Escola Politècnica Superior de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

MASTER THESIS

TITLE: Video Quality evaluation in IP videoconference between fixed and mobile devices

MASTER DEGREE: Master in Science in Telecommunication Engineering & Management

AUTHOR: Josep Antoni Canadell Pulido

DIRECTOR: Jesus Alcober Segura

SUPERVISOR: Rubén González Blanco

DATE: November, 6th 2008

Overview

The integration of the mobile network with the fixed network, by using IP, has allowed the appearance of packet-based videoconference between fixed and mobile devices. Current mobile IP networks are sized for http traffic and its behavior is best-effort. In this context, it becomes compulsory to evaluate and measure the video quality when packets are transmitted through these networks. By doing this, results obtained could be used to size future networks and/or define different Quality of Service politics in future IMS networks.

The objective of this final career project is to evaluate possible video quality degradation in videoconference applications through IP integrated network (fixed + mobile). A study of the currently available video quality evaluation tools will be made and, as well, a specific system for the video quality evaluation in communications between the IP fixed network and the 3.5G IP mobile network, using own methodology, will be developed.

Several evaluation scenarios will be defined, with different mobile – fixed network configurations in order to identify the elements susceptible of degradate the quality of the video.

The developed system must be able to process the video sequences in origin and in reception, so it will be a referential system.

Resumen

La integración vía IP de la red móvil con la red fija ha hecho realidad la videoconferencia por paquetes entre dispositivos móviles y fijos. Las redes IP móviles actuales están dimensionadas para tráfico http y su comportamiento es best effort. En este contexto se hace necesario la evaluación y medida de la calidad de video, cuando se transmite por paquetes a través de estas redes, a fin de disponer de referencias que puedan ser utilizadas para dimensionar las redes a futuro y/o definir diferentes políticas de QoS en las futuras redes IMS.

El objetivo de este proyecto final de carrera es evaluar una posible degradación de la calidad de vídeo en aplicaciones de videoconferencia a través de la red IP integrada (móvil + fija). Se realizará un estudio de las herramientas para la medida de la calidad de video disponibles actualmente en la industria para detectar dichas degradaciones y se desarrollará un sistema especifico, con metodología propia, para la evaluación de la calidad de video en comunicaciones a través de la red IP móvil 3.5G y la red IP fija.

Se definirán diferentes escenarios de evaluación, con diferentes configuraciones de red móvil y fija, a fin de identificar los elementos que son susceptibles de degradar la calidad de video en las transmisiones IP a través de dichas redes.

El sistema desarrollado para le medida de la calidad de video debe de ser capaz de capturar las secuencias transmitidas en el origen y recibidas en el otro extremo de la comunicación, por lo que se tratará de un sistema referencial.

This Master Thesis is dedicated
to all those who helped me
making it possible and
to all those who encouraged
me to finish it

CONTENTS

FIGURES INDEX

TABLES INDEX

GLOSSARY

3GPP
: Third Generation Partnership Project

AVC
: Advance Video Coding = MPEG4 Part10 = H.264

CIF
: Common Intermediate Format screen size

DSCQS
: Double Stimulus Continuous Quality Scale

DSIS
: Double Stimulus Impairment Scale

DVQ
: Digital Video Quality

ENG
: Electronic News Gathering

HSPA
: High Speed Packet Access

HSDPA
: High Speed Download Packet Acces

HSUPA
: High Speed Upload Packet Acces

IETF
: Internet Engineering Task Force

ITU-T
: Internacional Telecomunicationión Union – Telecommunications Statandards

MNG
: Mobile News Gathering = WENG.

MPEG-4 Part 10/AVC
: Moving Pictures Expert Group Part 10 / Advanced Video Codec

MPQM
: Moving Pictures Quality Metric

OVR
: Original Valid Region

PDM
: Perceptual Distortion Metric

PQR
Picture Quality Rating

PSNR
Peak Signal to Noise Ratio

PVQM
Perceptual Video Quality Metric

QCIF
Quarter CIF screen size

RMSE
Root Mean Square Error

RTP
Real Time Protocol

SIP
Session Initiation Protocol

SDTV
Standard Definition TV

SMPTE
Society of Motion Picture and Television Engineers

SSCQE
Single Stimulus Continuous Quality Scale

TME
Telefónica Móviles de España

UMTS
Universal Mobile Telecom System

VQM
Video Quality Metric

WENG
Wireless Electronic News Gathering = Mobile News Gathering

INTRODUCTION

The development and the growth of multimedia applications in a distributed environment have attracted the interest of the general audience in relation to the offered services. Digital video systems are replacing most of those analogical video sources enabling the appearance of new telecommunication services (high definition TV, high definition videoconferencing, telemedicine, e-commerce, Internet video) that are becoming essential part of the world-wide economy.
In this scenario, governments and standardization industries have established objective metrics for the evaluation of the video quality in order to:

- Measure the performance of the system.
- Determine the minimum requirements to guarantee a good performance.
- Compare the different offered services.
- Optimize the use of the limited network resources such as the transmitting band.

With the arrival of the digital video and the introduction of compression mechanisms, storage and transmitting systems, traditionally-used video-quality-measuring techniques show limitations. Parameters such as the differential gain, waveform distortion, etc, have been used for long and, in spite of not being invalidated, they are not precise enough to evaluate the digital video quality, which is more complex.

Multimedia applications in distributed environments are subjected to strong limitations due to the best-effort profile of the IP protocol-based networks. In fact, these applications are extremely sensible to end-to-end delay and to the variation of this delay (jitter) which can cause undesirable effects on the quality of the service. One video transmission system can have a good performance when used in a videoconference but it could be unsuitable for transmitting high quality entertainment television. In the other hand, many parameters which affect digital video quality and are concerned to the communication network (bit-rate, error rate, dropped packet rate, jitter) are time-variable and so, can cause fluctuations of the quality produced.

This final career project is comprised in a major one (WENG) and has as main objective to evaluate the quality of the video transmitted in HSDPA mobile networks. HSPA (3.5G – 3.75G) stands for High Speed Packet Access and it is considered as the evolution of the 3G mobile networks, offering higher bandwidths and lower latencies and jitter. There are two stages of the HSPA networks: HSDPA (High Speed Downlink Packet Access or 3.5G) which improves the downlink speed up to 7.2Mbps and the HSUPA (High Speed Uplink Packet Access or 3.75G) which improves the uplink up to 2Mbps as well as the downlink up to 7.2Mbps.
By the time of making this project the only HSPA hardware available implemented only the HSDPA specification so all the tests performed are referred to this technology.

This final career project is divided in eight main chapters, each of them related to a specific aspect of the study.
In the first chapter, an introduction to the project specifications will be done. This will help in understanding the decisions taken during the making of this project as well as the features the project had to include.

In the second chapter, a study of the state of the art concerning video quality estimation methods is done. Here, several objective and subjective video quality evaluation techniques will be introduced in a theorical way so it will be easier to understand the way experts can analyze video degradation.

In the third chapter, the main features of the video codecs which will be used in this project will be introduced. These codecs will be Standard H.263 and H.264 and main differences and similarities will be shown. Most of the video transmitting applications, where bandwidth is a variable to minimize, take profit from these codec's main features.

In the fourth chapter, several objective video quality evaluation tools will be introduced, all of them aiming to obtain results concerning video quality degradation in the easiest possible way. However, a problem found in all of them will lead to the development of an own video quality evaluation tool.

This problem, as it will be explained in the fifth chapter, is the lack of synchronization between original and impaired sequences when packet loss happens, fact likely to happen when transmitting in an aggressive behavior network such as mobile network. The evaluation tool developed will avoid this problem making the results obtained be correct.

In order to evaluate video quality degradation, several scenarios will be designed, implemented and tested. These scenarios will be introduced in the sixth chapter. Besides, it will be introduced the methodology used for video quality evaluation.

The results obtained will be shown and commented in the seventh chapter and conclusions as well as future work will be introduced in the eighth and final chapter.

Finally, it must be highlighted that the making of this project will not have any kind of environmental repercussion as it will be developed entirely with personal computers in a controlled ambient such as a networking devices room.

CHAPTER 1. PROJECT ESPECIFICATIONS

WENG has been a project carried out by Telefonica I+D whose main objective has been to check whether an MNG (Mobile News Gathering) over mobile network solution can report similar results than a SNG (Satellite News Gathering) solution.
Traditional SNG requires a great infrastructure to be carried out:

- A modified (usually rented) truck with a satellite or radio frequency antenna to transmit multimedia signal to the central studio.
- A license for transmitting the signal in a determined time slot.
- A great number of human resources to make everything work properly.

All these facts make SNG an expensive, non-agile and human resource demanding solution to transmit news to the central studio. However, as mobile networks increase their bandwidth and cover, would it be possible to transmit high definition video and audio over mobile networks? This is the kind of question WENG project has looked for an answer.

In case it is possible to achieve similar video quality with the use of the mobile network several advantages rise up:

- Quickness in mobile unit establishment and transmission.
- Possibility to relate news as soon and as near as they are happening.
- Reducing costs as well as reducing human resources.
- Allowing spectators becoming reporters as they could transmit news through their mobile phones.

However, mobile networks currently suffer from several drawbacks which make them little attractive:

- Cover reduces as bandwidth increases.
- High speed connections mean, currently, high phone bills.

In spite of the drawbacks pointed before, a MNG solution could be very profitable for local television stations as they could challenge national television stations in big cities. Besides, carriers could offer reduced phone fares in case MNG solutions are contracted.

After making a state of the art study concerning new mobile technologies and possible MNG solution requirements, some premises were established. These premises were used as the guidelines for the project and were the initial point for the making of this final career project.

The initial requirements can be classified in two groups: functional and feature requirements. Only the requirements related to video quality or those which established the solution architecture will be explained.

Table 1.1. Functional Requirements

Functional Requirements	
Requirement	Description
Transmission through UMTS network	The solution must allow transmitting a digital audiovisual signal, on real-time, through a data connection established using the UMTS interface. This interface will be placed in a mobile unit equipped with a professional camera and HSPA transmission modules.
Variable video resolution depending on the bandwidth available	The solution must allow transmitting images in real-time at different resolutions and frames per second depending on the bandwidth available. By doing this, the received video quality will be enough to be transmitted through television.
Multimedia sessions based on SIP and RTP	The system must allow establishing multimedia sessions based on SIP (Session Initiation Protocol) and RTP (Real-Time Transport Protocol) between the mobile unit and the TV Studio.
Audio and video	The solution must allow transmitting audio and video from the mobile unit to the studio, and the returning audio to the reporter.

Table 1.2. Feature Requirements

Feature Requirements	
Requirement	Description
Maximum video resolution	The maximum video resolution delivered by the system must be “Standard Definition” (SD) - D1 (PAL: 720x576 pixels, NTSC: 720x480 pixels) - 4CIF (PAL: 704x576 pixels, NTSC: 704x480 pixels)
Minimum video resolution	The minimum video resolution must be QCIF (PAL: 176x144 pixels, NTSC: 176x120)
Maximum framerate	The maximum number of frames per second will depend on the video format used: PAL: 25 frames/sec, NTSC: 30 frames/sec.
Required end-to-end bandwidth for a SDTV connection.	The end-to-end transmission will require a 1.5-2 Mbps bandwidth, taking care of the audio / video codecs used and the radio access network capacity (in uplink).
Maximum delay	The maximum delay between the content capture and its display in the studio must be lower than 2 seconds (in order to guarantee a behavior equivalent to the Satellite News Gathering one).
Transmission latency	The average end-to-end transmission latency value must be lower than 150ms (according to eh G.114 ITU-T recommendation). The maximum value must be lower than 400ms.
End-to-end jitter	The average end-to-end RTP data flow jitter value must be lower than 30ms according to the G.114 ITU-T recommendation.
Packet loss ratio	The average end-to-end packet loss ratio value must be lower than 1%.

CHAPTER 2. VIDEO QUALITY EVALUATION (STATE OF THE ART)

2.1. INTRODUCTION

Digital video broadcasted through communication networks are subjected to several kind of distortion during acquisition, compression, transmission, decoding and reproduction. Such phases provoke a degradation of the video quality but a bigger efficiency and effectiveness of the multimedia service. As an example, compression is used to reduce the needed bandwidth to transmit video data but it provokes loses due to sampling and quantization. What remains fundamental in multimedia systems is the estimation of the video quality degradation in order to maintain, to control and to possibly improve such quality. To such purpose it turns out crucial to use video quality evaluation metrics.
In this chapter there will be introduced and analyzed some of the commonly proposed video quality evaluation metrics.

2.2. VIDEO QUALITY ESTIMATION METHODS

Based on the research done in the evaluation of the end-user perceived video quality it is possible to make some classifications using several criteria. According to the presence of a human judgment, these metrics are obvious:

- Objective: trying to estimate mathematically the impairment introduced in the video.
- Subjective: based in the human judgment of a specialized observer.

Another classification takes care of the existence of the original video and distinguishes between three approaches:

- Full reference: based in the comparison of images when the source is available for evaluation.
- Reduced reference: it is based on characteristics (parameters) extracted from the source; some parameters are extracted from the source and compared to the received sequence.
- No reference: useful for in-service applications as it is based in the lack of neither the source nor its characteristics.

2.3. SUBJECTIVE METRICS

Subjective metrics are based in the utilization of a human judgment made by video quality evaluation specialized observers. These metrics have been used for long but they are still valid, maybe because of international recommendations such as ITU-R BT.500-8-11 [2], it has been defined a number of tests and methodologies in order to assist researchers in several stages of the subjective evaluation, covering from the preparation of the viewers to the formalization of the obtained results.

The test methods are:

- SSCQE (Single Stimulus Continuous Quality Scale)
- DSIS (Double Stimulus Impairment Scale)
- DSCQS (Double Stimulus Continuous Quality Scale)

2.3.1. SSCQE

In the SSCQE, a series of video clips is introduced to the observer so they are watched only once. These video clips can contain more or less impairment and the observer evaluate the instant quality through a slider which has a continuous scale going from 0 (worst quality) to 1 (best quality). SSCQE reports, therefore, the evaluation quality in regular time intervals and enables capturing the perceived quality variation in the time. The evaluation is absolute in the sense that the reference video clip is not explicitly shown to the viewer. This corresponds to the typical situation of a spectator: staying at home and without the chance of viewing the original video sequence.

2.3.2. DSIS

In the DSIS, the observer evaluates the quality or the variability while watching two video sequences: the impaired and the original ones. In the DSIS particularly, copies of video clips are shown in sequence that represent the reference video and the impaired one, both of them approximately eight seconds long. The observer evaluates the amount of impairment referenced to the single shown copy, in a short interval of time. The evaluation scale is discrete and its five levels are:

- Level 5: Imperceptible
- Level 4: Perceptible, but not annoying
- Level 3: Slightly annoying
- Level 2: Annoying
- Level 1: Very annoying

Next figure shows the sequence of the original and impaired videos projection in DSIS.

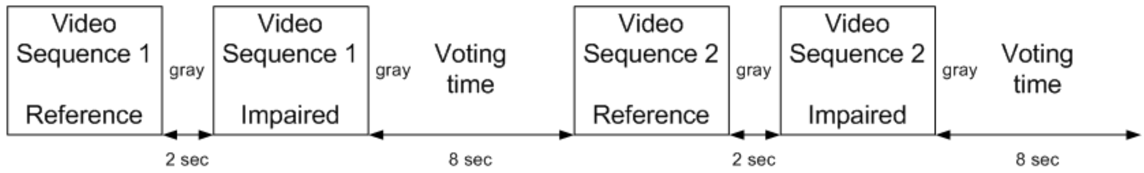

Figure 2.1. DSIS video sequence order

Particularly, it is possible to notice that the reference video is shown before the impaired one, so, with the DSIS method, the observer evaluates the degradation of the video quality referenced to the original one. The task of evaluating an impaired video while possessing the reference one turns out to be an easier task than evaluating the video quality in absolute terms as SSCQE does.

