People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

University of Adrar

Faculty of Science and Technology

Department of Mathematics and Computer Science

A Memoir Submitted for Master Degree in Computer Science (LMD System)

Networks and Computer Network Security Option

Theme:

Finding Images In A Sequence Of Videos

Prepared And Presented By: Sarah TAOUZA

Examination Committee:

Pr.	Mohamed OMARI	Univeristy of Adrar	President
Dr.	El Mamoun MAMOUNI	Univeristy of Adrar	Examinator #1
Dr.	Mohamed DEMRI	Univeristy of Adrar	Examinator #2
Mr.	Mohamed KOHILI	Univeristy of Adrar	Supervisor

October 2017

Abstract

Pattern recognition techniques have been widely used in a variety of scientific disciplines including computer vision, image understanding, biology and so on. Although many methods present satisfactory performances for image analysis, they still have several weak points and which leave a large space room for further improvements and enhancements.

For example, the linear discriminant analysis (LDA) algorithm is able to extract discriminative features, but the small sample size (SSS) problem makes its application scope limited.

In this work, we have proposed several feature extraction and learning algorithms in order to improve the classification performance in image analysis. At the beginning we decided to study the performance of the SIFT Operator aiming to find the proper image also with changing some parameters so that we can see its influence, then, our next objective was to combine both the SIFT Operator with the Harris Corner trying to produce a more effective hybrid method, better than each one of them.

For each proposed method, extensive experiments are designed in the interest of evaluating its effectiveness to find an image in a sequence of videos, then, comparing the performance of each one and the ability of minimizing the margin error.

Acknowledgments

First, I thank Allah the Almighty for giving me courage, strength and patience to complete this modest work. I wish to express my profound gratitude to Dr. KHOHILI Mohammed, my Respected supervisor, for his excellent guidance, caring, patience, I ask Allah to bless him.

My respect and gratitude to the jury members who gave me the honor of judging this work through their availability, observations and reports that have enabled me to enhance my scientific contribution.

Thanks to all the teachers of our faculty of Sciences and Technology.

I thank all those who participated directly or indirectly to the completion of this work.

Dedicates

I am dedicating this project to the spirit of my parent and to my aunt for raising me to believe that anything was possible.

To my husband for making anything possible.

To my sun the moon and the star of my live .

To my brothers, and sisters. Especially to my brother Nesro.

To anyone who have helped me in anyway.

Sarah

Table of Contents

List of Figures

List of Tables

General Introduction

Computer and database technologies swift advancement, understanding and mining useful information from huge amount of data attract numerous efforts from the areas of databases, machine learning, and statistics. Pattern recognition is the study of how computers sense their environment, learn from stored patterns of interest, and make decisions to categorize unseen data.

Recognizing patterns is an easy task for human beings; however, it is difficult for machines to accomplish. Nevertheless, since computers have several advantages on processing speed and data storage compared to humans, many pattern recognition techniques have been proposed and applied in a variety of scientific disciplines including computer vision, image understanding, speech recognition, computational biology and so forth. In this memoir, we concentrated on investigating pattern analysis techniques and their applications to visual learning where the inputs of the system are digital images.

In general, the representations of patterns in image analysis can be pixels, curves, or features from statistical models. Having the representations at hand, correlation, distance measure, or discriminant functions can be applied to distinguish among different patterns. Usually, a learning process on training samples is required, while the prediction of unknown patterns is accomplished using the training phase learned parameters.

Depending on the nature of the application, operations such as extracting features, matching, as long as finding the proper image in sequence of video (big database). Since classifiers are applied for prediction on testing data, this memoir emphasizes the stages of feature extraction and classification.

In our application we tried to evaluate a SIFT operator and to produce an application hybrid tow feature extraction methods, the Harris Corner extraction and the SIFT Detector. in the reason of comparing the performance of our hybrid method with the SIFT detector.

Each computer scholar has a specific objective which is developing applications that are capable accomplishing difficult tasks to human beings , hence, our objective was making an application that helps finding an image in a huge database by minimizing the errors using points of interest, this app can also be helpful in many fields like security security, surveillance, diagnosis ... etc.

This thesis is organized into three chapters:

1. The first chapter presents the structure and characteristics of digital images and Video.
2. The second chapter presents the general scheme of interested point extraction and presentation of the sift and the Harris Corner definition, concept and algorithm.
3. The third chapter presents a detailed description of our proposed method, and presents the implementation and the results obtained from each experiment.

Lastly, the thesis is closed with a general conclusion and some insights on future work.

Chapter 1

Image And Video Conceptions

1.1 Introduction

To begin with, in this chapter the basic idea of image will be stated as well as the discussion of the major aspects of digital image systems. Then, The main idea of video will be summarized. Finally, at the end of the chapter, the most important uses for video codecs are going to be mentioned.

1.2 What Is An Image

An image is a visual representation of an object, a person, or a scene produced by an optical device such as a mirror, a lens, or a camera. This representation is two-dimensional (2D), although it corresponds to one of the infinitely many projections of a real-world, three-dimensional (3D) object or scene [1].

1.2.1 Types of Images

Analog image: An analog image is a 2D image $a(x, y)$ which has infinite precision in spatial parameters x and y and infinite precision in intensity at each spatial point (x, y) [2].

Digital image: A digital image is a 2D image $a[m, n]$ represented by a discrete 2D array of intensity samples, each of which is represented using a limited precision [2].

1.3 Digital Image Representation

A digital image a[m,n] described in a 2D discrete space is derived from an analog image $a(x,\ y)$ in a 2D continuous space through a sampling process that is frequently referred to as digitization. The 2D continuous image $a(x,\ y)$ is divided into N rows and M columns.

The intersection of a row and a column is termed a pixel. The value assigned to the integer coordinates $[m, n]$ with $m = 0,\ 1,\ 2,\ ...,\ M - 1$ and $n = 0,\ 1,\ 2,\ ...,\ N - 1$ is $a[m,\ n]$. In fact, in most cases $a(x,\ y)$ - which we might consider to be the physical signal that impinges on the face of a 2D sensor - is actually a function of many variables including depth (z), color (δ), and time (t).

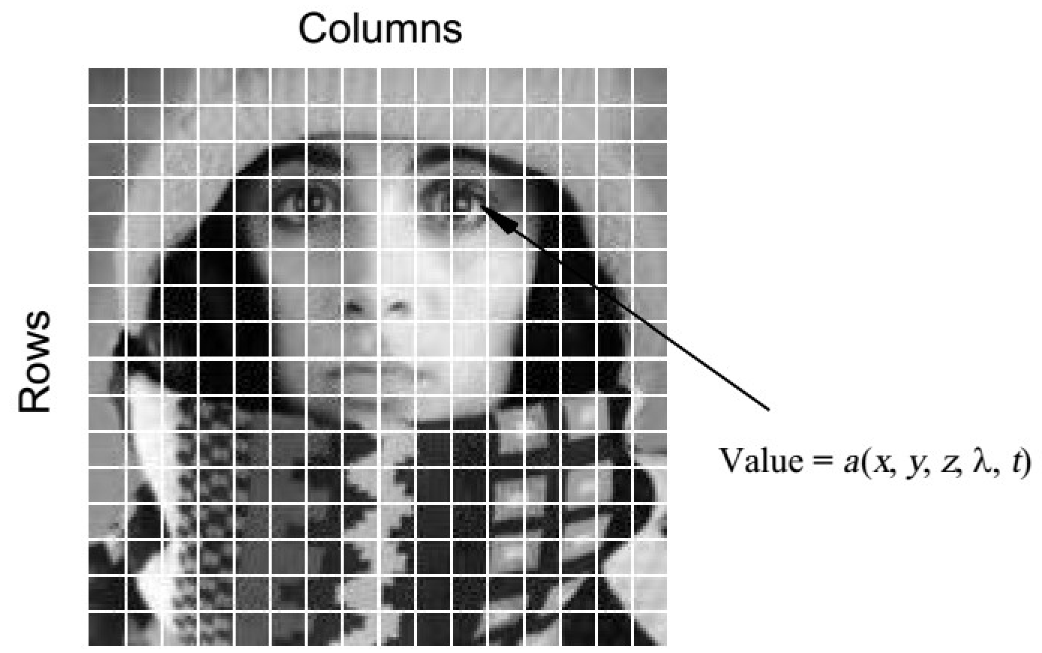


Figure 1.1 – *Digitization of a continuous image. The pixel at coordinates [m=10, n=3] has the integer brightness value 110 [3]*

The image shown in Figure 1.1 has been divided into N = 16 rows and M = 16 columns. The value assigned to every pixel is the average brightness in the pixel rounded to the nearest integer value. The process of representing the amplitude of the 2D signal at a given coordinate as an integer value with L different gray levels is usually referred to as amplitude quantization or simply quantization [3].

1.4 Digital Image Creation

To create an image which is digital, we need to covert continuous data into digital form. There are two steps in which it is done.

- Sampling
- Quantization

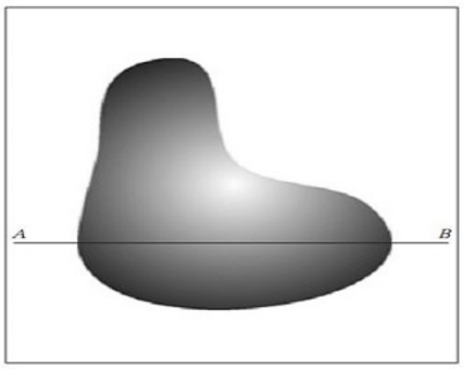
Since an image is continuous not just in its coordinates (x axis), but also in its amplitude (y axis), so the part that deals with the digitizing of coordinates is known as sampling. While the part that deals with digitizing the amplitude is known as quantization.

1.5 Basic Concepts in Sampling and Quantization

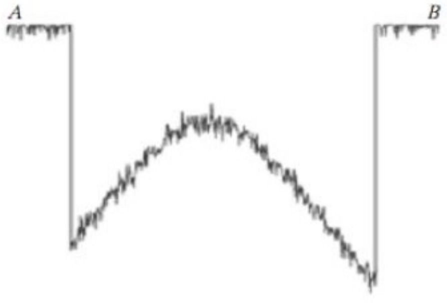
The basic idea behind sampling and quantization is illustrated in Fig.1.2 (a) shows a continuous image, $f(x,y)$, that we want to convert to digital form. An image may be continuous with respect to the x- and y-coordinates, and also in amplitude. To convert it to digital form, we have to sample the function in both coordinates and in amplitude. Digitizing the coordinate values is called sampling. Digitizing the amplitude values is called quantization.

The one-dimensional function shown in Fig.1.2 (b) is a plot of amplitude (gray level) values of the continuous image along the line segment AB in Fig.1.2 (a).The random variations are due to image noise. To sample this function, we take equally spaced samples along line AB, as shown in Fig.1.2 (c).The location of each sample is given by a vertical tick mark in the bottom part of the figure. The samples are shown as small white squares superimposed on the function. The set of these discrete locations gives the sampled function. However, the values of the samples still span (vertically) a continuous range of gray-level values. In order to form a digital function, the gray-level values also must be converted (quantized) into discrete quantities. The right side of Fig.1.2 (c) shows the gray-level scale divided into eight discrete levels, ranging from black to white. The vertical tick marks indicate the specific value assigned to each of the eight gray levels. The continuous gray levels are quantized simply by assigning one of the eight discrete gray levels to each sample. The assignment is made depending on the vertical proximity of a sample to a vertical tick mark. The digital samples resulting from both sampling and quantization are shown in Fig.1.2 (d). Starting at the top of the image and carrying out this procedure line byline produces a two-dimensional digital image.

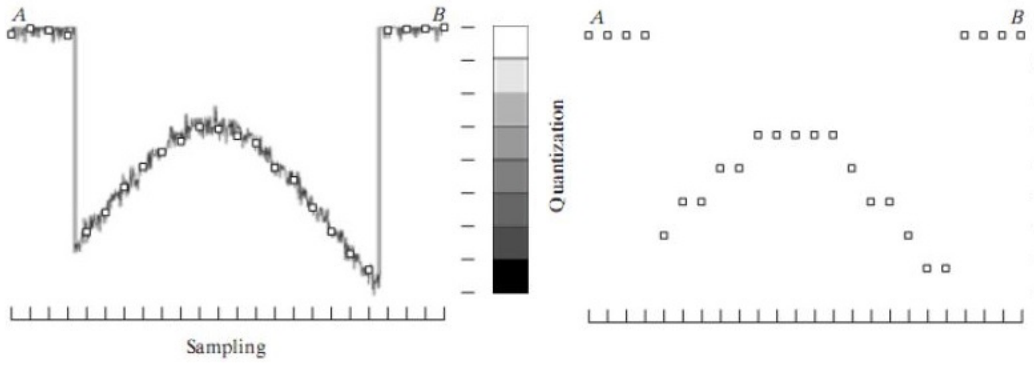
Sampling in the manner just described assumes that we have a continuous image in both coordinate directions as well as in amplitude. In practice, the method of sampling is determined by the sensor arrangement used to generate the image. When an image is generated by a single sensing element combined with mechanical motion, as in Fig.1.3, the output of the sensor is quantized in the manner described above. However, sampling is accomplished by selecting the number of individual mechanical increments at which we activate the sensor to collect data. Mechanical motion can be made very exact so, in principle; there is almost no limit as to how fine we can sample an image. However, practical limits are established by imperfections in the optics used to focus on the sensor an illumination spot that is inconsistent with the fine resolution achievable with mechanical displacements. When a sensing strip is used for image acquisition, the number of sensors in the strip establishes the sampling limitations in one image direction. Mechanical motion in the other direction can be controlled more accurately, but it makes little sense to try to achieve sampling density in one direction that exceeds the sampling limits established by the number of sensors in the other. Quantization of the sensor outputs completes the process of generating a digital image [4].



(a)



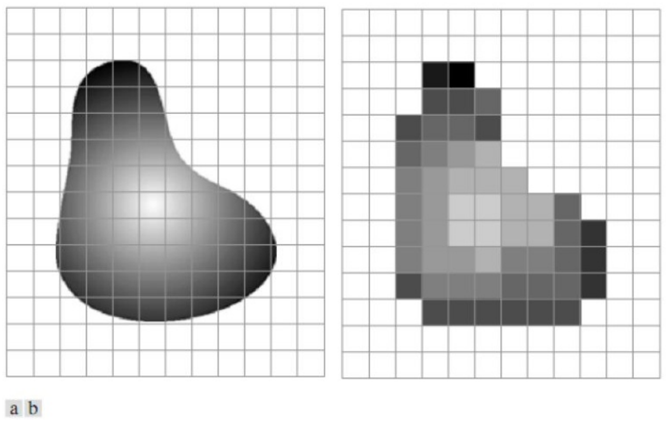
(b)



(c) (d)

Figure 1.2 – *Generating a digital image (a) Continuous image (b) A scan line from A to Bin the continuous image, used to illustrate the concepts of sampling and quantization (c) Sampling and quantization. (d) Digital scan line [4]*

When a sensing array is used for image acquisition, there is no motion and the number of sensors in the array establishes the limits of sampling in both directions. Figure 1.3 illustrates this concept. Figure 1.3 (a) shows a continuous image projected onto the plane of an array sensor. Figure 1.3 (b) shows the image after sampling and quantization. Clearly, the quality of a digital image is determined to a large degree by the number of samples and discrete gray levels used in sampling and quantization [4].



(a) (b)

Figure 1.3 – *(a) Continuous image projected onto a sensor array (b) Result of image sampling and quantization [4]*

1.6 Types of Digital Images

1.6.1 Binary

Each pixel is just black or white. Since there are only two possible values for each pixel, we only need one bit per pixel. Such images can therefore be very efficient in terms of storage. Images for which a binary representation may be suitable include text (printed or handwriting), finger prints, or architectural plans [5].

