Musical Slideshow

Design of an innovative presentation application for consumer photos

Bruno Alexandre Conde Tomás

Dissertation submitted for obtaining the degree of

Master in Electrical and Computer Engineering

Jury

Supervisor: Prof. Fernando Pereira
President: Prof. José Bioucas
Members: Prof. Joaquim Jorge

October 2009
First of all, I would like to thank my family, especially my father, mother, and grandparents, for the patience and support provided throughout my entire academic career so far, particularly in the last semester.

I would also like to express my gratitude to Prof. Fernando Pereira, who tutored this work, for his guidance, devotion and dedication, which along with his motivating skills have certainly made this work possible to accomplish. Needless to say, he is a unique reference and the cornerstone of this work.

I also want to dedicate a very special word to Joana Rebelo, who helped in choosing the songs that accompany this Thesis.

Last, but not least, a big thank you to all my friends and colleagues who have always been by my side throughout these years.

A deep “Obrigado” to all of the above mentioned as well as to everyone close to me, which, in their own way, contributed for the development of this Thesis.
Abstract

Photo storing devices, such as digital cameras and cell phones, have become an indispensable commodity for families and individuals in recent years. With the advance of digital storage technology, people can take photos at will and have become accustomed to recording almost everything with photos. Nevertheless, large amounts of photos without appropriate organization raise many potential problems in terms of information access; for example, people may spend too much time browsing and often get lost in massive photo collections, making it a painful task. Therefore, to help with organizing photos and to make the photo viewing process more enjoyable, it is useful to develop advanced analysis and presentation techniques that facilitate an effective photo organization and efficient photo browsing, eventually at some music pace to make the overall experience more pleasant.

As people’s time is getting more precious and scarce every day, an application capable of reducing the time spent in these tasks by automatically generating an enjoyable musical slideshow looks like something very useful and promising.

In this context, this Thesis describes the motivations for the development of a musical slideshow application, its architecture as well as the entire signal processing flow designed and implemented in the course of this work.

To evaluate the quality of the created musical slideshow, a user evaluation study was also conducted with encouraging results, showing that the developed application is able to fulfill people expectations.

The major novelty of this work regards the mixed time and content-based approach to the multimedia presentation problem to maximize the viewers’ excitement while watching a slideshow and the user evaluation study conducted to evaluate the overall system’s performance.

**Keywords:** Photo organization; Photo browsing; Musical slideshow; Music pace; Content-based; Time-based.
Resumo

Os dispositivos habilitados a armazenar fotos, como câmaras digitais e telemóveis, tornaram-se absolutamente indispensáveis para famílias e indivíduos nos últimos anos. Com o avanço da tecnologia de armazenamento digital, as pessoas podem tirar fotos indiscriminadamente e acostumaram-se a gravar quase tudo por esta via. No entanto, grandes quantidades de fotos sem uma organização adequada podem levantar muitos problemas em termos de acesso à informação, por exemplo, as pessoas podem passar muito tempo a pesquisar e muitas vezes acabam por se perder em enormes coleções de fotos, tornando-se assim numa tarefa penosa. Portanto, para ajudar a organizar as fotos e tornar o processo de visualização de fotos mais agradável, é útil desenvolver ferramentas avançadas de análise e técnicas de apresentação que facilitem a organização e consumo das fotos, tornando-a eficaz e eficiente, e eventualmente, coordenar este processo com o ritmo de uma música para tornar a experiência do utente ainda mais excitante.

Como o tempo das pessoas está a ficar cada vez mais precioso e escasso a cada dia que passa, uma aplicação capaz de reduzir o tempo gasto nestas tarefas, gerando automaticamente uma apresentação agradável de slides combinada com música assume-se como um objectivo muito útil e promissor.

Neste contexto, esta Tese começa por apresentar as motivações para o desenvolvimento de uma aplicação para a apresentação selectiva de fotos com fundo musical sincronizado, apresentando de seguida a sua arquitectura, bem como todo o fluxo de processamento de sinal concebido e implementado no decorrer deste trabalho.