2.3.3. DSCQS

The DSCQS method is the preferred one when there is not a big quality difference between two video clips. Therefore, this method has been widely applied when evaluating high definition TV sequences. The way sequences are introduced is similar to the DSIS one.
As it can be seen in next figure, the order of the first copy is pseudo-random and is preserved in the second copy.

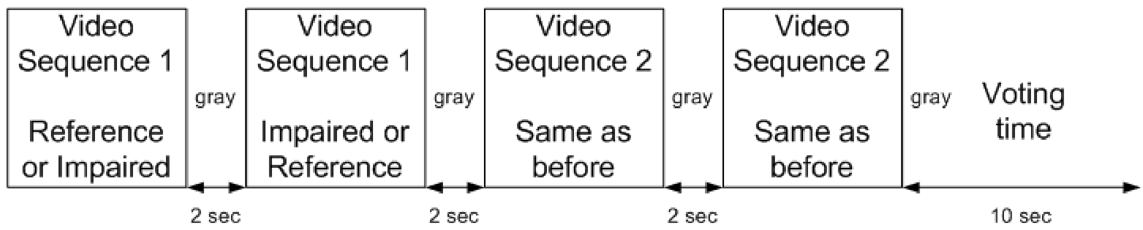

Figure 2.2. DSCQS video sequence order

At the end of the video viewing, the observer spends some time evaluating the quality between videos by using a graded scale. Observers are aimed to complete the voting task during a ten second gray period after the viewing of a sequence copy and the successive one.

At the end of the evaluation, DSCQS introduces two identically graded scales divided in five intervals that introduce these five judgments from high to low: Excellent, Good, Fair, Poor and Bad. This scale, related to the original and processed sequences is introduced twice in order to make the evaluation easier. The observer registers the evaluation of the total video clip quality using an answer sheet of paper or, digitally, by using specialized software. The

punctuations are highlighted in a vertical scale which becomes a continuous evaluation system in order to avoid quantization errors.
Typically, after the test has been done, the score is converted from a measure of length to a score, standardized in the range from zero to one hundred. Afterwards, the difference between the original and the impaired is calculated. Finally, there is the collection of all the data produced by all the spectators and for each sequence in order to draw statistical appraisals concerning the introduced video quality.

ITU-R BT.500-11 [2] international recommendations include specifications about how to execute many different kinds of subjective tests. Besides the fact that there are different ways of introducing the video sequences, subjective tests have also different evaluation scales, different words associated with these scales, and many other testing variables that change from a laboratory to another (for example, spectator competences, culture differences, variable test environments).

2.4. OBJECTIVE METRICS

As it can be intuited, subjective video quality evaluation techniques are complex as well as cost-demanding, in terms of money and time, in order to reach successful results. Then, in order to resolve the exposed problems it has been necessary to develop objective metrics which automates the process and reduce the costs associated to the subjective ones.
Objective metrics have the scope to determine the video quality of a video in absence of the participation (watching) of any observer; such metrics can be used in the “out of service” modality (when the sequence is fully available) or “in-service” (when the sequence is in streaming). These metrics have been previously applied to images and, later, to the video sequence, simply by applying the metric to every frame inside it. The objective model can be summed up in the next figure, where it can be seen that the source and the impaired video sequences are faced through an algorithm. It must be noticed that the objective evaluation method must achieve a result, which must be correlated, to the result obtained with the subjective evaluation method in order to get a mapping between the objective and the subjective results.
According to several references, objective metrics are divided according to the tools used to obtain the video quality evaluation. These tools are:

- Mathematic equations: SNR, PSNR, RMSE.
- Human sightseeing complex models: PQR, DVQ, KDD, VQM, PVQM, PDM, MPQM and MPQM(Q).

2.4.1. RMSE

RMSE (Root Mean Square Error): This parameter is obtained by calculating the average value from the subtraction between the original frame and the impaired one.

$$E_{RMS} = \sqrt{\frac{1}{M\ x\ N}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}[f'(m,n) - f(m,n)]^2}$$

(2.1)

Where f is the original frame and f' is the elaborated one.
N, M are the image dimensions.

2.4.2. SNR

SNR (Signal to Noise Ratio): It is the relation between the signal and the noise of an image.

$$SNR = \frac{\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}[f'(m,n)]^2}{\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}[f'(m,n) - f(m,n)]^2}$$

(2.2)

2.4.3. PSNR

PSNR (Peak Signal to Noise Ratio): The PSNR is a parameter used to measure the quality of the codification. In fact, it depends on the difference between the original image and the impaired one, known as the RMSE. It is a static quality measure and is obtained by applying this mathematic equation:

$$PSNR = 20\ x\ \log_{10}\left(\frac{255}{RMSE}\right)$$

(2.3)

Objective metrics based on mathematic equations have the characteristic of being used in an easy way but they can result unmatched with the subjective evaluation, producing inconsistent values with video sequences containing different spatial and temporal complexity. It must be noticed, but, that PSNR has been revalued in comparison to other objective metrics based in results obtained by the VQEG (Video Quality Expert Group). Because of barely exposed reasons it has been necessary the introduction of objective metrics based on the human sightseeing system (HVS). Some of these objective metrics are explained below:

2.4.4. PQR Metric:

The PQR metric is based on the proprietary algorithm HVS JNDmetrix from Sarnoff/Tektronix (Sarnoff Corporation). The flow diagram of this algorithm is shown below

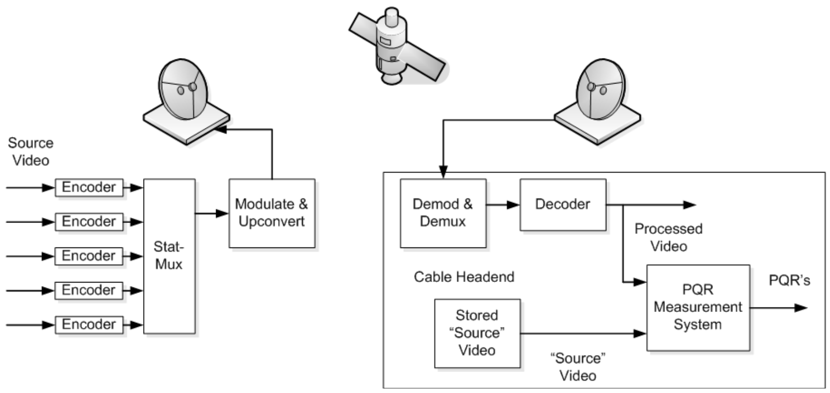

Figure 2.3. PQR flow diagram

It is used in order to extract parameters from the source sequences and create a PQR map. Next picture helps understanding this process:

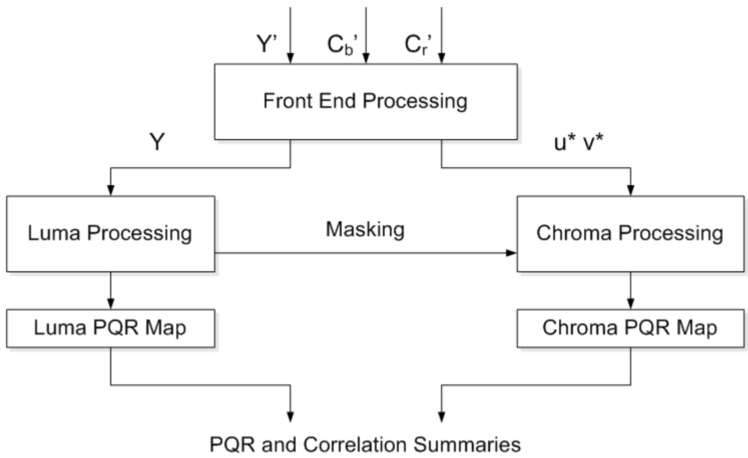

Figure 2.4. PQR map creation flow graph

Inputs are two video sequences with an arbitrary duration. For each frame of the input sequence three fields are labeled as Y', Cb' and Cr'. These data are transformed to R, G, B in order to increase the shown pixel value. The Front End Processing block is used to obtain the RGB tension: luminescence (Y) and (chrominance) bichromatic images (u*, v*) which are passed to posterior stages. The Luma Processing block accepts the Y parameter from the images (original and processed) as an input and creates a PQR map. This map is an image whose level in gray-scale is proportional to the difference of the two images related to the pixel. Finally, the Chroma Processing block takes as an input the bichromatic images and creates a PQR map related to their intensity.

The process of chrominance, intensity and luminescence are influenced by an input from the luminous channel known as "mask" that renders the difference more or less visible in relation to the image luminescence structure. Therefore, the luminescence and intensity maps are created from a small number of measures derived from these maps. The measured value gives a general vision of the distortion of the tested sequence while the maps give a more detailed view on the artifacts that have been had in the image. The PQR method demands a normalization process on the processed video sequence before carrying the comparison between sequences.

The normalization becomes useful to remove the time-invariant changes between the two video sequences before applying the HVS evaluation. The parameters adjusted in the normalization process are:

- Important horizontal and vertical movements of the image.
- Changes of color gain and luminescence.
- Changes of the continuous component of the brightness and the color.

These changes must be removed in order to supply an accurate comparison but it must be noticed that only the static time-invariant changes are removed. It is possible to define a general relation between PQR values and the subjective evaluation scales shown in next figure

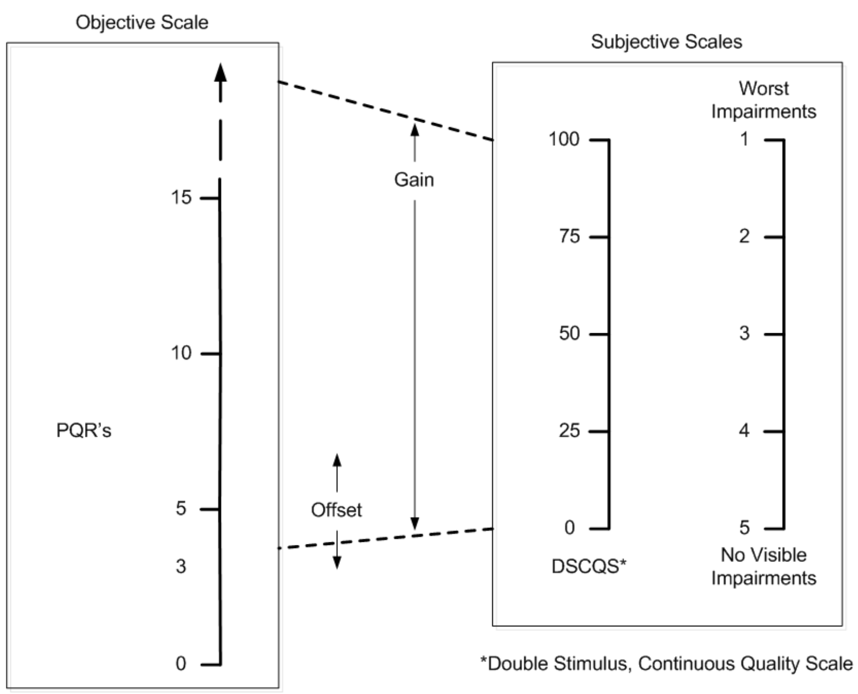

Figure 2.5. Relation between PQR and subjective values

As it can be seen, it exists a relation between the subjective and the objective scales. A DSCQS value of zero indicates that the damages are not noticeable and corresponds to a value of five in the impairment scale and to a value between three and five in the objective one.

A DSCQS value of one hundred indicates, instead, the presence of great defects (the maximum) and corresponds to a value of one in the damage scale and to a value between fifteen and twenty in the objective one.

2.4.5. VQM Metric

The VQM [19] metric has been developed by the NTIA/ITS which belongs to the US Commerce Department. The objective quality measurement algorithm of the VQM method provides approximations near the values obtained from subjective metrics.

Next figure shows the entire procedure of obtaining VQM results. As it can be noticed, both original and impaired video sequences go through a series of functional blocks which allow the sampling, the calibration, the extraction of perceived quality characteristics, the calculation of the quality parameters and, finally, the calculation of the VQM score.
Such metric shows the changes of quality due to the distortion caused by each one of the components of the digital video transmission system (the encoder, errors caused by the digital channel, the decoder). The analysis for the determination of the probable causes of degradation of the quality is lead using the RCA (Root Cause Analysis) model.
RCA lists the percentages associated to the different possible impairment: jerky motion, blurring and error blocks (talking about movement in jerks, obfuscating and block distortion, respectively). Where the 100% indicates that all the observers have perceived this impairment as the main cause of the total distortion, the 50% indicates that the observers have perceived such impairment as the secondary cause of the total distortion and, finally, the 0% indicates that the related impairment is not perceived. VQM carries out the sampling of the analogue video according to the ITU-R BT.601 recommendation. The Rec. 601 sampling is commonly known as a 4:2:2 since the components of the chrominance Cb and Cr are sampled to the same frequency, being this, half the frequency used to sample the luminance (Y) component. The sampled video is recorded according to a YUV format where Y means the luminance component and U and V mean the Cb and Cr chrominance components.
The calibration, previous to the extraction of the video quality characteristics, consists of 4 steps:

1. - Spatial registration, estimation and correction.
2. - Estimation of the valid region.
3. - Contrast and brightness (gain and offset level) estimation and correction.
4. - Temporal registration, estimation and correction.

1.- The spatial registration is used to determine the spatial movement in vertical and horizontal of the processed video respect the original one.

2.- The estimation of the valid region is executed in order to limit the extraction of the characteristics to the pixels that contain information of the frame. Besides, the PVR (Processed Valid Region) is determined as it is the region not affected by vertical nor horizontal movements. PVR is always contained in the OVR (Original Valid Region) which sets the size of the whole frame before the processing.

3.- Some video systems impose a gain different than 1 and an offset level different than 0. For this reason, a correction is mandatory in order for the two sequences to be compared. The luminance gain is what, in television terms, is called contrast and the offset level is called brightness. It must be noticed that the VQM software estimates the gain and the offset level for all the three components of the video signal used (luminance Y, chrominance U and V) but carries out the correction of the gain and the offset level of the Y component only.

4.- The temporal registration is used in order to estimate and to correct temporal shift (video delay) of the processed video sequence versus the original one. Concretely, the VQM software determines the interval inside the processed video which temporally matches the best with the original one. The next stage is the quality characteristic extraction one, where a quality characteristic is defined as an amount of information related to, or extracted from, a subregion spatial-temporal (S-T). The stream characteristics obtained are a function of the space and time.
From the comparison between the characteristics extracted from the processed video calibrated and the characteristics extracted from the original and calibrated video, it is possible to create a group of quality parameters which indicate the perceived video quality variations. This group of parameters characterizes the perceived changes in space, time and colors of the video stream. Usually, a perceptive filter is applied on the video stream in order to increase some of the properties of the perceived video quality, such as the edge information. After this perceptive filtering, the characteristics are extracted from “below region” (S-T) using a mathematical function such as the standard deviation.
At last, a perception threshold is applied to the extracted characteristics. The S-T regions are placed in order to divide the video streams within adjacent S-T regions. Since the processed video has been calibrated for every processed S-T region, it exists an original S-T region which corresponds to the identical spatial and temporal position inside the video stream. The characteristics are extracted from every S-T region. Each S-T region describes a pixel block. The size of the S-T region comes from the (1) number of horizontal pixels, (2) the number of lines of the vertical frame and (3) the number of video frames for a 30 fps video.
Next figure shows an S-T region composed by 8 horizontal pixels x 8 vertical lines x 6 video frames. When the 30 fps are applied, this region is equal to 1/5 of a second and contains 384 pixels. A fifth of a second is a desirable temporal unit, in fact, in such interval we have an integer number of frames for the video system functioning at 10, 15, 25 and 30 fps. It is possible to use the S-T region obtained from this method as it leads to a narrow correlation with the subjective quality index. The correlation decreases slowly as we move away from this S-T region size. From this “below region” the quality characteristics are extracted and the quality parameters are calculated in order to, finally, get the VQM score.