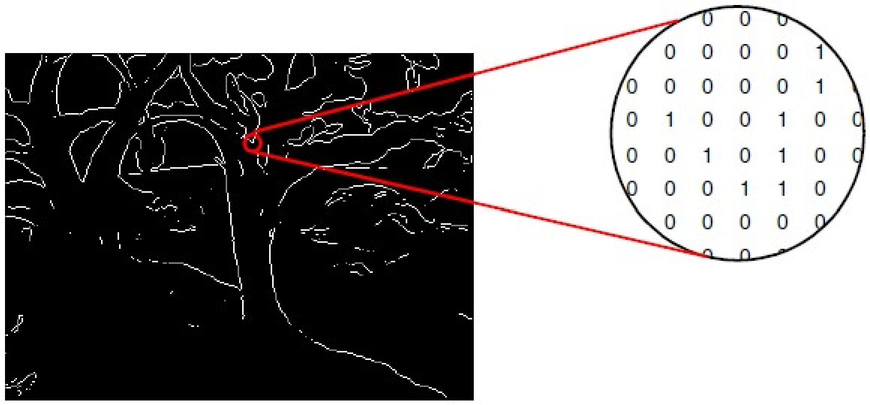


Figure 1.4 – *Binary image example [5]*

1.6.2 Grayscale

Each pixel is a shade of gray, normally from 0 (black) to 255 (white). This range means that each pixel can be represented by eight bits, or exactly one byte. This is a very natural range for image file handling. Other grayscale ranges are used, but generally they are a power of 2. Such images arise in medicine (X-rays), images of printed works, and indeed 256 different gray levels is sufficient for the recognition of most natural objects [5].

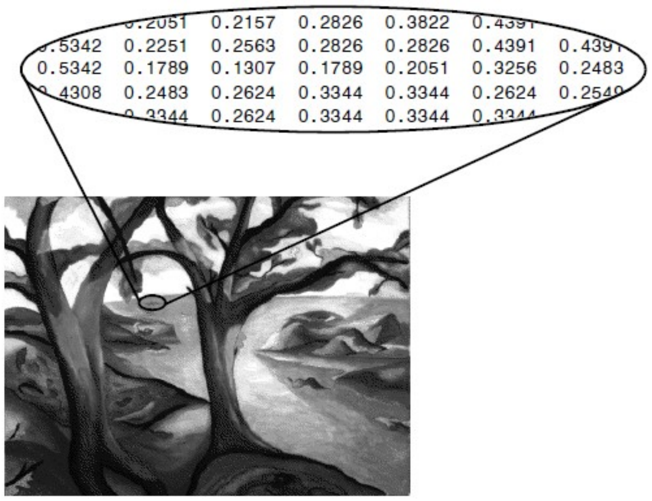


Figure 1.5 – *Grayscale image example [5]*

1.6.3 True color, or RGB

Here each pixel has a particular color; that color being described by the amount of red, green and blue in it. If each of these components has a range 0-255, this gives a total of 256^3 = 17,777,216 different possible colors in the image. This is enough colors for any image. Since the total number of bits required for each pixel is 24, such images are also called 24-bit color images. Such an image may be considered as consisting of a **stack** of three matrices; representing the red, green and blue values for each pixel. This means that for every pixel there correspond three values [5].

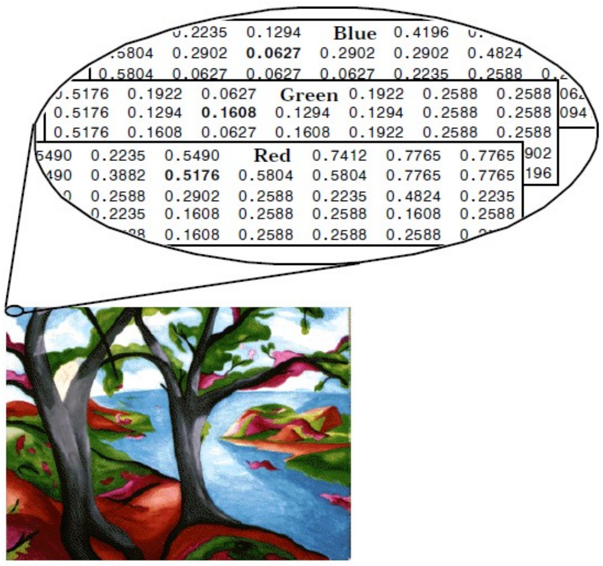


Figure 1.6 – *RGB image example [5]*

1.6.4 Indexed

Most color images only have a small subset of the more than sixteen million possible colors. For convenience of storage and file handling, the image has an associated color map, or color palette, which is simply a list of all the colors used in that image. Each pixel has a value which does not give its color (as for an RGB image), but an index to the color in the map. It is convenient if an image has 256 colors or less, for then the index values will only require one byte each to store. Some image file formats (for example, Compuserve GIF), allow only 256 [5].

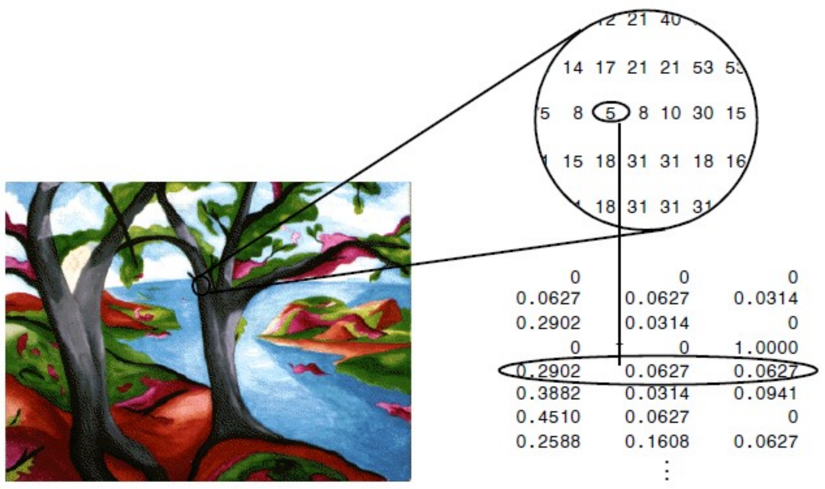


Figure 1.7 – *Indexed image example [5]*

1.7 Digital Image File Formats

There are a number of file formats in which one may store the images in files and retrieve them from files. These are known as image file format standards. Here we will present some of the most popularly used Image file format standards [6].

1.7.1 Tagged Image Format (.tif, .tif')

The .tif format is a very broad format, which can handle anything from bitmaps to compressed color palette images. The .tif format supports several compression schemes, but is often used for uncompressed images as well. This format is popular, relatively simple, and allows color [6].

1.7.2 Portable Network Graphics (.png)

This is an extensible file format that provides lossless, well compressed storage of raster images. This simple format covers the major functionalities of .tiff. Gray scale, color palette, and full-color (true-color) images are supported by this file format. It supports an optional alpha channel, and depths from 1 to 16 bits per channel [6].

1.7.3 Joint Photographic Experts Group (.jpg)

It is the most widely used standard for transmission of pictorial information and includes a variable lossy encoding as part of the standard, set by a quality parameter [6].

1.7.4 MPEG (.mpg)

This format is extensively used throughout the Web and is used only for motion images. This uses compression, yielding only lossy videos[6].

1.7.5 Graphics Interchange Format (.gif)

This format supports 8-bit color palette images and is not very popular among the image processing researchers [6].

1.7.6 RGB (.rgb)

This is an image file standard from Silicon Graphics for color images [6].

1.7.7 RAS (.ras)

This is an uncompressed scan flout of three color bands for Sun Raster images [6].

1.7.8 Postscript (.ps, .eps, .epsf)

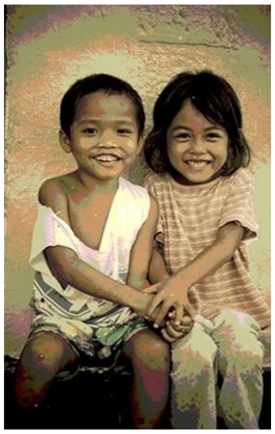
This image format is mainly used while introducing images or figures in a book or note and for printing. In postscript format, gray level images are represented by decimal or hex numerals encoded in ASCII [6].

(a) *Original file*

(b) *.tiff file*

(c) *.bmp file*

(d) *.jpg file*


(e) *.gif file*

(f) *.png file*

Figure 1.8 – *Les valeurs de l'effort normal après modalisation en logiciel sap 200*

1.7.9 Portable Image File Formats

Some of the most commonly used image file formats are Portable Image Formats, which include Portable Bitmap, Portable Graymap, Portable Pixmap, and Portable network map. The default suffixes for these formats are .pbm, .pgm, .ppm, and .pnm respectively. These formats are a convenient method of saving and reading the image data. These are some of the image formats which support all kinds of images of increasing complexity-from bits to gray levels to color pixmaps of various sorts.

1.7.9.1 PPM

A PPM file consists of two parts, a header and the image data. The header consists of at least three parts. The first part is a magic PPM identifier. The PPM identifier can be either P3 (for ASCII format image data) or P6 (data in binary format). The next part consists of the width and height of the image as ASCII numbers. The last part of the header gives the maximum value of the color components for the pixels. In addition to the above, a comment can be placed anywhere with a character; the comment extends to the end of the line.

1.7.9.2 PGM

This format is identical to the above except it stores gray scale in formation, that is, one value per pixel instead of three (r, g, b). The only difference in the header section is the magic identifiers which are P2 and P5 ; these correspond to the ASCII and binary form of the data respectively.

1.7.9.3 PBM

PBM stores single-bit pixel image as a series of ASCII 0 or 1's. Traditionally 0 refers to white while 1 refers to black. The header is identical to PPM and PGM format except there is no third header line (the maximum pixel value doesn't have any meaning). The magic identifier for PBM is P1 [6].

1.8 Video

1.8.1 Concepts and Definitions

In an analogue system a video camera produces an analogue signal of an image scanned from left to right and from top to bottom making up a frame [7]. The choice of number of scanned lines per picture is a trade-off between the bandwidth, flicker and resolution.

Any video consists of frames of a scene taken at various subsequent intervals in time. Each frame represents the distribution of light energy and wavelength over a finite size area and is expected to be seen by a human viewer [8]. Digital frames have a fixed number of rows and columns of digital values, called picture elements or pixels. These elements hold quantized values that represent the brightness of a given color at any specific point [9]. The number of bits per pixel is known as pixel depth or bits per pixel.

1.8.1.1 Frame rate

The number of frames displayed per second. The illusion of motion can be experienced at frame rates as low as 12 frames per second [10]. Standard-Definition television typically uses 24 frames per second, and HD videos can use up to 60 frames per second.

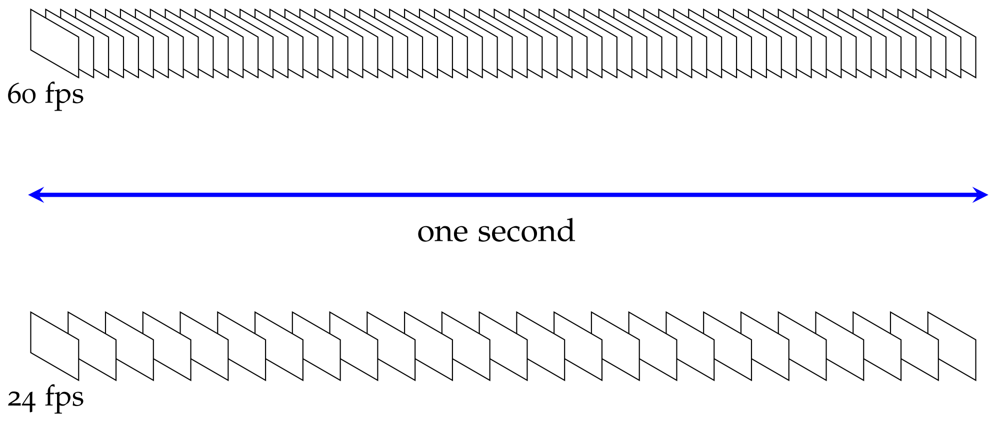


Figure 1.9 – *Example of frame rate [11]*

1.8.1.2 Frame dimensions

The width and height of the image expressed in the number of pixels. Some common formats include:

- CIF (Common International Format), defines a video sequence with a resolution of 352×288.
- QCIF (Quarter CIF), defines a video sequence with a resolution of 176×144.

1.8.1.3 Bit Rate (BR)

Numbers of bits that are processed per unit time are known as bit rate in computing. But more specifically in digital multimedia, we can say that the number of bits used in per unit time to represent a video or audio is known as bit rate. Higher bit rate will give higher video quality e.g. a VCD (Video Compact Disk) with the bit rate 1 Mbit/s has lower quality than a DVD (Digital Versatile Disk) with bit rate of 5 Mbits/s. There are two different types of bit rate.

- Variable Bit Rate (VBR): In any video section, there are different kinds of scenes. They can vary from very simple i.e. clear sky, to very complex i.e. a tree with a lot of branches and leaves. Therefore the number of bits required for a scene varies according to the scene complexity. The best way is to save bits from simple scenes and use them for complex one and that's how a variable bit rate decoder works. Although process for calculating bit rate is very complex.
- Constant/Fixed Bit Rate (CBR): For some applications, we need to transmit data on a constant bit rate. Mostly broadcast mediums like cable or satellite etc have the limitation of fixed bit rate. Compressors for constant bit rate are not as efficient as variable bit rate encoders are, but still there are very high quality encoders provided by MPEG-2 systems for CBR.

VBR is much more efficient for achieving good quality. It gives constant good quality as it can change bit rate for complex scenes. On other hand CBR gives us variable quality of a video, as it has same bit rate for each n every scene of a video[12].

1.8.1.4 Natural video scenes

A typical "real world" or "natural" video scene is composed of multiple objects each with their own characteristic shape, depth, texture and illumination. The color and brightness of a natural video scene changes with varying degrees of smoothness throughout the scene ("Continuous tone"). Characteristics of a typical natural video scene (Fig.1.10) that are relevant for video processing and compression include spatial characteristics (texture variation within scene, number and shape of objects, color, etc.) and temporal characteristics (object motion, changes in illumination, movement of the camera or viewpoint and so on)[11].

Figure 1.10 – *Still image from natural video scene [11]*

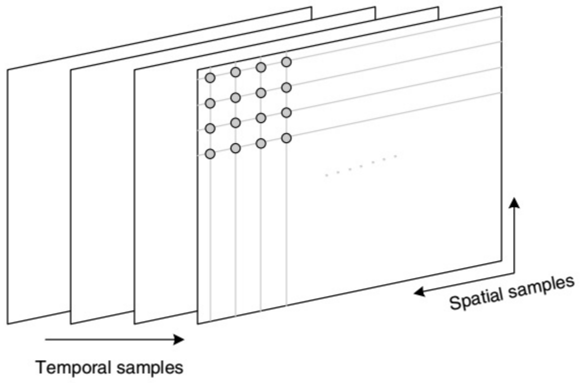


Figure 1.11 – *Spatial and temporal sampling of a video sequence [11]*

1.8.2 Capture

A natural visual scene is spatially and temporally continuous. Representing a visual scene in digital form involves sampling the real scene spatially (usually on a rectangular grid in the video image plane) and temporally (as a series of still frames or components of frames sampled at regular intervals in time) (Fig.1.11). Digital video is the representation of a sampled video scene in digital form. Each spatio-temporal sample (picture element or pixel) is represented as a number or set of numbers that describes the brightness (luminance) and color of the sample [11].