Para avaliar a qualidade da aplicação desenvolvida, foi realizado um estudo de avaliação de desempenho orientado para o utente que apresentou resultados encorajadores, mostrando que a aplicação desenvolvida é capaz de corresponder às expectativas das pessoas.

A grande novidade deste trabalho diz respeito à combinação de uma análise mista das imagens baseada em tempo e conteúdo, bem ao estudo de avaliação realizado para avaliar o desempenho do sistema como um todo.

**Palavras-chave:** Organização; Pesquisa; Slideshow; Ritmo da música; Tempo; Conteúdo.
# Table of Contents

CHAPTER 1 .................................................................................................. 1  
Context and Objectives ................................................................................ 1  
1.1 Motivation ......................................................................................... 1  
1.2 Objectives ......................................................................................... 2  
1.3 Thesis Structure ............................................................................... 3  
CHAPTER 2 .................................................................................................. 5  
Reviewing Technologies for Consumption of Personal Photos...................... 5  
2.1 Structuring the Photos Consumption Problem ...................................... 5  
2.1.1 First Dimension: Photo Browsing Versus Photo Presentation .......... 6  
2.1.2 Second Dimension: Dynamic Versus Static Layouts ......................... 8  
2.1.3 Third Dimension: Content Versus Temporal Filling Criterion .......... 9  
2.2 Most Relevant Solutions for Consumption of Personal Photos .......... 10  
2.2.1 Hierarchical Layouts for Photo Libraries ........................................ 10  
2.2.2 Genetic Algorithms for Album Page Layouts ................................... 17  
2.2.3 Content Based Photograph Slideshow with Incidental Music .......... 24  
2.2.4 An Audiovisual Tiling Slideshow Presentation ................................ 28  
CHAPTER 3 ................................................................................................ 37  
Architecture and Functional Description ....................................................... 37  
3.1 System Architecture ........................................................................ 37  
3.2 Functional Description by Module ..................................................... 41  
3.2.1 Photo Analysis ............................................................................ 41  
3.2.2 Photo Organization ..................................................................... 43  
3.2.3 Music Analysis ........................................................................... 44  
3.2.4 Composition ................................................................................ 44  
CHAPTER 4 ................................................................................................ 47  
Processing for Musical Slideshow Generation ............................................. 47  
4.1 Photo Analysis ................................................................................ 47  
4.1.1 Temporal Sorting ........................................................................ 47  
4.1.2 Similarity Detection and Filtering ................................................ 48  
4.2 Photo organization .......................................................................... 50  
4.2.1 Time-based Clustering .................................................................. 50  
4.2.2 Content-based Clustering ............................................................. 52
Index of Figures

Figure 1 - Classification tree for the personal photos consumption solutions. .............. 6
Figure 2 - Exemplification of the three classification dimensions............................... 7
Figure 3 - Dynamic (a) versus static (b) layouts......................................................... 9
Figure 4 - Family photo collection depicting the main subjects in a primary region surrounded by grouped photos of family members [7]. ...................................... 11
Figure 5 - Examples of quadrants and secondary regions set ups [7]. ......................... 12
Figure 6 - System architecture.................................................................................... 12
Figure 7 - Dividing secondary regions among three quadrants [7]............................... 15
Figure 8 - BRQ layout dynamic behavior showing user control over dragging and resizing the primary region [7]. ............................................................... 16
Figure 9 - Example of scrapbooking page..................................................................... 17
Figure 10 - System architecture [9]............................................................................ 18
Figure 11 - Flowchart for genetic algorithm processing [9]. ....................................... 19
Figure 12 - Genetic operators for arrays (a) and for trees (b). ................................. 20
Figure 13 - Genotype encoding for a page layout solution........................................ 21
Figure 14 - Page layout examples [9]......................................................................... 23
Figure 15 - System architecture[12]........................................................................... 25
Figure 16 - Interface for the prototype of the Photo Slide Show application [14]. ....... 27
Figure 17 - System architecture [17]......................................................................... 29
Figure 18 - Example of determining the timing for layout switching and photo display [17]................................................................................................................... 31
Figure 19 - Examples of various templates [17]......................................................... 33
Figure 20 - Architecture of the developed solution.................................................. 39
Figure 21 - Similarity detection screenshot................................................................. 42
Figure 22 - Time (3 clusters) and content-based clustering example......................... 44
Figure 23 - Composition screenshots: Cluster selection (top) Slideshow layout (bottom). ......................................................................................................................... 46
Figure 24 - Time-based clustering example. ............................................................... 51
Figure 25 - The CLD extraction process..................................................................... 55
Figure 26 - Clustering loop example........................................................................... 57
Figure 27 - Schematic view of the processing algorithm [4]. ................................. 58
Figure 28 - Example of a presentation sequence..................................................... 62
Figure 29 - Available slideshow layout templates..................................................... 63
Figure 30 - Example of a temporal composition sequence........................................ 65
Figure 31 - Example of player output layouts (3). .................................................... 66
Figure 32 - Application operations hierarchy............................................................ 69
Figure 33 - Application's GUI.................................................................................... 70
Figure 34 - Selection buttons.................................................................................... 71
Figure 35 - Analysis main tab................................................................................... 72
Figure 36 - Organization main tab.......................................................................... 73
Figure 37 - Composition main tab ................................................................. 75
Figure 38 - Slideshow and browsing control buttons ................................. 76
Index of Tables