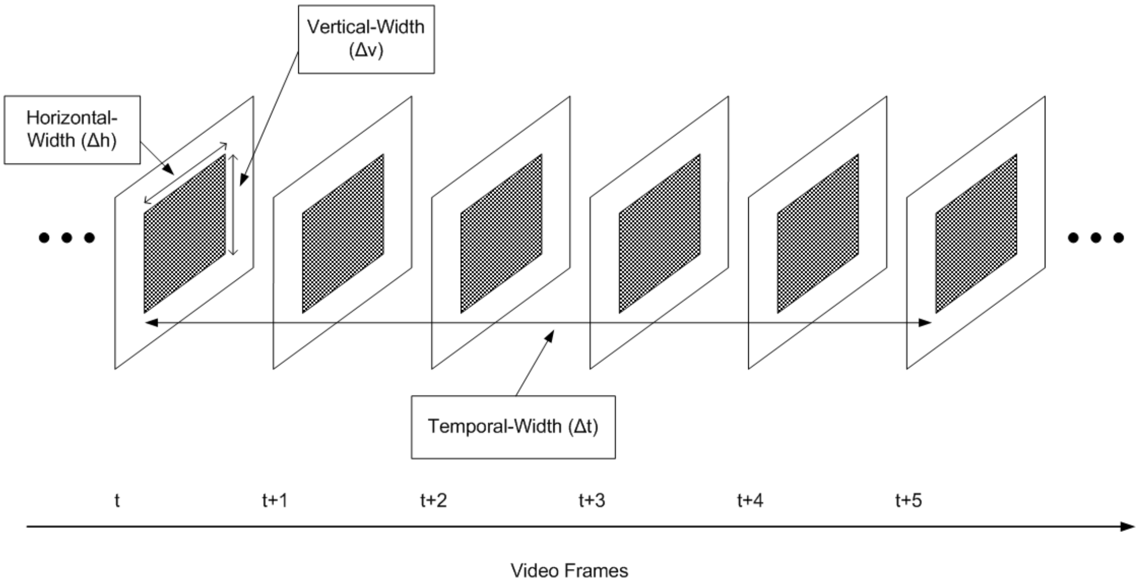

Figure 2.6. Example spatial-temporal (S-T) region size for extracting features

2.4.6. MPQM Metric

The MPQM (Moving Pictures Quality Metric) metric is based on the acquaintance of the human visual system. A general model is shown in following figure.

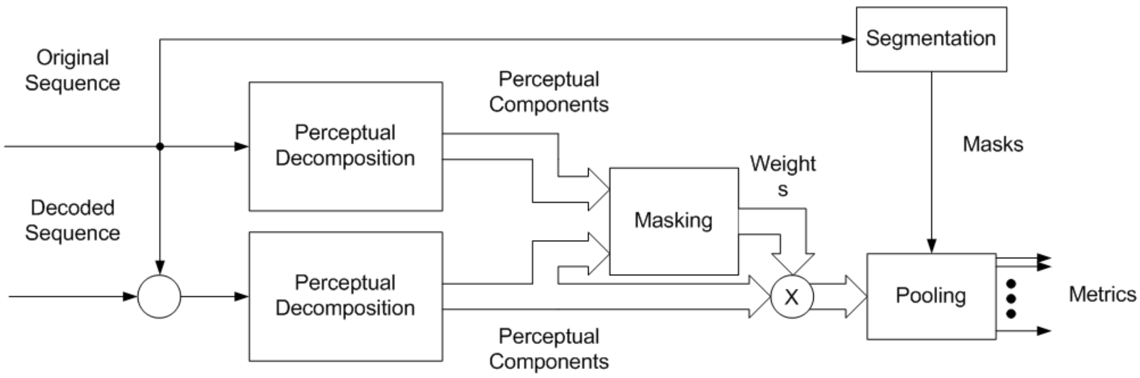

Figure 2.7. MPQM general scheme

The steps used from such metric in order to arrive to the judgment of the quality of an impaired video with the original corresponding one are the following ones:

1.- First, it is necessary to decompose the original sequence in smaller ones. The original sequence and the processed one are decomposed in their perceptive components using a bench of filters.

2.- The contrast sensitivity and the mask are brought in each channel. This is made by calculating pixel by pixel the threshold and considering that the original sequence will be a mask to the distortion. The strategy of the mask will consist, then, in dividing the error signal filtrated by the survey threshold. This result will express the *just noticeable differences,* that is, the hardly remarkable differences that represent parameters for the evaluation.

3.- Finally, all data is loaded on the channels in order to measure the distortion between the two input sequences.

It must be noticed that human observers do not observe in the same way different regions of a figure; this behavior is being caused by the attention paid and the distance. In order to take care of all this, a global metric has been developed, calculated on sequence blocks. By doing this, every block is three-dimensionally considered, being the dimensions the following ones:

- Temporal dimension: in order to take account the persistence of the image in the retina
- Spatial dimension: in order to take account characteristics related to the paid attention.

In the MPQM metric, the measure of the distortion is carried out for every block on every channel from the pooling. Therefore, the calculation of the distortion E for each block is calculated from next equation:

$$E = \left(\frac{1}{N} \sum_{C=1}^{N} \left(\frac{1}{N_x N_y N_t} \sum_{t=1}^{Nt} \sum_{x=1}^{Nx} \sum_{y=1}^{Ny} |e[x,y,t,c]| \right)^{\beta} \right)^{\frac{1}{\beta}}$$

(2.4)

Where $e[x,y,t,c]$ is the error signal masked in the (x,y) position and in the t time in the current block in the c channel; Nx, Ny are the horizontal and vertical dimensions and Nt is the block time. N is the number of channels. The exponent of the Minkowski summary is β with a value of 4 so as to close the summary.
The metric can be expressed in logarithmic scale and called MPSNR (Masked Peak Signal-to-Noise Ratio) and is calculated in this way:

$$MPSNR = 10 \, \log_{10} \left(\frac{255^2}{E^2} \right)$$

(2.5)

The scale is similar to that of the decibels and is called “visual decibels” (vdB).

CHAPTER 3. MULTIMEDIA INFORMATION CODIFICATION

3.1 INTRODUCTION

One of the key aspects when transmitting audio and video in a digital format through a channel is the use on advanced encoding techniques. One non-encoded digital video signal in SDTV (Standard Definition TV) format equivalent to the analogical television (ITU-R Rec BT.601) requires 270Mbps, which would be impossible to transmit through neither the existing broadcast channels nor current networks if none of the advanced encoding techniques were used. Next table shows an example of the resolutions achieved by Digital Video on SDTV for NTSC and PAL systems.

Table 3.1. SDTV Digital Video Resolutions

SDTV (4CIF)	**Pixels**	**Frames/s**
NTSC	704x480	24,30
PAL	704x576	25

In an ENG (Electronic News Gathering) solution, where transmission is done through the mobile network, the multimedia flows format must be suited according to the available bandwidth in the network (above all in the mobile radio channel) and the processing capabilities of mobile devices. Because of this, the choice of a coding/encoding algorithm adapted to these factors is a specially critic point.

In general terms, the main bandwidth requirement for this kind of applications comes from the need of transmitting real-time video which, with the help of specifically developed codecs, can be delivered through the mobile network.

Historically, a part from proprietary technologies (Microsoft, RealNetworks, etc.), the most important standards used in audio/video encoding have been the ones defined in the ITU-T (International Telecommunications Union) by the group ISO/IEC (inside the MPEG group).

3.2 CODECS USED IN WENG

3.2.1. STANDARD H.263

This standard was designed specifically for videoconferencing applications on communication networks with a low bit-rate and for sequences with little movement. It represents an advance respect H.261 as it produces a substantial improvement of the quality in rates lower than 64Kbps but, although the algorithm is really similar to the H.261 one, it contains some enhancements and changes that allow achieving the same quality h.261 offers but using half the necessary bits in the data flow.
H.263 [18] supports five formats:

- QCIF (176 x 144 pixels)
- CIF (352 x 288 pixels)
- subQCIF (128 x 96 pixels)
- 4CIF (704 x 576 pixels)
- 16CIF (1408 x 1152 pixels)

This standard is the base of the MPEG-4 encoding despite, later, new standards appearing which improved the original one. Some of these new standards are H.263+ and H.263++. H.263 is defined as the codec by default for mobile multimedia applications.

3.2.2. MPEG-4 Part10 / H.264 AVC

In 2001 and with the aim of developing an efficient compressing system, ISO/IEC and ITU-T Video Coding Experts Group (VCEG) joint their forces in the Joint Video Team (JVT), a workgroup whose objective was to develop and advanced encoding system, known as Advanced Video Coding or AVC. In 2003, AVC was integrated as Part 10 in the ISO/IEC 14496-10 MPEG-4 standard, named by the ITU as H.264.

This standard was developed with the both objectives of allowing the encoding of high quality video at low bitrates (reducing to half the required bitrates in previous standards, such as MPEG-2) without increasing much the complexity respect MPEG-2. By doing this, it was pretended to facilitate its implementation as well as its reusing by different kind of applications (videoconferencing, video broadcast, etc) in mobile or fixed networks.

The initial definition of the standard was centered in the video compression with quality enough for multimedia/entertainment applications. Later, the standard was extended in order to include professional encoding features, known as FRExt (Fidelity Range Extensions). Some of these professional applications could be distribution/contribution of contents or the edition and postproduction in television studios. The main features of this codec are shown in the table below:

Table 3.2. MPEG-4 AVC / H.264 FRExt main features

Features	**Bandwidth**	**Firms**
Video compression. Suitable for HD TV (High Profile). Better efficiency, above all, for high definition. Optimum visual quality. Advanced and optimized algorithms. More complex algorithms. Digital cinema profiles (4:2:2, 4:4:4 Sampling). High fidelity color encoding (digital cinema, 4:4:4 sampling)	Up to 3x MPEG-2, (up to 10% better than H.264 MP) SD-TV: 1.3-1.8 Mbps. 720p: 6-8 Mbps. 1080i: 7-10 Mbps. 1080p: 14-20 Mbps	MPEG Blu-Ray / HD-DVD DVB / Sky Tandberg Television Harmonic Broadcom ATI NVidia Apple …
Availability: Standardized in July 2004. Commercial Hardware devices start to appear		

In the same way MPEG-2 does, MPEG-4 defines several profiles, according to the encoding characteristics included and several levels according to the encoding parameters, each one defining the maximum bitrate the decoder must support:

- Baseline Profile: suitable for end-to-end and low-delay applications.
- eXtended Profile: suitable for mobile and e-streaming applications.
- Main Profile: suitable for broadcast SDTV applications.
- FRExt:
    - High Profile: suitable for end-user applications with high definition video. It supports 4:2:0 sampling with 8 bits per sample.
    - High 10 Profile (Hi10P): it supports 4:2:0 sampling with 10 bits per sample.
    - High 4:2:2 Profile (H422P): it supports 4:2:2 sampling rates with 10 bits per sample.
    - High 4:4:4 Profile (H444P): it supports 4:4:4 sampling including up to 12 bits per sample.

All the FRExt profiles, a part from the features defined in the Main Profile, support perceptual quantization matrices and size-adaptive transformation blocks.
High Profile is the candidate to be included as standard in the industry for new generation applications in which high definition video is required: HD-DVD, BD-ROM and DVB. It is introduced as the main substitute for the Main Profile as it

adds significant improvements in the encoding efficiency without adding much complexity in its implementation.

Next table sums up the transmission rates and the image size supported by the different H.264 AVC profiles and levels:

Table 0-1: Levels and bitrates in different H.264 AVC profiles [17]

Level Number	Max Macroblocks per second	Max frame size (macro-blocks)	Max video bitrate (VCL) for Baseline, Extended and Main Profile	Max video bitrate (VCL) for High Profile	Max video bitrate (VCL) for High 10 Profile	Max video bitrate (VCL) for High 4:2:2 and High 4:4:4 Profile	Examples for High resolution @ frame rate (max stored frames in Level)
1	1485	99	64 kbit/s	80 kbit/s	192 kbit/s	256 kbit/s	128x96@30.9 (8) 176x144@15.0 (4)
1b	1485	99	128 kbit/s	160 kbit/s	384 kbit/s	512 kbit/s	128x96@30.9 (8) 176x144@15.0 (4)
1.1	3000	396	192 kbit/s	240 kbit/s	576 kbit/s	768 kbit/s	176x144@30.3 (9) 320x240@10.0 (3) 352x288@7.5 (2)
1.2	6000	396	384 kbit/s	480 kbit/s	1152 kbit/s	1536 kbit/s	320x240@20.0 (7) 352x288@15.2 (6)
1.3	11880	396	768 kbit/s	960 kbit/s	2304 kbit/s	3072 kbit/s	320x240@36.0 (7) 352x288@30.0 (6)
2	11880	396	2 Mbit/s	2.5 Mbit/s	6 Mbit/s	8 Mbit/s	320x240@36.0 (7) 352x288@30.0 (6)
2.1	19800	792	4 Mbit/s	5 Mbit/s	12 Mbit/s	16 Mbit/s	320x480@30.0 (7) 352x576@25.0 (6)
2.2	20250	1620	4 Mbit/s	5 Mbit/s	12 Mbit/s	16 Mbit/s	320x480@30.7 (10) 352x576@25.6 (7) 720x480@15.0 (6) 720x576@12.5 (5)
3	40500	1620	10 Mbit/s	12.5 Mbit/s	30 Mbit/s	40 Mbit/s	320x480@61.4 (12) 352x576@51.1 (10) 720x480@30.0 (6) 720x576@25.0 (5)
3.1	108000	3600	14 Mbit/s	17.5 Mbit/s	42 Mbit/s	56 Mbit/s	720x480@80.0 (13) 720x576@66.7 (11) 1280x720@30.0 (5)
3.2	216000	5120	20 Mbit/s	25 Mbit/s	60 Mbit/s	80 Mbit/s	1280x720@60.0 (5) 1280x1024@42.2 (4)
4	245760	8192	20 Mbit/s	25 Mbit/s	60 Mbit/s	80 Mbit/s	1280x720@68.3 (9) 1920x1088@30.1 (4) 2048x1024@30.0 (4)

The MPEG-4 encoding efficiency is greater than the MPEG-2 one, being the most significant advantages next ones:

- Motion compensation:
  - H.264 AVC uses variable size and variable shape blocks in comparison to the 16x16 bits which MPEG-2 uses. Encoding efficiency can be increased up to a 15%.
  - Movement vector estimation is more precise in H.264 AVC (¼ pixel) in comparison to MPEG-2 (½ pixel). This allows an efficiency increase of the 20%.

  - o H.264 AVC uses up to five frames to estimate the movement in comparison to MPEG-2, which only uses two. This allows an efficiency increase between the 5% and the 10%.
- Spatial Redundancy Reduction:
  - o H.264 AVC uses DCT with integer numbers, reducing rounding off errors.
- Entropic encoding:
  - o H.264 AVC uses encoding techniques more complex than the static VLE included in MPEG-2. Concretely, it uses CABAC (Context Adaptative Binary Arithmetic Coding) and CAVLC (Context Adaptative Length Coding).
- Deblocking filter:
  - o H.264 AVC uses and adaptive filter for the blocks appearing in the MPEG-2 decoded image. These artifacts are one-block sized (because of the lack of DCT coefficients) or one-macro-block sizes (because of movement estimation errors) and they can seriously degrade image quality.