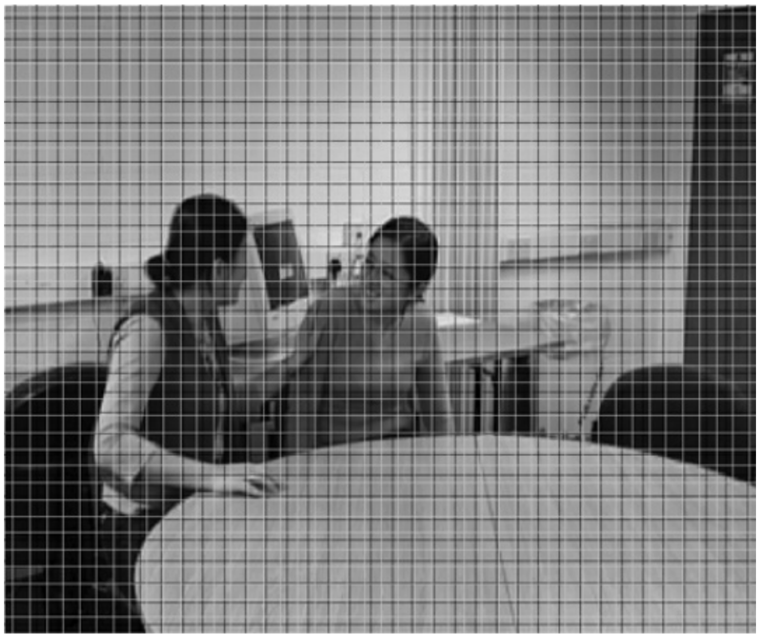


Figure 1.12 – *Image with 2 sampling grids [11]*

To obtain a 2D sampled image, a camera focuses a 2D projection of the video scene onto a sensor, such as an array of Charge Coupled Devices (CCD array). In the case of color image capture, each color component is separately filtered and projected onto a CCD array.

1.8.2.1 Spatial Sampling

The output of a CCD array is an analogue video signal, a varying electrical signal that represents a video image. Sampling the signal at a point in time produces a sampled image or frame that has defined values at a set of sampling points. The most common format for a sampled image is a rectangle with the sampling points positioned on a square or rectangular grid. Figure 1.13 shows a continuous-tone frame with two different sampling grids superimposed upon it.

Sampling occurs at each of the intersection points on the grid and the sampled image may be reconstructed by representing each sample as a square picture element (pixel). The visual quality of the image is influenced by the number of sampling points. Choosing a "coarse" sampling grid (the black grid in Figure 1.13) produces a low resolution sampled image (Fig.1.14) whilst increasing the number of sampling points slightly (the gray grid in Figure 1.13) increases the resolution of the sampled image (Fig.1.14) [11].

1.8.2.2 Temporal Sampling

A moving video image is captured by taking a rectangular "snapshot" of the signal at periodic time intervals. Playing back the series of frames produces the appearance of motion. A higher temporal sampling rate (frame rate) gives apparently smoother motion in the video scene but requires more samples to be captured and stored. Frame rates below 10 frames per second are sometimes used for very low bitrate video communications (because the amount of data is relatively small) but motion is clearly jerky and unnatural at this rate. Between 10 and 20 frames per second is more typical for low bit-rate video communications; the image is smoother but jerky motion may be visible in fast-moving parts of the sequence. Sampling at 25 or 30 complete frames per second is standard for television pictures (with interlacing to improve the appearance of motion, see below); 50 or 60 frames per second produces smooth apparent motion (at the expense of a very high data rate) [11].

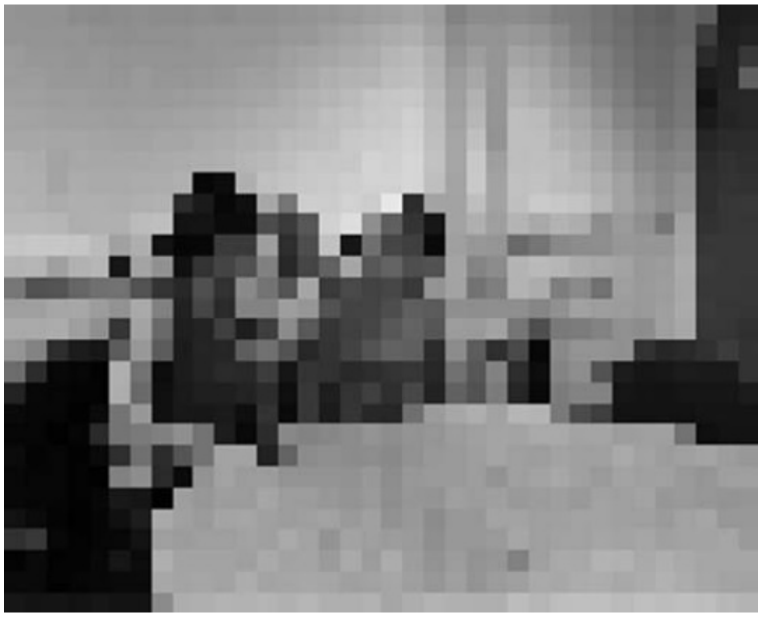


Figure 1.13 – *Image sampled at coarse resolution (black sampling grid) [11]*

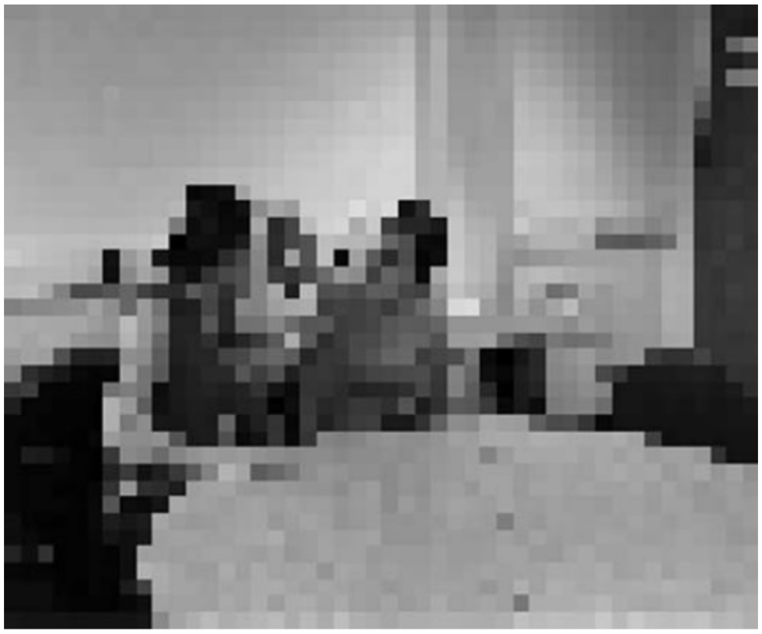


Figure 1.14 – *Image sampled at slightly finer resolution (gray sampling grid) [11]*

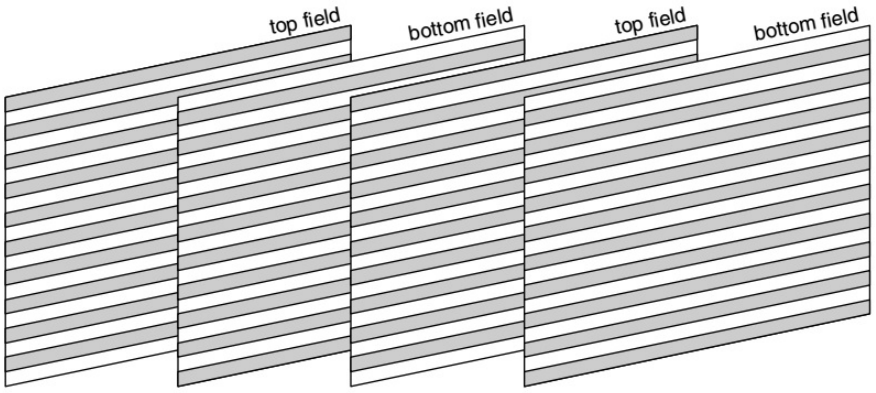


Figure 1.15 – *Interlaced video sequence [11]*

1.8.2.3 Frames and Fields

A video signal may be sampled as a series of complete frames (progressive sampling) or as a sequence of interlaced fields (interlaced sampling). In an interlaced video sequence, half of the data in a frame (one field) is sampled at each temporal sampling interval. A field consists of either the odd-numbered or even-numbered lines within a complete video frame and an interlaced video sequence (Fig.1.15) contains a series of fields, each representing half of the information in a complete video frame (e.g. Fig.1.16 and Fig.1.17). The advantage of this sampling method is that it is possible to send twice as many fields per second as the number of frames in an equivalent progressive sequence with the same data rate, giving the appearance of smoother motion. For example, a PAL video sequence consists of 50 fields per second and, when played back, motion can appears smoother than in an equivalent progressive video sequence containing 25 frames per second [11].

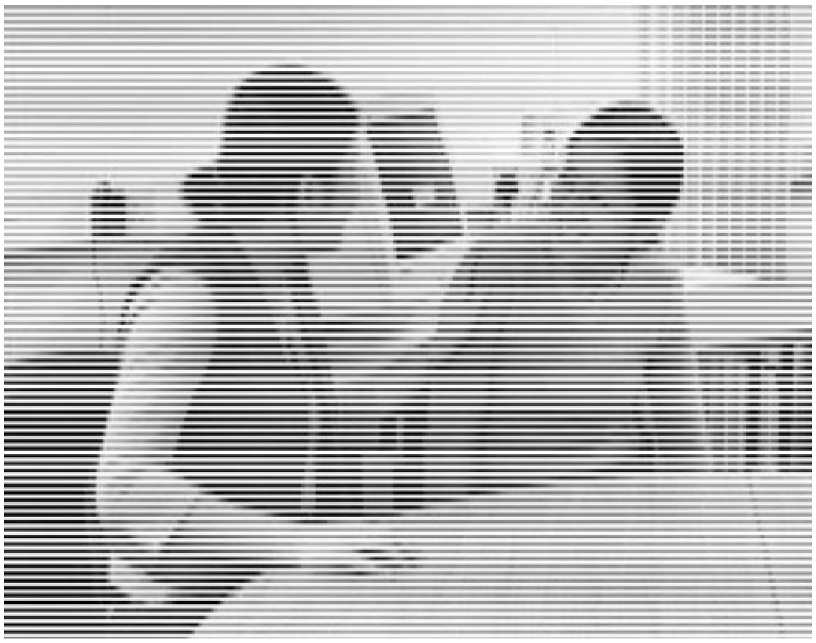


Figure 1.16 – *Top field [11]*

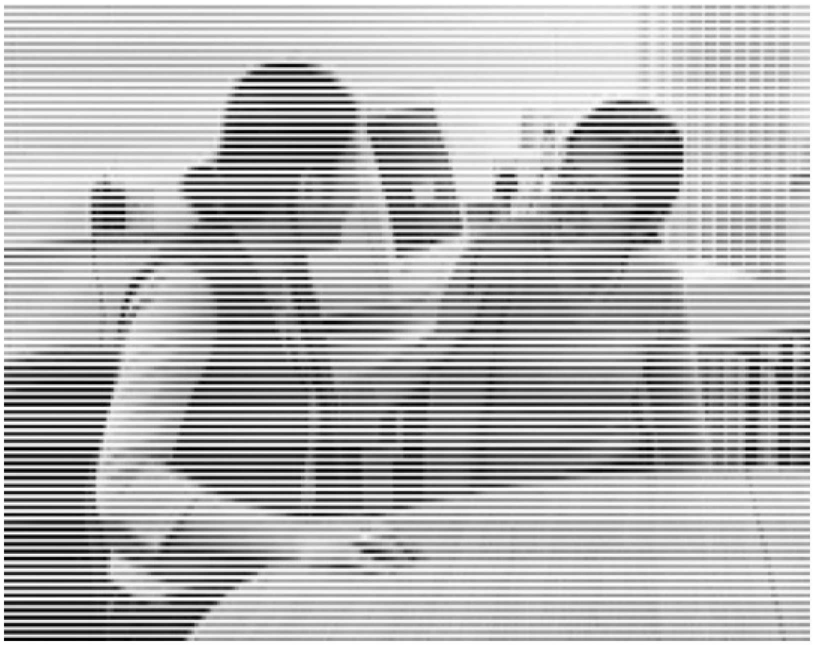


Figure 1.17 – *Buttom field [11]*

1.8.3 Video formats

A variety of video standards are there which define the resolution and colors for display. For a PC, both the monitor and the video adapter determine the support for a graphics standard. The monitor must be capable of displaying the resolution and the colors defined by the standard whereas the video adapter needs to transmit the appropriate signals to the monitor. Some of the popular video standards along with their respective parameters are listed here in tabular forms.

Table 1.1 displays the uncompressed bit rates of some video formats. It can be clearly observed that even QCIF at 15 fps (i.e., relatively low quality video suitable for video telephony) requires 4.6 Mbps for storage or transmission. Table 1.2 shows typical capacities of popular storage media and transmission networks [13].

Table 1.1 – *Video frame format (Uncompressed bit rates) [13]*

Video format	Color Resolution	Intensity Resolution	Bits per second (uncompressed)	Frames per second
QCIF	88 × 72	176 × 144	4.6 Mbps	15
CIF	176 × 144	352 × 288	36.5 Mbps	30
ITU-R 601	429 × 525	858 × 525	216 Mbps	30

1.8.4 Video standards

Basically there are two families of standards:

1) ISO/IEC (International Standards Organization and International Electro- Technical Commission)

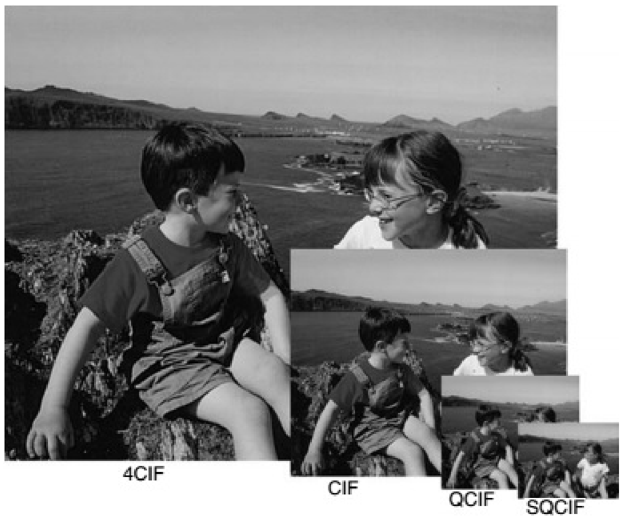


Figure 1.18 – *Video frame sampled at range of resolutions [11]*

Table 1.2 – *Typical storage capacities [13]*

Media/Network	Capacity
ADSL Typical	1-2 Mbps (downstream)
Ethernet LAN	(10 Mbps) Maximum 10 Mbps / Typical 1-2 Mbps
V.90 MODEM	56 kbps downstream / 33 kbps upstream
ISDN-2	128 kbps
CD-ROM	640 Mbytes
DVD-5	4.7 Gbytes

2) ITU (International Telecommunications Union)

- ISO/ IEC produced the MPEG standards which are the standard formats for video compression.
- ITU-T developed several recommendations for video coding such as H.261, H.263 starting from 1984 till it was approved in 1990 [13].

1.8.5 Explanation of File Formats

The different types of formats are technically referred to as "container formats". The differences between them lie in whether or not they compress the video and audio data they contain, and if so, how they go about compression [14]. Each container has its own specific set of compatibility. For instance, some containers were developed by Microsoft with Windows users in mind. Likewise, Apple has its own proprietary containers for use on its Macintosh Operating Systems.

