Universidade Nova de Lisboa
Faculdade de Ciências e Tecnologia

MSc in Informatics Engineering

Dissertation presented at the Faculty of Sciences and Technology of the New University of Lisbon for the Informatics Engineering Master's degree

# MovieGene

## A multimedia production system using Evolutionary Computation

Nuno Andrade da Cruz Henriques
nach@fct.unl.pt

Supervision:

Luís Correia
Associate Professor FC/UL

Nuno Correia
Assistant Professor FCT/UNL

2005

*To my mother, father and brother in Universe.*

*To Ana for making me a better being.*

*To all those particles that perpetually influence me...*

*...To the Universe for being.*

# Acknowledgements

Marília Andrade for the research stimulus, discipline and rigour; to CH for the Universal acceptance and creativity; and to Manuel for showing and maintaining the right to be different.

Ana for the critique to this Dissertation, care and patience.

Manuela Bentes for the new teaching of mathematics and formal thinking, discipline and friendship.

My supervisors Correia, Luís and Nuno, for their patience in showing me some more photons! :-)

To those friends always caring.

**Abstract**

This dissertation proposes a new approach regarding multimedia documents creation. To support this approach, an evolutionary module and a multimedia production system were conceived, integrated and implemented. A new paradigm is pursued for editing multimedia described documents. Cut and merge of video segments is applied by a Genetic Algorithm. The evolutionary module uses several descriptors, as genetic information, coded in an XML document following MPEG-7 standard.

The documents (video clips) are previously annotated with classifying descriptors. The search for the best clips editing is done accordingly to a fitness function using distance metrics for colour, camera and textual annotation descriptors. The user interaction is considered an additional selection operator doing elimination of less interesting videos. The evolution of the documents may be supervised in real time by the user or be done autonomously until some stop condition is reached, such as the generation number.

# Keywords

Artificial life, evolutionary computation, interactive genetic algorithm, multimedia, hypermedia, audio and video, described video, autonomous and human supervised video editing, mpeg-7.

# Contents

# List of Figures

# List of Tables

# Abbreviations

**A-Life** Artificial Life.

**API** Application Program Interface.

**AV** Audiovisual.

**AVI** Audio Video Interleaved.

**AWT** Abstract Window Toolkit.

**BB** Black Box.

**B.C.** Before Christ.

**BiM** Binary Format for MPEG-7 Description Streams.

**CSS** Cascading Style Sheets.

**DCT** Discrete Cosine Transformation.

**DDL** Description Definition Language.

**DS** Description Scheme.

**DNA** Deoxyribonucleic Acid.

**DVD** Digital Versatile Disc.

**EA** Evolutionary Algorithm.

**EC** Evolutionary Computation.

**ECJ** Evolutionary Computation and Genetic Programming Research System in Java.

**EDL** Edit Decision List.

**EXT3** The Third Extended Native File System of Linux.

**fps** frames per second.

**GA** Genetic Algorithm.

**GP** Genetic Programming.

**GSA** Genetic Segmentation Algorithm.

**GUI** Graphical User Interface.

**HMI** Human-Machine Interface.

**HMMD** Hue, Max, Min, Diff.

**HSV** Hue, Saturation, Value.

**HTTP** HyperText Transfer Protocol.

**IAS** Institute for Advanced Studies.

**IEC** International Engineering Consortium.

**IS** International Standard.

**ISO** International Organisation for Standardisation.

**JDK** Java Development Kit.

**JMF** Java Media Framework.

**JNI** Java Native Interface.

**JPEG** Joint Pictures Experts Group.

**JRMI** Java Remote Method Interface.

**JVM** Java Virtual Machine.

**MPEG** Moving Pictures Experts Group.

**MPEG-7** MPEG-7: Multimedia Content Description Interface (*see Glossary, p. xxiii*).

**PCM** Pulse Coded Modulation.

**PDF** Portable Document Format.

**QoS** Quality of Service.

**RGB** Red, Green, Blue.

**RMI** Remote Method Invocation.

**TV** Television.

**TZD** Time Zone Description.

**UML** Unified Modelling Language.

**VD** Video Descriptor.

**VDN** Video Description.

**VD** Video Descriptor.

**VRML** Virtual Reality Markup Language.

**Web** (*see WWW*).

**WWW** World Wide Web.

**XHTML** eXtensible HyperText Markup Language.

**XM** eXperimentation Model.

**XML** eXtensible Markup Language.

**YCbCr** Luminance, Chrominance Blue, Chrominance Red.

# Glossary

**Allele** ..... 8

One of the alternate forms of a gene specification of a trait. Both forms have the same locus on homologous chromosomes.

**Artificial Life** ..... 1

Synthetic systems which somehow behave like natural living systems.

**Being [Tipler03, chap. IV]** ..... 1

An entity that codifies information preserved by means of natural selection.

**Cellular automaton** ..... 8

(plural: cellular automata) Is a discrete model studied in computability theory and mathematics. It consists of an infinite, regular grid of cells, each in one of a finite number of states. The grid can be in any finite number of dimensions. Time is also discrete, and the state of a cell at time t is a function of the state of a finite number of cells called the neighbourhood at time $t-1$. These neighbours are a selection of cells relative to some specified, and does not change (Though the cell itself may be in its neighbourhood, it is not usually considered a neighbour). Every cell has the same rule for updating, based on the values in this neighbourhood. Each time the rules are applied to the whole grid a new generation is produced.

**Chromosome** ..... 8

The carrier of the genetic information of the cell at its nucleus; carrying genes in a linear order.

**Complex dynamic system** ..... 1

A particle set, usually numerous, at any scale of the Universe that constantly interact and in a way that constrains the ability to predict about the future state of any individually and all simultaneously. A non-linear dynamic system with a high and sensible dependency to its initial state is called chaotic.

**Deoxyribonucleic Acid** ..... 8

A long linear polymer found in the nucleus of a cell and formed from nucleotides and shaped like a double helix; associated with the transmission of genetic information.

**Evolutionary algorithm** ..... 2

A set of well defined steps necessary to reach some problem solution. The process is capable of, and produces, evolution in the natural sense of the term.

**Fitness** ..... 11

The measurement of the chromosome's environmental adaptation.

**Gene** ..... 8

Hereditary unit consisting of a sequence of Deoxyribonucleic Acid (DNA) and occupying a specific position within the genome.

**Genetic Algorithm** ..... 10

An algorithm used to find approximate solutions to difficult-to-solve problems through application of the principles of evolutionary biology to computer science. Genetic algorithms use biologically-derived techniques such as inheritance, mutation, natural selection, and recombination (or crossover). Genetic algorithms are a particular class of evolutionary algorithms based on information codification using gene emulated structures.

**Genome** ..... 8

The full complement of chromosomes, and extrachromosomal DNA coding for cellular proteins, contained within each cell of a given species.

**Genotype** ..... 8

The particular alleles at specified loci present in an organism; blueprint for the Phenotype.

**Hypermedia** ..... 2

A media in which related, possibly of several different formats, items of information are connected and accessible. The presentation of these documents may be like a unique and structured document with links to follow other nodes, such as eXtensible HyperText Markup Language (XHTML) pages, or imaging devices with mixed reality merging textual information linked with a geographical coordinate or visual reference.

**L-System** ..... 39

An L-System is an automaton designed by Aristid Lindenmayer in 1968 to model cell development. Cells are represented by symbols and cell subdivision is modelled by replacing these symbols with strings of symbols.

**Life [Tipler03, chap. IV]** ..... 1

Information preserved by means of natural selection.

**Locus** ..... 8

The specific site of a particular gene on its chromosome.

**MPEG-7: Multimedia Content Description Interface** ..... 3

ISO/IEC standard 15938: Multimedia Content Description Interface. An extensive framework to describe multimedia content.

**Panspermia** ..... 38

Is the name for the theory that life exists and is distributed throughout the universe in the form of germs or spores. `http://www.genarts.com/karl/panspermia.html`.

**Phenotype** ..... 8

What an organism looks like as a consequence of its genotype and life development; two organisms with the same genotype can have different phenotypes; physical expression of genetic variation due to environmental interactions.

# Chapter 1

# Introduction

This first chapter introduces the thesis subject. It reveals the original motivation and the main objectives. Also, the scientific context, from which ideas and development took place, is described. It briefly depicts the methodology followed and the work plan. Next, it exposes the main concepts for the adopted technology and, at the end, it explains the document structure.

## 1.1 Motivation

While searching for a thesis subject I have found Artificial Life (A-Life) as a field of interest for further exploration. A field that poses questions about the meaning of life and the Universe, from the particles chaos of the complex dynamic systems to intelligent life.

Creation of synthetic beings is one of several ways to study A-Life, any piece of information may be an individual. Visualisation of the beings is a must for any simulated ecosystem, therefore Multimedia Computation arose as a very interesting joining field.

The growth of multimedia documents production along with their availability either in small sized groups' networks, such as a television corporations, or big sized groups, such as the Web community, has motivated the search for new techniques of video annotation and description. At a lower level, traditional algorithms can automatically annotate or interpret simple descriptions, e.g., colour histograms, but, at a higher level, more complex semantic information must be aggregated and processed by algorithms in a feasible time, aiding human agents on the indexing, browsing, retrieval and editing of the multimedia documents.

Evolutionary algorithms, specifically genetic algorithms, are capable of search and optimisation in a huge search space of possible solutions and do the task in a feasible time [Goldberg89]. So, naturally, a Genetic Algorithm (GA) is chosen for the connection with multimedia computation needed for the production of multimedia documents.

## 1.2 Context

This thesis combines two computer science areas, Artificial Life and Multimedia Computation. Two supervisors are chosen from each scientific field and a natural mutual understanding arose between the leaders of the two research groups: Luís Correia for the GruVA – Artificial Life Group from Faculty of Sciences of the University of Lisbon (FC/UL); Nuno Correia for the IMG – Interactive Multimedia Group from the Faculty of Sciences and Technology of the New University of Lisbon (FCT/UNL). The scientific supervision is a joint one, but, naturally, Luís Correia helps on the evolutionary subjects while Nuno Correia the multimedia ones.

The facilities used are from two different places where each group is located: the LabMAg[1] – Laboratory of Agent Modelling at the Department of Informatics of the FC/UL where the GruVA is hosted; the CITI[2] – Informatics and Information Technology Centre at Department of Informatics of the FCT/UNL where the IMG is located.

This work includes both the evolutionary module development and the whole multimedia production system architecture design and implementation for testing an initial idea proposed by Nuno Correia, Jônatas Manzolli and Teresa Chambel [Correia02].

## 1.3 Objectives

The broader goal is to research new ways of editing and producing multimedia and hypermedia documents. The objective is to develop an evolutionary module to integrate in a multimedia production system in order to do automatic search and choice of video clips, interpret the descriptions, process and evolve video described segments to edit in a new multimedia document.

It foresees the ability to surpass temporal restriction barriers, such as the search and editing of a video clip among a huge repository of videos, all done by a human agent with

[1] `http://labmag.di.fc.ul.pt/`

[2] `http://www.di.fct.unl.pt/citi/`

the GA aid on a feasible time.

The evolutionary module, implemented as described below, intends to apply genetic algorithms based transformations, providing the evolution of previously annotated multimedia (video) documents with classifying descriptors using the MPEG-7: Multimedia Content Description Interface (*see Glossary, p. xxiii*) (MPEG-7) standard.

The resulting multimedia documents are presented using a Web interface, an XHTML page with a MovieGene's client application called MovieGoal. The user actions may influence the evolutionary process as a selection operator. The traditional rules for film editing will be used. A new paradigm of multimedia authoring is pursued as a result of several research areas.

## 1.4 Methodology and plan

The work's plan had three phases, being the Dissertation written simultaneously with the prototype development:

1. Requirements specification: development platform, libraries and environment. Annotation descriptors for the multimedia documents. This is the core phase for further refinement of the video editing semi-automatic process using a GA. Duration: 1 to 2 months.

2. Evolutionary module development and implementation, integration within MovieGene's system and MovieGoal's application and interface. System and unit testing in a loop process until reaching a satisfying result. Duration: 6 to 8 months.

3. Final stage for the Dissertation writing. Duration: 1 to 2 months.

## 1.5 Technology

The technology used is the most free, reliable and accessible: a Web based interface over the Internet distributed environment. Therefore the user may be anyone with access to a Web interface device, e.g., computer with a Linux operating system, and a XHTML+CSS and Java aware browser.

Specifically, Freeware and mostly also Free Software languages and tools are used; the

development platform with GNU[3]/Linux[4] distributed by Debian[5] is the main one. Java Software Development Kit and its Runtime Environment is another platform used, over Linux (or any of the several Operating Systems that support Java), for executing the prototype.

The persistence is handled through the local platform file system, in Linux is The Third Extended Native File System of Linux (EXT3). In the future, we foresee a relational database, e.g., PostgreSQL [PostgreSQL03], as a repository of all the data relating users, multimedia documents and sessions.

The programming languages used for the Web interface are XHTML [W3C03] + CSS [W3C04]; the client - MovieGoal - implementing the application logic for interaction between the MovieGene system and the user, is coded with Java [Sun04b, Sun04a] (using several add-on libraries, such as the Java Media Framework (JMF)).

The multimedia documents (video clips) are annotated and described using the MPEG-7 [ISO04] standard, well known and accepted by the multimedia community and industry. Therefore future portability for comparison with similar work will be easy to do if committing to the same standards. A very simple *ad hoc* library is developed for parsing the MPEG-7 eXtensible Markup Language (XML) video descriptors' files.

Java language and JMF Application Program Interface (API) from Sun Microsystems are used for the development of the video operators module.

The evolutionary module, inside MovieGene, is developed using Java with all the advantages of the existent libraries for evolutionary computation, such as Evolutionary Computation and Genetic Programming Research System in Java (ECJ)[6], the one chosen. An acknowledgement is made to Sean Luke[7], author of the ECJ 12, for the use of his work free of restrictions.

## 1.6 Structure

This first chapter is the introduction of the Dissertation and includes, along with this section, the Motivation, Context, Objectives, Methodology and plan, and the Technology

[3] http://www.gnu.org/
[4] http://www.linux.org/
[5] http://www.debian.org/
[6] http://cs.gmu.edu/~eclab/projects/ecj/
[7] http://cs.gmu.edu/~sean/

used.

The second chapter presents the Related Work with a brief description of paradigms and standards, a more broad description of some essential concepts related with MovieGene, such as GA, MPEG-7 and Video Description (VDN), and evolutionary multimedia.

The third chapter presents the fundamental concepts of evolutionary multimedia documents with focus on video clips. It describes the conceptual options that guide this work, also with some possible solutions and finally the chosen solution.

The fourth chapter describes the implementation, both of the evolutionary module and all the necessary parts complementing the MovieGene's architecture. In this chapter it is included the Human-Machine Interface (HMI) implementation of the prototype, also the restrictions and, at the end of the chapter, the system testing and results.

Finally, fifth chapter concludes the Dissertation with some Discussion and Future Work perspectives.

# Chapter 2

# Related work

This related work chapter includes a broad range of related subjects. It introduces the quest of Charles Darwin in the search for the origin of species, explains elementary genetics, travels through important artificial life historical facts and how a genetic algorithm conceptually works. Then, an introduction is made to media descriptors, the MPEG-7 working team motivation for the international standard specification of multimedia, audio and visual, descriptors, description schemes and description definition language is also described. Some examples from articles are presented and summarised. At the end of the chapter, systems that combine the use of multimedia and evolutionary computation are explained.

## 2.1 From Darwin to genetic algorithms

> *From so simple a beginning*
> *endless forms most beautiful and most wonderful*
> *have been, and are being, evolved.*
>
> Charles Darwin, *On Natural Selection* [Darwin04]

Charles Darwin visited the Galápagos Islands in 1835 and his ideas on natural selection were inspired by the unusual varieties of wildlife there. The isolation of these islands caused a rare example of a relatively independent evolutionary process which he was able to observe and record improving his research work. This visit was a decisive influence for the writing of his book called "On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life" (1859).

Biological living organisms consist of cells. In each cell there is the same set of chromosomes. Chromosomes are strings of DNA and serve as a model for the whole organism. A chromosome consists of genes, blocks of DNA. Each gene encodes a particular protein. Basically, it can be said that each Gene encodes a trait (figure 2.1), such as the colour of the eyes. Each possible setting for a trait, e.g., blue or brown eye colour, is called Allele. Each Gene has its own position, called locus, in the Chromosome.

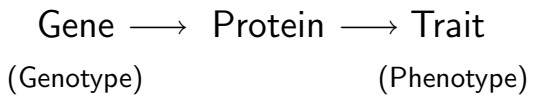


Figure 2.1: From Genotype to Phenotype: a living organism getting its traits.

The complete set of genetic material (chromosomes) is called Genome. A particular set of Genes in a Genome is called Genotype. The Genotype is, with later development after birth, the base for the organism's Phenotype, its characteristics, such as eye colour, hair colour and shape.

Biological evolution can be difficult to study because we have just one large example of life based on the genetic system of DNA and it progresses very slowly from our human short life point of view – life on earth has taken nearly four thousand million years to evolve. It has been impractical to perform experiments such as starting evolution over from scratch, or investigating alternative genetic systems. However, using the power of computers, it is now possible to simulate simplified 3D organisms in evolutionary systems, which can be observed from start to finish and run multiple times [Sims97].

Artificial life studies started some fifty years ago mainly by two men: John von Neumann (1903–1957) and Alan M. Turing (1912–1954). Both paths crossed during 1936 through 1938 when Turing was a graduate student in the Department of Mathematics at Princeton and did his dissertation under Alonzo Church supervision. Von Neumann invited Turing to stay on at the Institute as his assistant but he declined and returned to Cambridge. The Turing's 1934 paper publication "On Computable Numbers with an Application to the Entscheidungs-problem" which involved the concepts of logical design and the universal machine indicates that von Neumann knew of Turing's ideas, though whether he applied them to the design of the Institute for Advanced Studies (IAS) Machine ten years later is questionable.

Von Neumann created the field of Cellular Automata without computers, constructing the first examples of self-replicating automata with pencil and graph paper. The term von Neumann machine also refers to self-replicating machines. Von Neumann proved that

the most effective way large-scale mining operations, such as mining an entire moon or asteroid belt, can be accomplished through the use of self-replicating machines, to take advantage of the exponential growth of such mechanisms [Wikipedia05, Levy93].

During the final years of his life Turing was working on Morphogenesis, what would now be called A-Life. He used the Ferranti Mark I computer belonging to the Manchester University Computing Machine Laboratory to simulate a chemical mechanism by which the genes of a zygote may determine the anatomical structure of the resulting animal or plant [Copeland04, Hodges04].

We are now facing a set of complexness problems raised from the study of complex dynamic systems. Computer Science is developing new programming methods in order to respond to these new requests. Environments with large data diversity and variability may be described as populations of information elements. A new kind of algorithms may be developed to compute these populations in an evolutionary process. Thus, a new computational paradigm has been born known as Evolutionary Computation (EC).

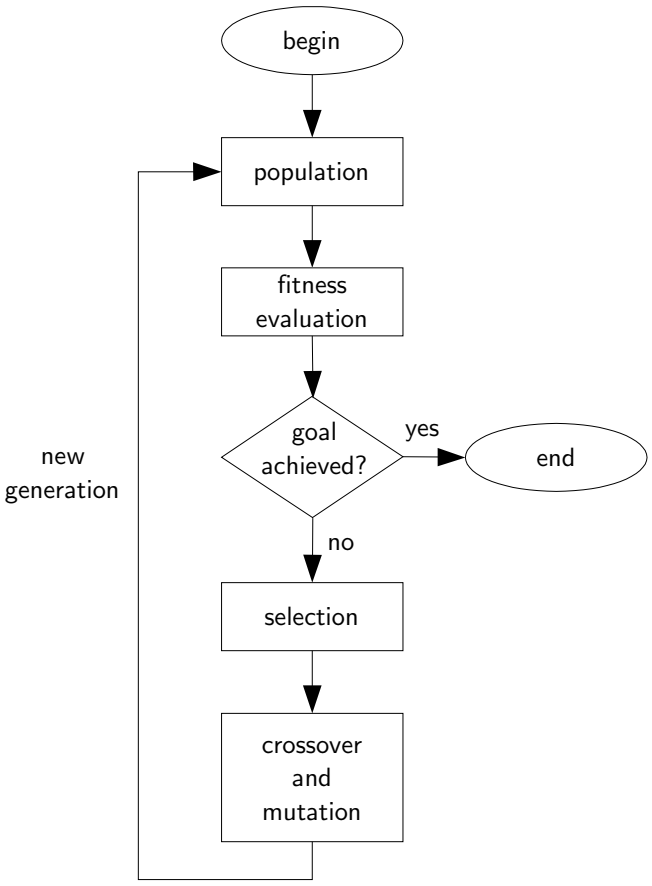


Figure 2.2: GA basic process flow.

The EC approach brought new algorithms based on genetics elementary concepts. Genetic algorithms were invented by John Holland and developed by him and his students and colleagues. This lead to Holland's book "Adaptation in Natural and Artificial Systems" published in 1975. The genetic algorithms have been used with success in search, optimisation and machine learning problems within several fields of science research and practical implementations, such as musical composition, stock market predictions, digital painting, weather forecast, and image recognition [Goldberg89].

In 1992 John Koza used a GA to evolve programs to perform tasks; Genetic Programming (GP) arose to support this new way of programming solutions for programs, mathematical functions or any tree encoding representable problem [Koza92].