A bigger H.264 AVC efficiency (defined as the reduction of the bitrates, keeping an equivalent subjective image quality) is not for free and implies a higher complexity in the encoder as well as in the decoder, as next table shows:

Table 3.3. Efficiency/Complexity of AVC/H.264 respect to MPEG-2 [16]

Profile	Efficiency with respect to MPEG-2	Increase in decoder complexity
Baseline	About 1.5 times	About 2.5 times
Extended	About 1.75 times	About 3.5 times
Main	About 2 times	About 4 times

An H.264 AVC encoder is up to eight times more complex than a MPEG-2 encoder, although already exist chips in the market that allow H.264 real-time encoding/decoding.

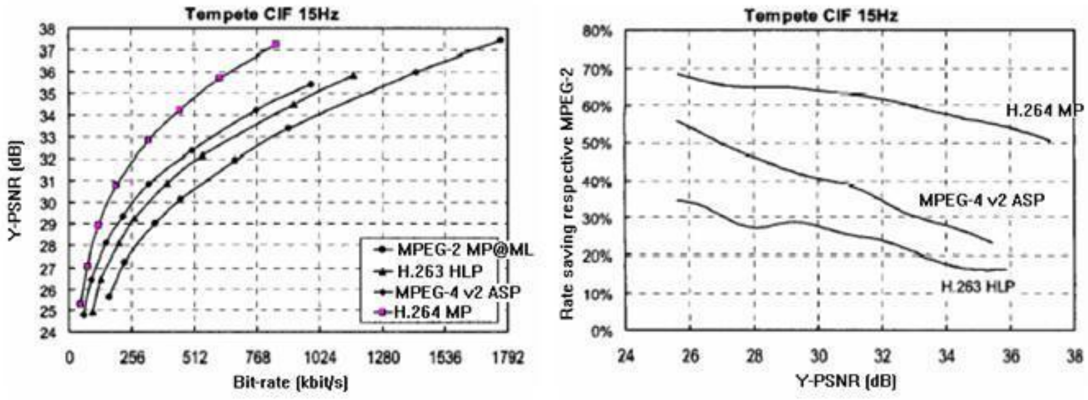

Figure 3.1. H.264 PSNR respect other codecs

FRExt profiles, particularly the ones in the High, are, currently, the most efficient tools for transmitting high definition video with bitrates equal to the 50%-60% of the bitrates offered by MPEG-2. Concretely, H.264 HP improves in a 10% the bitrate obtained with h.264 MP. The mix of advanced compression technologies, with visual quality enhancement algorithms, allows H.264 AVC to compress high definition contents without a significant quality loss. However, the algorithm requires much hardware resources to achieve this features what restrained its diffusion these last years. Nowadays, however, hardware technology advances are allowing to appear encoders/decoders with such features that enable this codec to become the favorite one when broadcasting real-time video.

CHAPTER 4. VIDEO QUALITY EVALUATION TOOLS

As some video quality evaluation tools were found while looking for documentation it was decided to test some of them in case they could be useful in our project. These tools are:

- Elecard Video Quality Estimator [12]
- Video Quality Studio 0.32 [13]
- MSU Video Quality Measurement Tool [14]
- VQM_pc [15]

Although claiming to be semi-professional evaluation tools, some limitations raised soon.

4.1. Elecard Video Quality Estimator.

Elecard company, founded in 1988, is a leading provider of software products for encoding, decoding, processing, receiving and transmission of video and audio data in different formats (MPEG-2, MPEG-4, H.264/AVC, MJPEG 2000 and others).

Elecard is one of the first companies, which has developed the video codecs for the newest H.264/AVC compression standard and whose MPEG-2 Decoder is considered to be one of the world's best.

Elecard Video Quality Estimator is a tool aimed to obtained different objective video quality metric results in an automated way. Next figure shows the application GUI.

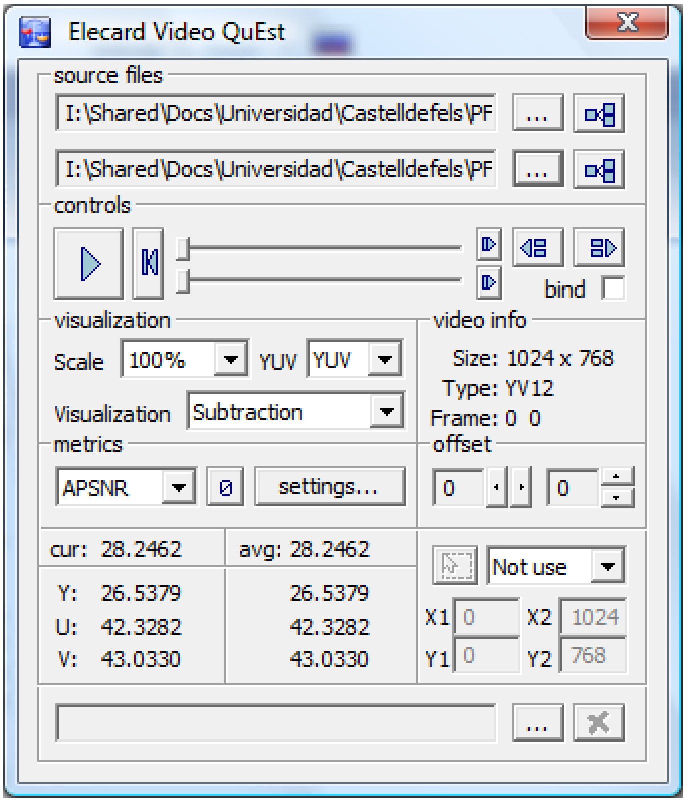

Figure 4.1. Elecard Video QuEst main window

This tool allows manual time synchronization by displaying both original and impaired video sequences. Besides, it displays a third screen showing the visualization operation between original and impaired video sequences.

Figure 4.2. Elecard Video QuEst video windows

Elecard Video Quality Estimation can provide results for several objective video quality metrics, including PSNR (APSNR or OPSNR). Results can be exported to a comma delimited value file in order to be lately analyzed.
However, this tool cannot override errors related to frame losing or frame freezing whose consequences will be explained in next chapter.

4.2. Video Quality Studio 0.32

Video Quality Studio is a software created by Georgio Diamantopoulos, which allows objective video quality estimation. The provided metrics are PSNR, SSIM and Feng Xiao's DCT-based VQM. Once you have chosen the original and the impaired video sequences, you can start the test and results are stored in a comma-delimited file which, later, can be analyzed.

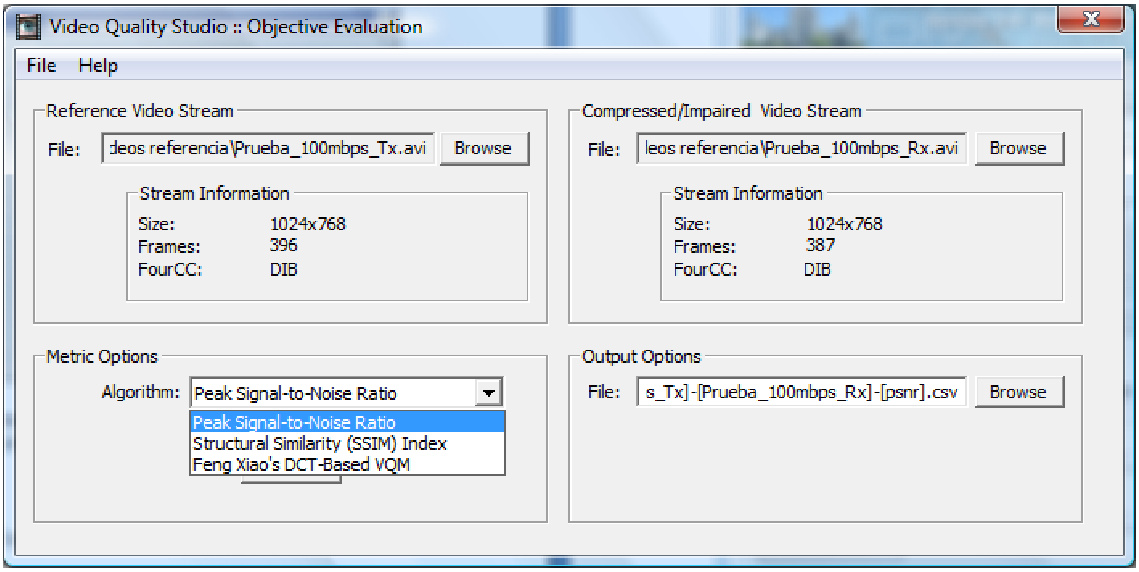

Figure 4.3. Video Quality Studio main window

Two of the limitations this software has is the lack of time synchronization feature and the impossibility of performing test whether the two sequences have different lengths.

4.3. MSU Video Quality Measurement Tool v.1.4

MSU Video Quality Measurement Tool is a software developed by the MSU Video Group, which allows objective video quality estimation. The MSU Video Group forms part of the Graphics and Media Labs [20] department in Lomonosov Moscow State University [21]. This department has developed a great amount of tools as well as publishing research papers about a wide range of subjects: from video filtering to rendering, passing by scientific visualization or image enhancement.
Speaking about the MSU tool, it raises as one of the most complete tool tested as it allows performing tests with nine different metrics. Besides, it allows choosing the component with which the test will be performed, for instance, one can choose the luminance component (L) of a YUV video sequence.

Results can be stored in a CSV file. Besides, the measure visualization video / image and the “bad frames” can be stored in disk for a posterior analysis.
Next figure shows MSU VQM tool user interface.

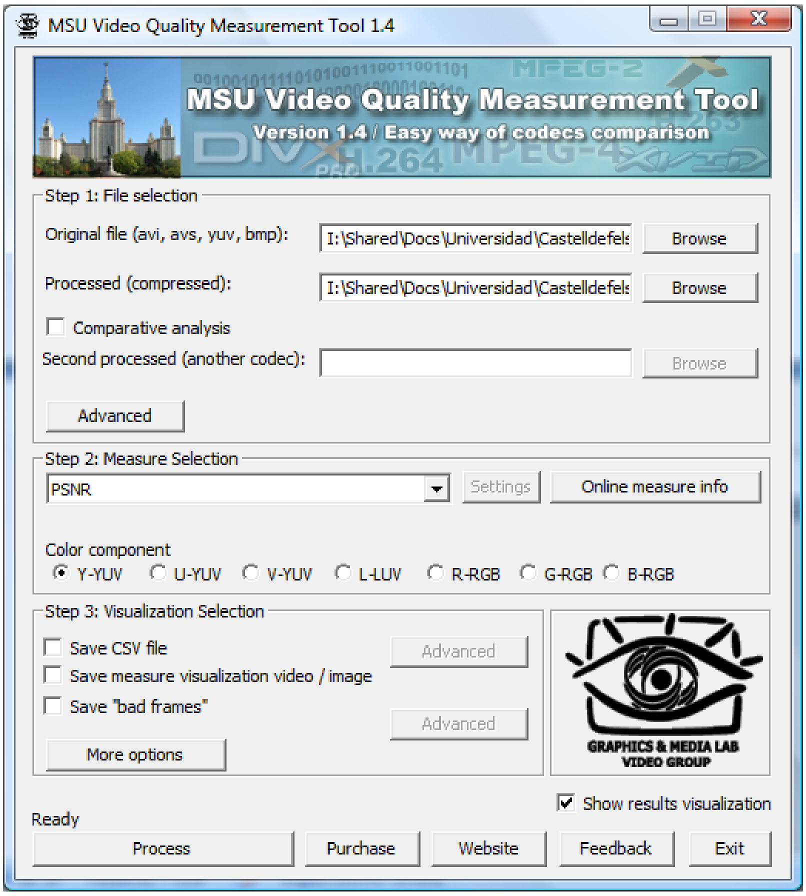

Figure 4.4. MSU Video Quality Measurement Tool main window

In spite of being one of the most complete video quality evaluation tool it has the limitation of not allowing length differences between the original and the impaired sequence.

4.4. VQM_pc

VQM_pc is a Windows program for performing detailed out-of-service / lab bench testing. The VQM_pc tool provides a graphical user-interface to video calibration algorithms, quality measurement models, video playback and conversion tools. The tool can also create a test sequence of multiple video scenes, which may then be sent through a video transmission system for quality analysis. VQM_pc operates on multiple senses at the same time (multi-threaded) for increased accuracy. Limited video capture capabilities (VfW) are included.

This tool is likely to be the most confusing software of all the tested ones as it requires completing several steps before obtaining results. These steps go from library creation (containing the video sequences to be evaluated) to calibration (although it can be chosen to perform an automatic calibration).

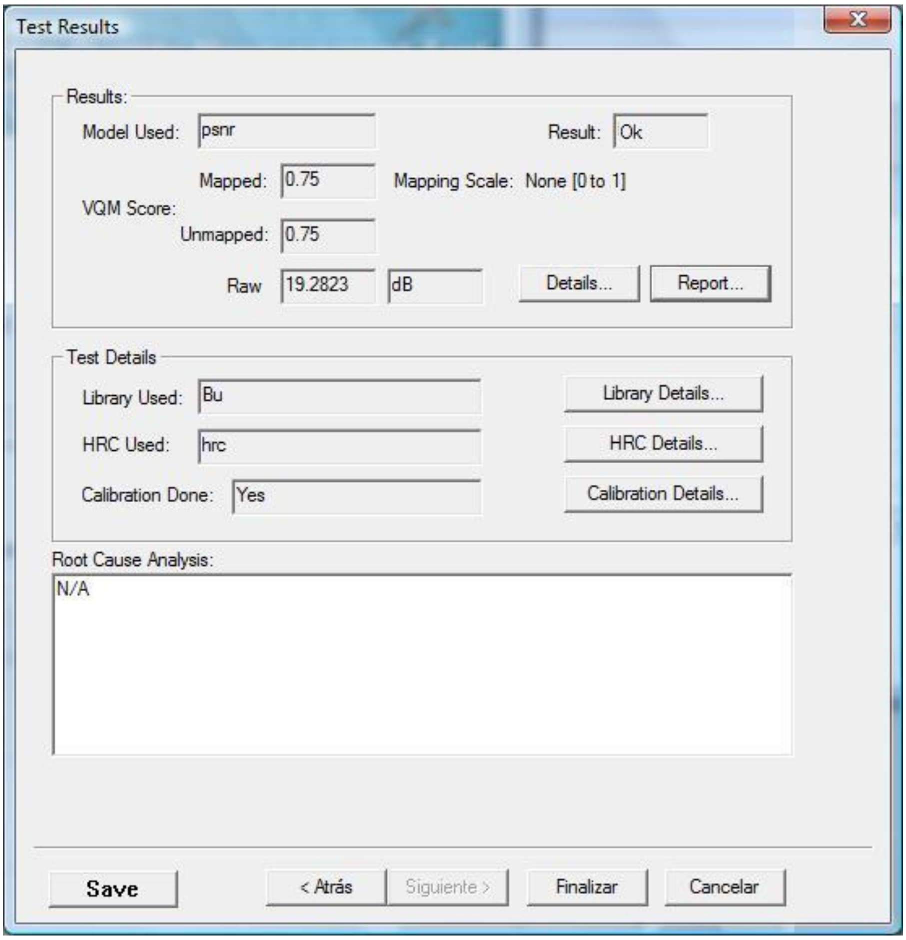

Figure 4.5. VQM_pc main window

However, this tool allows differences between sequences length as well as PSNR metric evaluation.

Next table shows the main features each video quality evaluation software has as well as the kind of license each tool has.