Lets list some of the common formats and their associations:

Table 1.3 – *Digital Video Containers and File Quick Chart Explanation [15]*

Option	Format	Description
AVI	Video	AVI stands for Audio Video Interleave. It is Microsoft's video container format. .AVI files are good for Windows Media Player (WMP) and other online video players. The video retains quality, but they're large files.
MOV	Video	A .Mov is a container file created by Apple to play QuickTime movies. .MOV and Mpeg 4 containers use the same MPEG-4 codecs and are usually interchangeable. MOV files do work on PCs. MOV files are large files, but they look great.
MPEG4/MPEG	Video	MPEG stands for Motion Picture Experts Group. Many video cameras output footage in the MPEG format. YouTube converts uploaded videos to either Flash (FLV) or MPEG (MPG) formats. MPEG4 was created for the internet. It gives excellent quality and a small file up to 5× smaller than MOV files.
MP4 1080p or 1080i(HD)	Video	Full 1920 × 1080 widescreen HD resolution perfect for HD-TV, Blu-Ray disks and high quality streaming. NBC and PBS broadcast in 1080i.
MP4 720p (HD)	Video	Standard 1280 × 720 HD resolution perfect for HD-TV, tablets, laptops and desktops. ABC, Fox and ESPN broadcast in 720p.
MP4 H.264	Video	Codec for Blu-Ray disks. Widely used for YouTube, Vimeo, iTunes, Flash and also for HD-TV. Sony's HD video cameras use AVCHD, which is H.264.
MP4 360p	Video	640 × 360 resolution for standard definition (non HD) devices with a much smaller file size.
MP4 240p	Video	320 × 240 resolution for standard definition (non HD) devices with a smaller file size. This is always a Flash Video format. Use the MP4 360p rather than MP4 240p.
FLV (FLASH)	Video	Flash is the most common file used online and they play in the Adobe Flash Player that almost all computer users have downloaded on their computers for free. Apple is moving away from Flash so these files don't work on iPhones and iPads. Many video sharing sites convert any video you upload to Flash files for sharing. The file sizes are of good quality and they're small.
FLV 480p	Video	640 × 480 pixel Flash Video Format best for movies on a tablet or similar large mobile device.
FLV 360p	Video	640 × 360 pixel Flash Video Format great for blogs and most mobile devices.
FLV 240p	Video	320 × 240 pixel Flash Video Format aimed at mobile devices and mobile websites.
NTSC 525	Video	Non-HD, analog standard definition video used in the U.S. All TVs in the U.S. broadcast NTSC from 1941 until the conversion to HD. Aspect ratio 4:3. 30 frames per second. 525 scanned lines. Not an online/digital format choice forstreaming.
WMV	Video	A WMV is a Windows Media Player format. They're very compressed and don't look great. They're also PC non Mac) oriented. The files are so tiny that you can email video.

1.9 Conclusion

First of all, We summarized in this chapter the not only the structure but also the characteristics of digital images and videos too, we also introduced some important concepts such as sampling and quantization, which are used in order to know more about digital images and videos.

Then, we explained some video codec related concepts, like the frame rate, the frame dimensions and the bit rate. Other basic concepts were deepened, among them we mention the following video formats: QCIF, CIF and ITUR601, along with two videos standard families.

Chapter 2

Features Extraction

2.1 Introduction

In this chapter we define and introduce the feature extraction, and explain his main idea and his algorithms that we use in our project the SIFT Detector and the Harris Detector. We tried to give the principle of those algorithm.

2.2 Feature Extraction

Feature extraction that is one of the main topics in Photogrammetry and Computer Vision (CV). This process consists of the extraction of features of interest from two or more images of the same object and of the matching of these features in adjacent images. In aerial and close-range photogrammetry, image features are necessary for automatic collimation procedures such as image orientation, DSM generation, 3D reconstruction, and motion tracking. In CV, features are used in various applications including: model based recognition, texture recognition, robot localization [16], 3D scene modelling [17], building panoramas [18], symmetry detection and object categorization. In the last 25 years, many photogrammetric and CV applications dealing with feature extraction have been developed. Photogrammetric research has led to feature extraction algorithms, called interest operators or point detectors, while many region detector techniques have been developed in CV.

2.3 Feature Extraction Algorithms

Interest operators extract salient image features, which are distinctive in their neighbourhood and are reproduced in corresponding images in a similar way [19]; at the same time, interest operators supply one or more characteristics, which can be used in the image matching. Region detector operators, instead, search for a set of pixels which are invariant to a class of transformations (radiometric and geometric distortions). The term region differs from classical segmentation since the region boundaries do not have to correspond to changes in image appearance such as colour or texture [20]. These operators have been developed for when the normal stereo image acquisition condition is not required. Region operators detect features that do not vary with different geometrical transformations (scale, affine transformation, etc.). A descriptor, which describes the extracted feature using a 2D vector that contains gradient pixel intensity information, is associated to each region. This information may be used to classify the extracted regions or to perform the matching process. Although region detector/descriptors are computationally slower than those of interest points, the experimental results show that these detectors have a wider application range. Interest in these detectors in the photogrammetric field is quickly increasing due to the introduction of new image acquisition techniques, which do not comply with the normal stereoscopic case. Images acquired through Mobile Mapping Technology [21] are usually extracted from video-sequences with low-resolution quality. Consequently, the orientation process is hampered by illumination problems, the limited dynamic range of the video-cameras, sensor noise, narrow baselines and projective distortions. Oblique photogrammetric images, which are commonly used for the generation of 3D city modelling [22], offer images that are affected by high projective distortions which must be carefully processed. Finally, image sequences acquired using low-cost UAV platforms [23] do not assure the normal taking geometry.

Interest point extractors and matchers, which are traditionally used in photogrammetry (Forstner operator [24], Harris operator [25], Cross-Correlation etc.), are usually inefficient for these applications as they are unable to give reliable results under difficult geometrical and radiometrical conditions (convergent taking geometry, strong affine transformations, lack of texture etc) [26].

2.4 The Scale Invariant Feature Transform (Sift)

The fundamental work on the SIFT descriptor is presented in the scientific paper by David G. Lowe ([27]). This paper describes image features that have many properties that make them suitable for matching differing images of an object or scene. The features are invariant to image scaling and rotation, and partially invariant to change in illumination and 3D camera viewpoint. They are well localized in both the spatial and the frequency domains, reducing the probability of disruption by occlusion, clutter, or noise. Large number of features can be extracted from typical images with efficient algorithms. In addition, the features are highly distinctive, which allows a single feature to be correctly matched with high probability against a large database of features, providing a basis for object and scene recognition ([27]). These features may be extracted very efficiently with minimal cost as more expensive operations are performed only in case of successful pass of an initial check. Here are the main computational stages of retrieving the image features set:

- scale-space extrema detection - first step is to go over all scales and image locations with difference-of-Gaussian function and identify potential scale and orientation invariant interest points.

  Computed from the difference of two nearby scales separated by a constant multiplicative factor k, scale-space extrema in the difference-of-Gaussian function convolved with the image, $D(x,\ y,\ \sigma)$ is used to efficiently detect stable feature point locations in scale space.

  $$D(x,\ y,\ \sigma) = (G(x,\ y,\ \sigma) - G(x,\ y,\ \sigma)) * I(x,\ y) = L(x,\ y,\ k\sigma) - L(x,\ y,\ \sigma) \tag{2.1}$$

  A powerful approach to the computation of $D(x,\ y,\ \sigma)$ is shown on Fig. 2.1.

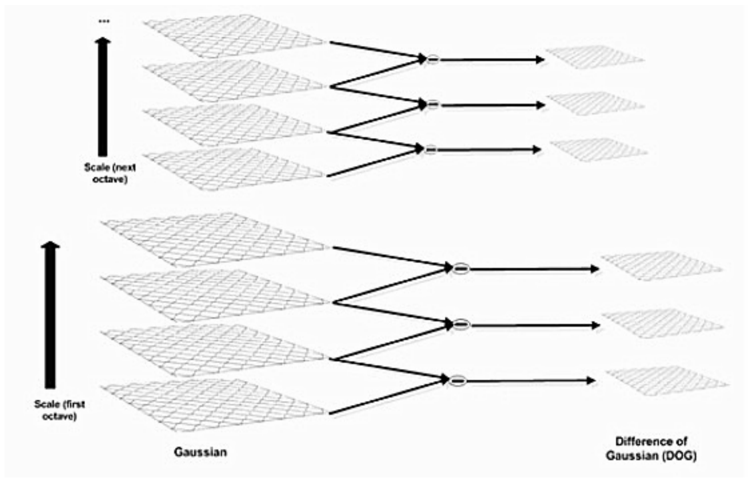


Figure 2.1 – *For each octave of scale space, the initial image is repeatedly convolved with Gaussians to produce the set of scale space images shown on the left. Adjacent Gaussian images are subtracted to produce the difference-of-Gaussian images on the right. After each octave, the Gaussian image is down-sampled, by a factor of 2, and the process is repeated. Image is adapted from ([27])*

In order to detect the local maxima and minima of $D(x, y, \sigma)$, each sample point is compared to its eight neighbors in the current image and nine neighbors in scale above and below (see Fig. 2.2). The point is considered to be a good candidate if and only if it is larger or smaller than all of them. An important issue is to determine the frequency of sampling in the image and scale domains that is needed to reliably detect the extrema, as the extrema that are close together tend to be sensitive to small image perturbations.

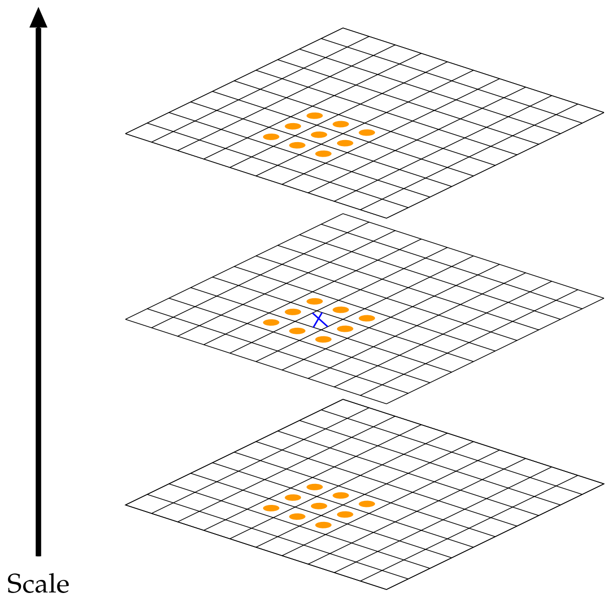


Figure 2.2 – *Maxima and minima of the difference-of-Gaussian images are detected by comparing a pixel (marked with X) to its 26 neighbors in 3×3 regions at the current and adjacent scales (marked with circles). Image is adapted from ([27])*

Lowe ([27]) defines a matching scale as being within a factor of $\sqrt{2}$ of the correct scale, and a matching location as being within σ pixels, where σ is the scale of the keypoint.

Also, additional little smoothing is needed prior to creation of the first octave of scale space.

- keypoint localization, basing on measures of keypoint stability - as we have a interest point candidate, we proceed to perform a detailed fit to the nearby data for location, scale, and ratio of principal curvatures. This way we reject the points that have low contrast (and are therefore sensitive to noise) or are poorly localized along an edge. Lowe ([27]) uses the Taylor expansion (up to the quadratic terms) of the scale-space function, $D(x, y, \sigma)$, shifted so that the origin is at the sample point:

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}}\mathbf{x} + \frac{1}{2}\mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2}\mathbf{x} \tag{2.2}$$

  where D and its derivatives are evaluated at the sample point and $\mathbf{x} = (x, y, \sigma)^T$ is the offset from this point. Such an interpolation method is also used in [28]. The location of the extremum, $\hat{\mathbf{x}}$, is determined by taking the derivative of this function with respect to $\mathbf{x}$ and setting it to zero, thus:

$$\hat{\mathbf{x}} = -\frac{\partial^2 D^{-1}}{\partial \mathbf{x}^2}\frac{\partial D}{\partial \mathbf{x}} \tag{2.3}$$

  The Hessian and derivative of D are approximated by using differences of neighboring sample points. If the offset $\hat{\mathbf{x}}$ is larger than 0.5 in any dimension, then it means that the extremum lies closer to a different sample point. In this case, the sample point is changed and the interpolation performed instead about that point. The final offset $\hat{\mathbf{x}}$ is added to the location of its sample point to get the interpolated estimate for the location of the extremum. Unstable extrema having low contrast are rejected by means of the function, $D(\hat{\mathbf{x}})$:

$$D(\hat{\mathbf{x}}) = D + \frac{1}{2}\frac{\partial D^T}{\partial \mathbf{x}}\hat{\mathbf{x}} \tag{2.4}$$

- orientation assignment - an invariance to image rotation is achieved by assigning consistent orientation to each keypoint based on local image properties, afterwards the keypoint descriptor is also represented relative to this orientation. The scale of the keypoint is used to select the Gaussian smoother image L with the closest scale, so that all computations are performed in a scale-invariant manner. For each image sample $L(x, y)$ at this scale, the gradient magnitude, $m(x, y)$, and the orientation, $\theta(x, y)$, is precomputed using pixel differences:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) + L(x, y-1))^2} \tag{2.5}$$

$$\theta(x, y) = tan^{-1}((L(x, y+1) - L(x, y-1))/(L(x+1, y) - L(x-1, y))) \tag{2.6}$$

  An orientation histogram is formed from the gradient orientations of sample points within a region around the keypoint. The orientation histogram has 36 bins covering the 360 degree range of orientations. Each sample added to the histogram is weighted by its gradient magnitude and by a Gaussian-weighted circular window with a σ that is 1.5 times that of the scale of the keypoint. Peaks in the orientation histogram correspond to dominant directions of local gradients. The highest peak in the histogram is detected, and then any other local peak that is within 80% of the highest peak is used to also create a keypoint with that orientation. Therefore, at locations with multiple peaks of similar magnitude, there will be multiple keypoints created at the same location and scale but different orientations ([27]).

- keypoint descriptor - by means of the keypoint scale, the Gaussian blur level of the current image is defined. The gradient magnitudes and orientations are sampled around the keypoint location on the image. Invariance is achieved by rotating the gradient orientations and descriptor coordinates in relation to the orientation of the feature point. A Gaussian weighting function with σ equal to one half the width of the descriptor window is used to assign a weight to the magnitude of each sample point. The purpose of the Gaussian window is to avoid sudden changes in the descriptor with small changes in the position of the window, and to give less emphasis to the gradients that are far from the center of the descriptor, as these are most affected by misregistration errors ([27]). The interest point descriptor is shown on figure 2.3. To achieve the illumination

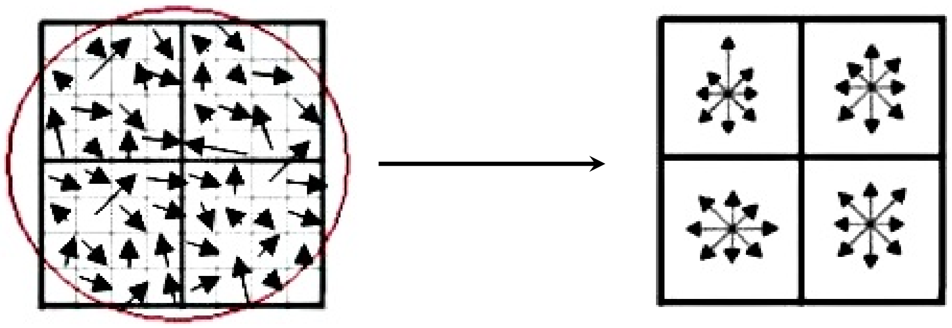


Figure 2.3 – *A keypoint descriptor is created by first computing the gradient magnitude and orientation at each image sample point in a region around the keypoint location, as shown on the left. These are weighted by a Gaussian window, indicated by the overlaid circle. These samples are then accumulated into orientation histograms summarizing the contents over* 4×4 *subregions, shown on the right, with the length of each arrow corresponding to the sum of gradient magnitudes near that direction within the region. This figure shows a* 2×2 *descriptor array computed from a* 8×8 *set of samples, whereas the experiments in this paper use* 4×4 *descriptors computed from* 16×16 *sample array. Image is adapted from ([27])*

invariance, the features vector is normalized to the unit length. The change in contrast will be canceled by vector normalization as the multiplying the each pixel value by a constant will also multiply gradients by the same constant.