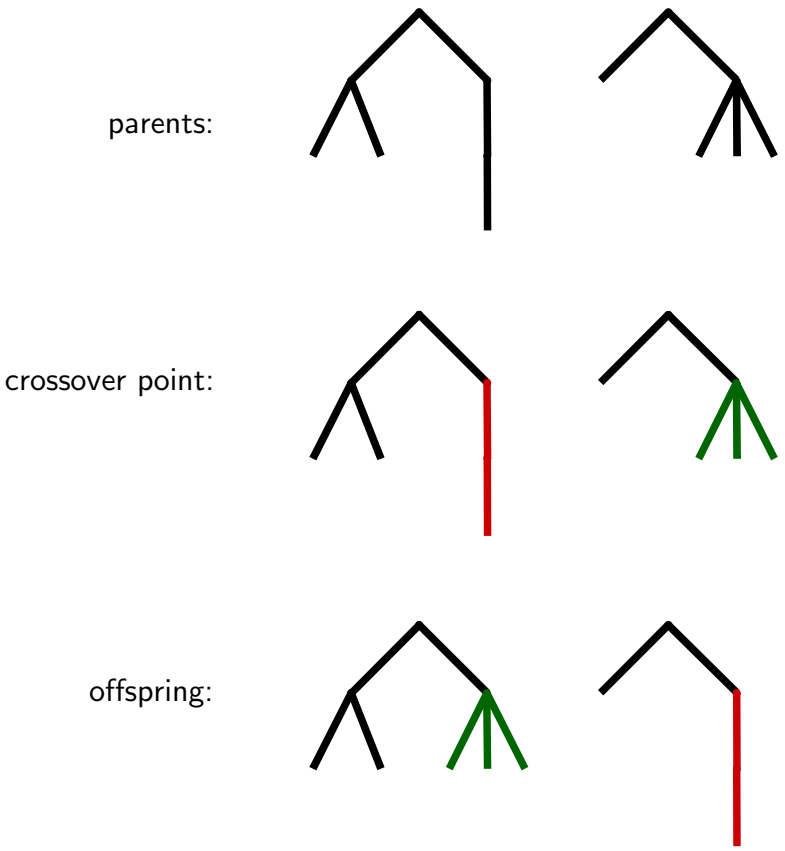


Figure 2.3: GP tree crossover example.

Genetic algorithms mimic natural evolution. Instead of evolving DNA and organisms, a GA evolves strings of symbols or hierarchies (trees). In all cases, two structures are selected to be parents. Each structure is, usually, broken into two parts at random, and one part from each parent is recombined into a new child. Crossover, also known as mating, is different for strings and trees. With strings of symbols, a random point in

each string is selected and the right-hand symbols from one parent are mated with the left-hand symbols from the other parent (figure 2.4). With trees, a sub-tree is selected in each parent. Then the selected sub-trees are exchanged (figure 2.3).

| | | |
|---|---|---|
| parents: | abcde | fghij |
| crossover point: | abc*de* | fgh*ij* |
| offspring: | abc*ij* | fgh*de* |

Figure 2.4: GA one point crossover example.

The simplest process flow (figure 2.2) of a GA is based on a few elementary concepts: Evolutionary loop, Chromosome, Fitness function, Selection (for mating and reproduction), Crossover and Mutation.

An Evolutionary loop starts by setting the initial population of chromosomes (individuals) then the Fitness function is applied to every single Chromosome evaluating the best ones among all. The next step is to test for the predefined goal achievement: on "yes" the loop ends, on "no" implies loop next step. Each new generation is the offspring of the previous one by selection of the fittest individuals (parents) for copy (reproduction) with mating by the application of Crossover and then Mutation of the resulting genetic material.

The Crossover operation produces two new chromosomes exchanging its contents with two parts swapping randomly at a breaking point. This operation may not happen causing the new offspring to be the parent's Chromosome exact copy.

The Mutation may or may not happen, i.e., with a higher or lower mutation probability which modifies one allele within the Chromosome.

Because Crossover operation is used to vary the Chromosome programming, several techniques exist besides One Point Crossover (figure 2.4), such as Two Point Crossover (figure 2.5), Cut and Splice (figure 2.6), Uniform Crossover and Half Uniform Crossover [Goldberg89, Wikipedia05, Obitko04].

**Two Point Crossover** Calls for two points to be selected on the parent organism strings. Everything between the two points, a chromosome segment, is swapped between the parent organisms, rendering two child organisms.

**Cut and Splice** This approach may result in a change in length of the children strings.

| | | |
|---|---|---|
| parents: | abcde | fghij |
| crossover points: | a*bcd*e | f*ghi*j |
| offspring: | a*ghi*e | f*bcd*j |

Figure 2.5: GA two point crossover example.

the reason for this difference is that each parent string has a separate choice of crossover point.

| | | |
|---|---|---|
| parents: | abcde | fghij |
| crossover points: | a*bcde* | fgh*ij* |
| offspring: | a*ij* | fgh*bcde* |

Figure 2.6: GA "cut and splice" example.

**Uniform Crossover and Half Uniform Crossover** In these schemes the two parents are combined to produce two new offspring. In uniform crossover scheme individual bits in the string are exchanged between two parents. The bits are swapped with a fixed probability, typically 0.5.

In the half uniform crossover scheme, exactly half of the non-matching bits are swapped. Thus, first the Hamming distance (the number of differing bits) is calculated. This number is divided by two. The resulting number represents how many of the bits that do not match between the two parents will be swapped.

Besides these basic work flow and useful operators, genetic algorithms have several options for fine tuning each and every problem to solve. The Selection operator can range from the simple "blind" picking of the $n$ first fittest individuals, to more creative solutions, such as Multi Selection. The first strategy selects the better parents in the hope that they will produce better offspring. The last strategy improves selection with complementary strategies and even the use of several selection methods randomly picked for each evolutionary iteration (new generation) – Multi Selection.

Elitism is not a method of selection but a characteristic able to be composed with any method. Due to its importance and usefulness, it is described below.

**Elitism** When creating a new population by crossover and mutation, a big chance exists in loosing the best chromosomes – best individual.

Elitism is the name of the method that first copies the best chromosome (or few best chromosomes) to the new population. The rest of the population is constructed in ways described below. Elitism can rapidly increase the performance of GA, because it prevents a loss of the best found solution. But, can also be nasty to the diversity as it always looses the less fittest.

Several of the more popular and interesting selection methods were chosen to be analysed, used and are described below.

**Fitness Proportionate selection** In this method, also known as Roulette-Wheel selection, possible solutions or chromosomes are assigned a fitness by the fitness function. In fitness proportionate selection, this fitness level is used to associate a probability of selection with each individual chromosome. While candidate solutions with a higher fitness will be less likely to be eliminated, there is still a chance that they may be. Contrast this with a less sophisticated selection algorithm, such as truncation selection, which will eliminate a fixed percentage of the weakest candidates. With fitness proportionate selection there is a chance some weaker solutions may survive the selection process; this is an advantage, as though a solution may be weak, it may include some component which could prove useful following the recombination process.

The analogy to a roulette wheel can be envisaged by imagining a roulette wheel in which each candidate solution represents a pocket on the wheel; the size of the pockets are proportionate to the probability of selection of the solution. Selecting $n$ chromosomes from the population is equivalent to playing $n$ games on the roulette wheel, as each candidate is drawn independently.

**Greedy over-selection** Individuals are selected based on their performance but this method biases selection towards the highest performers. This selection method first divides individuals into two groups: the "good" ("top") group, and the "bad" ("bottom") group. The best, e.g. 20%, top percent of individuals in the population go into the good group. The rest, e.g. 80%, go into the "bad" group. With a certain probability, usually one half, an individual will be picked out of the "good" group. Once we have determined which group the individual will be selected from, the individual is picked using fitness proportionate selection in that group, that is, the likelihood it is picked is proportionate to his fitness relative to the fitness of others in its group.

**Rank selection** The previous type of selection (Fitness Proportionate) will have problems when big differences arise between the fitness values. For example, if the

best chromosome fitness is 90% of the sum of all fitness values then the other chromosomes will have very few chances to be selected. Rank selection ranks the population first and then every chromosome receives fitness value determined by this ranking. The worst will have the fitness 1, the second worst 2 and so on. The best will have fitness $n$ (number of chromosomes in population).

Now all the chromosomes have a chance to be selected. However this method can lead to slower convergence, because the best chromosomes do not differ so much from other ones.

**Tournament selection** Runs a tournament among a few individuals and selects the winner (the one with the best fitness) for crossover.

Selection pressure can be easily adjusted by changing the tournament size. If the tournament size is higher, weak individuals have a smaller chance to be selected.

Deterministic tournament selection selects the best individual in any tournament. A 1-way tournament selection is equivalent to random selection. The chosen individual can, optionally, be removed from the population to avoid duplicates, otherwise individuals can be selected more than once for the next generation.

Tournament selection has several benefits: it is efficient to code, works on parallel architectures and allows the selection pressure to be easily adjusted.

A very interesting work for GAs design using the World Wide Web (WWW) as an accessible interface for every human agent is described in [Smith97]. This article presents a tool for the design of both sequential and distributed (or parallel using islands of sub-populations and migration of individuals between them) GAs using the Java language and also the Virtual Reality Markup Language (VRML) to graphically aid the visualisation of the solution for a specific problem of motion planning of an autonomous underwater vehicle.

The main user interface (figure 2.7) for the GA construction is very complete and permits a design–simulation loop of continuous tuning. Also, the Live3D plug-in is used for the 3D graphical path view of the autonomous vehicle working example (figure 2.8).

This tool has achieved a nice merging of a Java based evolutionary engine for GAs construction, an interface on the Web and an additional browser media plug-in for 3D motion representation.

From: `http://www.ics.hawaii.edu/~sugihara/research/ga-updates.html`

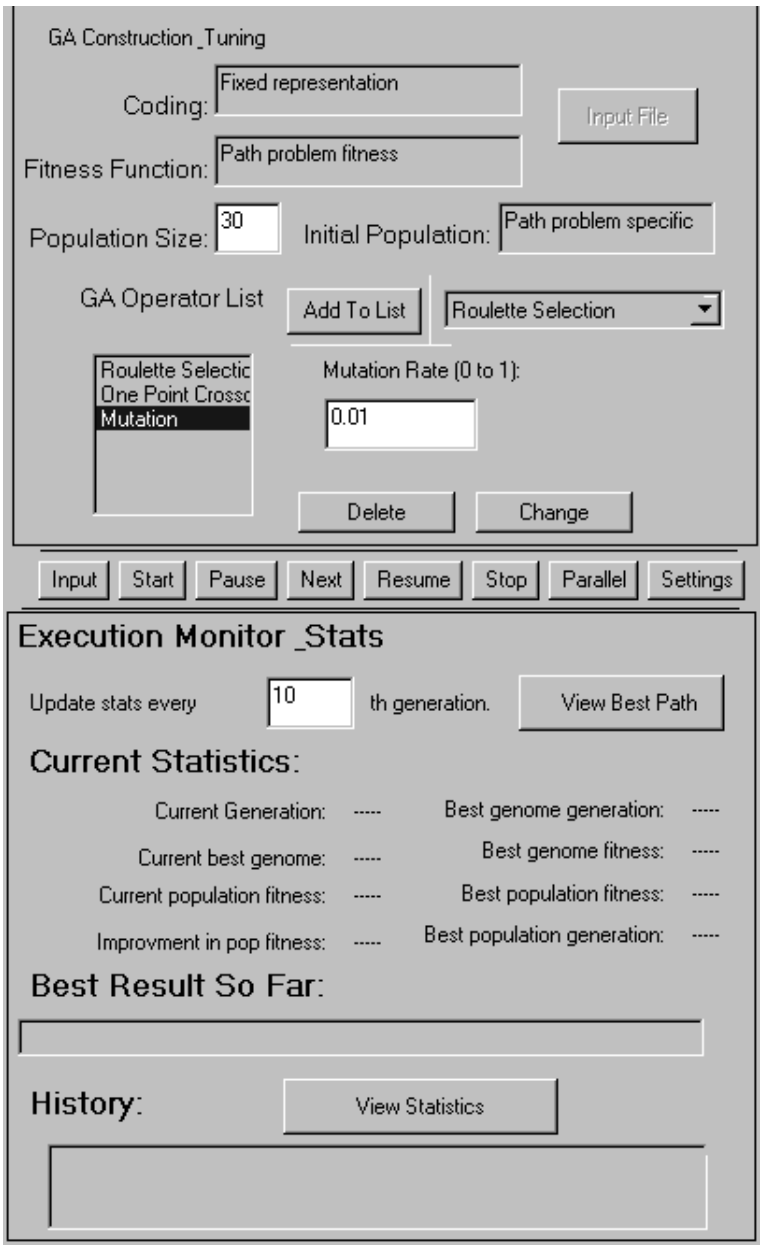


Figure 2.7: GA tool on the Web: main user interface [Smith97].

Another interesting work is the Java based learning environment and visualisation tool, also Web based using Java Applets, described in [Obitko04]. Here, two main objectives are fulfilled: one is the Web site as a tutorial for the GAs learning; the other is the application (Applet) that aids the user on the construction of a GA to solve a specific problem. Several typical problems, such as the Travelling salesman problem are exemplified with an explanation accompanying an Applet so the user can interact and learn by experimenting different parameters. Selection method for crossover and mutation probabilities are possible parameters to change.

From: `http://www.ics.hawaii.edu/~sugihara/research/ga-updates.html`

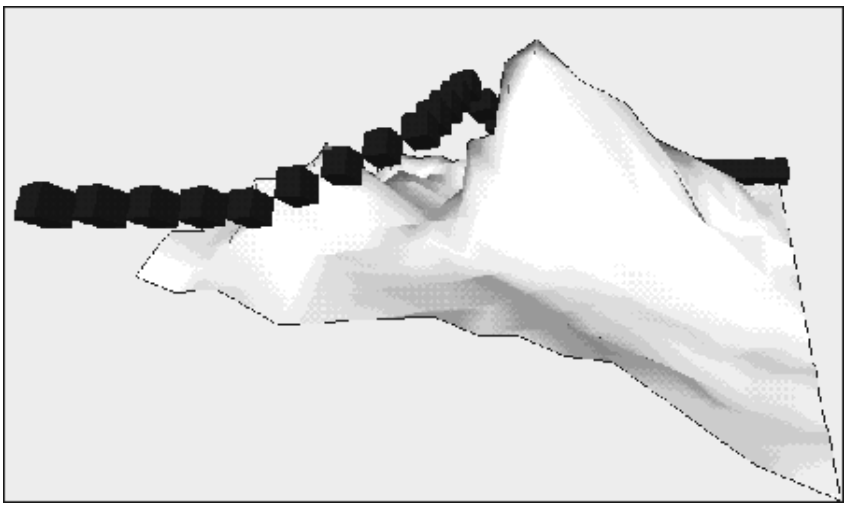

Figure 2.8: GA tool on the Web: 3D path VRML interface [Smith97].

This work by Obitko is a very important resource as an introduction to GAs, due to the broad range of specific subjects, accuracy of the information and pedagogic language and interface (figure 2.9).

From: `http://cs.felk.cvut.cz/~xobitko/ga/index.html`

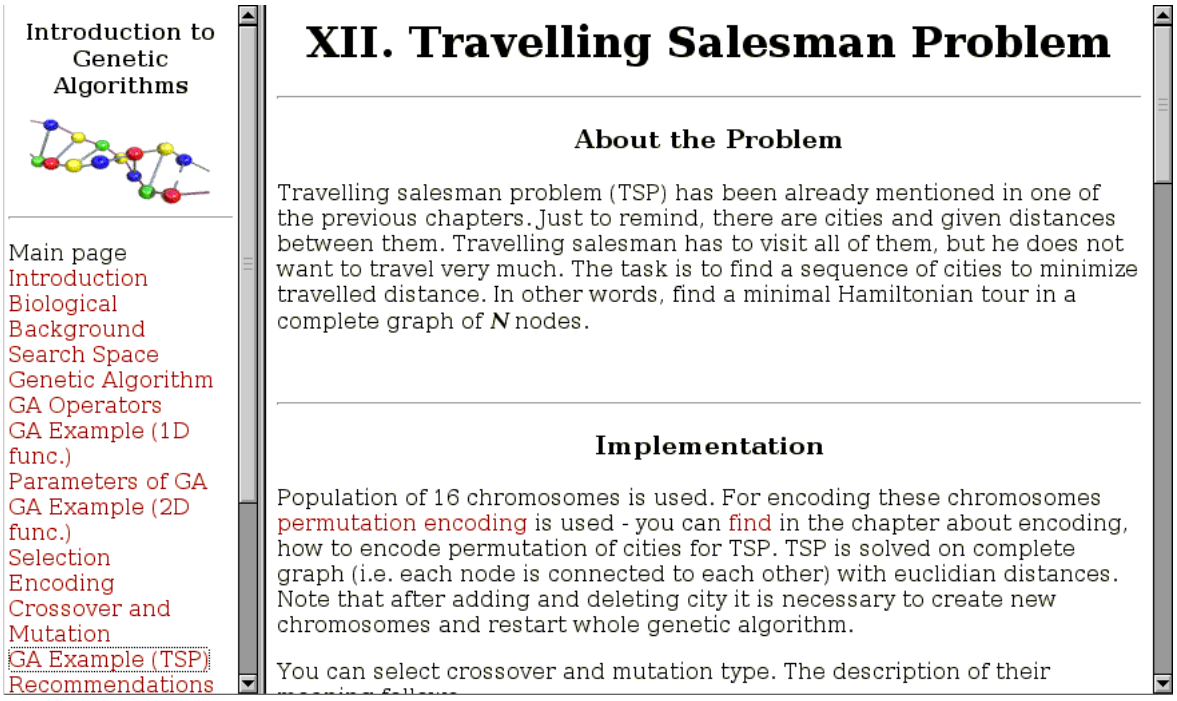


Figure 2.9: GA learning tool on the Web [Obitko04].

## 2.2 Cinema and video editing

The several shots taken for film and video clips production need composition. The segments combining method in a sequence with transitions between them is named editing. The editing styles are usually influenced and restricted by historical moments, technological developments, or national schools. The three main styles are [Prunes05]:

**Continuity editing** A cutting method to maintain continuous and clear narrative action. Continuity editing relies upon matching screen direction, position, and temporal relations from shot to shot. The film supports the viewer's assumption that space and time are contiguous between successive shots.

**Elliptical editing** Shot transitions that omit parts of an event, causing an ellipses in plot and story duration. Elliptical editing is not confined to the same place or time. It can be used, for example, to drive the viewer through discrete time steps (from minutes to years leaping) of someone's life.

**Montage** An approach to editing developed by the Soviet film makers of the 1920s such as Pudovkin, Vertov and Eisenstein; it emphasises dynamic, often discontinuous, relationships between shots and the juxtaposition of images to create ideas not present in either shot by itself. Soviet Montage influenced film makers around the world. In a famous sequence from "The Godfather" (Francis Ford Coppola, USA, 1973), shots of Michael attending his son's baptism are inter cut with the brutal killings of his rivals. The word montage is also a synonym for editing.

Cinematic styles can be divided in transitions, matches and duration. These are sets of characteristics that each director and editor use for the creation of a narrative.

The several styles of transitions are as follows [Prunes05]:

**Cheat cut** A cut which intents to show continuous time and space from shot to shot but which actually mismatches the position of figures or objects in the scene.

**Cross cutting or Parallel editing** Editing that alternates shots of two or more lines of action occurring in different places, usually simultaneously. The two actions are therefore linked, associating the characters from both lines of action.

**Cut in, cut away** An instantaneous shift from a distant framing to a closer view of some portion of the same space, and vice versa.

**Dissolve** A transition between two shots during which the first image gradually disappears while the second image gradually appears; for a moment the two images blend in superimposition.

**Iris** A round, moving mask that can close down to end a scene (iris-out) or emphasise a detail, or it can open to begin a scene (iris-in) or to reveal more space around a detail. For instance, in this scene (figure 2.10) from "Neighbours" (Buster Keaton, 1920), the iris is used with the comic effect of gradually revealing that the female protagonist is 1) ready for her wedding and 2) ready for her not-too-luxurious wedding.

Figure 2.10: Iris cut transition.

**Jump cut** An elliptical cut that appears to be an interruption of a single shot. Either the figures seem to change instantly against a constant background, or the background changes instantly while the figures remain constant.

**Establishing shot / Reestablishing shot** A shot, usually involving a distant framing, that shows the spatial relations among the important figures, objects, and setting in a scene. Usually, the first few shots in a scene are establishing shots, as they introduces us to a location and the space relationships inside it.

**Shot / Reverse shot** Two or more shots edited together that alternate characters,

typically in a conversation situation. In continuity editing, characters in one framing usually look left, in the other framing, right. Over-the-shoulder framings are common in shot / reverse shot editing.

**Superimposition** The exposure of more than one image on the same film strip. Unlike a dissolve, a superimposition does not signify a transition from one scene to another.

**Wipe** A transition between shots in which a line passes across the screen, eliminating the first shot as it goes and replacing it with the next one. A very dynamic and noticeable transition, it is usually employed in action or adventure films. It often suggest a brief temporal ellipsis and a direct connection between the two images. In this example (figure 2.11) from Kurosawa's "Seven Samurai" (Sichinin No Samurai, Japan, 1954), the old man's words are immediately corroborated by the wandering, destitute samurai coming into town.

Figure 2.11: Wipe cut transition.

The three editing matches are as follows [Prunes05]:

**Eyeline match** A cut obeying the axis of action principle, in which the first shot shows a person off in one direction and the second shows a nearby space containing what he or she sees. If the person looks left, the following shot should imply that the looker is off screen right.

**Graphic match** Two successive shots joined so as to create a strong similarity of compositional elements, e.g., colour and shape. Used in transparent continuity styles to smooth the transition between two shots.

**Match on action** A cut which splices two different views of the same action together at the same moment in the movement, making it seem to continue uninterrupted.

Quite logically, these characteristics make it one of the most common transitions in the continuity style.

The three editing duration techniques are as follows [Prunes05]:

**Long take or Plan-sequence** A shot that continues for an unusually lengthy time before the transition to the next shot. The average length per shot differs greatly for different times and places, but most contemporary films tend to have faster editing rates. In general lines, any shot above one minute can be considered a long take.

**Overlapping editing** Cuts that repeat part or all of an action, thus expanding its viewing time and plot duration. Most commonly associated with experimental film making, due to its temporally disconcerting and purely graphic nature, it is also featured in films in which action and movement take precedence over plot and dialogue, such as sports documentaries, musicals and martial arts. Overlapping editing is a common characteristic of the frenzied Hong Kong action films of the 80s and 90s. When director John Woo moved to Hollywood, he tried to incorporate some of that style into mainstream action films, such as "Mission: Impossible 2" (figure 2.12).

Figure 2.12: Overlapping editing.

