.

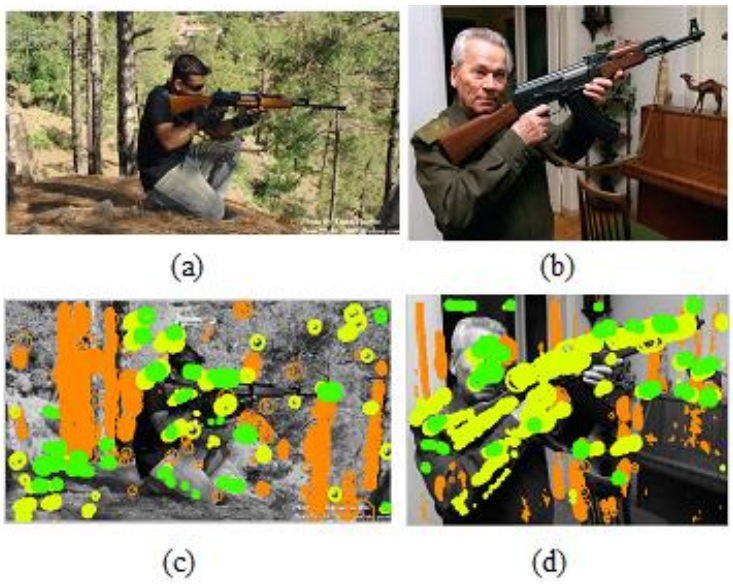


Figure 9. Three Clusters, Corresponding to Three Visual Words are displayed with Different Color for each Visual Word after K-Means Clustering

The cluster in orange (the first one), shows patterns such as walls and tree bark, the second one shows patterns with a much higher frequency, including sleeves, weapons and soldiers, and the third one represents patterns with extremely high frequencies. The histogram, which actually counts the number of visual words from each image, is created once the k-means clustering was applied. Figure. 10 presents each histogram for both mixed and clear images. We have decided to only show 10 visual words for each of the selected images, which represent the 10 centers that were assigned with the help of the k-means clustering. Keep in mind that we have decided to choose the kd-tree, in the histogram calculation, to represent the hierarchical structure for similar queries and neighbor computation. Also, we have used the kd-tree as the most suitable method to search the description that is closest to the k-means center, and manage to compare the current center with the descriptor. Having this in mind, we decided to adopt a randomized kd-tree algorithm, which was initially presented by FLAAN [17].

4.2 Reducing the Matching Outliers and Locating Fire Weapons

Locating the weapon coordinates is an important factor in surveillance systems, as the automatic detection helps locate the weapons. When it comes to the BoWSS, the SIFT features from each of the two images must be compared in order to obtain matching items, exactly as it is presented in Figure. 11. For this, we decided to use a cluttered image containing a referred gun, so we can detect the location much easier. In reality, in BoWSS, we are searching for certain types of weapons, and even though the types may be changed, if there is a type that is not used, a lot of outliers will become visible. A perfect match appears only if there are the same types of weapons in the resulting cluttered scene. In order to detect the weapon location, we are comparing more than 40.000 SIFT descriptors, but in the video sequence there is no proper algorithm for succeeding to locate weapons, mainly because of the high amount of descriptors comparisons. Figure. 11 (a) shows that there are 3 correct matches and 2 outliers after the SIFT matching. We have successfully managed to implement the RANSAC in order to eliminate he outliers, and the result is perfectly shown in Figure. 11 (b). In case we decide to use different types of weapons as a comparison, even if we use RANSAC, the outliers will not be eliminated.

As a conclusion, the most efficient comparison will only be obtained while using the same type of weapon.

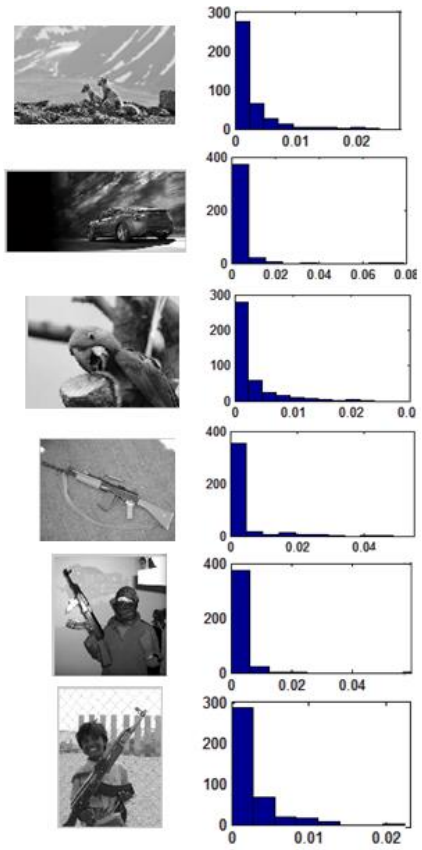


Figure 10. Mixed and Clear Images with their Corresponding Spatial Histograms

5. Conclusion

In order to detect the fire weapons efficiently in cluttered scenes, we have introduced a surveillance technique called Bag of Words Surveillance System (BoWSS), and we have managed to apply properly visual words to help detect those fire weapons. We have also included RANSAC as a solution to remove the outliers that matched, As a result, the BoWSS presented high accuracy results when it comes to detecting fire weapons in the resulting cluttered scenes. In the future work, we are planning to use the same algorithm for real-time surveillance security system with dedicated camera. However, this leads us to explore more efficient and faster classification especially in case of clustering and histogram calculations.