The SIFT keypoints described in the paper ([27]) are particularly useful due to their distinctiveness, which enables the correct match for a keypoint to be selected from a large database of other keypoints. This distinctiveness is achieved by assembling a high-dimensional vector representing the image gradients within a local region of the image. The keypoints have been shown to be invariant to image rotation and scale and robust across a substantial range of affine distortion, addition of noise, and change in illumination. A large number of keypoints may be extracted from typical image, which leads to robustness in extracting small objects among clutter. The fact that keypoints are detected over a complete range of scales means small local features are available for matching small and highly occluded objects, while large keypoints perform well for images subject to noise and blur.

2.5 Harris Corner Detection

A corner can be considered as the intersection of two well-defined edges. The Harris corner detection algorithm searches for corner points by looking at regions within an image which contains high gradient values in all directions. A window is iteratively scanned across the X and Y gradients of the input image, and if high changes in intensity exist in multiple directions, then a corner is inferred to exist within the current window. Figure 2.4 shows the different types of regions that can exist within an image.

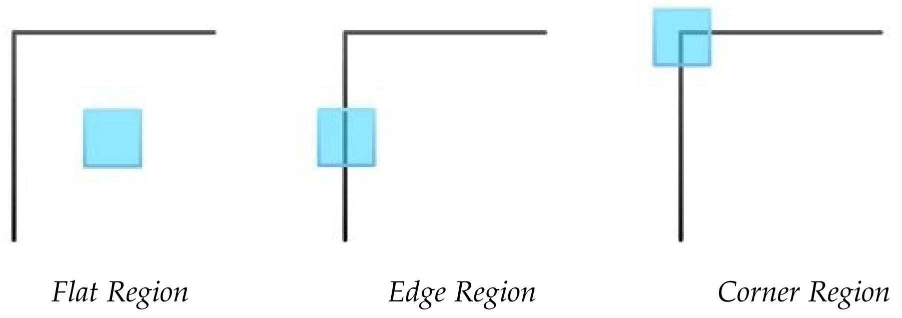


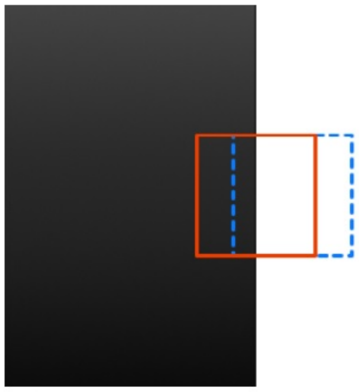
Figure 2.4 – *Directional Intensity Change Types*

2.5.1 Corner Detection Mathematical Description

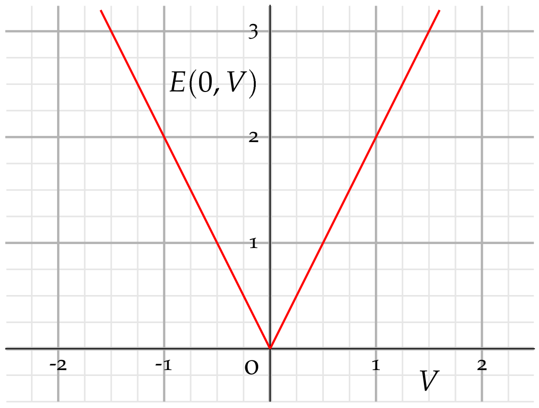
A corner within a region of interest (ROI) can be identified by calculating the sum of squared difference (SSD) between the ROI and shifted nearby regions. The SSD formula, shown in Equation 2.7, quantifies the difference between ROI and shifted region by summating the squared differences pixel by pixel. The function I, in Equation 2.7, represents the input image. The $(x,\ y)$ coordinates specify the ROI, and the $(\Delta u,\ \Delta v)$ coordinate specifies the offset of the shifted region from the ROI.

$$E(\Delta u,\ \Delta v) = \sum_{(x,\ y) \in ROI} I(x + \Delta u,\ y + \Delta v) - I(x,\ y)^2 \tag{2.7}$$

Figures 2.5-2.7 (a) show the ROI (red box) containing an edge, defined by the $(x,\ y)$ coordinates, and the shifted region (blue dashed box), defined by the $(\Delta u,\ \Delta v)$ offset. Consider iterating the shifted region away from the ROI in only the horizontal direction, shown in Figure 2.5 (a), thus only varying the Δu coordinate. When the Δu coordinate is at zero, the ROI and shifted region are the same region, thus resulting in a SSD of zero. As the shifted region iterates farther from the ROI in the horizontal direction the SSD increases significantly, shown in Figure 2.5 (b). This implies that the ROI and shifted region become more different as the shifted region iterates in the horizontal direction.



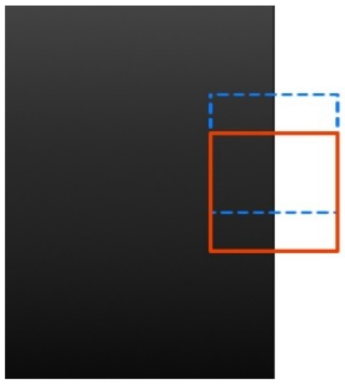
(a) *Shifted Region Horizontally*



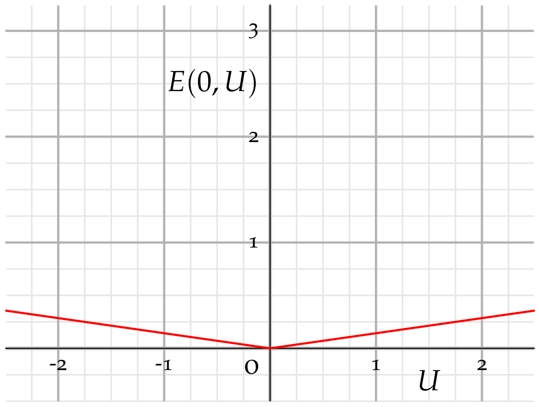
(b) *SSD Increases Significantly*

Figure 2.5 – *SSD When Shifting Region Horizontally Away From ROI*

Now consider iterating the shifted region away from the ROI in only the vertical direction, shown in Figure 2.6 (a), thus only varying the Δv coordinate. As the shifted region iterates farther away from the ROI in the vertical direction, the SSD does not increase much, shown in Figure 2.6 (b). This implies that the ROI and shifted region stay similar as the shifted region iterates in the vertical direction.



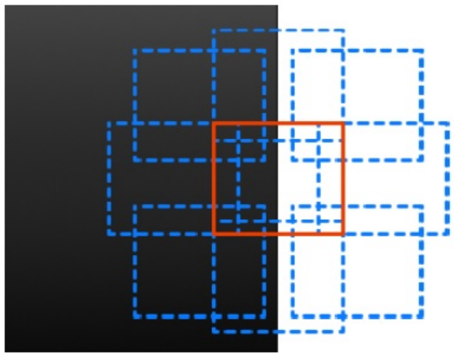
(a) *Shifted Region Horizontally*



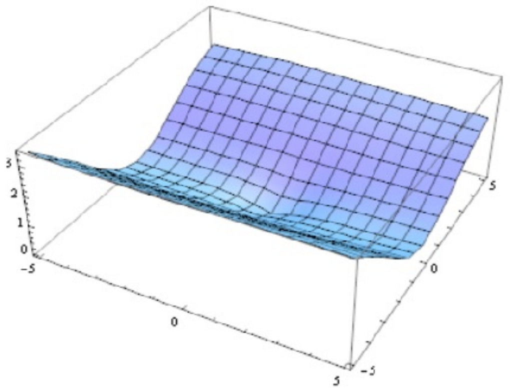
(b) *SSD Increases Minimally*

Figure 2.6 – *SSD When Shifting Region Vertically Away From ROI*

Now consider iterating the shifted region in all directions away from the ROI, shown in Figure 2.7 (b), thus varying both the Δu and Δv coordinates. Figure 2.7 (a) shows the SSD surface produced by iterating the shifted region in all directions. Since the SSD is only significant when iterating the shifted region way from the ROI in the horizontal direction, the SSD surface resembles a canyon shape, which implies the existence of an edge within the ROI.



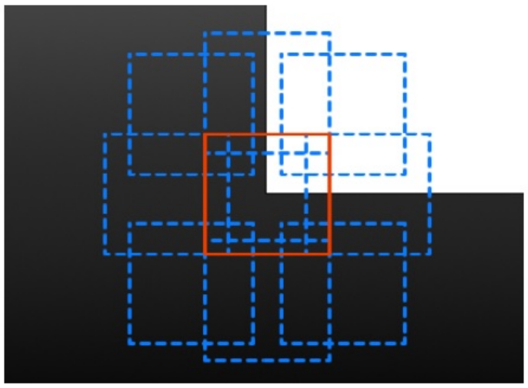
(a) *Shifted Region in All Directions*



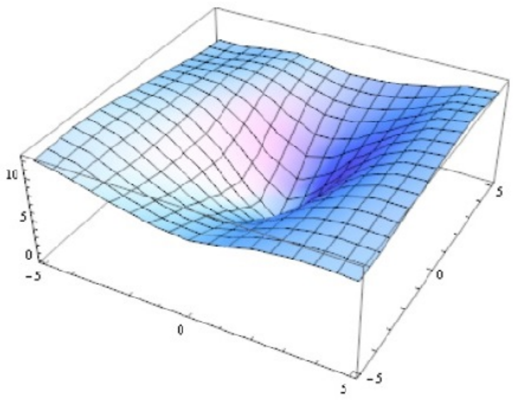
(b) *SSD Increases in Only One Dimension*

Figure 2.7 – *SSD Surface for ROI Containing an Edge*

If a ROI contains a corner, as shown in Figure 2.8 (a), the SSD will increase significantly regardless of the shifted window direction. The SSD for a corner existing within the ROI will have the surface shape shown in Figure 2.8 (b). The concave surface is zero-valued at the origin and increases in all directions away from the origin. A corner can be identified within a ROI based solely on the shape of the SSD surface.



(a) *Shifted Region in All Directions*



(b) *SSD Increases in Only One Dimension*

Figure 2.8 – *SSD Surface for ROI Containing an Edge*

The shape of the SSD surface can be accurately approximated by its behavior at the origin. The Taylor series expansion can be utilized to approximate the surface behavior by expanding the SSD equation near the origin. The Taylor series states that a function's behavior at a specific point can be approximated by the infinite sum of that function's derivatives. Equation 2.8 shows the 1D Taylor series expansion about point a.

$$f(x) = f(a) + \frac{df}{dx}(x-a) + \frac{1}{2!}\frac{d^2y}{dx^2}(x-a)^2 \cdots \tag{2.8}$$

Under the assumption that the shifted window offsets Δu and Δv are minimal, the Taylor series can be utilized to accurately approximate the SSD surface. By utilizing Taylor series expansion, the pixel intensities within the shifted region can be approximated by the ROI gradients, shown in Equation 2.9 I_x is the partial derivative of the ROI in the X (horizontal) direction, and I_y is the partial derivative of the ROI in the Y

$$I(x + \Delta u,\ y + \Delta v) \approx I(x,\ y) + I_x(x,\ y)\Delta u + I_y(x,\ y)\Delta v \tag{2.9}$$

The SSD equation can be reduced to only be dependent on the gradients of the ROI. Equation 2.10 shows the approximated SSD equation by substituting the Taylor series approximation, shown in Equation 2.9, into the SSD Equation 2.8.

$$E(\Delta u,\ \Delta v) \approx \sum_{(x,\ y) \in ROI} I_x(x,\ y)\Delta u + I_y(x,\ y)\Delta v^2 \tag{2.10}$$

The SSD approximation only depends on the ROI gradients I_x and I_y, and not the ROI's pixel intensity values. The SSD approximation can be converted to matrix form by factoring non-summation dependent variables Δu and Δv (derivation shown in the Equation 2.11).

$$\begin{aligned}
E(\Delta u,\ \Delta v) &\approx \sum_{(x,\ y) \in ROI} \Delta u^2 I_x^2(x,\ y) + 2\Delta u \Delta v I_x(x,\ y) I_y(x,\ y) + \Delta v^2 I_y^2(x,\ y) \\
&\approx \sum_{(x,\ y) \in ROI} \left\{ [\Delta u\ \Delta v] \begin{bmatrix} I_x^2(x,\ y) & I_x(x,\ y)I_y(x,\ y) \\ x(x,\ y)I_y(x,\ y) & I_y^2(x,\ y) \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix} \right\} \\
&\approx [\Delta u\ \Delta v] \sum_{(x,\ y) \in ROI} \left\{ \begin{bmatrix} I_x^2(x,\ y) & I_x(x,\ y)I_y(x,\ y) \\ x(x,\ y)I_y(x,\ y) & I_y^2(x,\ y) \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix} \right\} \\
&\approx (\Delta u \quad \Delta v)\, H \begin{pmatrix} \Delta u \\ \Delta v \end{pmatrix}
\end{aligned} \tag{2.11}$$

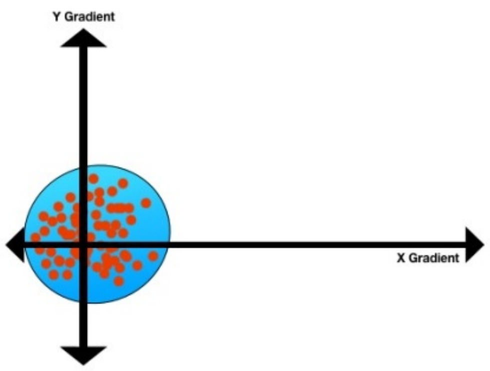
$$H = \sum_{(x,\ y) \in ROI} \begin{pmatrix} I_x^2(x,\ y) & I_x(x,\ y)I_y(x,\ y) \\ x(x,\ y)I_y(x,\ y) & I_y^2(x,\ y) \end{pmatrix} \tag{2.12}$$

The 2×2 matrix H-Harris matrix-is defined in Equation 2.12. The Harris matrix describes the gradient distribution within the ROI, therefore the Harris matrix can be used to classify corner features. The gradient distribution is variant to corner rotation within the ROI, therefore the eigenvalues of the Harris matrix are used to create a rotationally invariant description of the gradient distribution. The eigenvalues of the Harris matrix define the shape of the eclipse which encapsulates the horizontal and vertical gradient distribution of the ROI, shown in Figure 3.11. The eigenvalues of the Harris matrix are invariant to rotation, intensity scaling, and affine transformations, thus the eigenvalues of the Harris matrix are used as the characteristic for detecting corners.