**Rhythm** The perceived rate and regularity of sounds, series of shots, and movements within the shots. Rhythmic factors include beat (or pulse), accent (or stress), and tempo (or pace). Rhythm is one of the essential features of a film, for it decisively contributes to its mood and overall impression on the spectator. It is also one of the most complex to analyse, since it is achieved through the combination of mise-en-scene, cinematography, sound and editing. Indeed, rhythm can be understood as the final balance of all the elements of a film. Rhythm can radically alter the treatment of a similar scene.

### Soft Cinema

All the above techniques for film editing are well known and used on film making. Besides the, more or less, traditional approach, there are several artistic projects proposing new ways of making films. Soft Cinema is a very interesting one [Manovich05]:

> Soft(ware) Cinema is a dynamic computer-driven media installation. The viewers are presented with an infinite series of narrative films constructed on the fly by the custom software. Using the systems of rules defined by the author, the software decides what appears on the screen, where, and in which sequence; it also chooses music tracks. The elements are chosen from a media database which at present contains 4 hours of video and animation, 3 hours of voice over narration, and 5 hours of music.

From: `http://www.manovich.net/cinema_future/toc.htm`

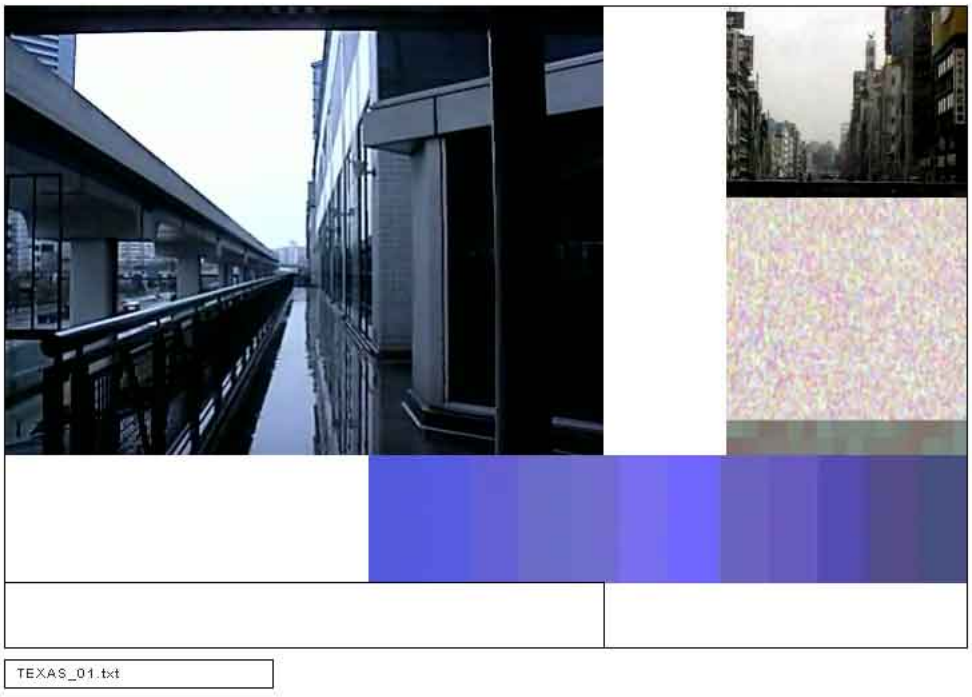


Figure 2.13: Soft Cinema example.

Soft Cinema database contains a collection of short movies in different styles. Each video clip follows Dogma 95[1] rules: it was shot in continuous takes without edits using a hand-held camera. Some of the clips are simulated, i.e., a still image was animated to look like a video shot on location. The length of each movie corresponds to the average length of a music track (3 to 7 minutes).

Typically, a story has been divided into a number of sequential parts, each part becoming a short movie. At the beginning of each segment, the software generates a new screen layout, which can be comprised of two to six different windows. The software also selects which video clips and animations will play in these windows and in what order. This process is repeated for each part of the narrative. Following the same modular logic, different voices are used for different parts of each story.

The small window that always appears in the bottom left corner identifies the part of the story currently playing (for instance, texas_01.txt). A narrow, horizontal window presents scrolling sentences selected from the same story segment (figures 2.13 and 2.14).

From: `http://www.manovich.net/cinema_future/toc.htm`


Figure 2.14: Soft Cinema layout explanation.

## 2.3 Multimedia and video description

Printed document's annotation has a long and solid tradition and is being researched, developed and already in use for digital documents, despite its youth, such as Web documents in XHTML and Portable Document Format (PDF) formats.

[1] `http://www.dogme95.dk/` A movement in film making developed in 1995 by the danish directors Lars von Trier, Thomas Vinterberg, Kristian Levring, and Søren Kragh-Jacobsen.

Until now, the annotation effort has mainly been made for static information, e.g., text and still images, while temporal based media, such as video segments (figure 2.15), was a paradigm left for specific academic research or proprietary *ad hoc* enterprise solutions in order to describe a subset of the search space solving a few problems and producing a commercial asset.

Technology evolution promotes cultural changes with more or less impact in society. Multimedia production devices are now more accessible than ever before which boosts the multimedia production itself. Joining this reality with the easier content support and distribution, such as DVDs, mobile devices and the Web, results in a huge quantity in numbers and bytes of available and exchangeable multimedia documents. All these multimedia documents, such as video and audio clips, continuously pressed all agents, producers and consumers, for the urgency of a solution.

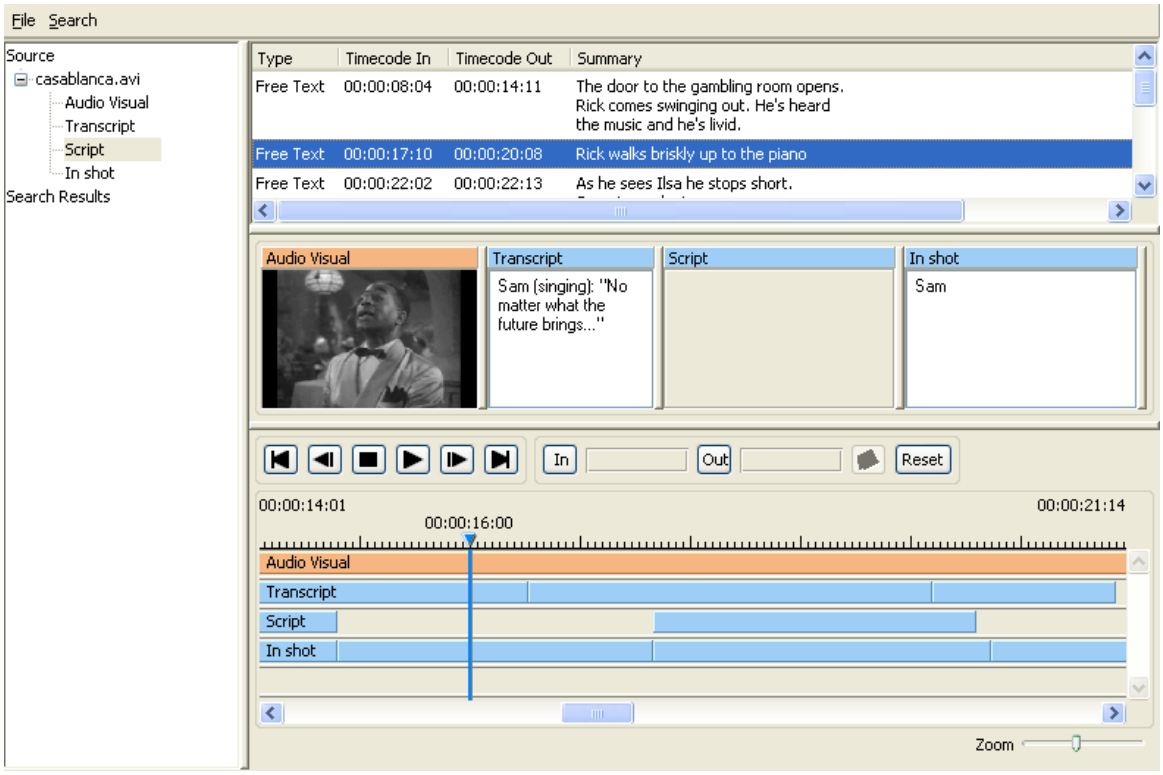


Figure 2.15: The VAnnotator application [Costa02].

A relevant effort is being made and several trends are being followed, on the multimedia research and development groups around the world, towards efficient techniques for video content analysis and retrieval. Several algorithms are being developed for video structure parsing, content representation and content based abstraction. Also, application tools are being produced for content based video indexing, retrieval and browsing. These techniques

and algorithms focus on automatic processes instead of human agent annotation. This is needed to comply with the formal (objective) content based descriptions of documents with a very high rate of production.

Any temporal based document must be partitioned prior to the parsing and analysis. Video segments may be divided atomically in frames, shots and scenes. Next, the extraction of key frames or key segments is the way to produce the entries for the scenes (figure 2.16, p. 24). Therefore, automated indexing of video will require the support of tools that can detect such meaningful segments and extract content features of any video source. Content analysis may then be performed on individual segments in order to identify appropriate index terms. The three major steps are explained below [Zhang97]:

From: `http://jodi.tamu.edu/Articles/v02/i04/Lee/`

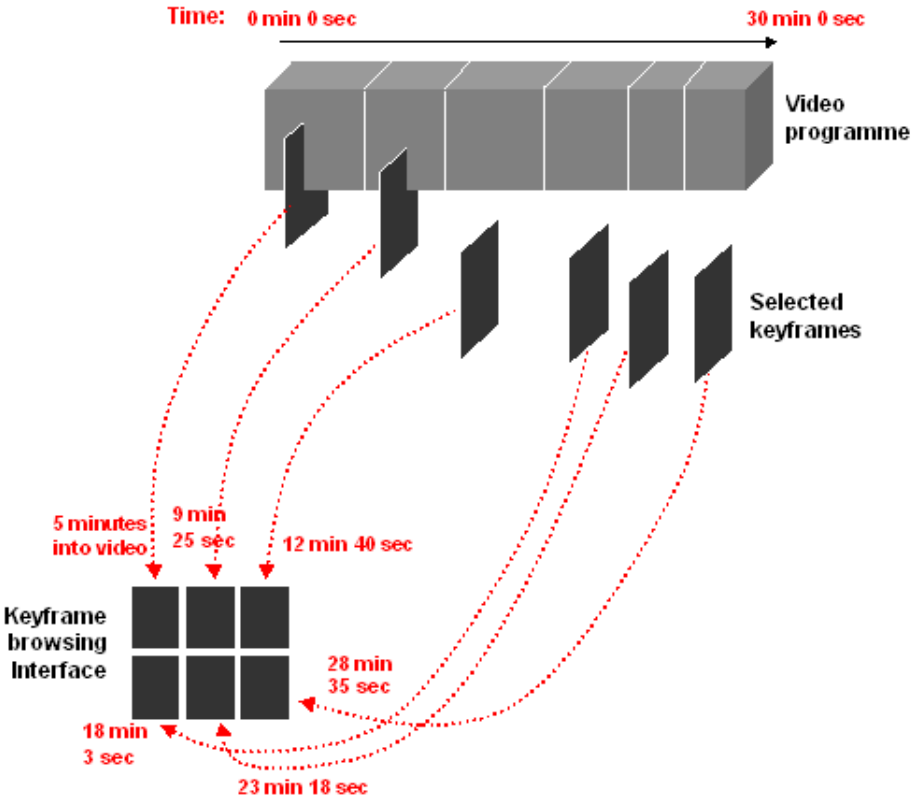


Figure 2.16: Key frames selected from video to be displayed on a browsing interface [Lee02].

**Parsing** This process will partition a video stream into generic video segments (shots or clips) of different levels of granularity and extract structure information of the video. These clips will be the units for representation and indexing.

**Content analysis, abstraction and representation** Individual clips will be decomposed into semantic primitives based on which a clip can be represented and indexed with a semantic description. In practice, the abstraction process will generate a visual abstract of clips and low level visual features of clips will be used to represent their visual content.

**Retrieval and browsing** Indexes are built based on content primitives or meta data through, for instance, a clustering process which classifies shots into different visual categories or indexing structure. Schemes and tools are needed to search and browse large video databases in order to retrieve the desired video clips. These tools use the generated content representation and indices.

### Parsing

The shot, consisting of one or more frames recorded contiguously and representing a continuous action in time and space, is the basic temporal unit for video indexing and editing. A collection of one or more adjoining shots that focus on an object or objects of interest may comprise a scene. Shots have transitions or boundaries between them that may be of different types: a simple cut is an abrupt shot change occurring between two frames; a fade is a slow change in brightness of images finishing in or starting by a solid black frame; a dissolve occurs when the images of the first shot get dimmer and the images of the second one get brighter, with the frames at the middle of the transition showing two superimposed images; a wipe occurs when pixels from the second shot replace those of the first shot in a regular pattern, such as if a vertical line from the right edge of the frame travels to the left edge.

The metrics for shot boundaries detection and efficient video partitioning may be divided into two major groups: those based on local pixel feature comparison, such as pixel values and edges; and those based on global features, such as pixel histograms and statistic distributions of pixel to pixel change. Metrics in both groups should be able to detect the following three factors of image change [Zhang97]:

- Shot abrupt or gradual change.
- Motion, including camera and object motion.
- Luminance changes and noise.

Thus, the techniques for shot cut detection can be divided into:

**Pixel value detection** This is an easy way to detect a qualitative change between two frame images, comparing the spatially corresponding pixels in the two frames and determining how many have changed.

**Histogram comparison** Is a more efficient alternative to direct comparing of the frames pixel, using statistics and global features of an entire image. Pixel value (colour) histograms are used instead of pixel to pixel comparison.

**Edge pixel comparison** Changes in the edge distribution between frames are an indication of content changes. When a cut or a gradual transition occurs, new intensity edges appear but far from the locations of old edges and similarly, old edges disappear far from the locations of new edges.

An example of the sample key frames used for the algorithms applied on performance analysis of shot change detection is depicted in figure 2.17.

Figure 2.17: Sample transitions of a video sequence [Gargi00].

For gradual transition detection there are the following techniques: Twin comparison

approach, Pixel change classification, Edge pixel comparison (see definition above), and Editing model fitting.

Video partitioning has evolved with the Joint Pictures Experts Group (JPEG), Moving Pictures Experts Group (MPEG) and H.26X standards for compressed pictures and motion pictures in mind. With so many multimedia production and exchange being done in each one of these three formats, the operation directly on compressed representations could save resources and be more efficient while avoiding the decompressing effort. The three basic types of algorithms for video partitioning are the Discrete Cosine Transformation (DCT) coefficient based comparison, the Motion vector based comparison, and an hybrid approach using both.

#### Content analysis, abstraction and representation

A very important technique for shot content analysis and classification is the camera work and object motion analysis. Object motions usually represent locomotion activity, by humans or not, and major events in video shots. There are classical problems in the computer vision field, that remain unsolved, about the accurate discrimination between camera work induced motion and object movement induced motion. However there are several algorithms that perform the detection with satisfactory accuracy and speed [Zhang97]. Camera work includes panning and tilting (horizontal or vertical rotation of the camera), zooming (focal length change) without the changing of the camera position, tracking and booming which are the horizontal and vertical movement of the camera, and dollying (horizontal lateral movement of the camera). Of course, any combination of these operations is possible.

After parsing, a visual abstraction of the video clip is needed before the video content representation. There are three main techniques to concisely abstract a relevant representation of the video document: key frames, video icons and skimmed highlights. Key frames are the images extracted from the video which best represent the whole. Video icons are constructed static images based on a key frame and with extra information, such as synthetic signs to relevant objects or motion. Video skimming is a very interesting technique which does a visual summary of the whole video, i.e., generates a trailer or a teaser of short duration from hours of video.

Next to the abstraction, the video content representation can be done, automatically, but only for low level features, such as colour, texture and shape. Key frames are used for the shot content representation. For the representation of the shot content temporal features, temporal variation, camera operations and also statistic motion features are used. With all these features it is possible to try shot similarity and improve retrieval of a specific

kind of video shot.

For a video sequence analysis shot clustering method is needed. A contiguous group of shots may constitute a scene unit from the narrative point of view, with shots in the same location and sharing some visual or content theme. The clustering can be based on image and motion features of the shots as referred above. Of course, video scene analysis requires higher level content analysis than what was mentioned until now. An efficient browsing and retrieval tool for multimedia content should be able to deal with indexed low level features and with high level semantic information.

#### Retrieval and browsing

At present, informatics and continuous technological advances in the physics of storage and transport of information lead to information supports like the Digital Versatile Disc (DVD) and its respective player. Being very cheap and with reasonable quality for a small screen projection at home, DVD rapidly promoted a way of enjoying a good movie without going to a cinema theatre. Video-on-demand is a growing reality where Television (TV) networks can stream the movies by cable or satellite to any home thanks to broad band communications networks. So, all these "travelling bits", i.e., video information needs efficient ways for indexing, browsing and retrieval.

#### MPEG-7

A common meta data definition was needed for multimedia description and, naturally, to ease the tasks of delivering, trading or exchanging, cataloguing and searching of the multimedia documents. Therefore the MPEG Committee started to work on the objectives and requirements for the MPEG-7 Call for Proposals issued in October of 1998. In the beginning of 1999, proposals were evaluated and in March the first version of the MPEG-7 eXperimentation Model (XM) Software was released. After two more years of work by the Committee, in September of 2001, the ISO/IEC 15938 International Standard (IS) is committed. The period after the IS release date and early 2003 continued the work on the first Amendment which was committed in May of that year.

Before the MPEG-7 requirements, applications where first identified for scoping the work. Relevant applications included both new and existing ones with the same priority for shaping the standard. Several application domains were identified (table 2.1, p. 29) with the purpose to give the industry a good set of examples for the MPEG-7 kick off, being also expected new unforeseen applications of the standard.

| | |
|---|---|
| Architecture, real estate, interior design | Investigation services, forensics |
| Audiovisual content production | Journalism |
| Biomedical | Remote sensing |
| Cultural services | Shopping |
| Education | Social |
| Entertainment | Surveillance |
| Film, video and radio archives | Tourist information |
| Geographic information systems | |

Table 2.1: MPEG-7 standard's application domains [Manjunath02].

The applications' examples were organised into three sets:

**Pull applications** Such as storage and retrieval in audiovisual databases, delivery of pictures and video for professional media production, commercial musical applications, sound effects libraries, historical speech database, movie scene retrieval by memorable auditory events and registration and retrieval of trademarks.

**Push applications** Such as user agent driven media selection and filtering, personalised television services, intelligent multimedia presentations and information access facilities for people with special needs.

**Specialised professional applications** The ones that are particularly related to a specific professional environment, notably interactive television shopping, biomedical, remote sensing, educational and surveillance applications.

The MPEG-7 standard requirements have been extracted from the identified applications' examples and are divided into five categories which will only be briefly[2] named below:

**Descriptors** A descriptor is a representation of a feature, where a feature is a distinctive data characteristic that has some meaning to an agent (human or not). Defines the syntax and the semantics of the feature representation and its value allows an evaluation of the corresponding feature. Independent from the way content is stored or coded, descriptors require Cross-modality, Direct data manipulation, Data adaptation, Language of text-based descriptions, Linking, Priority ordering of related information and Unique identification.

[2]It's out of the scope of this work to describe all the requirements and descriptors. For more information refer to [Manjunath02, ISO04]

**Description Schemes** A Description Scheme (DS) specifies the structure and semantics of the relationships between its components, which may be both descriptors and description schemes. Provides a solution to model and describe the structure and semantics of multimedia content. For instance, a movie, temporally structured as scenes and shots, including some textual descriptors at the scene level and colour, motion and audio amplitude descriptors at the shot level. It requires DS relationship, Priority ordering of descriptors, Hierarchy of descriptors, Scalability of descriptors, Description of temporal range and Data adaptation.

**Description Definition Language** With the purpose to ease the creation, modification and extension of the Description Schemes and, eventually, the Descriptors. The requirements are Compositional capabilities, Unique identification, Primitive data types, Composite data types, Multiple media types, Various types of DS instantiations, Relationships within a DS and between Description Schemes, Relationship between description and data, Link to ontologies, Platform independence, Grammar, Validation of Constraints, Human readability, Real-time support, and Forward and backward compatibility.

**Descriptions** These requirements on Descriptions have common aspects to the requirements for Descriptors, Description Schemes and the Description Definition Language (DDL). Descriptions are obtained by instantiation of one or more Description Schemes and are themselves not defined by the standard. The requirements are clustered in:

- General: Types of features, Abstraction levels for multimedia material, Management of descriptions, Translations in text descriptions, Associated information, Referencing analogue data and Associate relations.
- Functional: Retrieval effectiveness stored descriptions, Distributed multimedia databases, Interactive queries, Browsing, Interactivity support, User preferences, User usage history, Key items, Ordering keys and Temporal validity.
- Coding: Description-efficient representation, Description extraction, and Robustness to information errors and loss.
- Visual-specific: Types of features, Data visualisation using the description, Visual data formats, and Visual data classes.
- Audio-specific: Types of features, Data sounding using the description, Auditory data formats, and Auditory data classes.
- Text-specific: Text retrieval and Consistency of text description tools.

**System Tools** Are the tools related to the binary codification, synchronisation, transport and storage of descriptions, as well as to the management and protection of intellectual property. The System Tools requirements are clustered in:

- General: Multiplexing of descriptions, Flexible access to partial descriptions at the systems level, Temporal synchronisation of content with descriptions, Synchronisation of multiple descriptions over different physical locations, Physical location of content with associated descriptions, Transmission mechanisms for MPEG-7 streams, MPEG-7 file format, Robustness to information errors and loss, Quality of Service (QoS), Carried descriptions, Partition of descriptions, Efficient parsing, Efficient updating of descriptions, and Timed updating.
- Intellectual property management and protection: No legal status of descriptions, Describing content rights, Relationship to content management and protection measures, Applications distinguishing between legitimate and illegitimate content, Authentication of descriptions, Management and protection of descriptions, Management and protection of descriptors and Description Schemes, Usage rules, Usage history, identification of content, Identification of content in descriptions, and Identification of descriptions.
- Binary Format for MPEG-7 Description Streams (BiM): Compactness, Streaming, transfer and storage, Parsing, Applicability to individual descriptors and Description Schemes, Mapping with the DDL, Well-formedness and validation, and Easy wrapping.
- Textual representation: Description tree updating, Adding and deletion, Scheduling of update executions, and Description tree coherency.