Table 4.1. Video quality evaluation tools comparison

Software	Metrics	Temporal Sync	Matched lengths needed	License
Elecard Video QuEst	APSNR OPSNR NQI VQM SSIM Delta MSE MSAD	Yes	No	21-day evaluation version
Video Quality Studio	PSNR SSIM VQM	No	Yes	Freeware
MSU	PSNR SSIM VQM MSU Blocking MSU Blurring Delta MSE MSAD MSU BFM	No	Yes	Freeware
VQM_pc	PSNR	No	No	Needs registering

According to the table above, it seems all this software tools were designed to check video quality degradation when encoding / compressing video sequences. In these operations frame loss is very difficult to appear as well as desynchronization. However, in our scenario, it is very probable to have frame loss and, as it will be explained later, synchronization between original and impaired video sequences will be hand-made.
These are the main reasons why a custom-made objective video quality evaluation tool has to be developed.

CHAPTER 5. PROBLEM ANALYSIS AND SOLUTION PROPOSED

5.1. INTRODUCTION

In previous chapter it has been shown the problematic related to the limits of the software tools tested for the objective quality evaluation between two sequences, the original and the impaired one. Now, it will be made a more detailed description of the problem found and the proposed solution will be shown in order to guarantee a correct comparison between videos as well as being able to get valid and reliable video quality results.

5.2. PROBLEM ANALYSIS

Objective metrics, and so VQM, calculate the judgment on the video quality without resolving in an accurate way temporal synchronization problems between the original and the impaired video sequences. It is raised, therefore, the need to develop a valid method in order to resolve the problems related to the temporal calibration in VQM. By doing this, it will be possible to get correct as well as consistent results in the process of evaluation of the impairment introduced in a video sequence and, later, introduced in a video stream.

A short description of the problem will be done so it will help in understanding the problem.
In the optimal case, the impaired video has a lower quality than the original one but it does not introduce frame loss. The next figure shows that the two videos have the same length in terms of frame and the VQM tool can make a correct evaluation as it confronts synchronized frames.

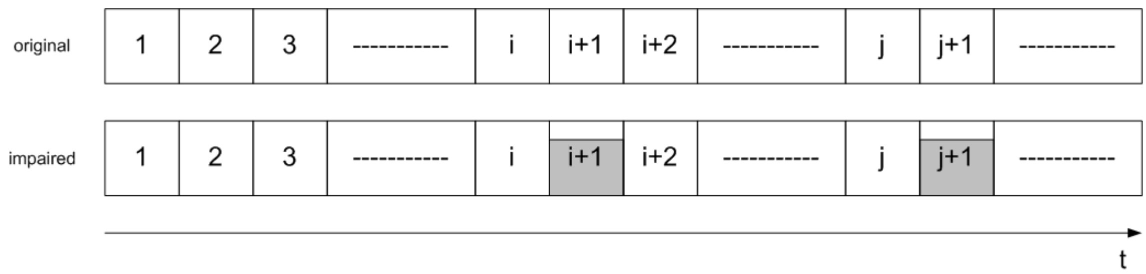

Figure 5.1. Correct VQM evaluation

In this case, the impairment evaluated is the one really present in the degraded frame (the gray frame shows it is degraded) as, by doing it frame by frame, the evaluation and a series of percentages that describe the distortion are extracted. The synchronization problem between the two sequences is verified when the impaired video lacks a frame. As it can be noticed in next figure, the loss of one or more frames (the “i” frame) provokes a shift to left of all the successive ones. This means the impaired video will have a lower number of frames.

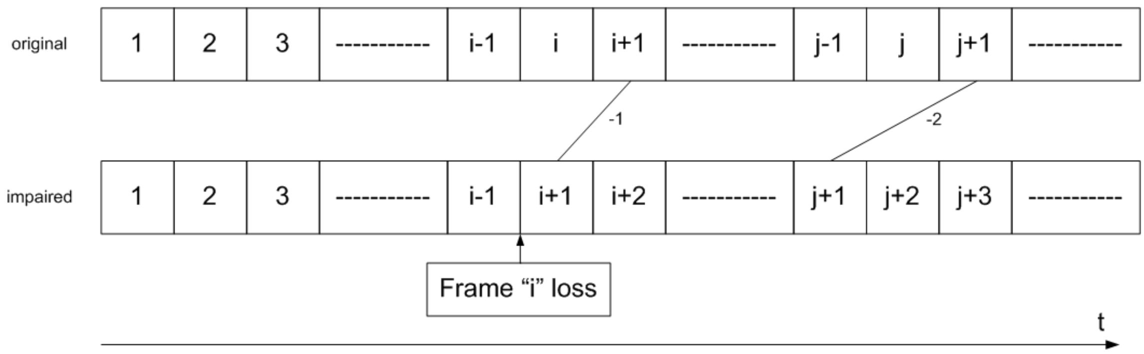

Figure 5.2. VQM incorrect evaluation as packet loss happens

The VQM tool, in its recovery strategy, carries out padding by inserting gray frames at the end of the impaired video in order to get the same length than the original video sequence.

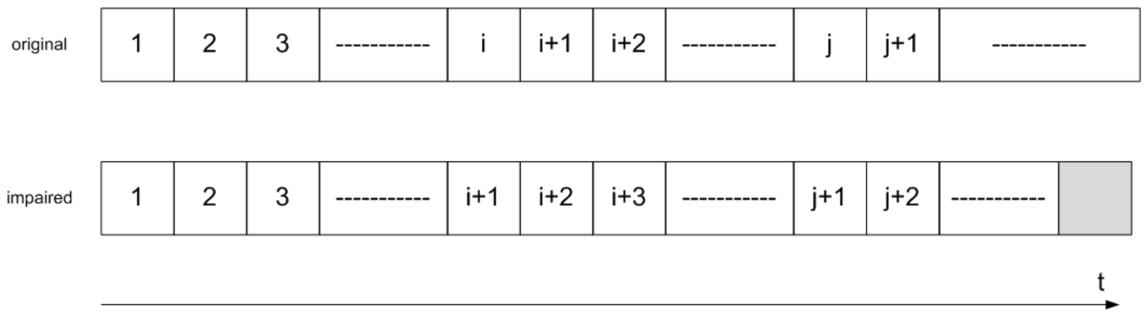

Figure 5.3. Padding added for video length matching

This padding lets the VQM tool comparing the two sequences but it does not treat in an accurate way the temporal shift leading to inconsistent results as this method is based in the temporal matching between frames. In the problem analysis stage, it appears the need to deal and solve an ulterior problem related to a codec wrong behavior: the “freezing” effect. In fact, it has been noticed than in case of a hard injection on the net, the RTP traffic could delay the arrival to the receiver. In this case, the codec, not having enough bytes for creating the frame, uses the freezing technique (resetting the preceding frame of the video stream) to guarantee the continuity of the reproduction. This behavior has been analyzed with the support of the VQM player allowing the simultaneous visualization of the two evaluated video sequences. Therefore, it has been developed a method of resolution of the problems found with the help of the Matlab development environment.

5.3. VIDEO QUALITY METRIC USED: PSNR

As a support for the understanding of the proposed solution to the previously shown problems it will be explained some of the properties related to images and to the using of PSNR metric.
The main property of an image is the modality. There are three possible modalities: RGB, gray-scale and indexed. RGB, English acronym for Red-Green-Blue, indicates that every point of the image is represented from three levels red, green and blue. Since every color perceived by human being can be represented as a combination of these three colors, RGB images are complete and colored. Whether for each color channel 8 bits are used, this leads to 256 possible intensity levels which, in the same way, lead to 16.7 million colors (256x256x256, known as True Color).
By other hand, in a gray-scale image, each point is represented with a luminance value comprised between the 0 value (black) and the 255 value (white) with all the other possible values representing different gray levels.
Essentially, the differences between a gray-scale and a RGB image is the number of color channels: a gray-scale has only one while a RGB image possesses three. A RGB image can be conceived as an overlap of three images in gray-scale, each one with a red, green or blue colorful filter respectively.
In the making of this thesis it was decided to use the Matlab development environment because of the image analysis Matlab allows. Matlab allows decomposing a video dealing with its single frames where each of them is representative of an image of the video in question. Next figure shows the way one frame is represented by using three matrixes with 16.7 million colors.

As it can be seen, the images are represented by three matrixes where each of them corresponds to one of the primary RGB color channels. Each value contained in the matrix represents a pixel. For instance, for a 176x144 pixel image, three matrixes are obtained which have same dimensions and whose elements are the values of the color intensity for each pixel.

PSNR, as described in Chapter X, is a mathematic tool used for evaluate de level of matching between two images. In our case, this metric has been used to estimate the video quality degradation.

In general terms, the developing of the Matlab mathematic tool and the use of the PSNR has allowed us to evaluate video quality degradation avoiding all the synchronization drawbacks seen in the VQM tool.

5.4. VIDEO QUALITY EVALUATION TOOL DEVELOPED

As it has been highlighted before, all the video quality evaluation tools tested were unable to produce satisfactory results when frame freezing or frame losing phenomena appeared. This is the main reason for developing a custom-made tool, able to solve all the problems found before.

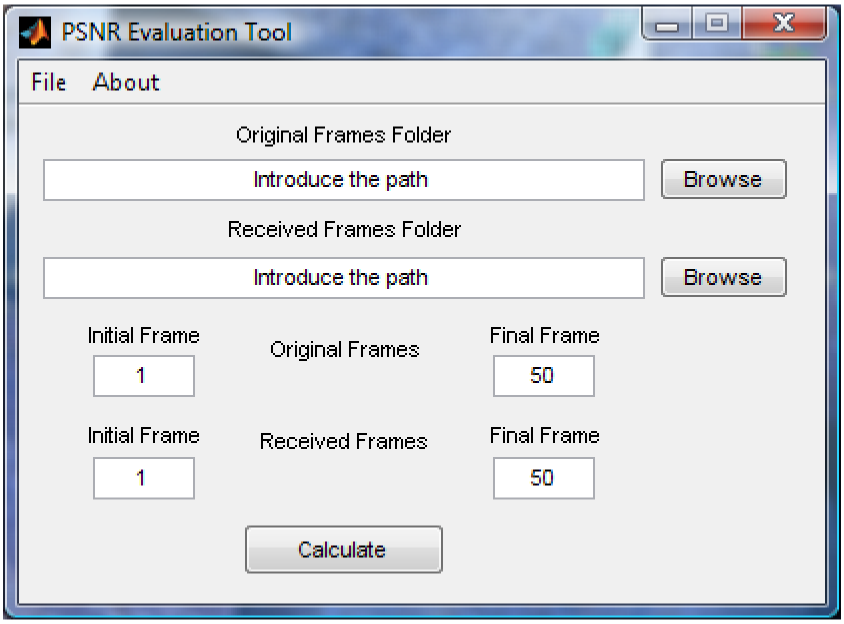

Figure 5.4. Custom PSNR Evaluation Tool GUI

This tool has been created using Matlab [8] developing environment as it offers optimized functions when operating with matrixes and an easy-to-start programming language. Moreover, the metric chosen to evaluate the video quality degradation is PSNR as its results provide an easy comprehension and its algorithm can be easily implemented.

Next figure shows the block diagram of the tool.

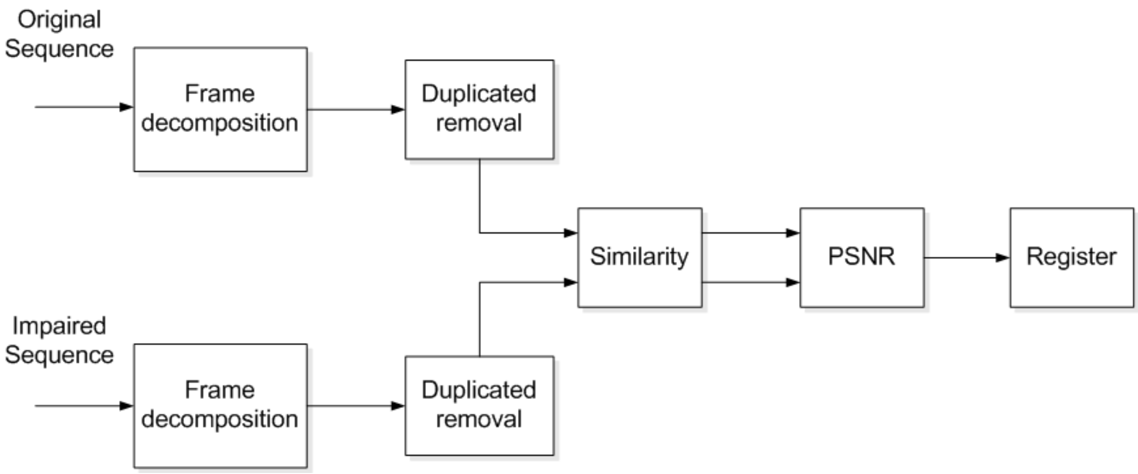

Figure 5.5. Custom video quality evaluation tool diagram block

The first function to be performed is to load the chosen frames to memory. Once all the frames are loaded, it must be checked whether identical frames are

present among the loaded frames. These frames must be removed before starting the evaluation as they can cause desynchronization between the two video sequences leading to devirtuated results.
Once we have the two sequences without any identical frame we start comparing the impaired frames with the original ones. The process is like this:

- We take the first impaired frame and start comparing it with all the original ones as it is possible for the two sequences not to start in the same instant of time.
- In order to compare the frames we make the subtraction between the impaired frame and all the original ones and, next, calculate three consecutive “average value” operations with each original-impaired couple of frames:
    - As the result of the subtraction is an NxMx3 matrix (RGB), the result of the first “average value” operation will be the average value of the three image components.
    - The result of the first operation is an NxM matrix. The second “average value” operation will correspond to the average value of all the columns of the frame.
    - The result of the second operation is an M-length row with the average values of all the columns of the matrix. The third value obtained from the “average value” operation will have as a result a single number.
- It is assumed that those original-impaired frames with a lower number, result of the previously explained operations, will be those frames more similar between them.
- Once we have found the first original-impaired couple, we can continue comparing the consecutive frames obtaining, finally, a correspondence table. This table will contain which impaired frame corresponds to an original one so; afterwards, PSNR can be calculated with this couple of frames.
- When the correspondence table is created, we have to pass all the original-impaired couples to the PSNR module.
- The result of the PSNR is a number, expressed in logarithmic scale, showing how much similar these two images are. In case both images were identical, PSNR value for this couple would be infinite. This can be noticed from the PSNR mathematical equation:

  If the subtraction between the impaired and the original frames is equal to zero, the logarithm is equal to infinite.

- Finally, a graphic representation of the PSNR according to the frames checked is created allowing an easier analysis of the obtained results.

CHAPTER 6. IMPLEMENTATION (TESTING SCENARIOS)

These initial premises became, finally, the specifications for this final career project. These specifications are:

Table 6.1. Final project specifications

Final project specifications	
SIP Client	The SIP client chosen was Polycom PVX version 8.02
SIP Proxy	OpenSER
Video codec	H.264
Audio codec	Polycom Siren14
Computers	2x Dell Latitude D620
Operating system	Windows XP
Camera	Sony Handycam
DV adapter software	WebCamDV
HSDPA hardware	Huawei modems

Once all these specifications have been satisfied, some testing must be carried out. Testing has consisted in three different scenarios (network topologies) and the elaboration of a methodology which has allowed evaluating the video quality degradation in these scenarios.