Table 1 - Evaluation for each criterion ................................................................. 22
Table 2 - Results of the user study [12] ................................................................. 28
Table 3 - Data from photo set evaluation [17] ......................................................... 34
Table 4 - Clustering performance evaluation [17] .................................................... 34
Table 5 - Cropping performance evaluation [17] ...................................................... 34
Table 6 - Test material ......................................................................................... 81
Table 7 - Algorithms’ parameters .......................................................................... 82
Table 8 - Evaluation results for Question 1 ............................................................ 83
Table 9 - Evaluation results for Question 2 ............................................................ 83
Table 10 - Evaluation results for Question 3 ........................................................... 84
Table 11 - Evaluation results for Question 4 ........................................................... 84
List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BRQ</td>
<td>Bilevel Radial Quantum</td>
</tr>
<tr>
<td>CI</td>
<td>Cluster Importance</td>
</tr>
<tr>
<td>CIE</td>
<td>Commission Internationale de l'Eclairage/International Commission on Illumination</td>
</tr>
<tr>
<td>CLD</td>
<td>Color Layout Descriptor</td>
</tr>
<tr>
<td>DCD</td>
<td>Dominant Color Descriptor</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>LUV</td>
<td>L*, u*, v*</td>
</tr>
<tr>
<td>IDE</td>
<td>Integrated Development Environment</td>
</tr>
<tr>
<td>PC</td>
<td>Photo Conformance</td>
</tr>
<tr>
<td>PPM</td>
<td>Photos Per Minute</td>
</tr>
<tr>
<td>ROI</td>
<td>Region Of Interest</td>
</tr>
<tr>
<td>RGB</td>
<td>Red, Green, Blue</td>
</tr>
<tr>
<td>SDK</td>
<td>Software Development Kit</td>
</tr>
<tr>
<td>WMP</td>
<td>Windows Media Player</td>
</tr>
</tbody>
</table>
CHAPTER 1

Context and Objectives

This chapter intends to present the scope and objectives of this Thesis, after providing its motivation and context. Finally, the structure of this Thesis is presented.

1.1 Motivation

Photo storing devices, such as digital cameras and cell phones, have become an indispensable commodity for families and individuals in recent years. With the advance of digital storage technology, people can take photos at will and have become accustomed to recording almost everything with photos. Nevertheless, large amounts of photos without appropriate organization raise many potential problems in terms of information access. People spend too much time browsing and often get lost in massive photo collections. As people’s time is getting more precious and scarce every day, an application capable of saving the time spent in these tasks by automatically generating an innovative and enjoyable musical slideshow looks like something very useful and promising. Therefore, to help with organizing photos and making the photo viewing process more enjoyable, it is useful to develop advanced analysis and presentation techniques and applications that facilitate the effective photo organization and efficient photo browsing, eventually combined with some music pace.