The MPEG-7 standard is an important contribution of the MPEG community to the area of meta data or descriptions. It can be characterised by [Manjunath02]:

- Its generality, related to its capability to consistently describe content from many application domains;
- The integration of low-level and high-level descriptors into a single architecture, allowing to combine the power of both types of descriptors;
- Its object-based data model, providing the capability to independently describe individual objects within a scene; and

- Its extensibility, provided by the DDL, which allows users to augment MPEG-7 to suit their own specific needs and the standard to keep evolving, integrating novel description tools.

There are too many descriptors on the MPEG-7 IS to refer them all here, so only the Audiovisual (AV) relevant ones, for the implementation of the prototype, will be referred.

The description of multimedia content using natural language text is called Text Annotation. This kind of annotation is well known and widespread, thus, MPEG-7 supports it in several ways: Free text, Keyword, Structured, Dependency structure, Classification schemes and terms, Defining classification schemes, Using terms, Graphical classification schemes, Peoples and places, Affective response and Ordering descriptions.

Among all these text annotation descriptors, three are selected as relevant for a brief comparison presented below, in figures 2.18, 2.19 and 2.20, as instantiations (in MPEG-7's XML) of some multimedia document. An important evidence of refering is that the `KeywordAnnotation` descriptor may be a special case of the `FreeTextAnnotation` where all the words are separated. However, the proper use of keywords may be used to increase the relevance of some distinct words. Thus, using both descriptors is still an advantage for high level semantic annotations. `StructuredAnnotation` is a descriptor for even more semantic information addition. Some user agents and systems of annotation and search, such as a newspaper database, may prefer to explicitly restrict semantics for well known meanings.

```
<TextAnnotation>
  <FreeTextAnnotation xml:lang="en">
    The moon shadows by the sea...
  </FreeTextAnnotation>
</TextAnnotation>
```

Figure 2.18: `FreeTextAnnotation` example.

Along with the text annotation, time representation is very important and MPEG-7 supports it with two different kinds, although, both representations are ISO 8601[3] based

[3]The ISO 8601 format defines [W3C98, ISO00]: `YYYY-MM-DDThh:mm:ss.d`[+|−]`hh:mm` (`d` is a second fraction with infinite precision (digits))

```
<TextAnnotation>
  <KeywordAnnotation xml:lang="en">
    <Keyword>sea</Keyword>
    <Keyword>moon</Keyword>
    <Keyword>shadows</Keyword>
  </KeywordAnnotation>
</TextAnnotation>
```

Figure 2.19: `KeywordAnnotation` example.

```
<TextAnnotation>
  <StructuredAnnotation xml:lang="en">
    <Where><Name>sea</Name></Where>
    <Who><Name>moon</Name></Who>
    <What><Name>shadows</Name></What>
  </StructuredAnnotation>
</TextAnnotation>
```

Figure 2.20: `StructuredAnnotation` example.

with World time having Time Zone Description (TZD) added:

**Media time** Is the time measured or stored within the media. Media data times represent time intervals using a start time point (`mediaTimePoint` data type) and a duration (`mediaDuration` data type). The syntax is [−]`YYYY-MM-DDThh:mm:ss:nFN` where − is for dates Before Christ (B.C.), T date/time separator, F fraction/total-fractions-in-a-second separator.

The `mediaDuration` data type uses the format [−]P`n`*D*T`n`*H*`n`*M*`n`*S*`n`*N*`n`*F* where P is the separator indicating the beginning of a duration and each part of the duration contains a count `n` followed by a letter indicating the unit being counted for days, hours, minutes, seconds, fractions and total fractions in a second.

On top of the `mediaDuration` and `mediaTimePoint` data types, MPEG-7 builds three kinds of media time representation:

- Simple time: the basic representation of an absolute time.
- Relative time: specifies a media time point relative to a time base. Useful if a media segment, such as a story in a news sequence, is inserted dynamically. To update the story's description (time), only the time base needs to be changed

- Incremental time: specifies a time interval by counting predefined time units. When dealing with sampled data, the sample points or ticks of the reference clock are often equidistant in time.

  For example, a start frame for some movie scene can be represented by `711` units of a predefined time unit like `PT1N30F`, meaning $711 \times 30$ frames from the start. On this example each second has 30 fractions (frames) and the predefined time unit is $\frac{1}{30}$ of the second.

**World time** or Generic, which is the global time measured in the World.

MPEG-7 descriptors are designed for a broad range of descriptions and types of information: low level audiovisual features, such as colour, texture, motion or audio. High level features of objects with semantics, events and abstract concepts; content management processes; information about the storage media. The majority of the low level descriptors should be extracted automatically, contrasting with the high level descriptors where human intervention will be required.

The visual (the audio ones will not be used) descriptors that were developed can be broadly classified into general visual descriptors and domain specific visual descriptors. The domain specific descriptors are application dependent, such as Face descriptor for face recognition, while general descriptors are Colour, Texture, Shape and Motion visual descriptors. There are hundreds of descriptors (data types) within the standard, so a focus on the visual relevant ones for the implementation of the prototype is taken, despite the existence of others which can be studied and added at any time.

In the Colour group of descriptors, the `ColorSpace` descriptor is an important one that specifies a selection of a colour space to be used in another colour descriptor. The colour spaces specified in the MPEG-7 are Red, Green, Blue (RGB), Luminance, Chrominance Blue, Chrominance Red (YCbCr), Hue, Saturation, Value (HSV), Hue, Max, Min, Diff (HMMD), Monochrome and Linear transformation matrix with a reference to RGB. In addition, a flag is provided to indicate reference to a primary colour and mapping to a standard reference white value. The associated `ColorQuantization` descriptor specifies the partitioning of the given colour space into discrete bins. These two descriptors are used in conjunction with other colour descriptors.

The RGB colour space is defined as the unit cube in the Cartesian coordinate system. For the Monochrome colour representation, Y component alone in the YCbCr is used. The colour space HSV, is defined as the following (algorithm 2.1) non-linear transformation

from RGB colour space:

$$
\begin{aligned}
&max = max(R, G, B);\\
&min = min(R, G, B);\\
&v = max;\\
&\text{if } (v == 0)\ s = 0;\ \text{else } s = (max - min)/max;\\
&\text{if } (max == min)\ h = 0;\ //\text{Achromatic colour}\\
&\text{else if } (R == max\ \&\&\ G >= B)\ h = 60 \times (G - B)/(max - min);\\
&\text{else if } (R == max\ \&\&\ G < B)\ h = 360 + 60 \times (G - B)/(max - min);\\
&\text{else if } (G == max)\ h = 60 \times (2.0 + (B - R)/(max - min));\\
&\text{else } h = 60 \times (4.0 + (R - G)/(max - min));
\end{aligned}
\tag{2.1}
$$

The YCbCr is a legacy colour space of the precedent MPEG standards. It is defined by a linear transformation of RGB colour space as specified below (equation 2.2 [Manjunath02]):

$$
\begin{aligned}
Y &= 0.229 \times R + 0.587 \times G + 0.114 \times B\\
Cb &= -\ 0.169 \times R - 0.331 \times G + 0.500 \times B\\
Cr &= 0.500 \times R - 0.419 \times G - 0.081 \times B
\end{aligned}
\tag{2.2}
$$

| Field | Number of bits | Meaning |
|---|---|---|
| `NumberofColors` | 3 | Specifies the number of dominant colours |
| `SpatialCoherency` | 5 | Spatial Coherency value |
| `Percentage[]` | 5 | Normalised percentage associated with each dominant colour |
| `ColorVariance[][]` | 1 | Colour variance of each dominant colour |
| `Index[][]` | 1–12 | Dominant colour values |

Table 2.2: MPEG-7 `DominantColor` descriptor [Manjunath02].

The dominant colour (`DominantColor`) descriptor is a low level one that provides a compact description of representative colours in an image or image region. It is useful for automatic processing, with five fields of information and the specification presented in table 2.2, p. 35. The `DominantColor` descriptor is defined as $F = \{(c_i, p_i, v_i), s\}, (i = 1, 2, ..., N)$ where $N$ is the number of dominant colours. Each dominant colour value $c_i$ is a vector of corresponding colour space component values. The percentage $p_i$ ($\sum_i p_i = 1$) is the fraction of image or region pixels corresponding to colour $c_i$. The optional colour variance $v_i$ describes the variation of the colour values of the pixels in a cluster around the

corresponding representative colour. Spatial coherency $s$ is a single number representing the overall spatial homogeneity of the dominant colours in the image. [Manjunath02]

Also in the Colour group there is the colour layout (`ColorLayout`) descriptor that captures the spatial layout of the representative colours on a grid superimposed on a region or image. Representation is based on coefficients of the Discrete Cosine Transformation. This is a very compact descriptor, highly efficient in fast browsing and search applications. It can be applied to still images as well as to video segments. The functionalities of this descriptor can be achieved using a Grid Layout data type of MPEG-7 and the `DominantColor` descriptor. However, this combination would require a large number of bits, and a more complex and expensive matching than `ColorLayout` descriptor. The possible number of coefficients are 3, 6, 10, 15, 21, 28 or 64. The actual values are represented by the arrays `YCoeff`, `CbCoeff` and `CrCoeff`. The lengths of each of these are either five or six bits depending on the coefficient. The representation is summarised in the table 2.3, p. 36.

| Field | Number of bits | Meaning |
|---|---|---|
| `CoefficientPattern` | 1–2 | Specifies the number of DCT coefficients |
| `NumberofYCoeff` | 3 | Number of DCT coeffs for the luminance |
| `NumberofCCoeff` | 3 | Number of DCT coeffs for the chrominance |
| `YCoeff[]` | 5–6 | The DCT coeffs values for the luminance |
| `CbCoeff[]` | 5–6 | The DCT coeffs values for the chrominance |
| `CrCoeff[]` | 5–6 | The DCT coeffs values for the chrominance |

Table 2.3: MPEG-7 `ColorLayout` descriptor [Manjunath02].

The scalable colour (`ScalableColor`) descriptor (figure 2.21) is also a very useful one, and may be interpreted as a Haar transform based encoding scheme applied across values of a colour histogram in the HSV colour space, which is defined (algorithm 2.1, p. 35) as a non-linear transformation from RGB colour space. The histogram values are extracted, normalised and non-linearly mapped into four bit integer values along the histogram bins. The output representation of the extraction process is scalable in terms of the number of bins, by varying the number of coefficients used. Besides the scalability in the number of histogram bins, another form of scalability is achieved by scaling the quantised (integer) representation of the coefficients to different number of bits. The difference coefficients in the Haar transformation can take either positive or negative values. The sign part is always retained, whereas the magnitude part can be scaled by skipping the least significant bits. Using only the sign bit (one bit / coefficient) leads to an extremely

compact representation, while good retrieval efficiency is kept.

```
<VisualDescriptor xsi:type="ScalableColorType" numOfCoeff="16"
  numOfBitplanesDiscarded="0">
  <Coeff>
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6
  </Coeff>
</VisualDescriptor>
```

Figure 2.21: `ScalableColor` example.

The group of frame or group of picture colour (`GoFGoPColor`) descriptor (figure 2.22) is used for the joint representation of colour based features for multiple images or frames in a video segment.

This descriptor is an extension of the scalable colour one with an additional field of information: aggregation, which stores the type of operation done to the histograms of the multiple images or frames of a video segment. The aggregation of the histograms of multiple images or frames may be done by average, median or intersection. The representation is a scalable colour descriptor with the additional field of the aggregation process type.

```
<VisualDescriptor xsi:type="GoFGoPColorType" aggregation="Average">
  <ScalableColor numOfCoeff="16" numOfBitplanesDiscarded="0">
    <Coeff>
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6
    </Coeff>
  </ScalableColor>
</VisualDescriptor>
```

Figure 2.22: `GoFGoPColor` example.

## 2.4 On Multimedia Evolutionary Computation and Annotation

Pursuing a new paradigm implies that there is not much work done or published that can be used for comparison purposes. Nevertheless, similar work has been and is being developed around adaptive and evolutionary hypermedia and multimedia annotation.

### Smart Snakes

In the past fifteen years some work has been done connecting multimedia and EC despite the missing relation with multimedia annotation. In [Heap95] a GA is used to do a global image search, in order to approximately locate an image feature, before the system do the specific task of hand tracking and gesture recognition using Smart Snakes (Active Shape Models). This is a typical hybrid usage of EC and traditional deterministic algorithms where an approximated solution is needed before the intervention of the traditional technique. When the search space is huge and there are strict time constraints the GA may be used with very good results.

### Karl Sims

From: `http://www.genarts.com/karl/panspermia.html`

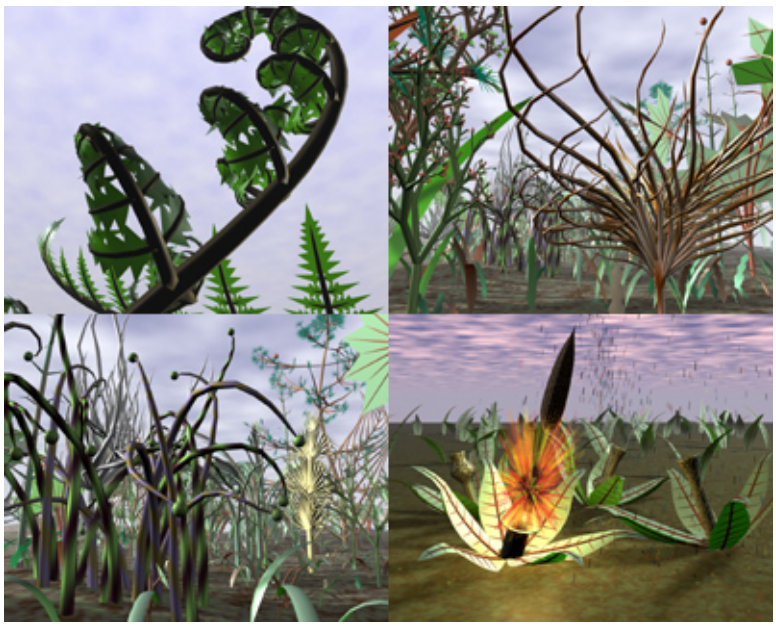

Figure 2.23: Panspermia: ferns, jungle, stalks, shooters.

In [Sims91] a very impressive work has been made proposing artificial evolution for the

production of computer graphics and animation in order to create forms, textures and motions that are not bounded by a fixed space of possible results (figure 2.23).

In this work, Karl Sims does a demonstration of how evolutionary techniques of variation and selection can be used to create simulated and complex structures. Interactive selection, based on visual perception of procedurally generated results, allows the user to direct simulated evolutions in preferred directions, while achieving flexible complexity with a minimum user knowledge of details. One of the interesting techniques applied was the usage of genotypes consisting of symbolic expressions as an attempt to surpass the limitations of fixed-length genotypes with predefined expression rules. Symbolic Lisp language expressions are used as genotypes. A set of Lisp functions and a set of argument generators are used to create arbitrary expressions which can be mutated, evolved, and evaluated to generate phenotypes by means of GP.

The developed examples in Karl Sims work range from evolving 3D plant structures, using a technique like L-Systems, to evolving images and animations using symbolic expressions as genotypes. The mating of symbolic expressions has two main methods: the first method (figure 2.24) requires the two parents to be similar in structure. The nodes in the expression trees of both parents are simultaneously traversed and copied to create the new expression. When a difference is encountered between the parents, one of the two versions is copied with equal probability.

The second method is somewhat more "liberal" in allowing the combination of the parents in a less constrained way. A node in the expression tree of one parent is chosen at random and replaced by a node chosen at random from the other parent. This "crossing over" technique permits parts of even dissimilar expressions to be combined thus generating many more offspring than the first method.

```
parent 1:  (* (abs X) (mod X Y))
parent 2:  (* (/ Y X) (* X -.7))
 child 1:  (* (abs X) (mod X Y))
 child 2:  (* (abs X) (* X -.7))
 child 3:  (* (/ Y X) (mod X Y))
 child 4:  (* (/ Y X) (* X -.7))
```

Figure 2.24: Symbolic expressions mating: method 1 [Sims91].

The temporal evolution of animations was done by extending the image evolution system to evolve moving images. Several methods for the inclusion of a temporal dimension in

symbolic expressions are proposed in [Sims91] and briefly enumerated:

1. The addition of a new input variable Time to the list of available arguments. Expressions can be evolved that are functions of X, Y and Time such that different images are produced as the value of Time is smoothly animated.

2. Genetic Cross Dissolves can be performed between two expressions of similar structure. Interpolation between two expressions is performed by matching the expressions where they are identical and interpolating between the results where they are different.

3. The addition of an input image to the list of available arguments to make functions of (X,Y,Image). The input image can be animated and processed by evaluating the expression multiple times for values of Image corresponding to frames of another source of animation such as hand drawn or traditional 3D computer graphics.

4. Animation of the images that use the pixel coordinates (X,Y) to determine the colours at each pixel, altering the mappings of X and Y before the expression is evaluated.

5. Evolved expressions can be adjusted and experimented with by hand. If parameters in expressions are smoothly interpolated to new values, the corresponding image will change in potentially interesting ways.

From: `http://www.genarts.com/karl/evolved-virtual-creatures.html`

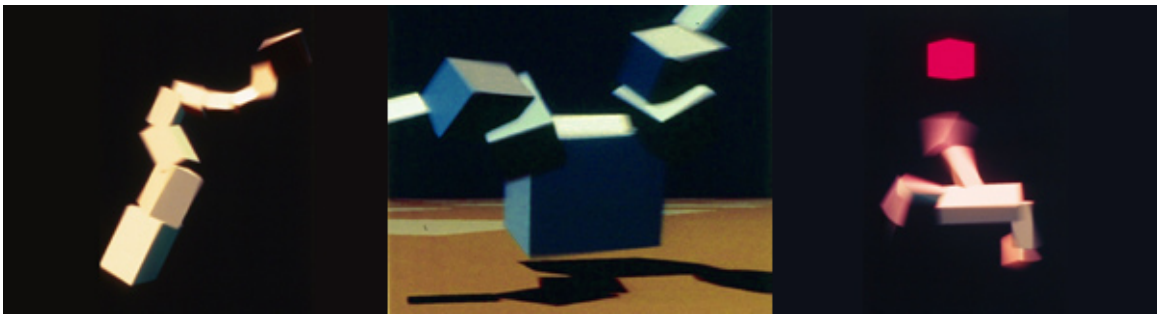

Figure 2.25: Creatures morphology: swimmer, hopper, follower.

In [Sims94b, Sims94a] Karl Sims extended his previous work in EC and developed a system for creating virtual creatures that move and behave (figure 2.25), also compete (figure 2.26) in one-on-one contests for a common resource, in physically simulated three

dimensional worlds. The creatures' morphologies and neural systems for controlling their muscle forces are both generated automatically using genetic algorithms. Different fitness evaluation functions are used to direct simulated evolutions towards specific behaviours, such as swimming, walking, jumping, or following using any method of locomotion. The genetic language for the representation of the creatures with directed graphs of nodes and connections allows an unlimited space of possible creatures and behaviours to be explored. All this is obtained without requiring cumbersome user specifications, design efforts or knowledge of algorithmic details.

From: `http://www.genarts.com/karl/evolved-virtual-creatures.html`

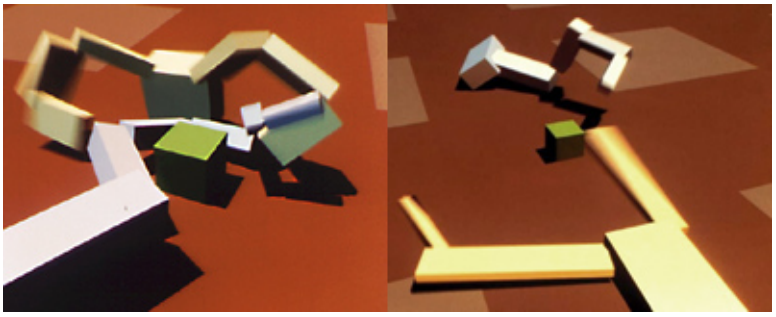

Figure 2.26: Creatures competition: crab vs arm and sweeper vs arm.

In the creature morphology, the phenotype embodiment is a hierarchy of articulated three dimensional rigid parts. The genetic representation of this morphology is a directed graph of nodes and connections. Each node in the graph contains information describing a rigid part. The information is composed of dimensions, joint type, recursive limit, a set of local neurons and a set of connections to other nodes. Each connection also contains information. The placement of a child part relative to its parent is decomposed into position, orientation, scale and reflection. The genotype is represented by directed graph and the phenotype is the hierarchy of the 3D parts.

Evolutions performed with populations of competing creatures promote the emergence of interesting, diverse and opposed strategies to reach the goal. The competition is restricted to one-on-one by pair of individuals. On each generation individuals are paired up by some pattern and a number of competitions are performed to eventually determine the value for every individual. Several different pair-wise competition patterns for one and two species are devised: all vs all within species, all vs all between species, random within species, random between species, tournament within species, all vs best within species, all vs best between species. Because the all vs all competitions require an intensive computational operation, Karl Sims suggests that random or tournament patterns should be chosen in

order to obtain results more quickly or if the supporting hardware can not handle the load giving results in a feasible time.

### Rhetorical Patterns

A recent new conceptual notion, Rhetorical Patterns, has been introduced [Rocchi04] to support an approach to the adaptive composition of video documentaries. The adaptation is based on templates that encode rules for the dynamic selection, sequencing and composition of video shots. A new language was created, XASCRIPT, for the definition of adaptation rules and constraints, specifying a schemata of solutions by means of templates programming, intensional descriptions of a set of potential documentaries, with multiple choice points on user dependent parameters. Adaptation rules enable authors to state constraints and strategies to select shots and apply transition effects. An adaptation rule is a `<condition, action>` pair, where the condition tests the requirements and the action composes the pieces of the documentary.

Relying on the notion of pattern when writing templates for adaptive presentations, designers have to face recurrent problems. For example:

**Deepening** if the user has already been exposed to a topic "t", how to select and present material related to "t" (e.g. highlight its features)?