6.1 TEST SCENARIOS

6.1.1. Introduction

Testing process has consisted in the setting of a series of scenarios or prototypes whose complexity has been increased in an evolved way until reaching the implementation of a MNG solution system. This system must be able to establish a videoconference with a television studio through several mobile links with a video quality as acceptable as it can be broadcasted.
Main testing objectives have been:

- To evaluate the performance provided by SIP client with H.264 video encoding and fullscreen working.
- To evaluate several design and scenarios alternatives. By making the tests in an evolutive way has allowed identifying design weak points which has been solved along testing stage.
- To identify the maximum audio/video quality it can be reached / offered while using this client depending on the transmission technology used.

- To analyze video and audio quality in different points of the transmission chain, adding, progressively, elements which can degradate this quality.
- To compare audio / video transmission effects through multiple radio interfaces to a single transmission interface (end-to-end latency increase, jitter, packet loss, etc.)

All these tests will help to demonstrate the technical viability of a solution with these characteristics.

6.1.1. Prototypes description

The conceptual design, or logic view, of an MNG system is quite simple:

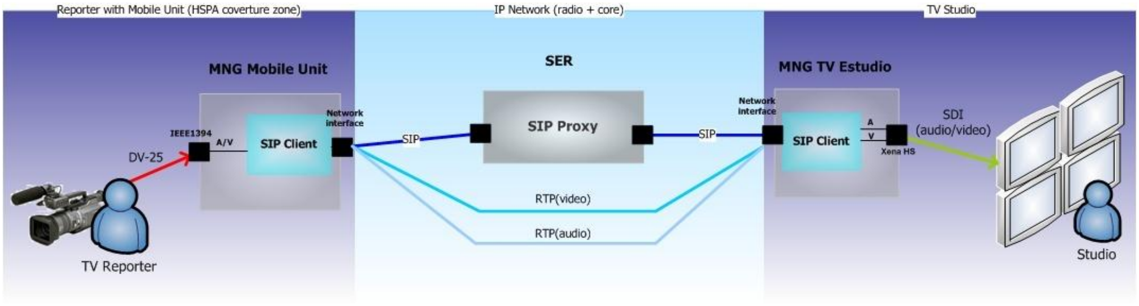

Figure 6.1. Basic MNG system logic view

The mobile client consists of a professional camera connected to a module with a FireWire interface which has a network interface to transmit captured data over IP towards the television studio.

The connection is established through a SIP client installed in the device which encodes DV captured signal (which includes audio and video components) and transforms it in audio and video independent RTP/UDP/IP flows.

In order to establish a call, a SIP signaling exchange between the mobile unit and the studio is performed. This allows both endpoints setting video codecs as well as any necessary data to assure the correct reception / transmission between end-points. This will be made using a SIP Proxy.

Once the session is established, RTP packets are distributed in the IP network towards a network interface placed in the television studio. Here, another SIP client receives RTP packets and transform them in a signal suitable to be accepted by television studio ingest system (SDI signal).

This system, easy in concept, becomes a physic complex design as several implementation solutions exist and, these require integrating heterogeneous components as well as environments, not conceived for capture and transmission of SDTV quality signals.

6.1.2. Scenario 1: Videoconference without HSDPA/HSUPA and with transmitter and receptor in the same network.

Tests started in the simpler scenario which would be equivalent to a MNG system (without taking care of the final SDI signal transformation stage) which is shown in next figure.

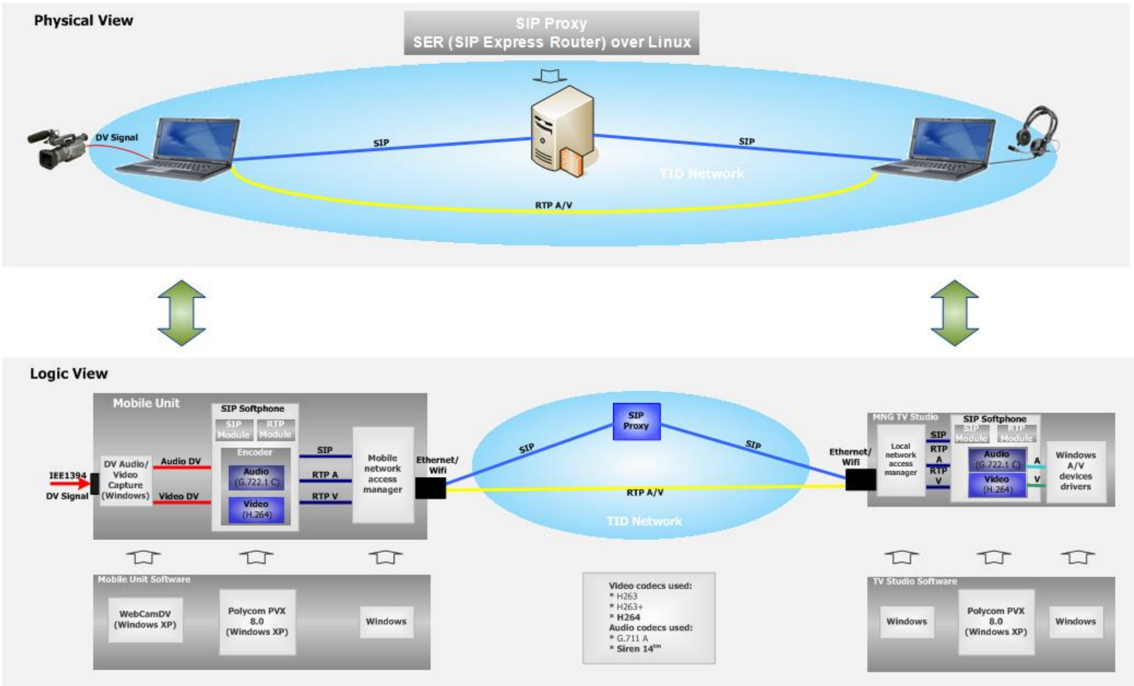

Figure 6.2. Scenario 1 physical and logic view

A professional camera connected to a laptop by its FireWire interface. This laptop has a Polycom PVX SIP client installed which allows capturing audio and video components from the DV signal. This laptop, moreover, is connected to TID network through its Ethernet interface (100Mbps).In reception, another laptop, connected in the same network is used.

The main objective of this scenario is to check the maximum video quality which can be provided by SIP clients, using a low-rate packet-loss and high bandwidth medium in comparison to the medium in which application will be deployed (3G mobile network).
Besides, tests in this scenario have been useful as they have allowed us checking whether a first approaching of the final solution was viable or not. It has been really important to make most of the components of the final solution work together as limitations have risen up.

6.1.3. Scenario 2: Mobile unit with HSDPA transmission

This scenario establishes the first prototype with transmission through mobile network links. In it, the mobile unit establishes a data connection with Telefonica Moviles network using Movistar desktop and a HSDPA modem. This mobile unit gets registered against a public SIP Registrar/Proxy.
The equipment placed in the other communication extreme simulates the MNG TV Studio and it is connected through an ADSL and gets registered against the same SIP Registrar/Proxy than the mobile unit.
In this case, the SIP Proxy is placed in TID network but it has, as well as the mobile unit and the MNG TV Studio, a public IP address.
This scenario is the one which shows most similarities with a final deployable scenario whether HSUPA technology was used:

- The mobile unit has a SIP client which makes the call to another SIP client placed in the TV Studio.
- The connection is made by the mobile unit through a single UMTS/HSPA data link.
- MNG TV Studio is connected to the rest of the MNG chain through an ADSL connection or a dedicated data connection in which quality of service (QoS) for the videoconference service is guaranteed.

Next figure shows the physical design and the corresponding logic modules of the testing scenario.

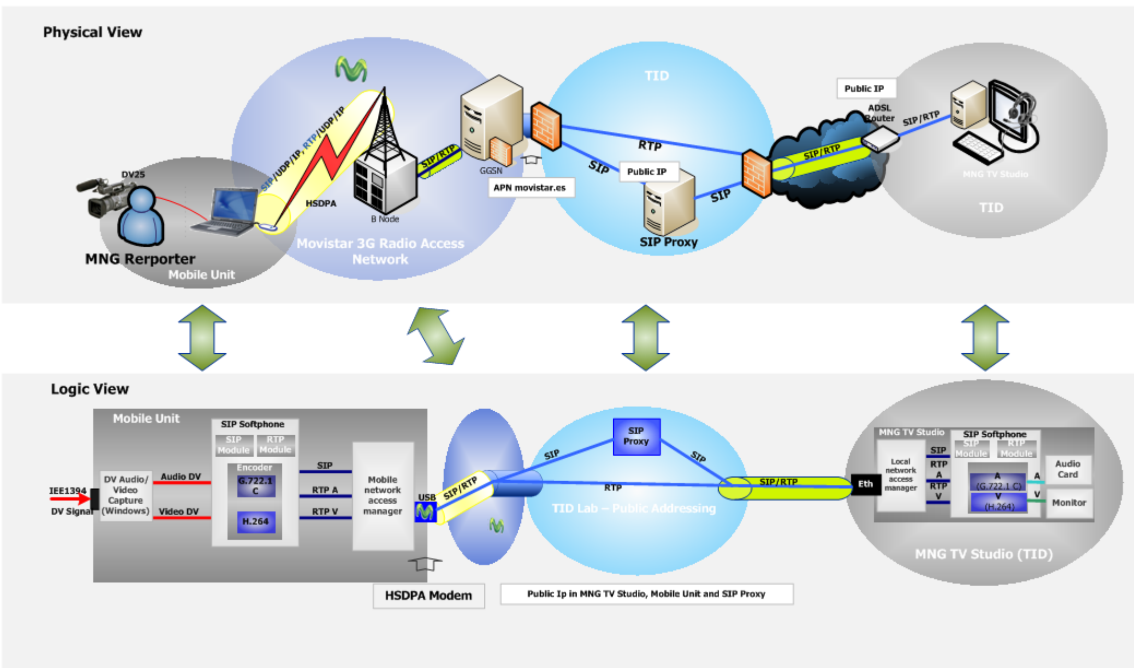

Figure 6.3. Scenario 2 physical and logic view

6.1.4. Scenario 3: Bandwidth aggregation mechanism with HSDPA in mobile unit and ADSL in reception.

This scenario is a direct evolution of the scenario 2. In it, the mobile unit uses several UMTS/HSDPA connections for the RTP flows transmission through the bandwidth aggregation mechanism developed in WENG project.

In this scenario appears the MUNIBA multiplexor whose objective is to unify audio/video RTP received packets, reorder them according to the flow type (and according to the order they were transmitted in order to avoid media synchronization problems) forwarding them towards the studio through the ports negotiated in the mobile unit – studio SDP negotiation process.

Next figure shows the physic and logic structure of this scenario.

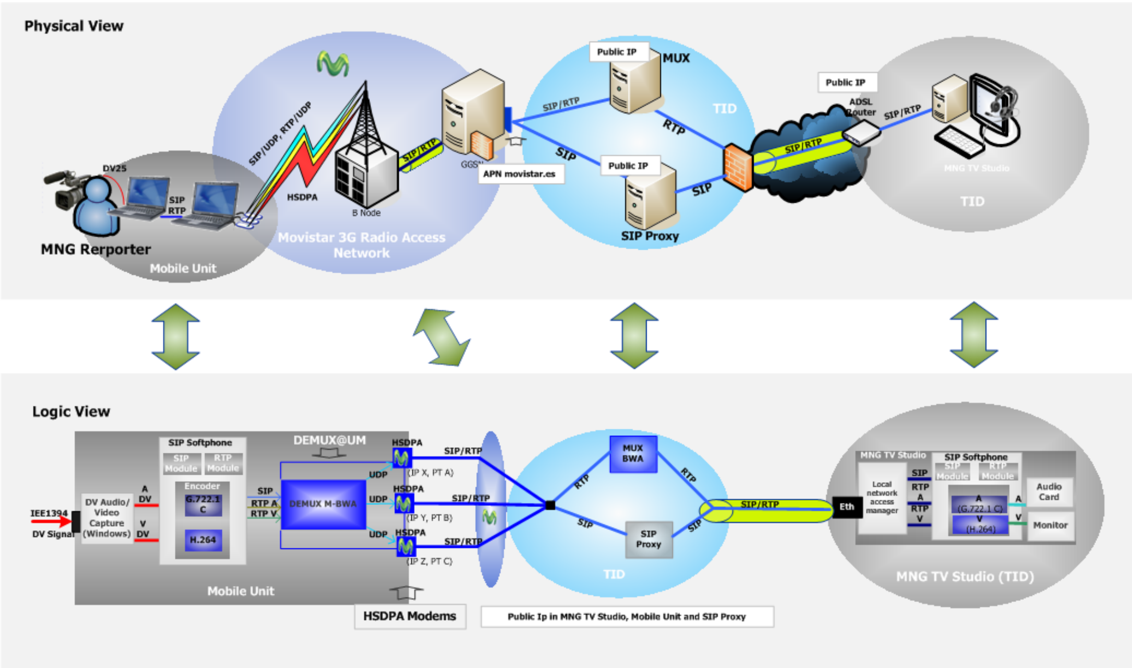

Figure 6.4. Scenario 3 physical and logic view

As MUNIBA technology is out of the scope of this project as it is a patent-pending technology developed by Telefonica, the reason of this scenario is to evaluate video quality with a higher uplink bandwidth as MUNIBA allows aggregating uplink connections bandwidth among several HSPA network interfaces.

All the video quality evaluation results are exposed and explained in chapter 7.5 Video quality evaluation results

6.2 VIDEO QUALITY EVALUATION METHODOLOGY

As evaluating video quality in each scenario is not an automatic process (many scenario components, different video sequences, etc.), a testing methodology was implemented. This methodology can be separated in three phases:

scenario settling, video capture and results analysis, each one with its own advantages and drawbacks.

6.2.1 Scenario Settling stage

In this stage, all the scenario components must be set up to make a videoconference call. Concretely, the steps are:

- Both laptops must be turned on and with the videoconference software running.
- The camera must be connected to the transmitting laptop through its firewire connection and WebCamDV software must recognize the camera and pass the video signal to Polycom PVX software.
- Depending to the scenario, network adapters or HSDPA modems must be initialized. In the second case, HSDPA modems must be connected to the 3G mobile network.
- Connection parameters such as the desired bandwidth and the video quality must be specified prior to the videoconference call. This is done by setting the required options in Polycom PVX software.
- SIP Express Router server must be running and both clients, transmitting and receiving one, must be registered to it.
- Wireshark, NetperSec and Windows Performance Register software must be idle and prepared to write down all the performance indicators (CPU utilization, Bandwidth used, packet loss, etc.)
- Camtasia software must be idle in both transmitting and receiving laptops as it will record original and impaired video sequences.

Once all these steps are checked, videoconference can start.

6.2.2 Video Capture stage

In this stage, a videoconference will take place and both original and impaired video sequences, as well as all the performance indicators, will be recorded and stored. The processing steps are:

- Wireshark, NetPerSec and Windows Performance Register are started.
- Any of the two clients starts the videoconference by calling the other part.
- Once the videoconference is established, both clients must set the video in full screen mode.
- Camtasia Recording tool software is started by setting the recording region as the full screen video.
- Video sequences have 4 differenced parts, each one with a specific performance behavior and lasting, all of them, about thirty seconds:
    - First sequence shows a steady image without movement neither in the background nor in the foreground.
    - Second sequence shows objects moving in the foreground of the image while the camera stays fixed.
    - Third sequence shows objects moving in the foreground of the image while zoom is varied. So, in this sequence, foreground and background image is modified. The camera remains still.

- In the fourth sequence, reporter takes the camera and, while moving around, varies the zoom and focus on moving objects.

  All these sequences are aimed to evaluate the system performance in different recording situations as system should not perform the same way while recording steady or high-movement situations.

- Once all the sequences have been performed and recorded, any of the two laptops finishes the videoconference as well as Camtasia recording tool.
- Finally, Wireshark, NetperSec and Windows Performance Register tools are stopped and results are saved and tagged for later analysis.