Typically, commercial photo browsers provide thumbnail functionalities to scale down photos so that users can browse multiple photos at a glance. There also other rather simple solutions available for managing and accessing photo collections, such as the sequential presentation of the photos based on their time labels; however, this is still rather short in terms of maximizing the user experience while consuming the photos. Some critical problems that significantly limit the users’ presentation experience are worthy of further investigation:
1. **Photo analysis** - Since most photos are taken by amateurs who are not familiar with photography, many photos may suffer from quality degradation, such as blur derived from hand shaking and underexposure (overexposure) caused by bad light or shutter control. Although some solutions have been already developed [1], techniques of quality estimation and photo filtering are not widely applied in photo presentation and management.

2. **Photo organization** – The popularity of photo capturing devices creates massive and disordered photo sets, which stuff user’s storage and make photo browsing and access tedious. Therefore, the automatic organization of photos by time and content is urgently demanded. Some studies have been conducted for photo clustering based on temporal context [2][3], while relatively few works have exploited clustered data for creating organized presentations.

3. **Music analysis** - One of the most popular ways to present photos is the photo slideshow which is provided in many photo management systems as an indispensable function. However, for large amounts of photos, sequentially browsing photos often takes much time and makes users weary. Some works have been conducted to improve the browsing experience by cooperating with music [4], but the lengthy displaying time problem is still not addressed.

4. **Photo composition** - In conventional photo slideshows, photos are displayed one-by-one, according to alphabetical or temporal order. Therefore, photos taken in the same scene or having the same topic are separated into different slots, and the browsing experience is cut off. Hua et al. proposed an approach to convert two-dimensional (2D) photos into three-dimensional (3D) videos [5] to create a new browsing approach. However, this system deals less with photo organization and does not emphasize the browsing experience via audio-video synchronization.

All this motivates the development of a new photo presentation solution with dynamic image layouts determined by both temporal and content-based analysis, and synchronized with some music pace. The value of this photo presentation approach in the context of the above mentioned systems and situations proves to be rather significant, motivating the application designed, implemented and evaluated in this Thesis which objectives are explained next.

### 1.2 Objectives

The main objectives of this Thesis are to design, implement and evaluate an application capable of, based on some generic input audio and visual content, producing an audiovisual presentation system able to address the short comings listed above. With the help of image analysis, the proposed system should automatically filter out photos according to a series of criteria; the remaining photos are then organized in terms of time and content characteristics so that similar photos can be displayed together; finally, the proposed system should manipulate photos and generate audiovisual slideshows that follow the music’s pace to improve the user’s experience. More precisely, the system automatically generates music-driven photo slideshows, in which multiple photos having similar characteristics would be displayed in the same frame, with the presentation of the photos proceeding at the pace of the incidental music; this approach justifies the name adopted for the developed application: musical slideshow.
Therefore, this Thesis targets the development of a generic photo presentation solution with dynamic layouts and both temporal and content-based criteria combined with audio content in order to maximize the user experience.

1.3 Thesis Structure

This Thesis is organized in seven chapters, including this first one introducing the Thesis, and the seventh dedicated to the conclusions and future work.

Chapter 2 presents a detailed review of the photo consumption problem as well as some main technologies and systems already developed to address the problem. First, a classification tree for photo consumption solutions is presented, with the target to structure the photo consumption problem; after, four different solutions which have been considered more representative and relevant are reviewed.

Chapter 3 introduces the solution developed in this Thesis, by presenting its architecture and a functional description of each module.

Chapter 4 presents an in-depth description of all the architecture’s modules, notably the processing algorithms implemented, in order to allow the reader to get a complete understanding of the entire process proposed for the musical slideshow generation.

Chapter 5 is dedicated to the application developed, presenting an implementation’s overview with a brief description of the application’s implementation operation diagram, the software frameworks and libraries used as well as an installation guide and an application’s Graphic User Interface guide. This chapter intends to provide sufficient information for a proper and easy usage of the musical slideshow application by any user.