**Comparison** maximise the extent to which a visitor's understanding of an exhibit coheres with her other knowledge, and help to prevent the hearer from forming misconceptions. How to refer to previously mentioned material "m", related to the current topic?

**Suggestion** suggestions are complementary to comparisons. If the user has not visited an exhibit "e" and the designer thinks that "e" – considering the current context – is worth a visit, how to lead the user to visit "e"?

**Exemplification** if the topic "t" is generic, say "a painting technique", how to provide visual and aural explanations, so that the user can more effectively understand "t"?

Rocchi tries an approach to surpass the recurrent problems stated above. This notion of "Rhetorical Patterns" has the advantage of extensibility and easy adaptation to any problem related to hypermedia authoring, such as video documentaries, where identification of new patterns is constantly happening. It improves the focus on the interaction with the information space and user's needs; and also copes with the usually large amount of content to be organised during the authoring of a hypermedia document.

### Video segmentation and summarisation

A GA for video segmentation and summarisation is presented in the work of Chiu *et al* [Chiu00] as a feasible solution with advantages over other non-evolutionary ones, such as clustering, for the effective search of the image frames within segments space. It is very difficult for standard non-evolutionary algorithms to search in feasible time huge spaces of potential solutions. Three advantages are stated: First, the genetic mechanism is independent of the prescribed evaluation function and can be tailored to support a variety of characterisations based on heuristics depending on genre, domain, user type or others. Second, evolutionary algorithms are naturally suited for doing incremental segmentation that can be applied to streaming media, such as video over the Web. Third, it can support dynamically updated segmentation that adapts to usage patterns, like adaptively increasing the likelihood that frequently accessed points will appear as segment boundaries.

In that work, similarity adjacency functions are defined for the task of browsing and summarising the segments of video, with varying degrees of function complexity: the simplest one is only able to account image differences, and in the more complex one, information retrieval concepts are used. Preprocessing is used to reduce the size of the set of images by only looking at those that are not too similar, a video clip can usually have no less than thousands of frames (images) and a large number of adjacent ones are likely to be similar. So, for a typical rate of 30 frames per second (fps) video, first a subsample at a lower rate (near 2 fps) is done and reasonable enough to capture the action in most domains. Then, only the least similar images are extracted using the colour histograms standard technique for differences measurement.

The evaluation is made with similarity adjacency functions as follows. Let $S_k$ be a subset of $k$ selected images (after the preprocessing stage): $f(S_k) = \sum_{i,j \in S_k} \alpha(i,j)h(i,j)$ where $h(i,j)$ is the histogram difference between $i$ and $j$ and $\alpha(i,j)$ is a function for weighting the histogram differences.

Information retrieval concepts may be applied by weighting each element by its importance. The definition, for example, of factors in the length of an element with its commonality, so that the longer and the less common elements have greater importance. Extending this notion of importance one can define the precedence of a frame as another factor, so that earlier appearing frames are more heavily weighted than later ones in the same similarity class. The importance can be defined as $I_i = P_i \log(\delta(i)) \log(1/W_i)$ where $W_i = |C_i|/|F'|$ being $C_i$ a set of elements similar to $i$ and $F'$ the reduced set after the preprocessing, $\delta(i)$ is the number of frames in the original set $F$ from $i$ to the next element

in $F'$. The precedence is defined by $P_i = |B_i|/|C_i|$ where $B_i = \{j \in C_i | i \leq j\}$.

Then each term is weighted with the average importance: $\alpha(i,j) = (I_i + I_j)/|i-j|^2, i \neq j, i \neq 0$ and finally we can define a more interesting evaluation function:

$$f(S_k) = \sum_{\substack{i,j \in S_k \\ i \neq j, i \neq 0}} h(i,j) \frac{(I_i + I_j)}{|i-j|^2} \tag{2.3}$$

This function is a similarity adjacency that makes nearby images more dissimilar and permits a certain amount of repetition in the overall summary to capture the rhythm of the video.

This evaluation function is the fitness function for the GA which has as input a video clip and an integer $k$ for the desired number of segment boundaries. It serves as access points for indexing and summarisation, and the output is a sequence of $k$ boundary images plus their importance scores. The encoding of the chromosome is a string of ones and zeros; the bit position of the chromosome string is an index for an element of the data stream, i.e., a video frame in $F'$, read left to right. The length of the string is the number of images $|F'|$ and ones (`1`) denote the segment boundaries. The Fitness proportionate selection method is used. One point crossover is the method for the offspring generation with a slight change: the point must be at a boundary (`1`). Mutation is used after the crossover but when flipping a zero to one another one must be flipped to zero so the total number of segments is kept constant.

A test was made with this Genetic Segmentation Algorithm (GSA) for summarising an hour long seminar video. The GSA was applied with $k = 5$, population size of 2000 and over 100 generations. The result was excellent, the three distinct topics of the video (title frames) have been selected along with the speaker and a view of the room. It would be difficult for a person to select a much better set of representative images for a summary.

This is a promising technique for video segmentation and summarisation, while $k = 5$ is a value where brute force algorithms can be applied (and were to verify the global maximum reached by GSA), and $k = 12$ or $k = 24$ makes the combinatorial explosion in equation 2.3 infeasible to be computed by the traditional algorithms.

# Chapter 3

# Conceptual model

In this chapter the concepts underlying the system are presented. The approach to the concept is clearly explained along with the objectives and the reason for the utilisation of the two computational paradigms: Evolutionary Computation and Multimedia Computation. Next, a description of possible solutions to the implementation of the conceptual model is made. Here a couple of solutions are exemplified with brief explanations about the combinations of selection methods, mutation actions and all the relevant issues. The chosen solution, the way the presented problem is going to be solved, is thoroughly explained, including all the gene codification, fitness function and its associated metrics with the respective formulas. Finally, the diagram and description of the whole specific GA is presented.

## 3.1 Approach

Video and audio documents digitised and stored in a computer system become material for the application of several computational techniques. One can divide the application of the techniques in two scale levels: micro and macro. At the micro level is where the information is represented by non-structured bytes sequences, which may be processed in many different ways. This kind of representation allows many transformations but hides contextual characteristics of the material. Cultural, narrative, continuity and linearity aspects need a macro (high) level processing for its study. There is a built-in contextual complexity to be indexed and act upon [Correia02]. Therefore the approach objectives are:

- The use of Evolutionary Computation for the creation of a new paradigm for

multimedia production.

- To develop video annotation as fitness evaluation of a set of image shots, verifying the relation between video annotations and fitness evaluation.
- Use of the multimedia production system in an interactive way for the audience, so that the evolutionary process may be affected and changed.

A new computational approach is being developed by this research for the ability to deal with AV information complexity. Therefore two existent paradigms are merged: Evolutionary Computation and Multimedia Computation.

In Multimedia Computation, Video Description Computing MPEG-7 standard [ISO04] is used, and a small subset of its descriptors is applied to the multimedia segments for classification and the segments are combined and subject to the evolutionary process.

In this new approach, the evolutionary algorithms studied are the ones derived from Genetics' concepts, or simply Genetic Algorithms that have been successfully [Goldberg89] applied in search, optimisation and machine learning in several fields of art, science and engineering, including mathematical equation solving, robotics and musical composition. Therefore, audiovisual information contextual complexity with the EC potential is joined. The GAs usage is proposed for the creation of a new narrative editing process.

The EC concepts are translated and directly mapped to the annotated AV materials concepts. Namely a full video segment (video clip) is mapped to a chromosome. A scene (video segment) to a *super*-gene (with several alleles – characteristics) as the descriptors codification.

To do a proof of concept an AV database is populated with short duration footage from several places. As a starting point, one place – Quinta da Regaleira, Sintra, Portugal – is used for the footage of seven short (less than one minute) clips. These clips are mainly shots of exterior landscapes within a forest and with some centenary monuments near water courses. GAs are applied to edit the multimedia material. The goal is to develop a system with the ability to produce new scene's sequences, integrating different AV landscapes but, eventually, with similarities.

The first prototype developed is Web based thus accessible to the public. An important characteristic of this work is the existence of "interaction windows", spaces where the visitor can interact, intervene in the selection operation, and interfere in the narrative editing, evolution and final result.

## 3.2 Possible solutions

There are several general observations about the generation of solutions via a GA that should be stated:

- The fitness function should be carefully defined to prevent GAs from a tendency to converge towards local optima rather than the global optimum of the problem.
- It is essential to tune the parameters such as mutation probability and crossover probability, to find reasonable settings for the problem class that we are handling.
- Operating on dynamic data sets is difficult, as genomes begin to converge early on towards solutions which may no longer be valid for later data. Several methods have been proposed to remedy this by increasing genetic diversity somehow and preventing early convergence, either by increasing the probability of mutation when the solution quality drops (called Triggered Hypermutation), or by occasionally introducing entirely new, randomly generated elements into the gene pool (called Random Immigrants).
- GAs can rapidly locate good (sub-optimal) solutions, even for difficult very large search spaces. For some specific problems with more restricted search spaces there are traditional algorithms more efficient in finding an optimal solution than GAs.

Working with descriptions, such as MPEG-7 Video Descriptors, of multimedia documents that describe a broad range of features, suggests that one should tailor the existent descriptors set to a smaller subset easier to deal with and implement in a prototype. A possible set for the video descriptors should have macro level semantic information, such as summarisation, and also micro level, such as colour histograms. Therefore the set of video descriptors chosen are both the low level simple and automatically generated and high level human agent summarised semantic information about the same segment.

One concern about the boundaries is undertaken. The case where the descriptor of a segment surpasses a narrative scene boundary, e.g., a segment summarisation description that starts or finishes (timely) in the middle of a scene. Thus for a complete information descriptor one should assume the time boundaries of the descriptor instead of a more complex analysis of the possible scene borderlines.

The video segments may have any size (duration $\times$ frame rate) and the number of segments by video clip is variable. Therefore the size of the video document is variable. The relative

start time and the duration of each segment, within a video document, is of interest to explore (evolve) hence to code into the chromosome.

The fitness function evaluates layout colour similarities values using colour histograms from descriptors, such as `ColorLayout` (table 2.3, p. 36) or `GoFGoPColor` (figure 2.22, p. 37) between genes (within video segments) of different chromosomes at a micro level. At a macro level text annotation is required in order to have some semantic information; the `FreeTextAnnotation` (see figure 2.18, p. 32), `KeywordAnnotation` (figure 2.19, p. 33) and `StructuredAnnotation` (see figure 2.20, p. 33) descriptors may be used. Along with these two descriptors plus the start time and duration of the segment, referred above, there are two more characteristics, chromosome coded, useful for a first approach to a fitness function evaluation: segment presentation (a simple yes or no for playing) and shot distance type (close-up, mid-shot and long-shot). The evaluation may have several approaches: from the simple user choice of the video clips to a more complex evaluation of similar descriptors' values.

For the process of generating a new population with individuals from an existing one, using a GA, a possible sequence is presented in figure 3.1, p. 48. From the OLD population a selection method is applied with a probability for mating and crossing over. The individuals that do not mate are transfered without modification. The ones that are reproduced with crossover are next subject to the mutation operator with a very low probability. At the end every individual is presented for elimination or not as an additional selection operator which enforces some specific characteristics, e.g., this operator may be the human agent decision, for elimination, after visualisation of the resulting document (video clip) at any evolutionary step.

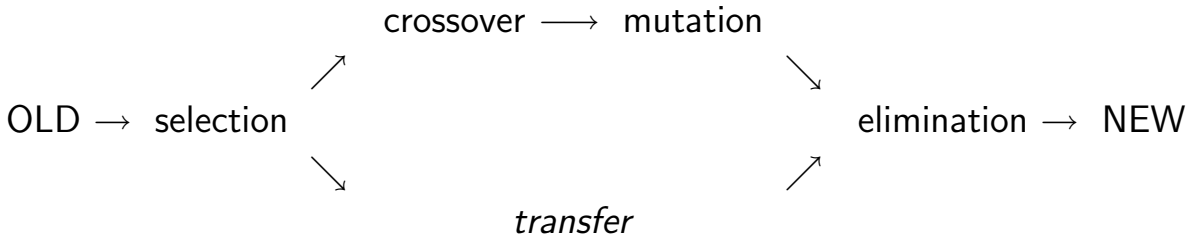


Figure 3.1: GA population generation sequence.

There are too many options that can be devised, developed and implemented. As a proof of concept, one must focus on a trail that leads to results, despite being aware of all those possible options. Nevertheless, a summary of the most significant combinations for two strategic possibilities is presented in the tables 3.1 and 3.2 below. These are examples of

strategies with combinations towards the approach explained earlier.

| **Descriptors** | |
|---|---|
| Low level | High level |
| GoFGoPColor | KeywordAnnotation |
| Texture | FreeTextAnnotation |
| | StructuredAnnotation |
| | Shot distance type |

| **Chromosome** |
|---|
| Codification |
| (Descriptors) |
| Segment presentation (yes/no) |
| Relative time of segment begin and duration |
| Document identifier (URI) |

| **GA Selection** | |
|---|---|
| Method | Probability ($p_S$) |
| Tournament (7-way) | 0.1 – 1.0 |
| + Elitism (10%) | |

| **GA Mating** | |
|---|---|
| Method | Type |
| Crossover | Two point |

| **GA Mutation** | |
|---|---|
| Method | Probability ($p_M$) |
| Segment presentation (yes/no flip) | 0.01 – 0.10 |

| **Elimination** |
|---|
| Method |
| By human agent explicit intervention |

| **Fitness** |
|---|

| Evaluation |
| --- |
| Congruent descriptors (distance to goal) |
| Segment relative duration |
| Number of segments in chromosome |

Table 3.1: Strategic option I.

For the strategic option I (table 3.1) a full use of the main textual annotation descriptors is applied, along with two relevant low level descriptors for colour and texture (see [ISO04, Manjunath02] for specific information) of the document. The chromosome codification is made using the descriptors and also additional information relevant for the final editing and presentation of the document. When preparing an Edit Decision List (EDL) the unique identifier of the temporal based multimedia document, the start time and the duration of the play are essential. The traditional steps of a simple GA are taken, but the selection is with Elitism which intends to improve convergence towards the optimal solution maintaining good solutions. The mating is a Two point crossover for the improvement of the "creativity" in exchanging segments. This way the segments may be edited in any sequence order, avoiding the caveat of One Point where genes before crossover point are always first. The mutation is a simple option of temporarily removing the gene (segment) from the individual (EDL) by hiding it in order to offer more diversity. Finally, a new step of elimination is added in a relevant way because of the multimedia production scope. Here, a human agent intervention may happen in choosing the better multimedia documents, i.e., removing the unwanted from the pool. The fitness evaluation merges two approaches: a quantitative one based on the number of segments and duration; and a qualitative one of congruence validation; the second approach computes the resulting sum of the descriptors distance to the goal: the closer the better.

This is an ambitious strategy where a significant number of descriptors are used and the fitness evaluation is rather complex. For a first approach, a simpler strategy should be used, one with only the relevant descriptors for the kind of multimedia documents in use (video clips) and a fitness equation that computes in feasible time and allows to early analyse the results.

Therefore, another strategy is presented below in table 3.2 where descriptors and fitness are engineered to cope with the implementation of the prototype.

| **Descriptors** | |
|---|---|
| Low level | High level |
| GoFGoPColor | KeywordAnnotation<br>FreeTextAnnotation<br>Shot distance type |

| **Chromosome** |
|---|
| Codification |
| (Descriptors)<br>Segment presentation (yes/no)<br>Relative time of segment begin and duration<br>Document identifier (URI) |

| **GA Selection** | |
|---|---|
| Method | Probability ($p_S$) |
| Tournament (7-way)<br>+ Elitism (10%) | 0.1 – 1.0 |

| **GA Mating** | |
|---|---|
| Method | Type |
| Crossover | One point |

| **GA Mutation** | |
|---|---|
| Method | Probability ($p_M$) |
| Segment presentation (yes/no flip) | 0.01 – 0.10 |

| **Elimination** |
|---|
| Method |
| By human agent explicit intervention |

| **Fitness** |
|---|
| Evaluation |
| Congruent descriptors (distance to goal) |

Table 3.2: Strategic option II.

The strategic option II (table 3.2) does a tailored use of the main textual annotation descriptors, along with the more relevant low level descriptor for the colour analysis of the document. The chromosome codification maintains the use of the descriptors and the additional information relevant for the final editing and presentation of the document. The traditional steps of a simple GA are taken, but the selection is a simple deterministic tournament with a probability ranging from one tenth to one half with the additional 10% of Elitism. The mating is a One point crossover to maintain the simplicity in exchanging segments. The mutation is a simple option of temporary removing the gene (segment) from the individual by hiding it in order to offer more diversity. Finally, a new step of elimination is added in a relevant way because of the multimedia production scope referred above (p. 50). The fitness evaluation is a qualitative one of congruence validation: the similar the better.

As stated before, there is a relatively large number of variables and its possible values for this kind of problems. Several methods of selection for mating do exist and many strategies can be implemented using only one or combining two or more methods. The tuning of the probability for selection and mutation operators offers many variations. The mutation can be a simple probabilistic step in the GA sequence or be implemented as a more evolved strategy, such as the Triggered Hypermutation that introduces mutation occasionally. Also islands of populations evolving in a distributed manner and Random Immigrants may be used in order to increase diversity avoiding sooner convergence. Therefore one may see all the natural examples that exist in our own planet and learn with nature on how to evolve species and naturally select them. One should focus on a feasible solution that does not restrict future work improvements and addition of new features.

## 3.3 Chosen solution

The strategic option II (table 3.2, p. 51) is the one chosen for the implementation of the prototype. Seven described video clips (individuals) will be used as the initial population. Each video clip has three segments (number of *super*-genes). The `GoFGoPColor` descriptor is the lower level descriptor for objective comparisons between segments of evolving individuals. The keyword and free textual annotation descriptors along with the shot distance type are the subjective ones used for evaluation of the evolution. The chromosome genes are coded with the descriptors plus the segment duration and the relative time of the segment start. The data types for the implementation are the most adequate for each descriptor. The representation of time is the one explained in section 2.3, p. 33, and based on the well known ISO 8601:2000 International Standard [ISO00]. The selection

method is a deterministic 7-way tournament, with a size of seven as a first approach to a tournament where all (seven) the individuals can compete at the same time. Each individual's probability of being at the tournament ranges from one tenth to one, and is selected for a one point crossover using Fitness Proportionate method. Exception is made for the top 10% of the individuals where Elitism is applied. Mutation is used with a very small probability value (ranging from 0.01 to 0.1) to decide if the segment is presented or not. An additional elimination method is applied using the interaction with the human agent, asking for a judgement on which individuals (multimedia documents) should prevail for the next generation and which are eliminated.

The individual, a chromosome gene/descriptors bounded, elected to be the best solution is played using the chromosome genes extra information about times, duration and presentation as an Edit Decision List. This is a way of playing a final result, as if an editing and rendering was made on the document, but avoiding the resources consumption of such a heavy task. Besides the document (video clip) preview available by playing the segments continuously, a rendered copy of the best viewable solution may be generated in order to be portable to another system or mobile device.

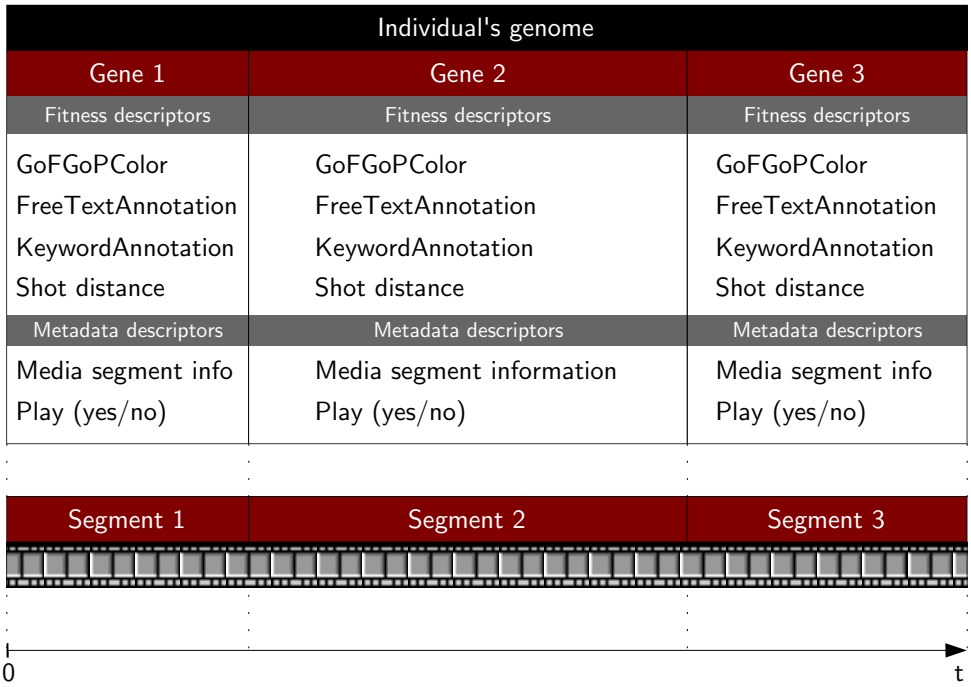


Figure 3.2: Three atomic segments (genes) example.

The stop condition can be one of the two options: a numerical value as a parameter for limiting the number of possible generations; or explicitly enforced by the user agent at

any evolutionary step (generation) if a desired result has been achieved.

A gene or an allele, and a document segment (the descriptor time limits are the segment bounds) are considered atomic for all the processing by the system. This is a restriction in the scope of our solution, nevertheless, it can be changed, for example, to non-atomic segments in order to improve diversity in the crossover points. But, for this solution, each gene contains the full set of descriptors, for the characterisation of a document (video) segment (figure 3.2, p. 53).

An extra descriptor should be used, chromosome coded, to bound the video segments (genes). A sophisticated algorithm could be used, as in [Chiu00] and described in section 2.4, p. 43, for automatic segmentation purposes. The solution chosen for solving this problem is a simple one that eases the prototype development: no extra descriptor will be used to bound the segments and restrict the other descriptors values' mapping to a specific (video) segment. The MPEG-7 XML description language already provides tags to explicitly divide the video document into several video segments and, thus, describing each one separately.