6.2.3 Result Analysis stage

In this stage, video sequences as well as performance indicators are analyzed to obtain video quality evaluation results. The steps are:

- Both the original and impaired video sequences must be separated in their own frames as Video Quality evaluation tool developed takes video frames as an input. This process is performed by the Blaze Media Pro software, which allows setting the output frames size and color scheme. As video sequences have been recorded in a 1024x768x3 (RGB) resolution, video frames will be extracted with the same characteristics.
- Once video frames are available, they must be passed to the Video Quality Evaluation tool. This tool processes frames as it has been explained in chapter 5.4 VIDEO QUALITY EVALUATION TOOL DEVELOPED.
- Results are stored in a spreadsheet allowing a graphical representation which eases the result interpretation process. A template has been created to make this last step simpler.
- Performance indicators are processed in a separate spreadsheet as well.

Once all these steps are performed for each scenario and obtained data is processed and analyzed, only conclusions leave to be extracted.

CHAPTER 7. RESULTS AND JUSTIFICATIONS

After finishing all the tests and collecting all the relevant data, it is time for commenting the obtained results.

7.1 Network performance results

Next table shows main characteristics concerning network performance parameters such as transmission bandwidth, maximum frames per second achieved or packet loss.

Table 7.1. Network performance results

Scenario	SIP Client	Max Transmission Bandwidth	Max Resolution	Max frames/s	Latency	Audio Jitter	Video Jitter	Packet loss
1	Polycom PVX	620 Kbps	QVGA	30 fps	<1 ms	2-10 ms	6 ms	0%
2	Polycom PVX	30-317 Kpbs	QVGA	24 fps	1 s	12-22 ms	17-34 ms	0,28%
3	Polycom PVX	150-700 Kbps (4 modems)	QVGA	30 fps	2-6 s	11-16 ms	16-30 ms	0,19%

Watching the results, some conclusions can be extracted.

7.2 First scenario

In the first scenario, where both clients where transmitting / receiving videoconference flows through a 100Mbps Ethernet connection, results are optimal. For instance, packet loss during tests was almost inexistent as network technology as well as network protocol used (TCP) assures a really low packet loss ratio. Although UDP turns up to be the most efficient protocol for transmitting multimedia streams through the network, Polycom PVX used TCP as transport protocol.

Another thing which must be highlighted is the fact that, in spite of being configured to make videoconference calls with a bandwidth of 1920Kbps, Polycom PVX only reached 620Kbps as the maximum transmission bandwidth.

The maximum resolution Polycom PVX SIP client allowed us to make a videoconference was QVGA (320x240 pixels). While looking for information about this illogical result, it was found that this SIP client established the maximum resolution according to the model of processor (CPU) the computer was equipped with. As this software did not have our laptop's (Dell Latitude

D620) processor (Intel Centrino Duo) typified, it used the lower resolution available.

The results obtained in this scenario where established as the reference ones for future comparisons.

7.3 Second scenario

In the second scenario, where the mobile unit transmitted videoconference flows through a single HSDPA mobile network interface, all the measures turned out to be worse than the first scenario, logically.

The maximum transmission bandwidth achieved by a HSDPA modem, theorically, is 384Kbps while, in our case, was of 317Kbps. This transmission bandwidth can vary, theorically again, between 64 - 384Kbps as it strongly depends on the number of modems connected to a BTS and on the HSDPA bandwidth assignation discipline implemented in Telefonica's BTS. According to Telefonica's Mobile Network managers, all the available bandwidth is distributed among the connected devices according to these premises:

- The first two HSDPA devices in connecting the BTS are assigned 384Kbps in uplink.
- Beyond this point, all the devices connecting the BTS are assigned 64Kbps in uplink.

Because of this important bandwidth limitation, most of the network performance measures get values according to it, that is to say, lower bandwidth, bigger packet loss ratio and bigger latencies than in first scenario.

7.4 Third scenario

In the third scenario, where the mobile network interface multiplexing takes place, the effect of this technique becomes notable, above all, in transmission bandwidth terms. The bandwidth aggregation allows better network performance parameters values but, as these are mobile network links, some of these parameters do not get any improvement. Concretely, latencies, jitters and packet loss do not get better values than first scenario ones as mobile network has an aggressive network profile.
Besides, the bandwidth assignation discipline makes the maximum available bandwidth vary between 700Kbps and 150Kbps, which can lead, in the worst case, to high video quality degradation.

Next figure shows a comparison between the average bandwidths, transmission as well as reception, for each scenario.

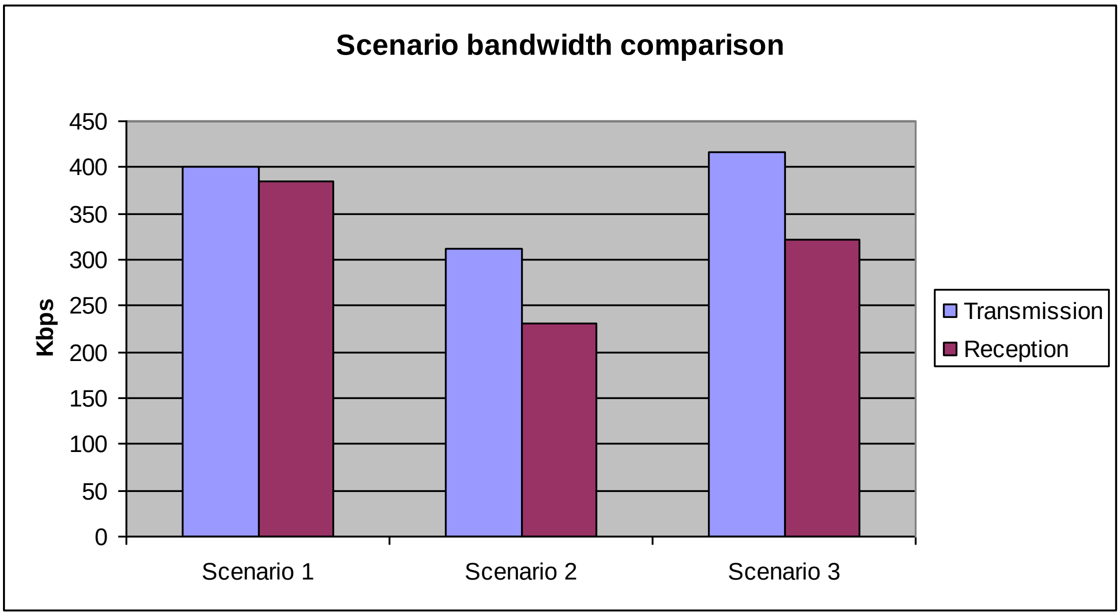

Figure 7.1. Scenario bandwidth comparison

It must be highlighted that this figure shows the bandwidth lost during a videoconference call as transmission values are the amount of bits transmitted, in average, in a call and the reception values are the amount of bits received, in average too, for the same calls.
Besides, it can be noticed the effect of the aggressive network profile as packet loss is greater while making a videoconference call through mobile network.

7.5 Video quality evaluation results

In video quality evaluation terms, results tendency is similar to the ones in network performance. When bandwidth is the highest and latencies, jitter and packet loss ratio are the lowest, video quality is the best.
Comparisons between scenarios have been done according to the four different video sequences established for this testing, as it has been explained in chapter 6.2.2 Video Capture stage

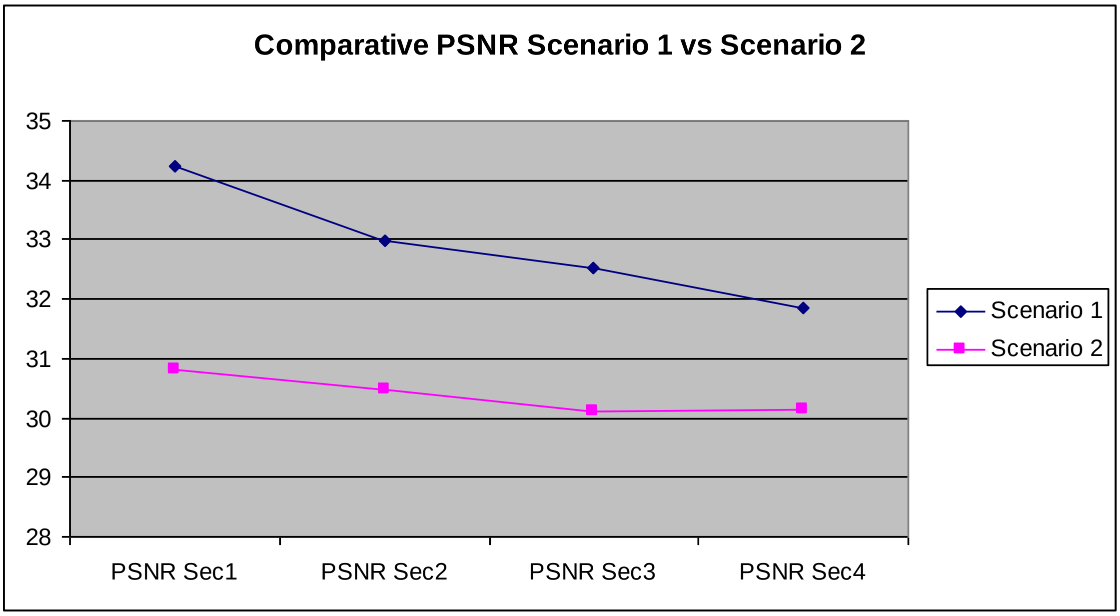

Figure 7.2. Comparative PSNR Scenario 1 vs. Scenario 2

The figure above shows the comparison between the PSNR of the first and second scenarios. As it can be seen, video quality degradation is lower in first scenario than in the second one, fact that matches network performance results for each scenario. Besides, it can be noticed that, although having, both scenarios, a constant transmission bandwidth, video quality gets worse as long as video sequence has more movement. This fact can be associated with encoding processing stage of the transmission as, when bigger the amount of information to transmit is, higher is the bandwidth required. In other words, as H.264 codec uses movement vector and spatial redundancy techniques, it requires a higher amount of information for encoding video sequences that introduce a lot of movement. This effect can be observed in the three scenarios.

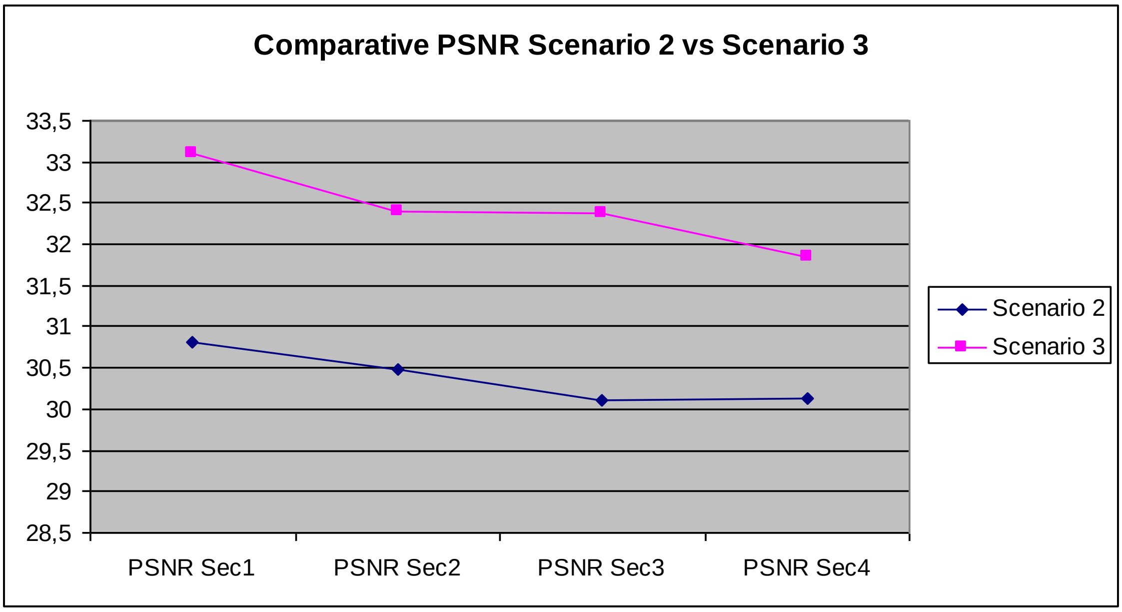

Figure 7.3. Comparative PSNR Scenario 2 vs. Scenario 3

The comparison between the second and the third scenarios follows the tendency of the first comparison: video quality is proportional to network performance. In this case, but, as these two scenarios have the same problems due to mobile network drawbacks, only the bigger bandwidth available in the third scenario provides a lower video quality degradation.

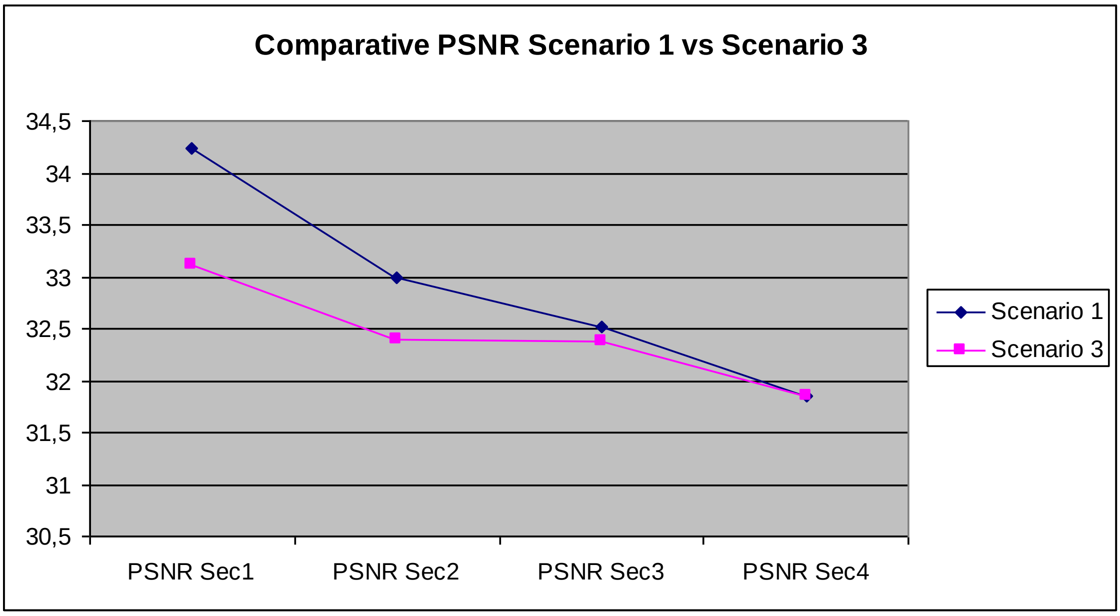

Figure 7.4. Comparative PSNR Scenario 1 vs. Scenario 3

Finally, comparing the first and the third scenarios one can see that video quality is very similar between them. In fact, only in “steady” video sequences video quality gets different values between scenarios. This can be due to better network behavior in the first scenario (100Mbps Ethernet) than in the third one (Mobile network), understanding “better behavior” as the better values of the network performance parameters.

CHAPTER 8. CONCLUSIONS AND FUTURE WORK

The main objective of this final career project was to create a methodology which allowed the objective video quality evaluation in mobile network environments.

This objective has been achieved.

First of all, it has been necessary to study the start-of-the-art in video quality evaluation. By doing it, it has been known that exist two main categories in video quality evaluation: subjective and objective techniques. In the first ones, a human observer evaluates the quality of a video sequence by comparing it to the impaired (or original, depending of the technique) version of this video sequence. It is important to have a minimum number of observers as tendency curves can be extracted from the results.
Objective video quality techniques are those where video quality is evaluated by using mathematical equations while comparing original and impaired video sequences.