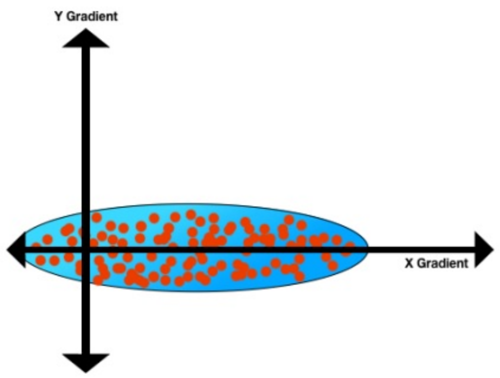
If a corner exists within a ROI, then both eigenvalues of the Harris matrix will have significant magnitude, which implies a large gradient distribution in the horizontal and vertical directions.

Based on the eigenvalues of the Harris matrix, a corner score is assigned to identify the likelihood of a corner existing within the ROI. A corner is considered detected when the corner score is significantly greater than zero, which implies a large encompassing eclipse over the gradient distribution. The corner score equation is described in Equation 2.13.

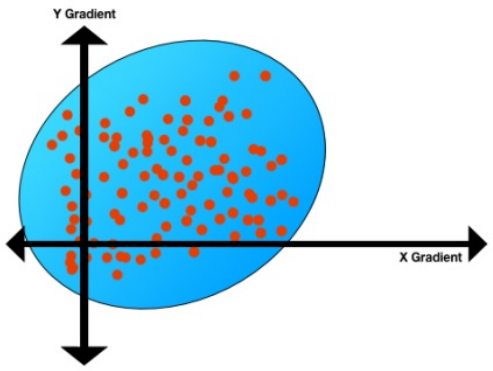
$$R = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2 \tag{2.13}$$



$\lambda_1 \approx 0$ *AND* $\lambda_2 \approx 0 \Rightarrow$ *Edge Exists*



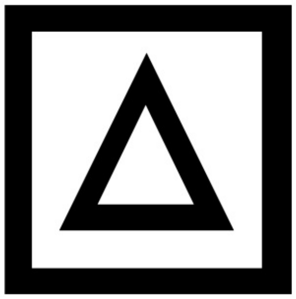
$\lambda 1 \gg 0$ *AND* $\lambda_2 \gg 0 \Rightarrow$ *Corner Exists*



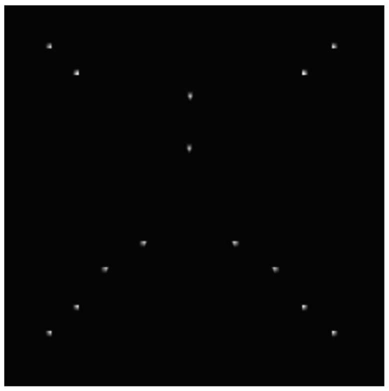
$\lambda 1 \approx 0$ *AND* $\lambda_2 \approx 0 \Rightarrow$ *No Feature*

Figure 2.9 – *Eclipse Representation for Harris Corner Detection*

The k term is considered the sensitivity parameter of the corner detector, which is manually adjusted, however it has been empirically shown that it typically ranges from 0.04-0.06. Figure 2.10 shows the corner response of an example geometric input image. The corner points of the input image in Figure 2.10 (a) produce a significant corner score greater than zero, while the edge and flat regions produce a smaller corner score.



(a) *Input Image*



(b) *Corner Response Threshold > 0*

Figure 2.10 – *Corner Response Example*

The Harris corner detection algorithm can be used to produce a corner response, which is computed by determining the eigenvalues of the Harris matrix at each ROI within the image. The Harris matrix is computed for every pixel within in the image, thus it is very computationally intensive. Once a corner response is created for an image, the corner locations can be extracted as feature locations for higher level computer vision algorithms.

2.5.2 Corner Detection Algorithm

Algorithm 2.1 Corner Detection Algorithm

```
1. Denoise Input Image Using Gaussian Smoothing Filter.
2. Compute Image Gradients $I_x$ and $I_y$
3. Compute Image Gradient Products $I_x^2$, $I_y^2$, and $I_x I_y$
4. For Every Pixel Location
    a. Define ROI Around Pixel
    b. Compute Harris Matrix from $I_x^2$, $I_y^2$, and $I_x I_y$ for ROI
    c. Compute Eigenvalues of Harris Matrix
    d. Assign Corner Score to Pixel
5. Threshold Corner Response
6. Perform Non-maxima Suppression on Corner Response
```

2.6 Conclusion

In this chapter we have given the interested point detector general idea of interested point detector, then, we have presented two detectors which were used in our application, the Harris Corner Detector and the SIFT, also, we have proposed the principal concept along with the algorithm of each one of them.

Last but not least, we have found that Harris detector uses a local autocorrelation analysis method [29] and is reported to be robust against noise, rotation and lighting [30], while, the SIFT approach that provides a robust feature detector that localizes interest points, called SIFT keypoints, which uses this same Harris point response function aiming to select keypoints and extracts keypoint descriptor based on the gradient orientation histogram local to each interest point [31].

Chapter 3

Experiments, Results And Discussion

3.1 Introduction

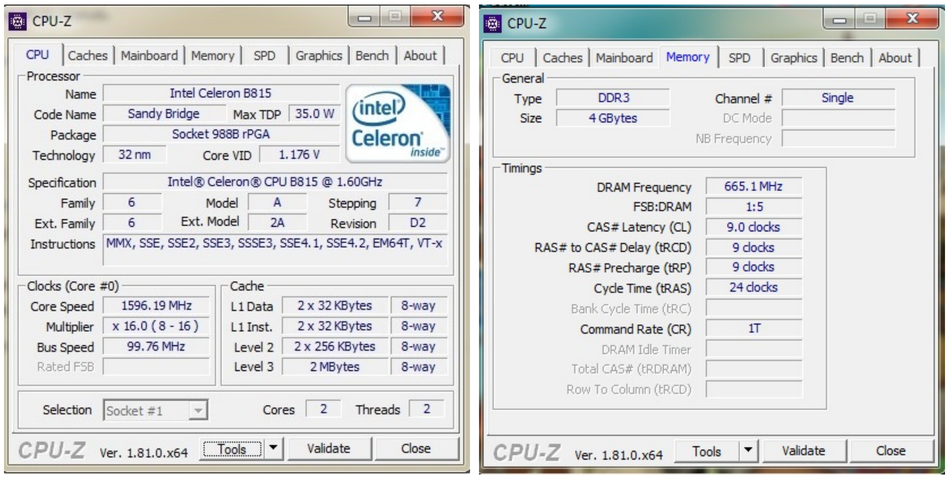
To begin with, in this chapter we are introducing our application made in order to test the proposed techniques, then, we mentioned some code source parts of our application.

Next, we proposed the images that have used to perform the experiments as long as with the tests parameters, the obtained results in both forms, numbers and graphs, and the decompressed images of each technique.

Last but not least, we discuss the results that have been obtained using the proposed techniques.

3.2 The Working Environment

This work has been done on a machine with the following specifications. This machine is laptop running on a Windows 7 Professional 64 bits edition, the CPU is an Intel Celeron B815 clocked at 1.60 *Ghz*, while the RAM is a 4 *GB* DDR3 with a frequency of 665 *Mhz*. The next figures show the machine specs.


CPU-Z
CPU | Caches | Mainboard | Memory | SPD | Graphics | Bench | About
CPU-Z Ver. 1.81.0.x64
Tools
Validate
Close

Processor
Name: Intel Celeron B815
Code Name: Sandy Bridge — Max TDP: 35.0 W
Package: Socket 988B rPGA
Technology: 32 nm — Core VID: 1.176 V
Specification: Intel® Celeron® CPU B815 @ 1.60GHz
Family: 6 — Model: A — Stepping: 7
Ext. Family: 6 — Ext. Model: 2A — Revision: D2
Instructions: MMX, SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, EM64T, VT-x

Clocks (Core #0)
Core Speed: 1596.19 MHz
Multiplier: x 16.0 (8 - 16)
Bus Speed: 99.76 MHz
Rated FSB:

Cache
L1 Data: 2 x 32 KBytes — 8-way
L1 Inst.: 2 x 32 KBytes — 8-way
Level 2: 2 x 256 KBytes — 8-way
Level 3: 2 MBytes — 8-way

Selection: Socket #1 — Cores: 2 — Threads: 2

General
Type: DDR3 — Channel #: Single
Size: 4 GBytes — DC Mode:
NB Frequency:

Timings
DRAM Frequency: 665.1 MHz
FSB:DRAM: 1:5
CAS# Latency (CL): 9.0 clocks
RAS# to CAS# Delay (tRCD): 9 clocks
RAS# Precharge (tRP): 9 clocks
Cycle Time (tRAS): 24 clocks
Bank Cycle Time (tRC):
Command Rate (CR): 1T
DRAM Idle Timer:
Total CAS# (tRDRAM):
Row To Column (tRCD):

(a) *The test machine CPU* (b) *The test machine RAM*

Figure 3.1 – *The test machine specifications*

3.3 The Application

This application has been developed using Matlab, which is one of the very important tools used for data analysis and Programming, because it is a high performance programming language for technical computing, developed by MathWorks in 1984, it integrates computation, visualization, and programming environment. Furthermore, MatLab is a modern programming language environment: it has sophisticated data structures, contains built-in editing and debugging tools, and supports object-oriented programming. These factors make MatLab an excellent tool for teaching and research. The used version of Matlab used to apply and test our proposed methods is Matlab R2010a(7.1.0.604).

3.4 KTH Database Description

KTH Databae had been introduced by Schuldt and al. ([32]) and it is widely used in Literature [33][34][35]. This low resolution database (160 $\times$ 120 pixels gray images), regroups 6 types of action: walking, jogging, running, boxing, hand waving and clapping. These actions had been done multiple times by 25 actors following four scenarios : outdoors scenes (s1), scale changing (s2), cloths changing (s3) and indoors scenes (s4) (Fig.3.7). The background in most scenes is homogeneous ; however, the presence of the fog may make background subtraction relatively difficult. In addition, many other variation factors like zooming for the second scenario, camera angle, lightning conditions and the duration of the actions [36].

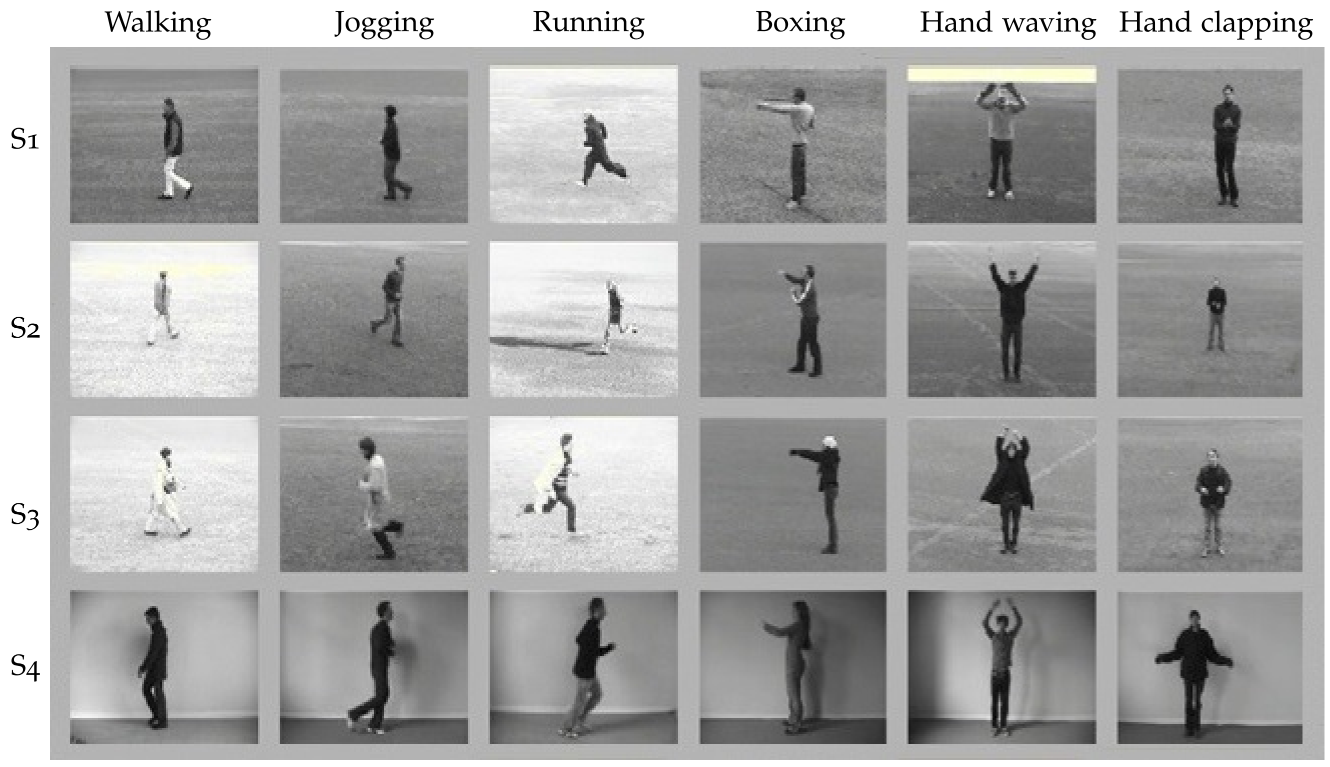


Figure 3.2 – *KTH illustration. A sample for each class of actions (columns) saved with different scenarios (rows).*

3.5 System architecture

Our application is based of two phases, the first one is The Offline Phase, and the second one is The Online Phase.

3.5.1 Offline Phase

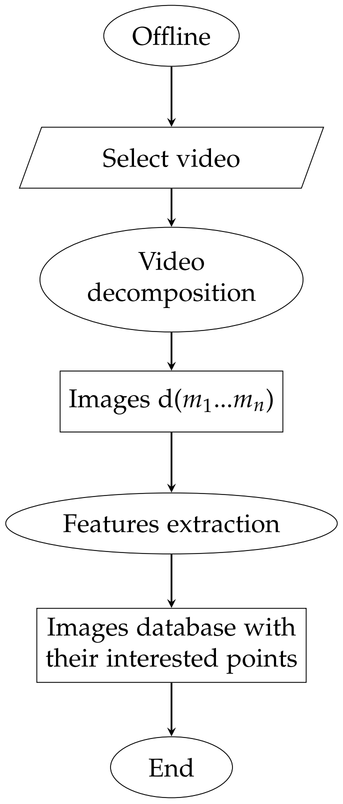


Figure 3.3 – *Offline phase flow chart*

In the offline phase we are treating our videos in the purpose of creating a database which contains interaction point for each frame of our videos using the following steps:

Video selecting: this is the first step of our recognition system. We insert videos in order to be decomposed into frames for detecting the interested point of each one. We insert videos from our database KTH in order to be decomposed into frames for detecting the interested point of each one.