Chapter 6 presents the conditions, methodology, and results of the subjective tests carried out to evaluate the proposed musical slideshow solution.

Finally, Chapter 7 is dedicated to the conclusions and eventual future work.
 CHAPTER 2

Reviewing Technologies for Consumption of Personal Photos

The main purpose of this chapter is to make a short overview on technologies and solutions for personal photos consumption. With that purpose in mind, it is first proposed a way to organize the most relevant types of approaches to address the personal photos consumption problem. After, some more relevant solutions are briefly reviewed, setting the context for the solution which design and implementation will be presented in the next chapters.

2.1 Structuring the Photos Consumption Problem

Since users should enjoy a nice experience while consuming their personal photos, it is essential to provide them an efficient and also pleasant user interface to consume the photos. However, designing a ‘good’ user interface for photo consumption brings some tough challenges: How can a user find the photos he/she is looking for in his/her collection? How can a user have an enjoyable presentation of his/her photo collection? While the first challenge usually asks for a photo browsing approach, the second asks for a photo presentation approach.

As for most technical problems, the various ways to address the personal photos consumption problem can be clustered and classified depending on the main approach, concepts and tools used. Although this type of classification is not unique, having some classification tree helps to better understand the relations between the various solutions and to get a more complete knowledge of the full solutions landscape.

Figure 1 presents the classification tree proposed in this Thesis for structuring the solutions addressing the personal photos consumption problem.
As shown in Figure 1, the proposed classification tree is based on three different classification dimensions, adopted to structure and organize the technologies and solutions for personal photos consumption which are associated to the following concepts (see Figure 2 in Section 2.1.1):

1. **Interaction** – This dimension regards the type of interaction the user has with the personal photos collection, notably through browsing or presentation.

2. **Spatio-Temporal Layout** – This dimension regards the temporal evolution of the spatial arrangement for the photos’ layout, notably a dynamic or static evolution.

3. **Filling Criteria** – This dimension regards the type of criteria used to fill the spatio-temporal layout.

In the next subsections, these three classification dimensions will be addressed in more detail.

### 2.1.1. First Dimension: Photo Browsing Versus Photo Presentation

The first and more important dimension adopted to classify the personal photos consumption solutions regards the way the users interact with the photos collection. In this dimension, two main cases have to be distinguished: photo browsing and photo presentation. The main motivation for selecting this dimension as the first was to highlight the importance of the user having or not relevant input to continuously set the pace and direction of the photo consumption process. In this context, the two clusters for this classification dimension regard active and passive photo consumption experiences since during a browsing experience and a presentation experience users act mostly actively and passively, respectively, in terms of driving the consumption process. These two clusters are described in the following:

- **Photo Browsing** - The work on photo browsing has largely concentrated on solving the problem of users interacting with a large, possibly labeled image database. Unfortunately, many of the lessons learned from that problem may not carry over directly to searching through one’s own personal photos. Unlike
interacting with non-personal databases, users have very intense memories associated to the photos within their personal collection. Also, users are often reluctant to spend effort annotating their own photos: the photos will often be stored in a small, shallow hierarchy of folders in a computer. Users can, therefore, spend a large amount of effort with browsing tools searching through disorganized collections for their photos. A possibility to address the photo browsing problem is to allow the users to interact with their personal photos by means of a photo browser which automatically organizes the user’s images. A compelling aspect of these layouts is the interactivity provided by the users. For example, there is the possibility to redisplay the photos, add and delete sections and photos. Most of the photo browsers available have taken the approach of providing thumbnails and photo management functionalities which help users in managing and accessing photo collections. Hence, the reason why photo browsing is assumed interactive.