Once the most appropriated technique has been chosen, it has been necessary to design, deploy and test different scenarios in order to check video quality in different points of the video transmission chain.
In this point, it has been noticed the importance of every element of the chains as, each one, has notable effects related to video quality. While WebcamDV software transformed a PAL video signal to a VGA one, the videoconference client transformed this signal to QCIF. It is necessary to highlight that each of these components has a very important role in the achievement of the project so none of them could be avoided.

While studying the state-of-the-art of the video quality evaluation techniques, some video quality evaluation software was met. A deeper study in this software tools allowed us to detect a flaw all of them had: none of them were able to make a good video quality evaluation when frames were lost, wherever they were captured. A time synchronization requirement appeared to be critical in video quality evaluation and, as none of the previously studied software tools were aware of this, a custom-made software tool had to be developed.

Although being able to perform the video quality evaluation properly, even with sequence desynchronization, the developed tool should be improved. Several proposed improvements are:

- Making the application use video sequences as information input instead of the images corresponding to video decomposing. This feature would help avoiding the use of a software tool (Blaze Media Pro) and would allow a more natural testing methodology. However, it must be highlighted that Matlab Developing Environment does not accept all kinds of encoded videos so, a trade-off should be achieved.
- Optimizing the memory usage. As Matlab must load in memory all the frames before starting the evaluation, memory should be freed as soon

as we know a frame is not used again. Besides, when performing video quality evaluation concerning big-sized frames (1024x768 pixels) the amount of simultaneously chosen frames gets dramatically reduced. In our tests, as images where 1024x768, we could only select two second video sequences (25 frames/sec * 2 seconds = 50 frames) for video quality evaluation, which limited our study in a notable way.

Making a wider lecture of the obtained results, it is easy to notice a clear one: video quality is much related to the network behavior: as soon as network performance values get worse, video quality degradation starts increasing. As Internet IP communications have, in most cases, a “best-effort” profile, constant bitrate communications, such as videoconference, can suffer from latencies, jitter and packet loss.

Implementing quality of service (QoS) disciplines in IP networks can provide better performance results in this kind of communication.

REFERENCES

[1] ITS: http://www.its.bldrdoc.gov/n3/video/

[2] ITU-R Recommendation BT.500, “Methodology for subjective assessment of the quality of television pictures,” Recommendations of the ITU, Radiocommunication Sector.

[3] Tektronix “A guide to maintaining video quality of service for digital television programs”, White paper: http://www.broadcastpapers.com/whitepapers/TektronixQOSforDTGuide.pdf

[4] ITU-R Recommendation BT.601, “Encoding parameters of digital television for studios,” Recommendations of the ITU, Radiocommunication Sector.

[5] Standard ANSI: http://www.ansi.org

[6] ANSI Video Quality Tutorial: http://www.its.bldrdoc.gov/n3/video/tutorial/

[7] VQEG – Video Quality Expert Group: http://www.vqeg.org

[8] Matlab, http://www.mathworks.com/index.html

[9] The H.264/AVC Advanced Video Coding Standard: Overview and Introduction to Fidelity Range Extensions: http://www.fastvdo.com/spie04/spie04-h264OverviewPaper.pdf

[10] ISO/IEC JTC/SC29/WG11 MPEG2003/N6321, December 2003 Waikoloa Hawaii, USA Report of the formal Verification Tests on H.264/AVC

[11] 3GPP TS 25.308 v.7.0.0 - High Speed Downlink Packet Access (HSDPA); Overall description

[12] Elecard Video Quality Estimator: http://www.elecard.com/products/products-pc/professional/video-quest/

[13] Video Quality Studio 0.32: http://www.visualchemia.com/vqstudio/#download

[14] MSU Video Quality Measurement Tool v.1.4: http://www.compression.ru/video/

[15] VQM_pc: http://www.its.bldrdoc.gov/n3/video/VQM_software_description.php

[16] MPEG Industry Forum: http://www.m4if.org/

[17] Wikipedia H.264/MPEG-4 AVC: http://en.wikipedia.org/wiki/H.264

[18] ITU H.263 codec features: http://www.itu.int/rec/T-REC-H.263/en

[19] Video Quality Measurement Techniques: http://www.its.bldrdoc.gov/pub/ntia-rpt/02-392/vqm_techniques_v2.pdf

[20] Graphics and Media Lab: http://graphics.cs.msu.ru/index.html.en

[21] Lomonosov Moscow State University: http://www.msu.ru/en/

[22] http://www.mobilab.unina.it/tesi/Tesi_Michele_Paolillo.pdf

ANNEXES

Annex 1.- Video Quality Evaluation Tool's Matlab source code

All the Matlab code used to develop the PSNR Evaluation tool is shown next. The code has comments so it is better understandable.

```
function varargout = PSNR_TOOL(varargin)
% PSNR_TOOL M-file for PSNR_TOOL.fig
%      PSNR_TOOL, by itself, creates a new PSNR_TOOL or raises the existing
%      singleton*.
%
%      H = PSNR_TOOL returns the handle to a new PSNR_TOOL or the handle to
%      the existing singleton*.
%
%      PSNR_TOOL('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in PSNR_TOOL.M with the given input arguments.
%
%      PSNR_TOOL('Property','Value',...) creates a new PSNR_TOOL or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before Interface_PSNR_OpeningFunction gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to PSNR_TOOL_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Copyright 2002-2003 The MathWorks, Inc.

% Edit the above text to modify the response to help PSNR_TOOL

% Last Modified by GUIDE v2.5 24-Sep-2008 13:17:04

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
   'gui_Singleton',  gui_Singleton, ...
   'gui_OpeningFcn', @PSNR_TOOL_OpeningFcn, ...
   'gui_OutputFcn',  @PSNR_TOOL_OutputFcn, ...
   'gui_LayoutFcn',  [] , ...
   'gui_Callback',   []);
if nargin && ischar(varargin{1})
   gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
   [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
   gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
```

```
% --- Executes just before PSNR_TOOL is made visible.
function PSNR_TOOL_OpeningFcn(hObject, eventdata, handles, varargin)

% Choose default command line output for PSNR_TOOL
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = PSNR_TOOL_OutputFcn(hObject, eventdata, handles)

% Get default command line output from handles structure
varargout{1} = handles.output;

function edit1_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit2_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit4_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit5_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit6_Callback(hObject, eventdata, handles)
```

```
% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit7_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

% --- Executes on button press in ori_folder_butt.
function ori_folder_butt_Callback(hObject, eventdata, handles)
ori_fold = uigetdir('c:\','Select the Original Frames Folder');
ori_fold2 = strcat(ori_fold,'\');
set(handles.edit1,'String',ori_fold2);

% --- Executes on button press in rec_folder_butt.
function rec_folder_butt_Callback(hObject, eventdata, handles)
rec_fold = uigetdir('c:\','Select the Received Frames Folder');
rec_fold2 = strcat(rec_fold,'\');
set(handles.edit2,'String',rec_fold2);

function File_Callback(hObject, eventdata, handles)

function about_menu_Callback(hObject, eventdata, handles)
msgbox('PSNR Evaluation Tool v0.1, developed by Josep A. Canadell','About','help');

function quit_menu_Callback(hObject, eventdata, handles)
button = questdlg('Ready to quit?', ...
    'Exit Dialog','Yes','No','No');
switch button
    case 'Yes',
        disp('Exiting MATLAB');
        quit force;
    case 'No',
        quit cancel;
end

%This is the main code as it is executed when the
%Calculate button is pressed.
%At the end, results are stored in a comma-delimited-file
%which, later, will be processed.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
Folder_Ori=get(handles.edit1,'String');
Folder_Rec=get(handles.edit2,'String');
Frame_Ori_start=get(handles.edit4,'String');
Frame_Ori_start2=str2double(Frame_Ori_start);
```

```
Frame_Ori_stop=get(handles.edit5,'String');
Frame_Ori_stop2=str2double(Frame_Ori_stop);
Frame_Rec_start=get(handles.edit6,'String');
Frame_Rec_start2=str2double(Frame_Rec_start);
Frame_Rec_stop=get(handles.edit7,'String');
Frame_Rec_stop2=str2double(Frame_Rec_stop);

if (strcmp(Folder_Ori,'Introduce the path')==0 && strcmp(Folder_Rec,'Introduce the path')==0)
   Ori_Matrix = Load_frames(Folder_Ori,Frame_Ori_start2, Frame_Ori_stop2);
   Rec_Matrix = Load_frames(Folder_Rec,Frame_Rec_start2, Frame_Rec_stop2);
   if (Frame_Ori_stop2 - Frame_Ori_start2)>(Frame_Rec_stop2 - Frame_Rec_start2)
      Num_Frames2=Frame_Rec_stop2 - Frame_Rec_start2;
   else
      Num_Frames2=Frame_Ori_stop2 - Frame_Ori_start2;
   end
   [Desc_Matrix_ORI, Num_Frames_ORI, Pos_Vector_ORI, Duplicated] = Remove_Duplicated
(Ori_Matrix, Num_Frames2);
   fprintf('%d Original Frames duplicated\n', Duplicated);
   [Desc_Matrix_REC, Num_Frames_REC, Pos_Vector_REC, Duplicated] =
Remove_Duplicated (Rec_Matrix, Num_Frames2);
   fprintf('%d Received Frames duplicated\n', Duplicated);
   clear Ori_Matrix;
   clear Rec_Matrix;
   [v, Num_Frames_total] = Similarity(Desc_Matrix_ORI, Num_Frames_ORI, Pos_Vector_ORI,
Desc_Matrix_REC, Num_Frames_REC, Pos_Vector_REC, Frame_Ori_start2,
Frame_Rec_start2, handles);

   %Results are stored in a CSV file
   A = fopen('Result.csv','w');
   if A==-1
      error('Error: The result file has not been created')
   else
      fprintf(A,'Original Frame,Received Frame,PSNR (dB)\n');
      PSNR_Matrix = zeros(1,Num_Frames_total);
      for i=1:Num_Frames_total;
         if (v(i)~=0)
            PSNR_Matrix(i) = PSNR(Desc_Matrix_ORI(:,:,1,i), Desc_Matrix_REC(:,:,1,v(i)));
            fprintf(A,'%d,%d,%5.2f\n',Pos_Vector_ORI(i)+(Frame_Ori_start2-
1),Pos_Vector_REC(v(i))+(Frame_Rec_start2-1),PSNR_Matrix(i));
         end
      end
      fclose(A);
   end
   ['Finished']
   msgbox('Finished','Calculate','help');
else
   msgbox('Please, provide the frames location','Routes in blank','error');
end

%This function loads the original and the received frames and
%stores them in a Matrix
function [F] = Load_frames(Folder,Frame_start2, Frame_stop2)
for j=Frame_start2:Frame_stop2
   k=Padding(j);
   filename=strcat(Folder,'Frame',k,'.bmp');
   filename
   try
   F(:,:,:,j-Frame_start2+1)=imread(filename,'bmp');
   catch
     msgbox('Error: file cannot be loaded','Error loading images','error');
```

```
      break
    end
end

%This function adds zeros to the counter in order to match
%filenames of the type FILE000X.jpg
function [out] = Padding(j)
if j<10
    out = ['000', int2str(j)];
    return;
elseif j>=10 && j<100
    out = ['00', int2str(j)];
    return;
elseif j>=100 && j<1000
    out = ['0', int2str(j)];
    return;
else
    out = int2str(j);
    return;
end

%This function searches duplicated frames among the ones read at
%the beginning. They are matched by comparing the mean value
%of all the components
function [Desc_Matrix, Num_Frames_total, Pos_Vector, Duplicated] = Remove_Duplicated
(Matrix,Num_Frames2)
i=2;
k=2;
Duplicated=0;
Desc_Matrix(:,:,:,1)=Matrix(:,:,:,1);
Pos_Vector = zeros(1,Num_Frames2);
Pos_Vector(1)=1;
while i<=Num_Frames2;
    mean(mean(mean(((Matrix(:,:,:,i))))));
    mean(mean(mean(((Matrix(:,:,:,i-1))))));
    if (mean(mean(mean(((Matrix(:,:,:,i)))))) == mean(mean(mean(((Matrix(:,:,:,i-1)))))));
        Duplicated=Duplicated+1;
    else
        Desc_Matrix(:,:,:,k)=Matrix(:,:,:,i);
        Pos_Vector(k)=i;
        k=k+1;
    end
    i=i+1;
end
Num_Frames_total=k-1;

%This function searches, among all the received frames, which is the one
%that is more similar to the original one. This is done by comparing
%the mean value of all components of each frame. The pair of original-received
%frames with a lower mean value are considered the same.
function [v1,Num_Frames_total] = Similarity(Desc_Matrix_ORI, Num_Frames_ORI,
Pos_Vector_ORI, Desc_Matrix_REC, Num_Frames_REC, Pos_Vector_REC,
Frame_Ori_start2, Frame_Rec_start2, handles)
min=255;
ref=0;
if (Num_Frames_ORI < Num_Frames_REC)
    Num_Frames_total=Num_Frames_ORI;
else
    Num_Frames_total=Num_Frames_REC;
end
```

```
for i=1:Num_Frames_total;
   for j=1:Num_Frames_total;
      x=mean(mean(mean((Desc_Matrix_ORI(:,:,:,i)-Desc_Matrix_REC(:,:,:,j)))));
      if x<min
         min=x;
         pos=j;
      end
   end
   min=255;
   v1 = zeros(1,Num_Frames_total);
   if (i>1) && (pos == ref)
      ['The original frame ',int2str(Pos_Vector_ORI(i)+(Frame_Ori_start2-1)),' has not
correspondent in reception']
      v1(i)=0;
   else
      v1(i)=pos;
      ref=pos;
      ['The original frame ',int2str(Pos_Vector_ORI(i)+(Frame_Ori_start2-1)),' looks like the
received frame',int2str(Pos_Vector_REC(pos)+(Frame_Rec_start2-1))]
   end
end

%This function calculates the PSNR between the matched original-received
%frames. The result is expressed in decibels.
function [decibels] = PSNR(A, B)
% PURPOSE: To find the PSNR (peak signal-to-noise ratio) between two
%        intensity images A and B, each having values in the interval
%        [0,1]. The answer is in decibels (dB).
%
% SYNOPSIS: PSNR(A,B)
%
% DESCRIPTION: The following is quoted from "Fractal Image Compression",
%           by Yuval Fisher et al.,(Springer Verlag, 1995),
%           section 2.4, "Pixelized Data".
%
%           "...PSNR is used to measure the difference between two
%           images. It is defined as
%
%                     PSNR = 20 * log10(b/rms)
%
%           where b is the largest possible value of the signal
%           (typically 255 or 1), and rms is the root mean square
%           difference between two images. The PSNR is given in
%           decibel units (dB), which measure the ratio of the peak
%           signal and the difference between two images. An increase
%           of 20 dB corresponds to a ten-fold decrease in the rms
%           difference between two images.
%
%           There are many versions of signal-to-noise ratios, but
%           the PSNR is very common in image processing, probably
%           because it gives better-sounding numbers than other
%           measures."
%

if A == B
   error('Images are identical: PSNR has infinite value')
end

max2_A = max(max(A));
max2_B = max(max(B));
```

```
min2_A = min(min(A));
min2_B = min(min(B));

if max2_A > 255 || max2_B > 255 || min2_A < 0 || min2_B < 0
   error('Input matrices must have values in the interval [0,255]')
end

error1 = A - B;
decibels = 20*log10(255/(sqrt(mean(mean(error1.^2)))));
```