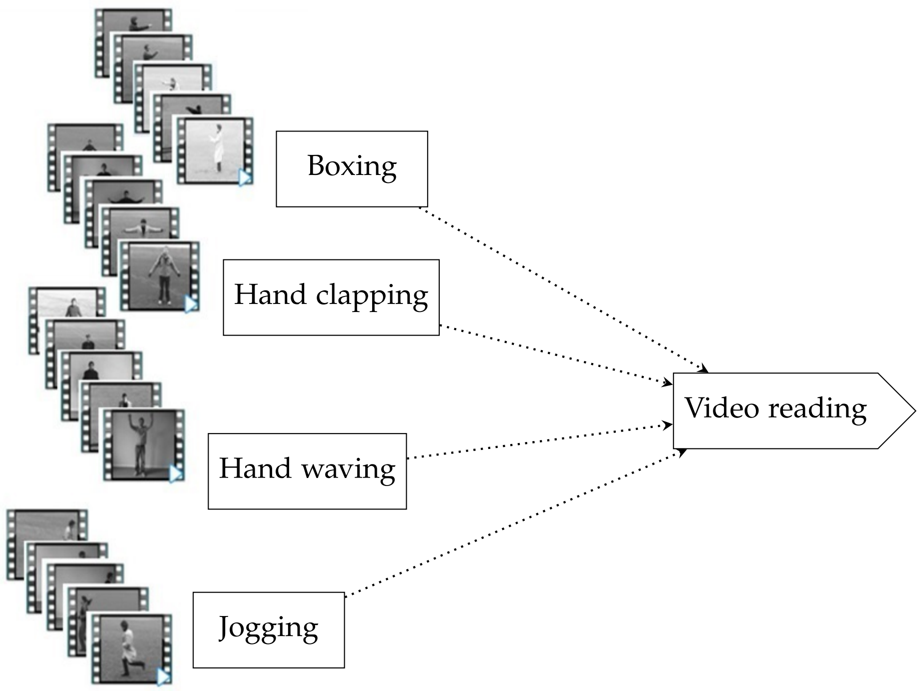


Figure 3.4 – *Video selection*

Video decomposition: we decompose the selected videos into frames with the use of a Matlab code, the next figure explained the process.

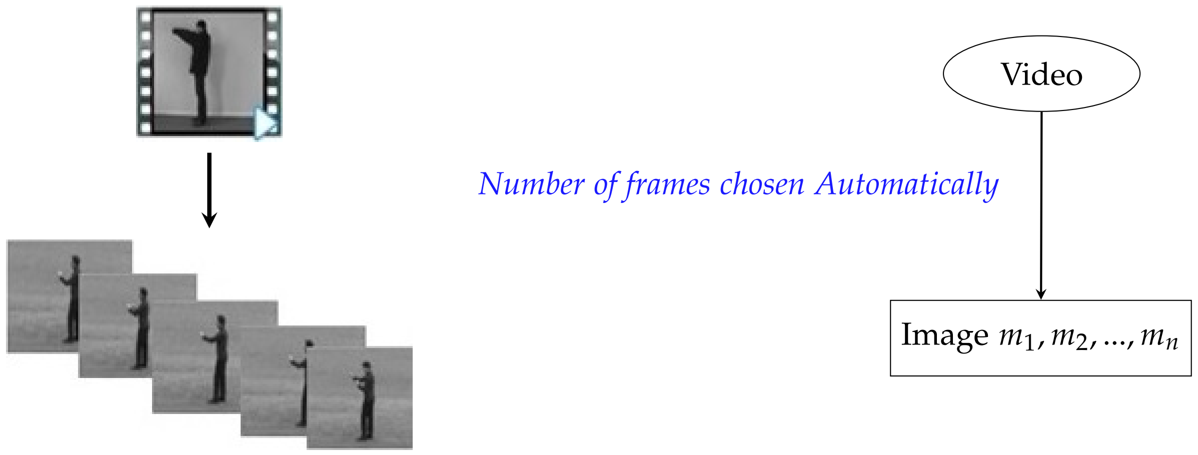

Video
Number of frames chosen Automatically
Image $m_1, m_2, ..., m_n$

Figure 3.5 – *Video decomposition process*

Naming method: each image takes its name as shown in the next figure.

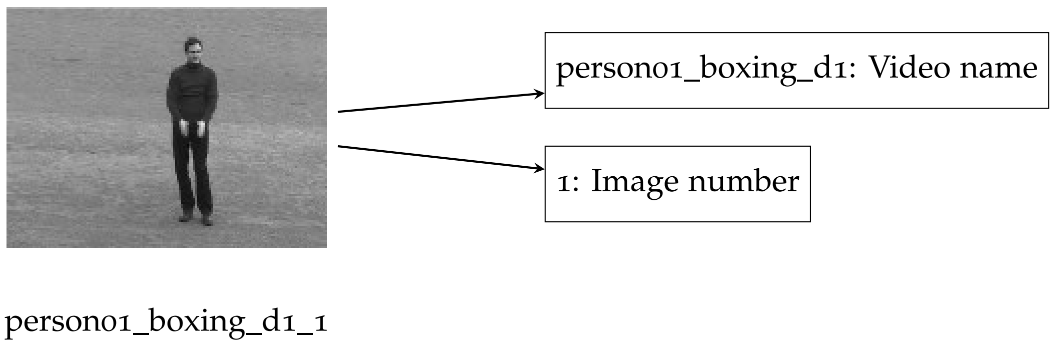


Figure 3.6 – *Naming method*

Feature extraction: we extracted the interesting point from each decomposed frame using Scale Invariant Feature Transform (SIFT).

3.5.1.1 Sift method

SIFT is quite an involved algorithm. It has a lot going on and can become confusing, So. Here's an outline of what happens in SIFT.

Algorithm 3.1 SIFT Algorithm

1. Constructing a scale space This is the initial preparation. You create internal representations of the original image to ensure scale invariance. This is done by generating a "scale space".
2. LoG Approximation The Laplacian of Gaussian is great for finding interesting points (or key points) in an image. But it's computationally expensive. So we cheat and approximate it using the representation created earlier.
3. Finding keypoints With the super fast approximation, we now try to find key points. These are maxima and minima in the Difference of Gaussian image we calculate in step 2
4. Get rid of bad key points Edges and low contrast regions are bad keypoints. Eliminating these makes the algorithm efficient and robust. A technique similar to the Harris Corner Detector is used here.
5. Assigning an orientation to the keypoints An orientation is calculated for each key point. Any further calculations are done relative to this orientation. This effectively cancels out the effect of orientation, making it rotation invariant.
6. Generate SIFT features Finally, with scale and rotation invariance in place, one more representation is generated. This helps uniquely identify features. Lets say you have 50.000 features. With this representation, you can easily identify the feature you're looking for (say, a particular eye, or a sign board).

Then, we save these frames with its interested point in Matlab data base.

3.5.2 Online Phase

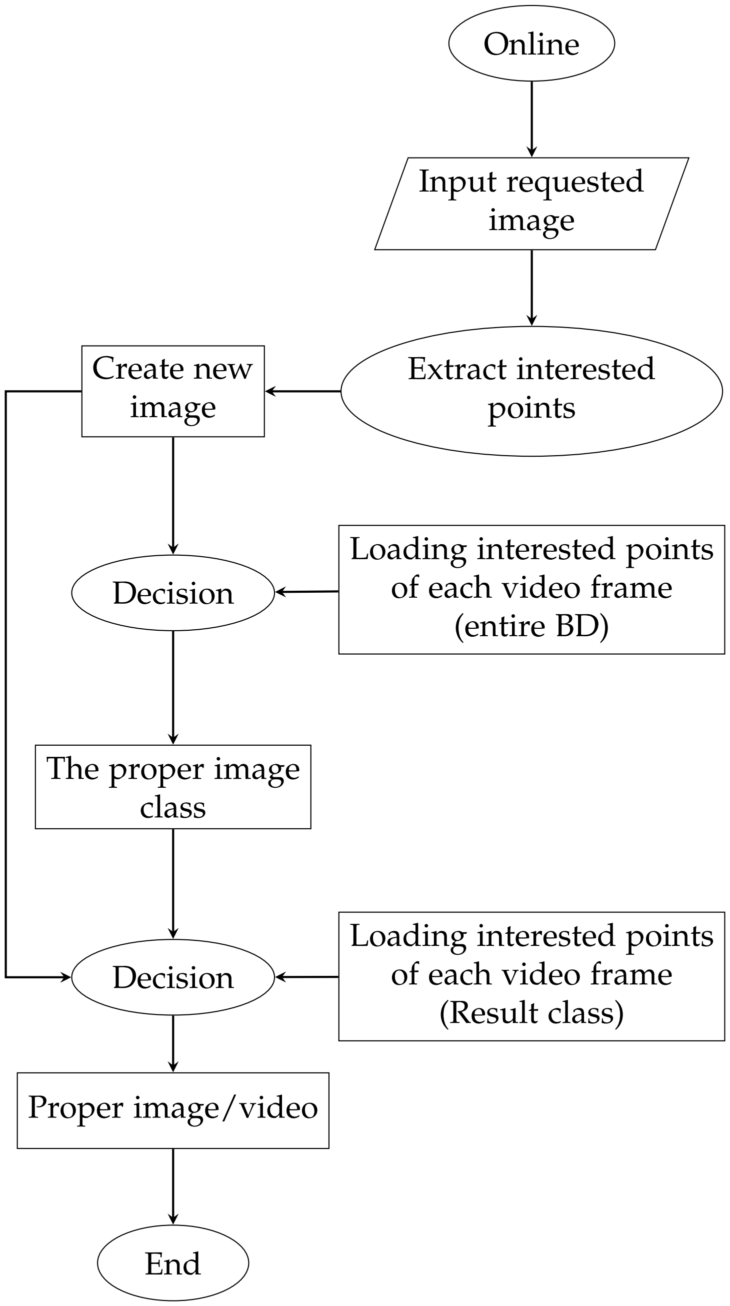

Online
Input requested image
Extract interested points
Create new image
Loading interested points of each video frame (entire BD)
Decision
The proper image class
Loading interested points of each video frame (Result class)
Decision
Proper image/video
End

Figure 3.7 – *Online phase flow chart*

This phase consists of using a procedure in order to obtain the proper image, of our requested image.

Extract interested point first of all, we extract corners from the target image by using the Harris algorithms, then, with the use of the Sift algorithms we calculate the interested point (defined in the first section).

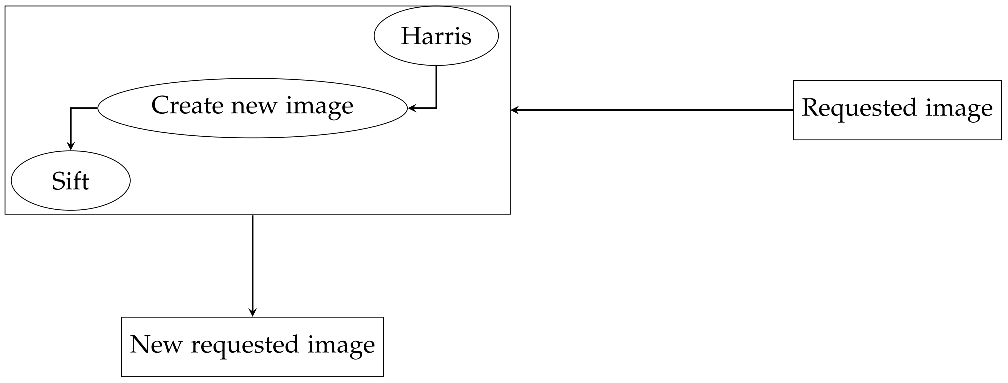


Figure 3.8 – *The process of corner point extraction from requested image*

New image creation: we create a new image which contains the corners of interested points from our requested image. Then, we use the second one in the matching process.

Loading interested point of all video frame (100): At this step our main objective is to load interested point of each database video which we already have extracted during the offline section, taking a frame, then, after 100 frames we take another until we reach the last possible frame.

Decision 01: this step consists of matching between the corner interested point of the new image requite and the corner interested point of video frames, this is done using the sift detector to matching between image and finding the class that had the proper image of the image requite. For the interested point of video frames we reload it from the Matlab database by step 100 (take the first frame, then, after 100 others we take a second one).

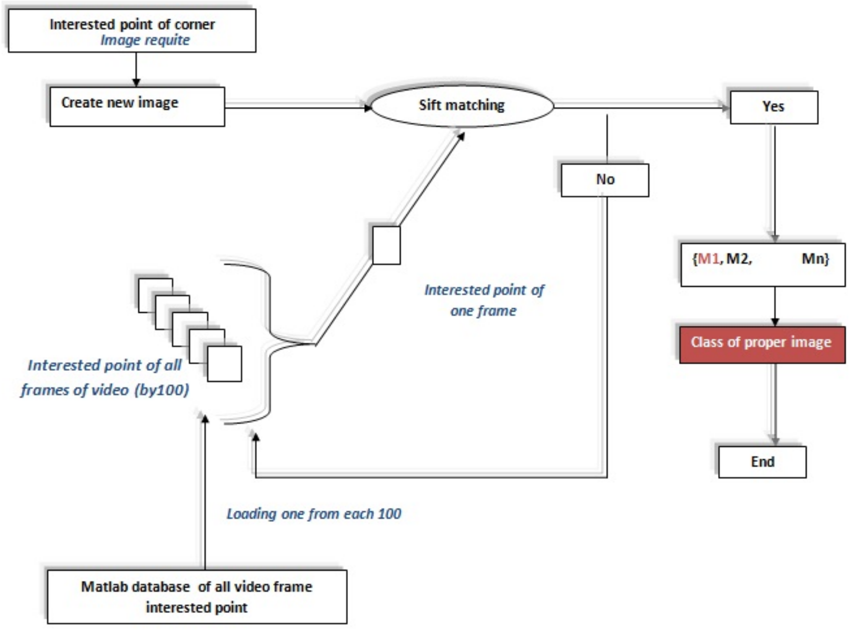

Interested point of corner
Image requite
Create new image
Sift matching
Yes
No
{M1, M2, Mn}
Class of proper image
End
Interested point of one frame
Interested point of all frames of video (by100)
Loading one from each 100
Matlab database of all video frame interested point

Figure 3.9 – *Finding the proper image class process*

Decision02: our objective in this step is to match between the corner interested point of the new requested image and the interested point of all video frames of the result class (decison01).

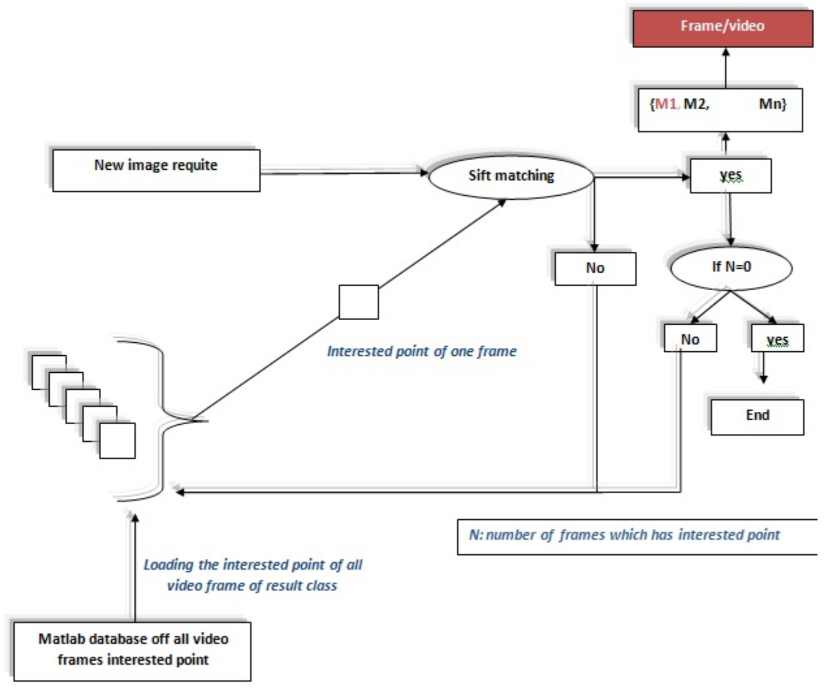

Frame/video
{M1, M2, Mn}
New image require
Sift matching
yes
No
If N=0
No
yes
End
Interested point of one frame
N: number of frames which has interested point
Loading the interested point of all video frame of result class
Matlab database off all video frames interested point

Figure 3.10 – *Finding the video and its proper image process*

3.5.2.1 Harris detector method

Here we study the Harris Corner algorithm. this latter allows the detection of the crossing line, and it was performed by Harris and Stephens. We take an image $I(x, y)$ We start by taken the derivative in line and in column of image:

$$I_x(x, y) = \partial I \partial x(x, y) \tag{3.1}$$
$$I_y(x, y) = \partial I \partial y(x, y) \tag{3.2}$$

We need those tree next matrix

$$A(x, y) = I_x^2(x, y), \tag{3.3}$$
$$B(x, y) = I_y^2(x, y), \tag{3.4}$$
$$C(x, y) = I_x(x, y) \cdot I_y(x, y), \tag{3.5}$$

They must be smooth by a Gaussian filter. You can choose the filter you want.

$$k(x, y) = k(x, y) * H_{gauss} \tag{3.6}$$

With $k = A,\ B\ or\ C$ and H_{gauss} is the Gaussian filter. Warning this is a convolution. Now, we can compute the corner strength for each pixel.

$$Q(x, y) = (AB - C)^2 - \alpha \cdot (A + B)^2 \tag{3.7}$$

Where α is the sensibility of the detector, and Q is called Corner Response Function (CRF) and return strong values when a corner is detected. Furthermore, α is high more the sensibility is small and less the CRF is high. Typically α ranged from 0.04 to 0.06 but never higher than 0.25. We make a thersholding to remove the weakest value of CRF.

$$Q(x,\ y) < th = 0 \tag{3.8}$$

Generally this threshold (th) is between 1.04 and 1.07. We can then store the corners detected in decreasing order and proceed with a detection of local maxima to remove the corners too close. Local maxima must be done from the corner of higher value and eliminate lower values.