References

[1] Z Jian, Marszalek, Marcin, Lazebnik, Svetlana, and Schmid, Cordelia. "Local Features and Kernels for Classification of Texture and Object Categories:A Comprehensive Study" *IJCV,73(2):* pp. 213-238, June 2007.

[2] S, Josef, Russell, Bryan C., Efros, Alexei A., Zisserman, Andrew, and Freeman, William T. "Discovering Objects and their Localization in Images". *Proceedings of ICCV*, pp. 370-377, 2005.

[3] A. Vedaldi and B. Fulkerson. VLFeat: "An open and portable library of computer vision algorithms" . [Online] http://www.vlfeat.org/, 2008.

[4] S. Lazebnik, C. Schmid, and J. Ponce. "Beyond bag of features: Spatial pyramid matching for recognizing natural scene categories". *In Proc. CVPR*, 2006.

[5] R Tudor Ionescu, M Popescu, C Grozea "Local Learning to Improve Bag of Visual Words Model for Facial Expression Recognition" ICML 2013 Workshop on Representation Learning, Atlanta, Georgia, USA, 2013.
[6] J. Sivic and A. Zisserman. "Video Google: A text retrieval approach to object matching in videos*". In Proc. ICCV*, 2003.
[7] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," in Readings in Computer Vision: pp. 726–740, Morgan Kaufmann, New York, NY, USA, 1987.
[8] D. Lowe. "Object recognition from local scale-invariant features*." In Proc. ICCV,* pages 1150–1157, 1999.
[9] T.Velmurugan and T.Santhanam" A survey of Partition based clustering algorithms in Data Mining: An Experimental Approach", *Information Technology Journal* Vol.10, Issue.3, pp.478-484. (2011).
[10] S. Lloyd. "Least square quantization in PCM." *IEEE Trans. on Information Theory*, 28(2), 1982.
[11] Doreswamy, Hemanth K. S.: "A Novel Design Specification Distance (DSD) Based K-Mean Clustering Performace Evluation on Engineering Materials Database." *CoRR* abs/1301.0179 (2013).
[12] C. Bishop, *Pattern Recognition and* Machine *Learning*, Springer, New York, 2006.
[13] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification*, Second Edition, Wiley, New York, 2001.
[14] Sanjeev R. Kulkarni and Gilbert Harman "Statistical Learning Theory: A Tutorial" Princeton University, February, 20, 2011. [Online]. http://www.princeton.edu/~harman/Papers/SLT-tutorial.pdf
[15] R. Baeza-Yates and B. Ribeiro-Neto. "Modern Information Retrieval". *ACM Press*, ISBN: 020139829, 1999.
[16] C. Elkan. "Using the triangle inequality to accelerate k-means." *In Proc. ICML*, 2003.
[17] M. Muja and D. G. Lowe. "Fast approximate nearest neighbors with automatic algorithmic configuration*." In Proc. VISAPP*, 2009.
[18] Dan Ventura "SVM Examples" (2009) [Online] http://axon.cs.byu.edu/Dan/678/miscellaneous/SVM.example.pdf Accessed: June, 2013.
[19] Olga Veksler (2010)"Lecture 11: Support Vector Machines" Pattern Recognition course, CS 434s/541a, Computer Science Department, University of Western Ontario, Accessed: July, 2013. [Online], http://www.csd.uwo.ca/~olga/Courses/CS434a_541a/
[20] C Thomaz "Lecture 18:Support Vector Machines" Intelligent Data Analysis and Probabilistic Inference course slides, Accessed July, 2013, [Online], http://www.doc.ic.ac.uk /~dfg/ProbabilisticInference /IDAPISlides18.pdf
[21] Freidman, J. H., Bentley, J. L., and Finkel, R. A. (1977). An algorithm for finding best matches in logarithmic expected time. ACM Trans. Math. Softw., 3:209–226.

Authors

Nadhir Ben Halima received his BSc in Computer Engineering from the National School of Computer Sciences (ENSI), Mannouba, Tunisia, the M.S degree in Communication Networks Engineering from SantAnna School of Advanced Studies, Pisa, Italy in 2006 and the PhD in Information and Communication Technology from the University of Trento, Italy, in 2009. In 2009 he was a visiting researcher at the Department of Electrical and Computer Engineering at North Carolina State University, USA. Since September 2011, he is an assistant professor at the College of Computer Science and Engineering at Taibah University , Yanbu Branch , Saoudi Arabia. His research interest include wireless and sensor networks, cognitive networks, image watermaking and multi-robot systems.

Osama Hosam Is an Assistant Professor in Research City for Scientific Research and Technology Applications, Alexandria, Egypt. In 2007 he received his MSc. In computer systems and engineering, He pursued his study in Hunan University, China and worked in parallel in Nanjing University of Technology; in 2011 he received his PhD in Computer Engineering. He moved then to Saudi Arabia and worked and Assistant Professor and then promoted to be the Head of Computer Science department, the Collage of Computer Science and Engineering in Yanbu. His research interests include, Computer Graphics, 3D Watermarking, Stereo Vision, and Pattern Recognition.