The fitness function $f$ is formulated (equation 3.11) as the sum of all chromosome genes' resulting values. Each gene (segment) value is the result of the descriptors' distance metrics weighted sum. The evaluations for these metrics are: the similarity matching of `GoFGoPColor` descriptors of a segment and the specified goal colour; the `KeywordAnnotation` similarity matching using a proposed specific algorithm for distance computation; the `FreeTextAnnotation` similarity matching using a developed hybrid algorithm, that uses Levenshtein[1] distance algorithm; and an *ad hoc* similarity matching function for the Shot Distance proposed descriptor. All descriptors distance formulas domains are normalised to a scale of $[0.0, 1.0]$, 0.0 is the worst and 1.0 the best, with the transformation $l_d = \frac{1.0}{1.0+l'_d}$ where $l'_d$ is the original descriptor value.

A fitness value of one (1.0) is the optimal solution and it means that the distance of the individual's evaluated parameters to the goal is zero, hence it reached the goal. As the distance of the individual's descriptors to the goal increases the smaller is the value of $f$ for the video segment, thus, less fitted.

The prior statements and considerations can be formally defined. The $I$ set includes all the individuals $i$ of the current generation step. The $D$ set is a chosen subset $m$ from all the possible MPEG-7 descriptors $M$ plus any *ad hoc* ones ($A$ set) not defined in the MPEG-7 standard:

[1] `http://www.nist.gov/dads/HTML/Levenshtein.html`

$$f \;:\; I \rightarrow [0,1]$$

$$\begin{aligned} I &= \{\text{population of individuals } i\} \\ D &= M \cup A, \text{ where } m \subset M \end{aligned}$$

Let $m$ be composed by $C$ – colour histogram of a group of frames, $F$ – free text annotation, $K$ – keyword annotation. The set $A$ has an unique *ad hoc* descriptor which is named $S$ – shot distance type:

$$D = \{C, F, K\} \cup \{S\} = \{C, F, K, S\}$$

The equations for the distance measurement between each individual's descriptor, at some segment, and the goal (purpose to reach) are presented below:

$$C = d(C_i, C_{goal}) \tag{3.1}$$
$$F = d(F_i, F_{goal}) \tag{3.2}$$
$$K = d(K_i, K_{goal}) \tag{3.3}$$
$$S = d(S_i, S_{goal}) \tag{3.4}$$

Where the formula, from the specification in [Manjunath02], for matching and measuring the distance between two distinct `GoFGoPColor` descriptors, $G$ and $G'$, is presented in the equation 3.5 where $C = d(G, G')$. The resulting coefficient order from the zigzag scanning after the DCT transformation is $n$. This is the number of coefficients for the histogram of colours.

$$C = \sum_{n} |G_n - G'_n| \tag{3.5}$$

This is a much more efficient descriptor for colour similarities. In the case of the colour layout descriptor, the resulting coefficient order from the zigzag scanning after the DCT transformation is $n$. This parameter is dependent of the `CoefficientPattern` that specifies the number of DCT coefficients and the number of luminance and chrominance coefficients `NumberofYCoeff` and `NumberofCCoeff` respectively. Thus, $n$ ranges from 1 to a variable number of 3, 6, 10, 15, 21, 28 or 64 (table 2.3, p. 36). The $w$ is a parametric weight for the coefficients $Y$, $Cb$ and $Cr$. For the implementation of the equation 3.6 in

the scope of video documents one should weight more the luminance $Y$ coefficient. The sum of all weights ($w$) should be equal to one. For two distinct colour layout descriptors, $L$ and $L'$, the formula for distance measurement is presented in equation 3.6 [Manjunath02].

$$C = \sqrt{\sum_n w_y(LY_n - LY'_n)^2} + \sqrt{\sum_n w_{cb}(LCb_n - LCb'_n)^2} + \sqrt{\sum_n w_{cr}(LCr_n - LCr'_n)^2} \tag{3.6}$$

So, the option for the `GoFGoPColor` descriptor instead of the `ColorLayout` is well justified.

The Levenshtein [NIST04] (algorithm 3.8) introduces a penalty for large strings. That kind of penalties are unwanted on the developed approach. Therefore, the metric used for measuring the free text similarity $F = d(s_1, s_2)$ of the `FreeTextAnnotation` descriptor is the hybrid algorithm 3.7 presented in three steps below.

(3.7)

1. 
```
s1 = bigger string of words.
s2 = smaller string of words.
if |s1| = 0 and |s2| = 0 then distance = 0
else if |s1| = 0 then distance = |s2|
else if |s2| = 0 then distance = |s1|
```
2. 
```
else
 distance = Integer.MAX_VALUE
 Loop (2 x (|s2| - 1) + (|s1| - |s2|))
  //              s1 subset
  //             |-|
  // s1          |-----------------------------|
  // s2 |---------| -> ...
  //             |-|
  //              s2 subset
         distance = min(distance,Levenshtein(s1 subset, s2 subset))
```
3. 
```
return distance
```

An optimisation may be implemented, being the two segments of equal size, using the Hamming algorithm instead of the Levenshtein for the segments matching. Also, other variations of the hybrid type, such as using the sliding segment, as in the hybrid 3.7, but only inside the boundaries of the bigger string, may be implemented.

The details of the Levenshtein distance algorithm are exposed below where a two-dimensional matrix, $m[0..|s_1|, 0..|s_2|]$, is used to hold the edit[2] distance values:

Adapted from: `http://www.csse.monash.edu.au/~lloyd/tildeAlgDS/Dynamic/Edit/`

$$
\begin{aligned}
m[i,j] &= d(s_1[1..i], s_2[1..j]), \qquad i = 1..|s_1|, j = 1..|s_2| \qquad (3.8)\\
m[0,0] &= 0\\
m[i,0] &= i, \qquad i = 1..|s_1|\\
m[0,j] &= j, \qquad j = 1..|s_2|\\
m[i,j] &= min(m[i-1,j-1] \quad + \quad \text{if } s_1[i] = s_2[j] \text{ then } 0 \text{ else } 1 \text{ fi},\\
&\qquad\qquad m[i-1,j] + 1,\\
&\qquad\qquad m[i,j-1] + 1), \qquad i = 1..|s_1|, j = 1..|s_2|
\end{aligned}
$$

The matrix $m$ can be computed row by row, using dynamic programming. Row $i$ depends only on row $i-1$. The time complexity of this algorithm is $O(|s_1| \times |s_2|)$. If $s_1$ and $s_2$ have an approximated length of $n$, this complexity is $O(n^2)$, which is better than exponential. The prototype `FreeTextAnnotation` descriptor distance function uses this approach for the metric computation part where Levenshtein is used.

The metric used for measuring the textual annotation distance of the `KeywordAnnotation` descriptor is a specific one. Two keywords distance $K = d(s_1, s_2)$ is computed using the following proposed algorithm 3.9:

(3.9)

```
1.  s1 = set of words (haystack).
    s2 = set of words (needles).
    #s2 = number of words in the s2 set.
    if |s1| = 0 and |s2| = 0 then distance = 0
    else if |s1| = 0 then distance = |s2|
    else if |s2| = 0 then distance = |s1|
2.  else
     distance = min(|s1|, |s2|)
     Loop #s2
      if s1.contains(s2.word) then distance = distance - |s2|
3.  return distance
```

[2]Levenshtein algorithm is also known as Edit Distance because it measures the number of character edits, insertions, removals, substitutions needed to transform one string into the other.

A "perfect" matching (distance=0) is considered in a loose sense if s1 contains all the s2 keywords or if all s1 keywords are contained in s2.

Camera shots may be classified with three main values for the respective domain and each one is an integer value of 0, 1 and 2 for the close-up, mid-shot and long-shot respectively. A similarity (distance) function may be defined as the absolute value of the algebraic difference between two shots. But, a camera shot can be combined with more than those three values, several pull and pushes, if a combination of different moments is made, may define a shot somehow between two of the three main values. Therefore the values should be between 0.0 and 1.0, where the 0.0 is for the close-up, 0.5 for the mid-shot and 1.0 for the long-shot.

$$\begin{aligned} \mathbb{D}_S &= \{s \in \mathbb{R} : 0.0 \leq s \leq 1.0\} \\ S &= d(s_1, s_2) \\ &= |s_1 - s_2|, \qquad s_1, s_2 \in \mathbb{D}_S \end{aligned} \tag{3.10}$$

Because the MPEG-7 standard [ISO04] does not specify a camera shot descriptor we propose to improve the `CameraMotionType` defined in the MPEG-7's Visual 2001 XML schema definition with the schema component defined in figure 3.4, p. 58. An example of how we are using the descriptor to describe the video segments is presented below (figure 3.3).

```
<VisualDescriptor xsi:type="CameraMotionType">
  <ShotDistance value="0.7"/>
```

Figure 3.3: `ShotDistance` example.

Let $V_i$ be the set of an individual's video segments with descriptions, $w_d$ the weight for the specific descriptor $d$, $g$ the number of genes/segments per individual, and $f_i(V_i)$ the fitness function applied to all those segments:

$$f_i(V_i) = \sum_{v \in V_i} \frac{1}{g} \sum_{d \in D_i} w_d d(v), \qquad i \in I \tag{3.11}$$

For the specific solution chosen with a restricted subset of descriptors, stated above, the equation 3.11 can be written as (where all the descriptors are normalised to a scale of

```
<complexType name="CameraMotionType" final="#all">
  ...
  <element name="ShotDistance">
    <complexType>
      <attribute name="value" type="mpeg7:zeroToOneType" use="required"/>
    </complexType>
  </element>
```

Figure 3.4: `ShotDistance` MPEG-7's Visual 2001 schema proposal.

$[0.0, 1.0]$, with the transformation $l_d = \frac{1.0}{1.0+l'_d}$):

$$f_i(V_i) = \sum_{v \in V_i} \frac{1}{g}\big(w_C C(v) + w_K K(v) + w_F F(v) + w_S S(v)\big), \qquad i \in I \tag{3.12}$$

Where $C$ is the `GoFGoPColor` computed distance (see equation 3.1 and 3.5), $K$ and $F$ are the `KeywordAnnotation` and `FreeTextAnnotation` respectively computed distances (equation 3.2, 3.3 and algorithm 3.7, 3.8, 3.9). Finally, $S$ is the camera shot computed distance value using the equation 3.10, p. 58. This equation, for the camera shot distance, is a simple metric able to, efficiently, obtain the information for the purpose of measuring distance differences.

The sum of all the descriptors' weight values must be equal to one ($\sum_{d \in D_i} w_d = 1$). The weight is needed for setting the relevance for the different descriptors relation. In this approach, the fitness function is implemented considering $w_K >> w_F > w_C = w_S$. The semantics are: if `FreeTextAnnotation` descriptor is omitted by the user then `KeywordAnnotation` weight $w_K = w_K + w_F$ and $w_F = 0$. And if `KeywordAnnotation` descriptor has no keywords inputed by the user then $w_F = w_F + w_K$ and $w_K = 0$. If the case where neither `FreeTextAnnotation` nor `KeywordAnnotation` are inputed then $w_F = 0$, $w_K = 0$, $w_C = 0.5$ and $w_S = 0.5$.

The proposed genetic algorithm flow diagram has the following description: it starts with an initial population where each individual initial fitness value is set, and then the evolutionary loop begins. The validation for the goal achievement is applied, and if any individual is the solution for the problem then the loop ends. If not, then a fitness evaluation is used for the selection step. At this step is applied a method of selection based on each individual fitness value and the probability of entering a tournament. Several individuals, depending on the selection probability, are elected for mating, and a crossing over technique is applied. This selection for crossing over is made by always pairing two

individuals. Elitism, if used, guarantees that, at least, some defined number of the best individuals are elected for mating. After the crossing over, mutation is the next step, individuals that were mated may be mutated but with a very low probability. At this step, one or a combination of several mutation techniques are applied. For the ones that weren't mated a shortcut towards the step of elimination is taken. At this step, one that every individual has to pass, a choice of individuals to be eliminated is made and the ones that are chosen are disposed. The ones that stay are the population new generation. Then the first step of the loop takes place again.

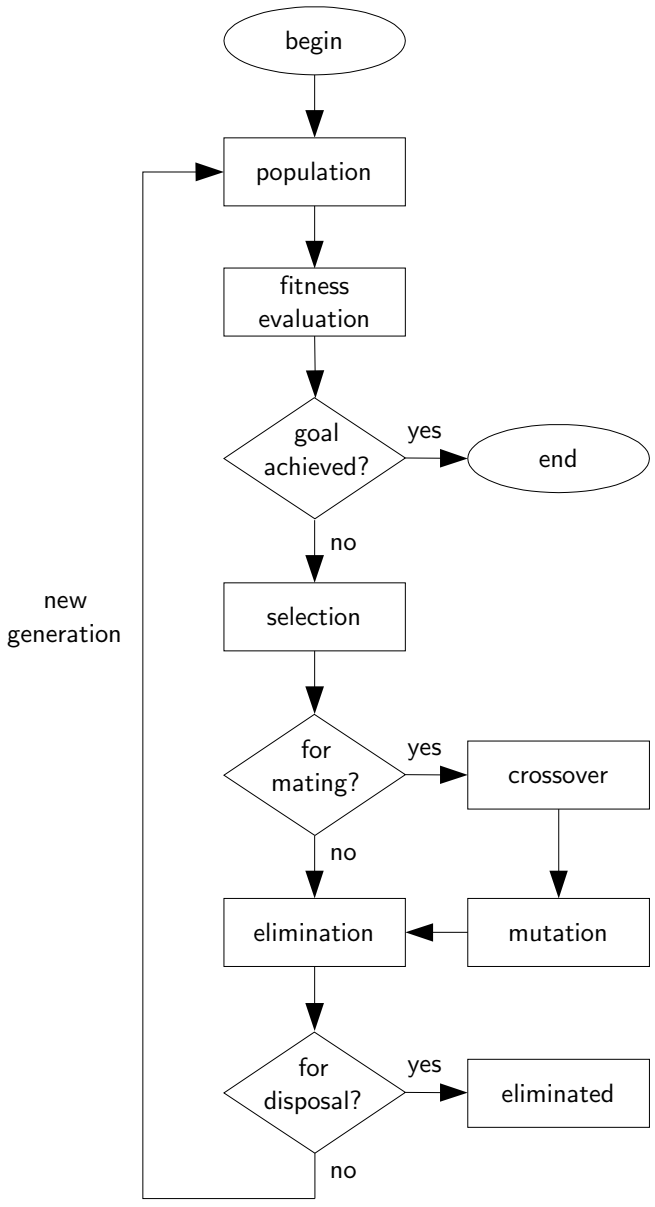


Figure 3.5: MovieGene's Genetic Algorithm.

To summarise, a complete approach ranging from the concepts, the fitness equation and

algorithms until the work flow of the algorithm for the evolutionary module is presented and described. This flow diagram is presented in figure 3.5, p. 60. After this presentation we are ready for the implementation of the operational prototype described in the next chapter 4. Here, the implementation of the metrics and algorithms strictly follows the conceptual approach described.

# Chapter 4

# Prototype implementation

This chapter describes the implementation of the MovieGene's prototype. It starts by presenting the evolutionary module integrated in the multimedia system architecture and the interaction between all the components of this multimedia production application. Next, the requirements and some restrictions relevant for this work are summarised, just before presenting the Human-Machine Interface of the whole system. The interface with the human agent is a complex task that deals with ergonomics. Finally, the tests' environment description, results and comments are exposed to complete the chapter.

## 4.1 Evolutionary module and system architecture

The MovieGene's system architecture (figure 4.1) is conceived to be platform (operating system under specific hardware) independent, Web distributed and accessible, as much as possible, due to technological restrictions and a MSc thesis' scope.

The whole system was divided into three major parts: Repository, Application and Interface. It is an agile view of conceptually independent ways of persistently storing system data, processing the data using a specific engine, and finally interacting with a human agent (or another kind of agent).

**Interface** It's the MovieGoal's Black Box (BB) which interacts with the Application and uses a graphical library for the HMI. The BB is developed as a Java application or an Applet. It is a window for the human agent interactive access to the evolving videos that are to be produced. The agent uses interface tools and options to first choose the environment goals for the final video and then defines a stop condition.

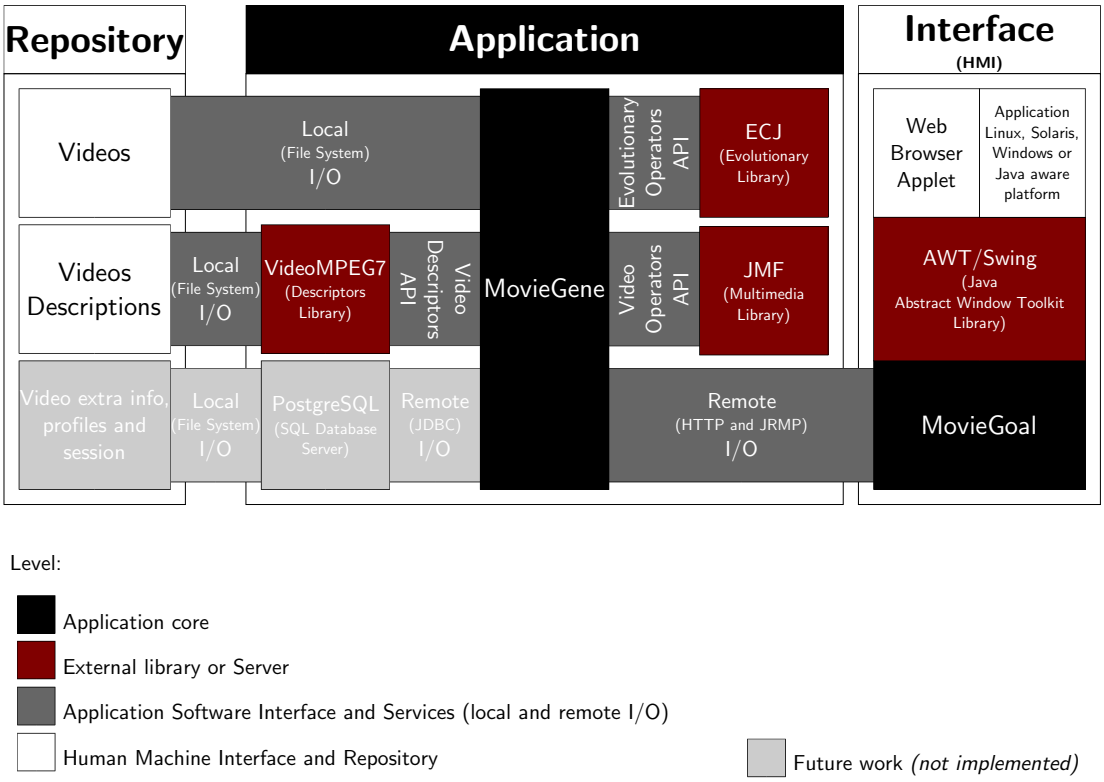


Figure 4.1: MovieGene's system architecture.

After that, the process may be supervised or left unattended until the stop condition is reached. The result is then showed as a new produced video document.

The HMI may serve as a way for the agent to interact as a genetic operator choosing which individuals (described videos) of the population should survive and which do not adapt to some semantic extra goals, not formalised at the goal stage, for the final video clip.

**Application** The core of the prototype is the MovieGene's BB linked with the three fundamental libraries for specific tasks: VideoMPEG7 the MPEG-7 library for reading and writing multimedia (video) documents descriptors in MPEG-7 format and coded in an XML file; JMF which provides low level methods for multimedia (video) objects reading, writing and editing; ECJ an excellent evolutionary Java based library for the implementation of the GA module which will be the core engine doing all the work evolving the descriptors of the media documents and updating data for serving the client agents (MovieGoal).

One can envisage the MovieGene's BB as an engine, powered by ECJ, being fed by the multimedia (video) documents, using the VideoMPEG7 at the same time

accessing the descriptors, running JMF for video manipulation, and serving the client's requests by generating new video productions as the result of the processing. All this work flow happens on a Java Virtual Machine (JVM) using the Object Oriented programming paradigm.

**Repository** There is always a need for a repository when the data must be persistently stored. In this system, the repository serves as the container for the original multimedia documents and also for the new produced documents along with the respective media descriptors.

It's planned as future work to implement this prototype as a production application, consequently the repository must be distributed and should evolve to a richer data warehouse, managing all the several user interactions and sessions, in order to promote concurrent access and guarantee data consistency and coherency. A relational database system, such as PostgreSQL, is foreseen as the best possible solution.

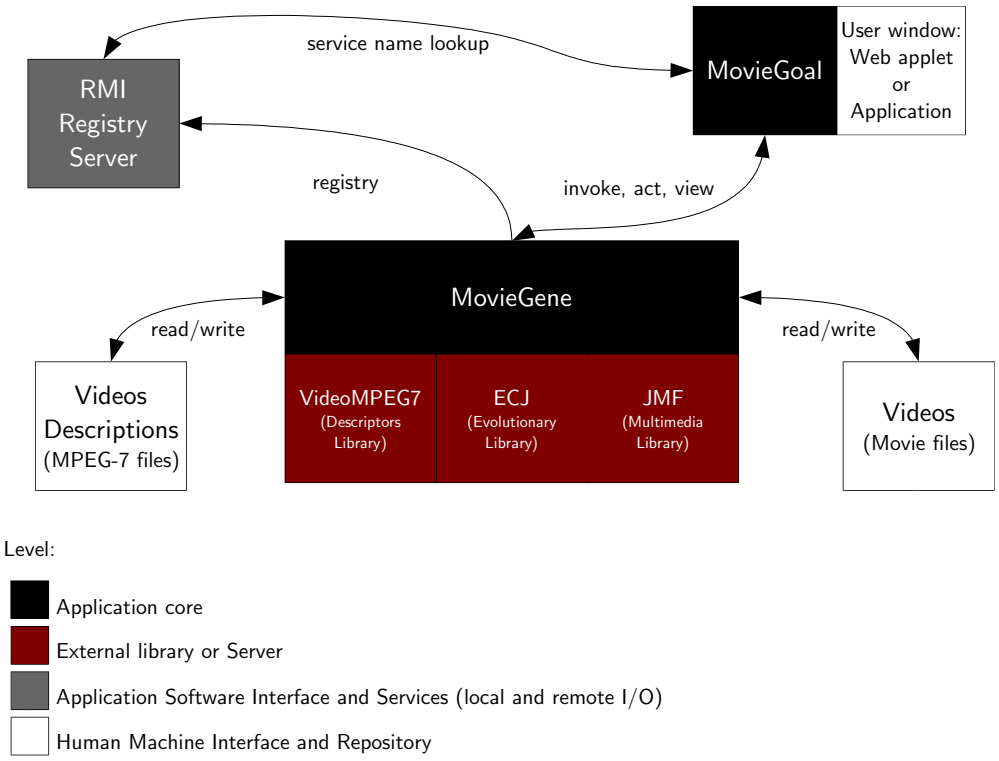


Figure 4.2: MovieGene's interaction architecture.