3.6 Experimentation and Results

In this section, we are illustrating the effectiveness of the proposed approach in image finding, this is done by presenting the obtained results of the performed experimentation.

3.6.1 Experimentation and Results(SIFT)

First of all, we use the SIFT method to categorize the video (between class), then, we used to find the similar image in the same class(intra class).

3.6.2 Between classes

In this section we try to defined the class that contain our image requite and the result is in the confusion matrix, we have like parameter:

The step: take one frame then after 100 take second one

The threshold: it's a maximum of distance between similar points (sift parameter)

Table 3.1 – *The first performed experiment between classes(sift)*

	Boxing	Handclapping	Handwaving	Joping	Running	Walking
Boxing	97.47%	0.80%	0.50%	0.20%	0.20%	0.80%
Handclapping	1.12%	89.28%	9.28%	0.10%	0.20%	0%
Handwaving	0.8%	17.5%	81.4%	0.2%	0%	1%
Joping	8.3%	3.3%	4%	56.6%	14.4%	13.5%
Running	7.3%	3.2%	4.2%	28.7%	46%	10.7%
Walking	7.7%	2%	2.6%	17.6%	9.4%	60.8%

The step: 100, the threshold: 20000

Table 3.2 – *The second performed experiment between classes(sift)*

	Boxing	Handclapping	Handwaving	Joping	Running	Walking
Boxing	94.12%	1.70%	1.60%	0.8%	0.7%	1%
Handclapping	3.12%	82.75%	10.56%	1.8%	0.80%	0.89%
Handwaving	7.5%	4.4%	4.8%	30.1%	49.2%	4%
Joping	11%	2%	3%	17.9%	9.8%	56.3%
Running	10.7%	4%	4.5%	29.3%	44.2%	7.3%
Walking	14%	4.5%	3.5%	17.5%	9.5%	50.9%

The step: 100, the threshold: 10000

Table 3.3 – *The third performed experiment between classes (sift)*

	Boxing	Handclapping	Handwaving	Joping	Running	Walking
Boxing	95.95%	0.80%	0.80%	0.90%	0.70%	0.80%
Handclapping	0.81%	90.40%	8.46%	0.10%	0.30%	0%
Handwaving	8.8%	3.4%	5%	28.1%	46%	8.8%
Joping	9.5%	2.3%	3%	19%	9.1%	56.7%
Running	8.8%	3.4%	5%	28%	46.1%	8.7%
Walking	9.5%	2.3%	3.1%	19%	9.2%	56%

The step: 100, the threshold: 15000.

3.6.3 Discussion

We have found good results for the classes boxing and handclapping, but starting from the handwaving class the rate of error increases, and the application has a difficulty in finding the exact class of the reference image. This is because of 3 reasons:

The first reason is: our decision the result of this application gives us a set of images. In this case we always chose to take the first image of this set as reference image. For that reason, the boxing class can be found easily because the first image must necessarily be in the boxing class. If we treat a boxing picture or in boxing or handclapping class if we treat a picture of handclapping. However, concerning the handwaving, jogging, running, walking classes, the first similar image can be in the boxing or handclapping class and here the error increases.

The second reason is: in the jogging, running and walking classes there is the empty background image which is similar for all the class even if we find an image which has a part of an object but the background takes the majority of the points of interest. Consequently, the score increases, and the error also increases.

The third reason: the great similarity between the class and the poor solution of the videos adds an important role. For the change of threshold, it has been found that it has a remarkable influence on the results. They change each time one has to change the threshold.Therefore, work with the correct parameter change too much.

3.6.4 Intra-class

In this experiment, our objective is to find the requested image in the used videos, then, we determine its class. Next, we try to find it using the SIFT algorithm. The next table summarizes the percentage of finding the correct image. The threshold: 20000.

Table 3.4 – *The 4th performed experiments intra-class (sift)*

Boxing	99.69%
Handclapping	99.89%
Handwaving	99.7%
Jogging	99.9%
Running	97.8%
Walking	98.7%

3.6.5 Discussion

Good results have been found for all classes, and that show the performance of sift operator for finding the correct proper image for the same classes. Some errors have occurred because of the similarity of sum of images in different videos.

3.7 Experimentation and Results(siftharris)

3.7.1 Between class

Table 3.5 – *The first performed experiment between classes (siftharris)*

	Boxing	Handclapping	Handwaving	Joping	Running	Walking
Boxing	97.77%	0.5%	0.90%	0.30%	0.10%	0.40%
Handclapping	1.73%	89.18%	9.08%	0%	0%	0%
Handwaving	2.3%	24%	73.3%	0.2%	0%	0.3%
Jogging	37.8%	7%	5.6%	31.7%	6.3%	11.7%
Running	36.6%	8%	5.5%	21.3%	20.6%	8%
Walking	38.2%	4.6%	5.8%	12.7%	7.1%	31.6%

the step: 100, the threshold: 20000.

3.7.2 Discussion

We have managed to achieve good results in boxing and hand clapping classes, for the other classes our application has to minimize the error rate compared to the results of sift, but the error rate remains high and that because of the same reasons cited for the sift.

The first reason: is the choice of the first image of sets of resulting images, the first resulting image can be in the boxing class but the similar image can be 2, 3 or 5 image sets.
For example in the jogging class 37.8% of similar image were found in the boxing class.

The second reason: the background as we previously mentioned the jogging, running, and walking classes on similar empty background is why we can find an image similar to our image but in another class.

The third reason: the great similarity of the class plays an important role especially the last 3 class jogging, running and walking.

3.7.3 Intra class

Table 3.6 – *The second performed experiment intra-class (siftharris)*

Boxing	97.97%
Handclapping	99.18%
Handwaving	98.4%
Jogging	98.9%
Running	98.3%
Walking	97.6%

The threshold: 20000.

3.7.4 Discussion

As can be seen in the previous table good results have been obtained for all classes, which shows the performance of our hybrid method. There is of course a margin error due to the big similarity between some images from different videos.

3.8 Comparative study

It has been found that the results obtained from class categorization (between classes) of our application is reliable compared with the sift results despite the same difficulties, even another benefit has been found. Our hybrid application minimizes search time and that's a big improvement. We did obtain good results of intra-class for the both sift and siftharris, and there is not a big differences in the performances of each one but the change was at the level of research time really diminishing.

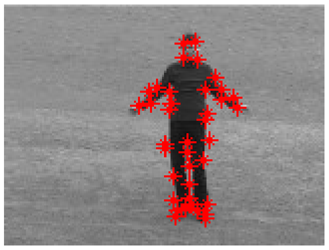
Table 3.7 – *Times variances between sift and siftharris (intra-class)*

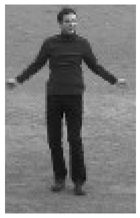
	Ratio	Time to find an image (sec)	Ratio	Time to find an image (sec)
Boxing	99,69%	41.426	97.97%	18.870
Handclapping	99.89%	32.179	99.18%	14.020
Handwaving	99.70%	50.194	98.40%	22.167
Jogging	99.90%	49.069	98.9%	30.831
Running	97.80%	65.094	98.30%	32.021
Walking	98.70%	65.568	97.60%	32.321

3.9 Hybrid method siftharris example

The following figures show some examples of figure using the proposed hybrid method siftharris.

(a) *Requested figure*


(b) *Figure after corner extraction*


(c) *Figure after creating new figure*

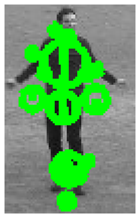

(d) *Figure after sift interested point calculation*

Figure 3.11 – *Example of the proposed hybrid method siftharris*

General Conclusion

In this thesis, we studied two tasks. The first task was testing and developing interest point detectors. This task was to evaluate the sift detectors produced by the original authors and to develop new software to implement combination between the Harris and the sift The validation methodology devised tested the detectors with images containing similar shapes. The results show that every detector can detect corners correctly in a rectangular shape and extract a variety of different points in real trademark images and find the correct image in our huge data base.

The second task was to measure the performance of the interest point detectors. The combination between the Harris detector and the sift has the best ability in matching between images and achieved good results with more than 90% correct result. To sum up, we found that the combination between the Harris and the sift interest point detectors give a good result then we use each one alone .

The performance of this method was presented firstly in the time of finding the proper image and,secondly in the percentage of finding the correct one.

As future work, and since we did not obtain good results, we are willing:

- making more hybridizing between the proposed techniques,
- performing more tests using other techniques related to interested point detection ,and to choose better algorithms than the used one, like Boundary based methods that extract shape boundaries within the input image and calculate interest points from image contours.
- Use the same method but looking only for paty from the obeject .

Bibliography

[1] Oge Marques. *Practical image and video processing using MATLAB*. John Wiley & Sons, 2011.

[2] Linda G Shapiro and George C Stockman. Computer vision, 2001, 279-325.

[3] Ian T Young, Jan J Gerbrands, and Lucas J Van Vliet. *Fundamentals of image processing*. Delft University of Technology Delft, 1998.

[4] Barry R Masters, Rafael C Gonzalez, and Richard Woods. Digital image processing. *Journal of biomedical optics*, 14(2):029901, 2009.

[5] Mohammed Omari, Nasreddine Karour, and Souleymane Ouled Jaafri. *Fractal Image Compression Based on Polynomial Interpolation*. Master memoir. Networks and intelligent systems, University of Adrar, Algeria, 2015.

[6] Tinku Acharya and Ajoy K Ray. *Image processing: principles and applications*. John Wiley & Sons, 2005.

[7] Roy Hoffman. *Data Compression In Digital Systems*. NewYork: Chapman & Hall, 1979.

[8] Barry G Haskell, Atul Puri, and Arun N Netravali. *Digital Video: An Introduction To MPEG-2*. NewYork: Chapman & Hall, 1997.

[9] Azriel Rosenfeld. Picture processing by computer. *ACM Computing Surveys (CSUR)*, 1(3):147–176, 1969.

[10] David A Cook. *A History of Narrative Film 2nd Edition*. WW Norton & Company, 1990.

[11] Iain E Richardson. *H. 264 and MPEG-4 video compression: video coding for next-generation multimedia*. John Wiley & Sons, 2004.

[12] Abdelrahman Abdelazim. *Fast Motion Estimation Algorithms for Block-Based Video Coding Encoders*. PhD thesis, University of Central Lancashire, 2011.

[13] Konapala Premshankar. *Efficient Video Compression Schemes by applying the DCT approach*. PhD thesis, 2014.

[14] Rhonda Callow. Video formats explained - understanding the different video formats. *http://www.brighthub.com/multimedia/video/articles/4761.aspx*, 2010. Online, visited on july 25th, 2017.

[15] Matt Buchanan. Giz explains: Every video format you need to know. *https://gizmodo.com/5093670/giz-explains-every-video-format-you-need-to-know*, 2008. Online, visited on october 5th, 2017.

[16] Stephen Se, David Lowe, and Jim Little. Vision-based mobile robot localization and mapping using scale-invariant features. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 2, pages 2051–2058. IEEE, 2001.

[17] Iryna Gordon and David G Lowe. What and where: 3d object recognition with accurate pose. *Toward category-level object recognition*, 4170:67–82, 2006.

[18] M Brown and DG Lowe. Iccv'03: Proceedings of the ninth ieee international conference on computer vision. *Washington, DC: IEEE Computer Society*, pages 1218–1225, 2003.

[19] V Rodehorst and A Koschan. Comparison and evaluation of feature point detectors. In *5th International Symposium Turkish-German Joint Geodetic Days*, 2006.

[20] Krystian Mikolajczyk, Tinne Tuytelaars, Cordelia Schmid, Andrew Zisserman, Jiri Matas, Frederik Schaffalitzky, Timor Kadir, and Luc Van Gool. A comparison of affine region detectors. *International journal of computer vision*, 65(1-2):43–72, 2005.

[21] H Bendea, Alberto Cina, Mattia De Agostino, A Lingua, and Marco Piras. Realizzazione di un gis stradale con veicolo rilevatore basso costo. In *Proceedings of XII Conferenza Nazionale ASITA*, 2008.

[22] D Poli and F Steidler. Reality-based 3d city models from aerial and satellite data. In *Urban Data Management: Urban Data Management Society Symposium 2007, Stuttgart, Germany, 10-12 October 2007*, page 47. Routledge, 2007.

[23] H Bendea, Filiberto Chiabrando, F Giulio Tonolo, and Davide Marenchino. Mapping of archaeological areas using a low-cost uav. the augusta bagiennorum test site. In *Proceedings of the XXI International CIPA Symposium*, pages 1–6, 2007.

[24] Wolfgang Förstner. A feature based correspondence algorithm for image matching. *International Archives of Photogrammetry and Remote Sensing*, 26(3):150–166, 1986.

[25] C Harris. Ms (1988). a combined corner and edge detector. In *Proceedings of the Alvey Vision Conference 1988*, pages 147–151, 1988.

[26] Andrea Lingua, Davide Marenchino, and Francesco Nex. Performance analysis of the sift operator for automatic feature extraction and matching in photogrammetric applications. *Sensors*, 9(5):3745–3766, 2009.

[27] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2009.

[28] Tony Lindeberg and Lars Bretzner. Real-time scale selection in hybrid multi-scale representations. In *scale-space*, volume 3, pages 148–163. Springer, 2003.

[29] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Manchester, UK, 1988.

[30] Cordelia Schmid, Roger Mohr, and Christian Bauckhage. Evaluation of interest point detectors. *International Journal of computer vision*, 37(2):151–172, 2000.

[31] Jose A Rodriguez and Florent Perronnin. Local gradient histogram features for word spotting in unconstrained handwritten documents. *Proc. 1st ICFHR*, pages 7–12, 2008.

[32] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: a local svm approach. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, pages 32–36. IEEE, 2004.

[33] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[34] Caroline Gilbert, Sophie De Winne, and Luc Sels. The influence of line managers and hr department on employees' affective commitment. *The International Journal of Human Resource Management*, 22(8):1618–1637, 2011.

[35] Piotr Bilinski and Francois Bremond. Contextual statistics of space-time ordered features for human action recognition. In *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on*, pages 228–233. IEEE, 2012.

[36] Mouna Selmi. *Reconnaissance d'activités humaines à partir de séquences vidéo*. PhD thesis, Institut National des Télécommunications, 2014.