The MovieGene's interaction architecture (figure 4.2) is simple: the MovieGene's appli-

cation acts as a server. When first running it registers itself in the local Java Remote Method Invocation (RMI) server. On the same machine (platform) an HyperText Transfer Protocol (HTTP) server must serve the Web interface to permit the human agent Web access to the MovieGoal's application interface. The MovieGoal finds the MovieGene's services doing a service name lookup in the platform from where it was first invoked. After that it starts dealing with MovieGene's services.

The MovieGoal is the HMI for accessing the multimedia production system, the actions requested or being allowed to the human agent are interfaced through MovieGoal. It shows samples of the multimedia documents (videos), requests user options and communicates them to the service.

The MovieGene access to the documents and the descriptors is mainly for reading in order to process them, but there are some stages where writing is needed for setting the population, converted from MPEG-7 descriptors, or changing the goal and setting a new objective for the final multimedia document production and storing results.

Although the implementation of the prototype is the first one and far from a production release, the major functionalities are implemented. MovieGene has the main components of an evolutionary aided multimedia production system: it includes setting population, goal and genetic parameters, besides the evolutionary engine itself. The development is made towards a general framework to promote evolution of the whole application. The `nach.moviegene.mpeg7` [Henriques04] package implements all the needed MPEG-7 descriptors, including the ones not used and the extra `ShotDistance` proposed by this thesis. While doing metrics research, several approaches were done to the textual distances. The work results are programmed and available for use in the `nach.moviegene.gene.MovieGeneProblem` class [Henriques04]. Some use the Levenshtein (also known as Edit Distance) well know algorithm for two strings (set of characters) distance computation. Others use hybrid approaches using the Levenshtein within proposed specific algorithms.

Finally, for a complete reference of all the development, including class diagrams using Unified Modelling Language (UML) notation and the API, see the "MovieGene's Reference Manual" [Henriques04]. The implementation is more than 8000 lines of Java code to accomplish the MovieGene engine and the MovieGoal client interface applications. The reference manual is rather large and not appropriate to be include within this document.

## 4.2 Requirements and restrictions

There are prototype requirements generally stated in the objectives section 1.3, p. 2. Additionally, there are some specific requirements for each prototype module improving the whole system to be more than the simple sum of its modules.

The MovieGene's BB is required to link to all the needed libraries, acting as the aggregator of all needed individual operations. It is required to include the developed evolutionary algorithm (as a GA) for solving the problem of evolving the videos as stated before. Thus, there are some important requirements to describe:

**ECJ** The evolutionary operators library:

1. API for GA development.
2. Hidden library with a method to retrieve at any evolutionary step the current results.
3. Support for several well known genetic selection methods, such as Tournament or Roulette Wheel, and also the Elitism characteristic.
4. Support for several well known genetic crossover methods, such as One point, Two points or Cut and splice.
5. Support for fixed-length and variable-length genomes and arbitrary representations.

**JMF** The video operators library[1]:

1. API for multimedia documents interaction: reading, writing, splitting and merging.
2. Support for well known multimedia formats (codecs), such as MPEG 1, 2 or 4 standards and Intel Video 5.0 for the video formats. Also, audio codecs, such as MPEG 3 and the raw Pulse Coded Modulation (PCM).
3. Support for well known multimedia file formats, such as Audio Video Interleaved (AVI).
4. Support for filtering in video editing.

[1] JFFmpeg (`http://jffmpeg.sourceforge.net/`) is being used to complement the lack of support, by the JMF library, for several relevant codecs. The two libraries are in an early stage of development, some codecs have random failures (*crashes*)!

**VideoMPEG7** The MPEG-7 operators library[2]:

1. API for MPEG-7 descriptors interaction: reading and writing.
2. MPEG-7 [ISO04] standard compliance.
3. Specific support for video descriptors read and write operations.

The whole application has some restrictions that are relevant to be stated here. These are the restrictions for the current version of the prototype. Of course these, mainly technological restrictions, such as a limited set of video formats[3], may in the future be surpassed:

1. The set of multimedia documents are only video clips.
2. Video codec used is MPEG-4 (DivX 5) in an AVI file format container.[4]
3. Video clips' duration has no restrictions, but the ones for testing are under 1 min.
4. Video frame size used for the testing clips is $160 \times 120$ pixels.
5. Video frame rate with no restriction, but the clips are all of 15 fps.
6. Audio breaks and structure will not be considered in the multimedia document editing.
7. The descriptors must be complete, i.e., in every video clip the time line is splited by the descriptors boundary of the segment.
8. Scenes will not be automatically detected hence not accurately considered.
9. The histogram for the GoFGoPColor MPEG-7 descriptor will not be automatically generated hence not accurately considered.
10. User session and profile management are left for future work. Currently it supports only one user and one session.

[2]The Avidemux (`http://fixounet.free.fr/`) application was used to assist, playing with time and frame counter, the manual videos annotation.

[3]The JMF formats: `http://java.sun.com/products/java-media/jmf/2.1.1/formats.html`

[4]MPlayer (MEncoder – `http://www.mplayerhq.hu/`) and also Avidemux were used to convert the videos from the original format.

## 4.3 Human-Machine Interface of the prototype

The human agent interface is Web based. Also it can be used as a stand alone client application running in any Java aware platform. It uses the Java Abstract Window Toolkit (AWT) and Swing (JFC/Swing) to design the graphical components for the interactive screen.

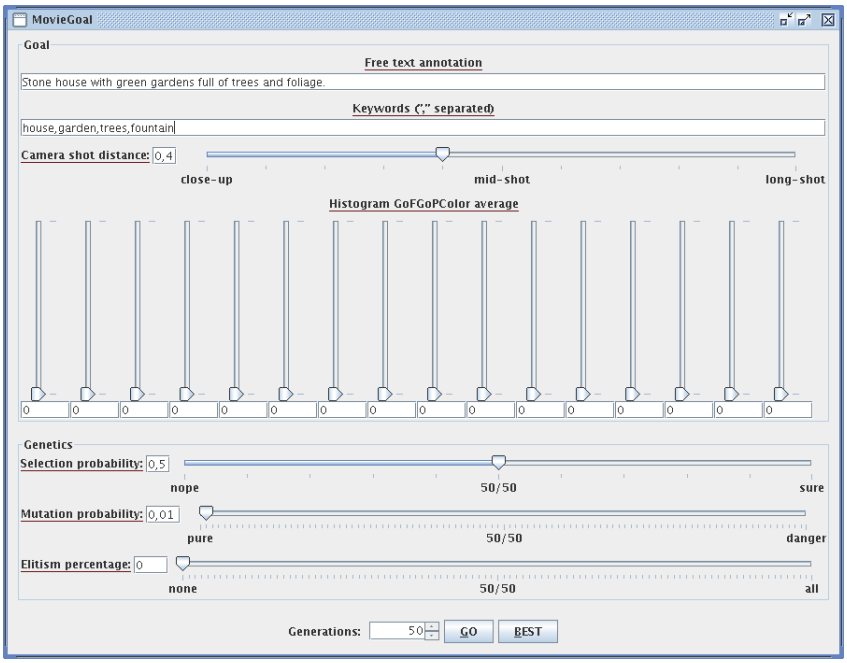


Figure 4.3: HMI goal and genetics screen.

The interaction with the agent is kept as simple as possible. An initial screen (figure 4.3) asks for the goal, using text boxes, histogram sliders and buttons where the user can input the intended characteristics for the final document (video clip). At this stage of the development the interface is directed to professional users for testing the presented concepts. For the general user it should be improved abstracting the inner details and specific terms, such as “goal”, “genetics” or “mutation”. Nevertheless, the graphical components are always very explicit to attend each human agent preferences. Redundancy of formatted text boxes along with scaled sliders is used for the values input. In the future, as mentioned, all this interface should be abstracted with graphical components and concepts for the general human agent, e.g., histogram components should be substituted by a painting board where the agent brushes some colours and next the application calculates the histogram coefficients.

The initial parameters for the goal are the free text box where the user can write sentences

with semantic meaning. The keywords text box is for refining semantic concepts for the search, restricted to single words separated by commas. The proposed camera shot distance descriptor offers a numeric box and a slider. The user may choose any of the two ways for inputing values. The slider restricts the values to decimals by snapping to the nearest decimal. The colour histogram, as the shot distance, offers two ways for the coefficients input.

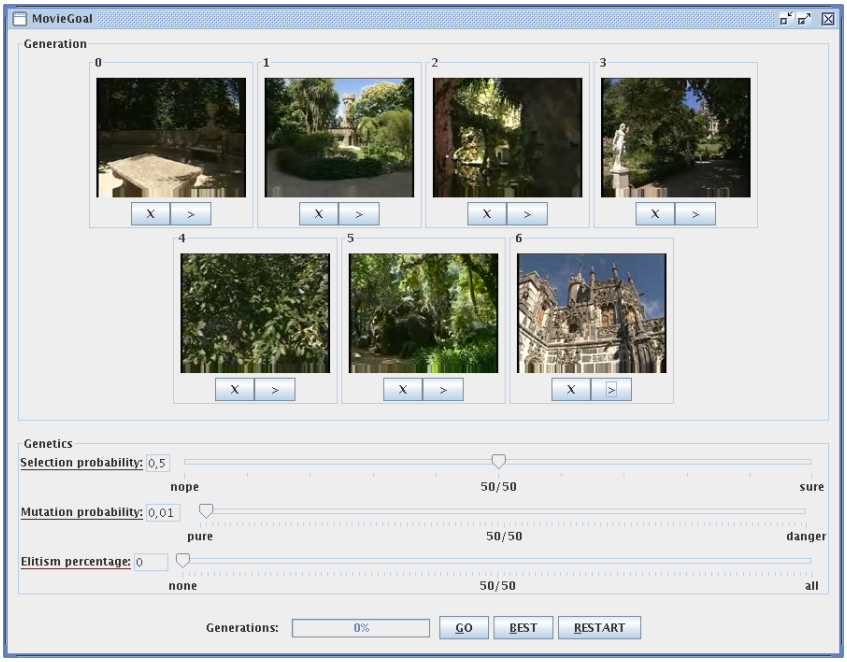


Figure 4.4: HMI evolutionary step screen.

Genetics parameters are the probabilities of selection for crossing over and mutation, and the percentage of elitism (from the population). Also, the number of generations may be inputed in the spinner at the bottom of the interface, near the "GO" button, in a range from one to one thousand. Future improvements are easy to made regarding the changing of the selection method and even the combination of several methods, or the continuous evolution until stop.

After the initial parameters setting, the evolutionary loop begins and at each step the resulting videos are presented (figure 4.4). The user has the like/dislike option, using the "X" button, to deprecate (eliminate) a specific video for the next round. A simple carry on (towards the best) option, using the "BEST" button at any stage (generation), is available to allow the process completion without user intervention.

Also, the user may play (figure 4.5) each one of the videos to make a better decision on elimination or not. If elimination is made, hitting the "X" button, then the video clip is

also removed from the screen, besides being eliminated from the population (figure 4.6).

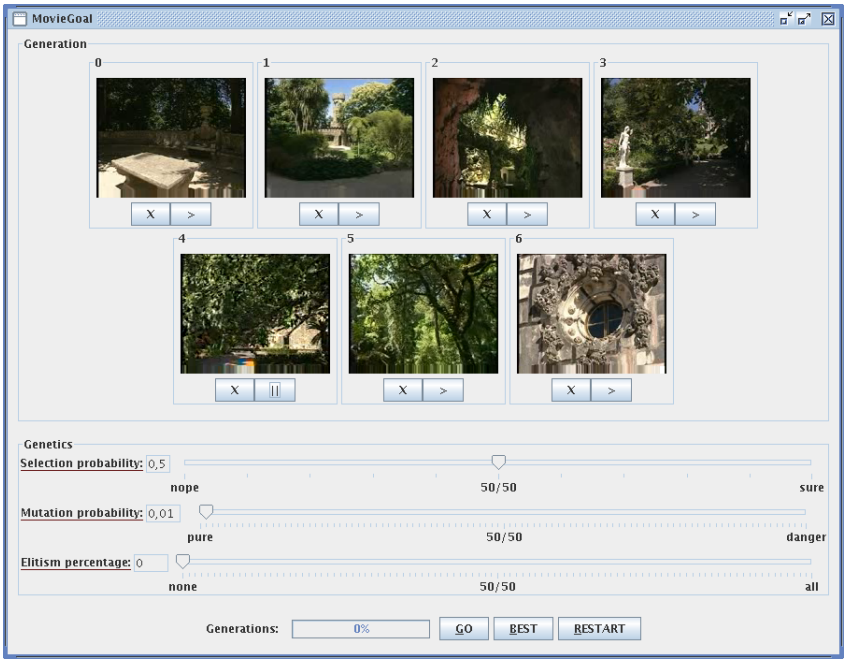


Figure 4.5: HMI playing video clip number 4.

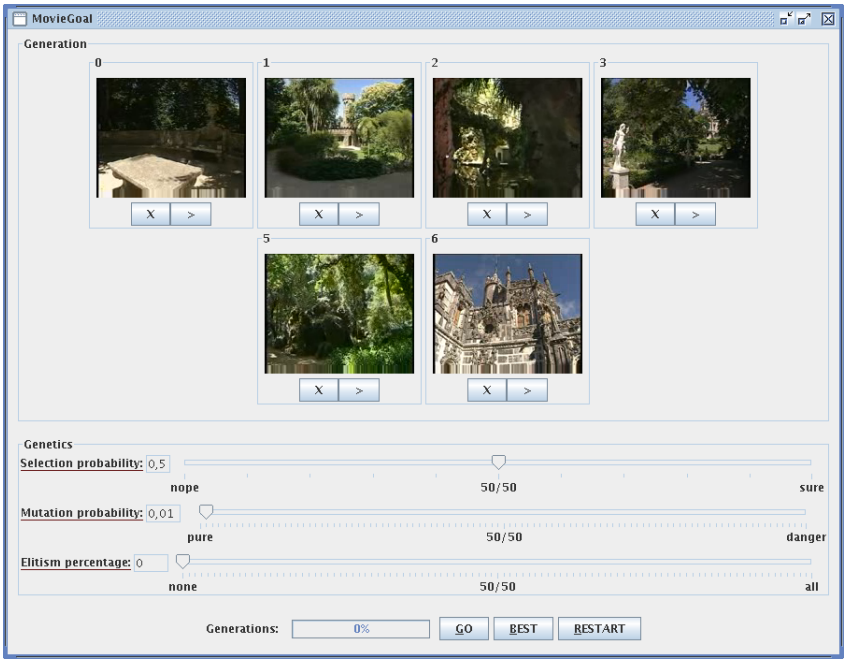


Figure 4.6: HMI after video number 4 elimination.

Another option is to run the process in an unattended way, instead of being asked at every step, until the best video clip is reached. A screen with the evolutionary process

progress is shown (figure 4.7). Of course, the user may cancel the process at any time by hitting the "RESTART" button.

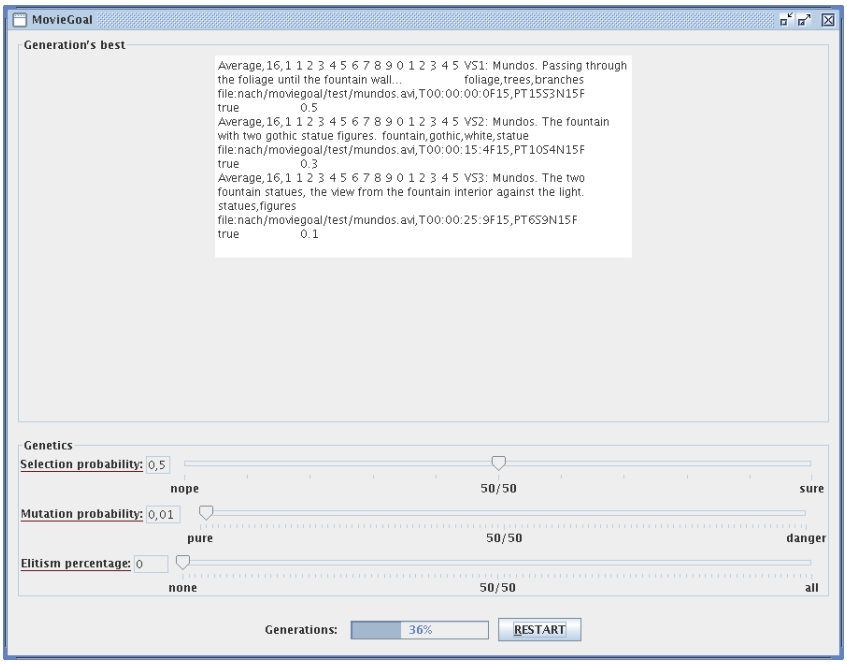


Figure 4.7: HMI running for the best.

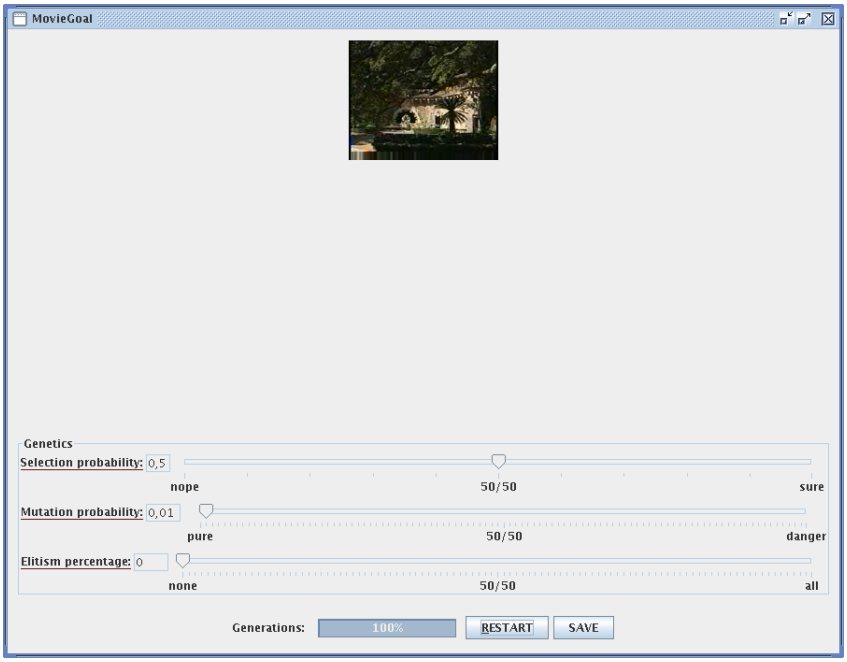


Figure 4.8: HMI best video play.

Finally, after the whole evolutionary process, the best video is presented for playing

(figure 4.8). Here, the user may simply visualise the continuous play of the resulting video or choice to store the clip in a local place. For this storing a rendering of all the segments composing the EDL is made.

## 4.4 Prototype system testing

### 4.4.1 Environment's description

The description of the environment where the tests take place is crucial for the analysis of the performance results. Future work may compare results if on similar hardware and software.

The hardware is composed by one computer for the preliminary tests. This machine acts as a server and a client. For simple testing the Web browser will be left aside and the stand alone application will be used. Nevertheless, the coding of all the application components, mainly the MovieGoal was made taking into account the Web access to the service, where MovieGoal is the client accessed by a Web page served by an HTTP server (Apache).

**Processor** Intel Pentium M (Centrino), mono, 32 bits, 1.4 GHz, 1 MB L2 cache, 32 kB Instructions and 32 kB Data L1 cache.

**Memory** 512 MB SDRAM of main memory with 496 MB available and 16 MB allocated (BIOS) for the video card.

**Disk** EIDE ATA100, 5400 RPM, 1740 kB of cache memory.

**Bus** System communication bus of 32 bits at 99 MHz. The bus between the processor and the video card is AGP.

**Video** 1024×768 pixels of frame size and 32 bits of colour depth.

**Network** Ethernet card at 100 Mbps.

The software platform for the development and testing will be Sun's JVM 1.5 (Java Development Kit (JDK) 1.5.0+02) over Debian GNU/Linux Testing distribution with Kernel 2.6.6. The libraries are the ECJ 12 and the JMF 2.1.1e cross-platform (JAR) version.

### 4.4.2 Tests comment and results statistics

The tests results are subjective because multimedia documents are evaluated by humans and may or may not be pleasant for distinct groups of humans. Nevertheless, unattended performance tests are made to measure the processing time without the interactive decisions. Also, quantitative results are obtained retrieving the fitness of the best individual, besides the computation time.

As shown in tables 4.1, 4.2, 4.3, 4.4, 4.5 and 4.6, several simple tests were done with preset the genetic parameters: population initial size to $I_s = 7$, selection probability to $p_S$= 0.5, mutation probability to $p_M$= 0.01, elitism to 10%. The number of generations can be limited or not (running *ad eternum*) and then it is the user agent who explicitly stops the evolution. For these tests, and in the current prototype, the run for a maximum of a thirty generations is defined by default, but with the possibility of choice by the human agent in a range from the minimum of one generation to one thousand. After some non-documented preliminary tests we concluded that convergence happens before the thirtieth generation. Therefore, all the tests are 30 generations preset.

Every parameter, including the frozen values, may be changed at any time and per test using the slider and numerical input boxes shown in section 4.3, p. 69. The selection method used is the Tournament (7-way) as explained before (section 3.3).

The Fitness function used is the equation 3.11, p. 59. The resulting fitness values ranges from zero to one ($f : I \rightarrow [0, 1]$). The value of the ideal (best) individual is 1.0, being the worst value equal to 0.0.

The notation used for the fitness value of the best individual is $f_{best}$. The initial weights are: $w_C = 0.1$, $w_F = 0.2$, $w_K = 0.6$ and $w_S = 0.1$.

For these first tests the `GoFGoPColor` results are omitted, because the descriptions in the video segments are not accurate for this specific descriptor. An extra tool is needed to accurately calculate the colour histogram average for groups of frames.

The names of the descriptors are abbreviated: F for the Free text annotation, K for the Keyword annotation and S for the Shot distance. The individuals (video clips) all have three segments, therefore, the genes of all the three segments are present in the table, column Best.

The goal for all the tests was inputted by a human agent. When no text was entered in one of the two textual annotation descriptors, the table shows a blank cell in the Goal

column of that descriptor. Although, that descriptor is not used for the evolutionary process, still the individual's genes are shown.

Every test case has a snapshot of each one of the video clip segments. The more relevant key frame of each segment is selected. The images (figures 4.9, 4.10, 4.11, 4.12, 4.13 and 4.14) are sorted by segment playing order and presented aligned from left to right.

**Test 1**

$f_{best} = 0.23380172$

$Time(seconds) = 0.518799$

| Descriptor | Goal | Best |
|---|---|---|
| F | | VS1: Mirante. Travelling through a garden trail with statues of human figures, many trees and shadows. |
| | | VS2: Mirante. Big gothic white and gray house made of stones. |
| | | VS3: Poco. The view from the bottom of the well. A ray of light. |
| K | trails,trees,foliage | trail,statues,trees,foliage |
| | | house,gothic,white,stone,foliage |
| | | well,gothic,bottom |
| S | 0.2 | 0.5 |
| | | 0.4 |
| | | 0.2 |

Table 4.1: Test 1: genes vs goal.

Figure 4.9: Test 1: Snapshots of the resulting video.

**Test 2**

$f_{best} = 0.37104774$

$Time(seconds) = 3.37283$

| Descriptor | Goal | Best |
|---|---|---|
| F | Christian cross and the word ESTASAM in ceramic | VS1: Lago2. A lake, dark water, with a stone path in between the shores made of rocks. Shadows by the foliage. Water reflection of tree leafs and branches. |
| | | VS2: Lago2. Following the path, made of stones, over the lake dark water. Shadows by the foliage. |
| | | VS3: Abundanca. Stone wall with painted ceramic, blue christian cross and the word ESTASAM. |
| K | estasam,cross,christian | lake,water,path,stone,rocks,foliage |
| | | lake,water,path,stone,rocks,foliage, reflection,leafs |
| | | cross,christian,ceramic,estasam |
| S | 0.0 | 0.3 |
| | | 0.0 |
| | | 0.0 |

Table 4.2: Test 2: genes vs goal.

Figure 4.10: Test 2: Snapshots of the resulting video.

**Test 3**

$f_{best} = 0.2097829$

$Time(seconds) = 2.653322$

| Descriptor | Goal | Best |
|---|---|---|
| F | gothic sintra houses with gardens and water fountain | VS1: Mirante. Travelling through a garden trail with statues of human figures, many trees and shadows. |
| | | VS2: Mirante. Big gothic white and gray house made of stones. |
| | | VS3: Terraco. Focusing the gothic house figures and statues. |
| K | gothic,water,house,garden | trail,statues,trees,foliage |
| | | house,gothic,white,stone,foliage |
| | | statues,figures,gothic,house |
| S | 0.5 | 0.5 |
| | | 0.4 |
| | | 0.2 |

Table 4.3: Test 3: genes vs goal.

Figure 4.11: Test 3: Snapshots of the resulting video.

**Test 4**

$f_{best} = 0.44468412$

$Time(seconds) = 2.105482$

| Descriptor | Goal | Best |
|---|---|---|
| F | Lake in the middle of the forrest | VS1: Lago2. A lake, dark water, with a stone path in between the shores made of rocks. Shadows by the foliage. Water reflection of tree leafs and branches. |
| | | VS2: Lago2. Following the path, made of stones, over the lake dark water. Shadows by the foliage. |
| | | VS3: Lago2. The lake dark water in between the rocky shores, shadows by the foliage. |
| K | | lake,water,path,stone,rocks,foliage |
| | | lake,water,path,stone,rocks,foliage, reflection,leafs |
| | | lake,water,rocks,reflection |
| S | 0.3 | 0.3 |
| | | 0.0 |
| | | 0.3 |

Table 4.4: Test 4: genes vs goal.

Figure 4.12: Test 4: Snapshots of the resulting video.

**Test 5**

$f_{best} = 0.20344159$

$Time(seconds) = 2.526132$

| Descriptor | Goal | Best |
|---|---|---|
| F | Gothic houses among the forrest. Lakes and gardens. | VS1: Mirante. Travelling through a garden trail with statues of human figures, many trees and shadows. |
| | | VS2: Cocheiras. Garden trails with stone wall boundaries in the middle of the trees. |
| | | VS3: Cocheiras. Old horse stables made with stone in the middle of the trees and flowers. |
| K | gothic,house,lake,garden | trail,statues,trees,foliage |
| | | road,trees,wall |
| | | stable,horse,trees,flowers |
| S | 0.8 | 0.5 |
| | | 0.7 |
| | | 0.6 |

Table 4.5: Test 5: genes vs goal.

Figure 4.13: Test 5: Snapshots of the resulting video.

**Test 6**

$f_{best} = 0.18234143$

$Time(seconds) = 0.588088$

| Descriptor | Goal | Best |
|---|---|---|
| F | | VS1: Terraco. Circular gothic window. |
| | | VS2: Terraco. Travelling around the gothic house with its windows, figures and statues. |
| | | VS3: Mundos. The two fountain statues, the view from the fountain interior against the light. |
| K | statues,water,gothic | window,glass,gothic,round |
| | | statues,figures,gothic,house,windows |
| | | statues,figures |
| S | 0.2 | 0.1 |
| | | 0.4 |
| | | 0.1 |

Table 4.6: Test 6: genes vs goal.

Figure 4.14: Test 6: Snapshots of the resulting video.

The statistical analysis summary is presented in table 4.7, p. 81, with the results. The time taken by the best fitness (solution) to reach the goal, is included. The maximum, average and the minimum are calculated for the above tests.

After these tests a comment must be made: the viewing (playing) of the videos is much more pleasant than the descriptions purview. Some less accurate video descriptions were detected, and also the need for fine tuning queries semantics, such as searching for plurals and case insensitive keywords. This is another field of interest, natural language search engines, for further development, but it is out of the scope to develop it further at this stage. The performance decreases dramatically when free text annotation is computed,

because a very complex spatial/temporal algorithm is used, should be improved.

| **Test** | $f_{best}$ | **time to goal** (`seconds`) |
|---:|---|---|
| 1 | 0.23380172 | 0.518799 |
| 2 | 0.37104774 | 3.37283 |
| 3 | 0.2097829 | 2.653322 |
| 4 | 0.44468412 | 2.105482 |
| 5 | 0.20344159 | 2.526132 |
| 6 | 0.18234143 | 0.588088 |
| **Maximum:** | 0.44468412 | 3.37283 |
| **Average:** | 0.274183 | 1.960775 |
| **Minimum:** | 0.18234143 | 0.518799 |

Table 4.7: Tests statistics.

# Chapter 5

# Conclusion

This chapter presents the conclusion of this dissertation including the discussion of the results and the proposals for the future. Future work is identified with all the improvements and new features that were envisaged but unfeasible for the current prototype due to time and resources constraints.

## 5.1 Discussion

A new approach is proposed regarding multimedia documents creation, by pursuing a new paradigm of evolutionary aided multimedia (EAM) documents production with human agent interaction.

The proposed objectives were accomplished. The tests results (section 4.4.2, p. 74) have shown good performance for reasonable (despite the lack of all segments fitness possible combinations) values of fitness. The results were obtained by unattended stop at the final (thirtieth) generation. The solution, the final video clip, was, at each run, an interesting one converging towards the user goal. Of course, subjective evaluation was taken here by an human agent, although one could try to formalise some characteristics able to be measured, such as the similarities of the genes shown in the test tables.

Disregarding the more or less subjective nice outcome, better results are pursued. Some fine tuning of the formula and metrics is needed. Also, a more consistent list of tests, including larger video clips with more well described segments (bigger genome), must be executed in order to analyse the caveats of the proposed formula.

There is a relatively large number of strategic combinations and their possible values for

this kind of problems. Several methods of selection for mating do exist and many strategies can be implemented using only one or combining two or more methods. The tuning of the probability for the selection and mutation operators offers many variations. The mutation can be a simple probabilistic step in the GA sequence or be implemented as a more evolved strategy, such as the Triggered Hypermutation that introduces high mutation occasionally. Islands of populations evolving in a distributed manner and random immigrants may be used in order to increase diversity and avoid early convergence.

One may see all the natural examples that exist in our own planet and learn with nature of how to evolve species and naturally select them. The human species are really diverse with ethnic crossing over and immigration between geographical distributed groups of populations. There is much more to research, learn and adapt.

Cinema is changing in many ways. There is much more to explore in cinema and video editing field, MovieGene is an example of an approach. A combination between the video editing by well known techniques (section 2.2, p. 17) used exclusively by humans, and the use of the same shot cuts but with computer aided evolutionary algorithms, is a promising path to trail.

## 5.2 Future work

Several improvements and research directions are foreseen, not only for the core system of the MovieGene's GA, but also for the whole system including a more ergonomic interface, data for statistical analysis and new methodologies implementation. In particular the ones we consider more important are:

1. Research for more, different and relevant descriptors for this specific problem of describing and searching the best video clips editing.

2. Research for different algorithm strategies of selection in order to improve the final results of the multimedia document editing. There are many possible combinations with the GA selection operator as a multi selection operator. One can even introduce the Triggered Hypermutation or Random Immigrants tactics in order to prevent early convergence and improve genetic diversity. Also, the fine tuning of the selection for crossover and mutation probabilities is a never ending task. Nevertheless, a dynamic automatic tuning may be implemented: a module for machine learning with the genetics probability values modified by the interaction of several user agents can be implemented.

3. Genome variable size (number of genes) with several $n$-point crossover techniques should be implemented.

4. Fitness formula and metrics needs improvement. Much more research has to be done in the metrics and fitness formula, following the addition of new descriptors. Mainly, the MPEG-7 semantic descriptors, such as `FreeTextAnnotation` and `KeywordAnnotation`, need better distance metrics. Maybe more semantic descriptors are needed...

5. Abstract, richer and more flexible interface:

   (a) Clustering of the clips, in order to aid the user interaction with the system, when dealing with a population size of a hundred ($I_s = 100$) or more individuals. Thus, individual (multimedia document) selection in between the evolutionary steps becomes easier.

   (b) A very important issue for the interface is to abstract the human user agent of the GA and other technical detail. Instead of presenting interaction options like "selection probability" in the "genetics" layout "creativity" could be used and for the "mutation" use of "strangeness" may be an option. All this with sliders with *more semantic* labels for the minimum and maximum scale values instead of numbers or technical words. For example, a painting board where the user brushes and mixtures some colours to express the kind of dominant colours for the goal, is a possible solution for the low level `GoFGoPColor` descriptor. The system will automatically calculate the colour histogram coefficients to use for the goal search.

   (c) The user should be able to select a specific group of videos from the whole library to start the production.

   (d) The ability to store the goal parameters for future use should be possible with simple interactions.

   (e) The user should be able to change the descriptor weights using the Graphical User Interface (GUI).

6. Add audio processing with rules for the ruptures and edition.

7. Concurrent interaction allowing several agents to act over one or several videos simultaneously in order to research new possible results or trends.

8. Relational database for the main repository, e.g., PostgreSQL, promoting a more distributed system and improving information management.

9. User profiles and session management including historical data and concurrent access. This is a very useful feature that offers, not only statistical data for analysis, but also an added value for the improvement of the concurrent access and creative collaboration between users.

10. Additional multimedia document information (characteristics) stored in the database and for presentation in the interface, such as frame size and rate, colour depth and clip duration. These are added value informations that may influence some aspects of the selection when supervised by a human agent. Also, it may be necessary to have this kind of data persistently stored in order to create a more sophisticated system that does evolve video documents with different frame sizes as isolated groups or mixed ones.

11. Support for several different languages for the same MPEG-7 descriptor. At this stage of the development, only the first descriptor of each type is stored, disregarding any following new annotations using any other human languages.

# Bibliography


[Apache02] Apache, HTTP Server Documentation Project.
*Apache HTTP Server Version 1.3 Documentation*.
`http://httpd.apache.org/docs-project/`, 2002.

[Chiariglione03] Chiariglione, Leonardo.
*Movies in bits?*.
`http://www.iso.ch/iso/en/commcentre/news/dcinema.html`, 2003.

[Chiu00] Chiu, Patrick, Girgensohn, Andreas, Polak, Wolf, Rieffel, Eleanor G., Wilcox, Lynn.
*A Genetic Algorithm for Video Segmentation and Summarization*.
In *IEEE International Conference on Multimedia and Expo (III)*, 1329–1332. 2000.
URL `http://citeseer.ist.psu.edu/article/chiu00genetic.html`.

[Copeland04] Copeland, Jack, Aston, Gordon.
*The Turing Archive for the History of Computing*.
`http://www.alanturing.net/`, 2004.

[Correia02] Correia, Nuno.
*Sistema Evolutivo para Produção Multimédia / Evolutionary System for Multimedia Production*.
*Research work proposal for GRICES/MCES funding*, 2002.

[Costa02] Costa, Miguel, Correia, Nuno, Guimarães, Nuno.
*Annotations as multiple perspectives of video content*.
In *MULTIMEDIA '02: Proceedings of the tenth ACM international conference on Multimedia*, 283–286. ACM Press, New York, NY, USA, 2002.
ISBN 1-58113-620-X.

[Darwin04] Darwin, Charles.
*On Natural Selection*.

Penguin Books, London, EN, 2004.
ISBN 0141018968.

[EC04] EC, European Commission.
*EuroDicAutom: European Commission Terminology Database*.
`http://europa.eu.int/eurodicautom/Controller`, 2004.

[Gargi00] Gargi, Ullas, Kasturi, Rangachar, Strayer, Susan H.
*Performance characterization of video-shot-change detection methods*.
*IEEE Trans. Circuits Syst. Video Techn.*, 10(1):1–13, 2000.

[Generation598] Generation5.
*Al Biles*.
`http://www.generation5.org/content/1999/biles.asp`, 1998.

[Goldberg89] Goldberg, David E.
*Genetic Algorithms in Search, Optimization and Machine Learning*.
Addison-Wesley Longman Publishing Co., Inc., 1989.
ISBN 0201157675.

[Heap95] Heap, A.
*Real-time hand tracking and gesture recognition using Smart Snakes*.
`http://citeseer.ist.psu.edu/heap95realtime.html`, 1995.

[Henriques04] Henriques, Nuno A. C.
*MovieGene´s Reference Manual*, 2004.
Version 0.1.

[Hodges04] Hodges, Andrew.
*The Alan Turing Home Page*.
`http://www.turing.org.uk/turing/`, 2004.

[IETF03] IETF, Internet Engineering Task Force.
*Uniform Resource Identifiers (URI): Generic Syntax, RFC 2396*.
`http://www.ietf.org/rfc/rfc2396.txt`, 2003.

[ISO00] ISO, International Organization for Standardization.
*ISO 8601:2000 – Data elements and interchange formats – Information interchange – Representation of dates and times*.
`http://www.iso.org/iso/en/prods-services/popstds/datesandtime.html`, 2000.

[ISO04] ISO, International Organization for Standardization.
*MPEG-7 Overview (version 9).*
`http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm`, 2004.

[Koenen03] Koenen, Rob.
*From MPEG-1 to MPEG-21: Creating an Interoperable Multimedia Infrastructure.*
`http://mpeg.telecomitalialab.com/documents/from_mpeg-1_to_mpeg-21.htm`, 2003.

[Kosch04] Kosch, Harald.
*Distributed multimedia database technologies supported by MPEG-7 and MPEG-21.*
CRC Press LLC, Boca Raton, Florida, US, 2004.
ISBN 0849318548.

[Koza92] Koza, John R.
*Genetic Programming: on the programming of computers by means of natural selection.*
MIT Press, Cambridge, MA, US, second printing, 1993 edition, 1992.
ISBN 0262111705.

[Lee02] Lee, Hyowon, Smeaton, Alan F.
*Designing the User Interface for the Físchlár Digital Video Library.*
*Journal of Digital Information*, 2(4), 2002.
Article no. 103.

[Levy93] Levy, Steven.
*Artificial Life: The Quest for a New Creation.*
Penguin Books, London, EN, 1993.
ISBN 0140231056.

[Manjunath02] Manjunath, B.S., Salembier, Philippe, Sikora, Thomas.
*Introduction to MPEG-7 Multimedia Content Description Interface.*
John Wiley & Sons, Ltd, West Sussex, EN, 2002.
ISBN 0471486787.

[Manovich05] Manovich, Lev.
*Soft Cinema.*
`http://www.manovich.net/cinema_future/toc.htm`, 2005.

[McWorter97] McWorter, William.
*Fractint L-Systems Definition*.
`http://spanky.triumf.ca/www/fractint/findex.html`, 1997.

[Member] Member, Kit-Sang Tang.
*Optimal File Placement in VOD System Using Genetic Algorithm*.
`http://citeseer.ist.psu.edu/498910.html`.

[Miller04] Miller, George A.
*WordNet: An Electronic Lexical Database*.
`http://www.cogsci.princeton.edu/~wn/`, 2004.

[Moore03] Moore, Kevin W.
*MediaWare Solutions´ EDL format version 3*.
`http://www.mediaware.com.au/ProductInfo/MWS-EDLpaper.html`, 2003.

[NIST04] NIST, National Institute of Standards and Technology.
*Dictionary of Algorithms and Data Structures*.
`http://www.nist.gov/dads/`, 2004.

[Obitko04] Obitko, Marek.
*An introduction to genetic algorithms with Java applets*.
`http://cs.felk.cvut.cz/~xobitko/ga/`, 2004.

[OMG03] OMG.
*Unified Modelling Language*.
`http://www.omg.org/uml/`, 2003.

[PostgreSQL03] PostgreSQL.
*PostgreSQL database*.
`http://www.postgresql.org/`, 2003.

[Prunes05] Prunes, Mariano, Raine, Michael, Litch, Mary.
*Film analysis*.
`http://classes.yale.edu/film-analysis/`, 2005.

[Rocchi04] Rocchi, Cesare, Zancanaro, Massimo.
*Rhetorical Patterns for Adaptive Documentaries*.
In *International Conference on Adaptive Hypermedia AH2004*. 2004.
URL `http://peach.itc.it/papers/ah2004.pdf`.

[Sims91] Sims, Karl.
*Artificial Evolution for Computer Graphics*.
In *Computer Graphics ACM SIGGRAPH'91*, 319–328. 1991.

[Sims94a] Sims, Karl.
*Evolving 3D Morphology and Behavior by Competition*.
In R. Brooks, MIT Press P. Maes, editors, *Artificial Life IV*, 28–39. 1994.

[Sims94b] Sims, Karl.
*Evolving Virtual Creatures*.
In *Computer Graphics ACM SIGGRAPH'94*, 15–22. 1994.

[Sims97] Sims, Karl.
*Galápagos*.
`http://www.genarts.com/galapagos/`, 1997.

[Smith97] Smith, John, Sugihara, Kazuo.
*A Tool for Design of Distributed Genetic Algorithms on WWW*.
`http://citeseer.ist.psu.edu/smith97tool.html`, 1997.

[Sun04a] Sun.
*Java 2 Media Framework*.
`http://java.sun.com/products/java-media/jmf/`, 2004.

[Sun04b] Sun.
*Java 2 Standard Edition*.
`http://java.sun.com/j2se/`, 2004.

[Sun04c] Sun.
*The Java Tutorial*.
`http://java.sun.com/docs/books/tutorial`, 2004.

[Tan] Tan, Theen-Theen, Davis, Larry, Thurimella, Ramki.
*One-Dimensional Index for Nearest Neighbor Search*.
`http://citeseer.ist.psu.edu/209287.html`.

[Timday04] Timday.
*Evolvotron*.
`http://www.bottlenose.demon.co.uk/share/evolvotron/index.htm`, 2004.

[Tipler03] Tipler, Frank.
*A Física da Imortalidade – Cosmologia Moderna, Deus e a Ressurreição dos Mortos / The Physics of Immortality – Modern Cosmology, God and the Resurrection of the Dead*.
Editorial Bizâncio, Lisboa, PT, 2003.
ISBN 9725301986.

[W3C98] W3C, World Wide Web Consortium.
*Date and Time Formats*.
`http://www.w3.org/TR/NOTE-datetime`, 1998.

[W3C02a] W3C, World Wide Web Consortium.
*Addressing the Internet*.
`http://www.w3.org/Addressing/`, 2002.

[W3C02b] W3C, World Wide Web Consortium.
*HTTP: HyperText Transfer Protocol*.
`http://www.w3.org/Protocol/`, 2002.

[W3C03] W3C, World Wide Web Consortium.
*XHTML 1.0: The Extensible HyperText Markup Language*.
`http://www.w3.org/TR/xhtml1/`, 2003.

[W3C04] W3C, World Wide Web Consortium.
*Web Style Sheets*.
`http://www.w3.org/Style/`, 2004.

[Watanabe03] Watanabe, Hiroshi.
*Information technology: Multimedia standards – what´s next?*.
`http://www.iso.ch/iso/en/commcentre/pdf/Multimedia0101.pdf`, 2003.

[WEBNOX04] WEBNOX, Corporation.
*Hyperdictionary: online technical dictionary*.
`http://www.hyperdictionary.com/`, 2004.

[Wikipedia05] Wikipedia, the free encyclopedia.
*Wikipedia, the free encyclopedia*.
`http://en.wikipedia.org/`, 2005.

[Zhang97] Zhang, Hongjiang.
*Handbook on Pattern Recognition and Computer Vision*, chapter 5.

World Scientific Publishing Company, 1997.

# Index