- **Photo Presentation** - The work on photo presentation has largely concentrated on conventional solutions that display photos one-by-one, according to alphabetical or temporal criteria. Therefore, photos taken in the same scene are separated into different time slots, and the viewing experience is cut off. Despite all different techniques used in photo presentation, the final purpose is to achieve more reasonable and pleasing results. Nowadays there are some innovative presentation approaches which use multi-photos display solutions, improving the viewing experience. To summarize, photo presentation systems automatically display consumer photos in order to provide good visual experiences. It is possible to distinguish them in relation to their final visual aspect, although the user input is absent in principle. Hence, the reason why photo presentation is assumed non-interactive.
2.1.2. **Second Dimension: Dynamic Versus Static Layouts**

In photo browsing and photo presentation applications, it is possible to distinguish different types of layout depending on the way the spatial arrangement evolves in time (see Figure 3) defining the interaction and presentation structure:

- **Dynamic Layouts** – In dynamic layouts, the spatial arrangement of the user interface changes along time, notably in terms of the number of windows, their positioning and size, etc. The dynamics of the layout structure may be defined by the user, by the application itself or by both in combination. Dynamic browsing layouts allow the user to get a better understanding of the distribution of his/her personal photos, notably through a browser which automatically organizes his/her photos. This distribution, usually in windows, defines a primary window and some subsequent, typically smaller, windows. Thus, large-sized topic photos are displayed in primary windows and smaller-sized supportive photos are displayed in smaller windows. Dynamic presentation layouts, e.g. as those used in a slideshow system, generate a descriptive presentation via more elaborate spatial arrangements which are typically algorithmically determined. One common approach is to spatially concatenate many photographs to convey the whole narration of a scene (e.g. similar photos taken in a single place/event). Concatenating many photos into the same layout strengthens the atmosphere of the viewing experience. As for dynamic browsing layouts, each layout may be composed of a larger-sized topic photo and several smaller-sized supportive photos. Therefore, the viewing experience can be highly attractive. It is assumed that, during both a browsing and a presentation session, the layout structure will change temporally, enriching the experience. In the first case, due to some possible user input, the collections can be re-arranged, e.g. using drag and drop actions. In the second case, during the presentation, the different algorithms decide the changes, which are mainly related to changes of the photo clustering, position, shape and size from layout to layout.

- **Static Layouts** – In static layouts, the spatial arrangement of the user interface does not change along time this means the same number of windows, their positioning and size is kept along the full session. For example, it is largely recognized that a mono window static layout is rather simple not to say boring. In browsing, given a set of photos, an algorithm makes a fixed distribution on individual windows. These windows are usually filled with the photos or their thumbnails which are connected to the photos. In presentation, static layouts are traditionally related to basic presentations where the photos are simply shown one by one and some music may be played in the background. In addition, this type of layout is the basis for conventional slideshows because a simple click on a button provides a sequential photo viewing experience. It is assumed that, during the browsing session, the layout structure will remain the same and thus static although the interface may have a multi-window/photo, or single-window/photo structure. The same applies to a presentation experience in which the structure is temporally the same as defined by the adapted structure for the photos initially determined.
2.1.3. **Third Dimension: Content Versus Temporal Filling Criterion**

Despite the importance of the different classification dimensions taken into consideration before, the last dimension of the tree is of major importance since it represents the final step to determine how rich the final user experience will be. In this context, there are two possible main ways of organizing the photos, thus filling the spatio-temporal layout determined above:

- **Content-based Criteria** – The purpose of taking a photo is mainly to register a special event. Since this applies for most of the cases, it is assumed that in a photo collection there will be a lot of photos with a similar visual appearance. Therefore, the extraction of some content description information from the photos may allow an organization based on low-level features (e.g. color and texture) or more high-level concepts, e.g. indoors and outdoors, used as criteria for filtering, clustering and sequencing the photos and fill the selected layout. Likewise, in order to achieve reasonable and pleasant results, that information can be manipulated in many ways, such as grouping similar photos, detecting low quality photos and correct photo orientation. These procedures will result in a more attractive and enjoyable organization.

- **Temporal-based Criteria** – Generally, photos from the same event are taken in relatively close proximity in time. Hence, it is assumed that organizing photos using a strict temporal criterion significantly improves the users’ experiences in a series of browsing and presentation tasks. The algorithms associated with this approach are solely based on temporal labels similarity although they may seldom also regard the directory structure of the photo collection. This is possible since nowadays digital photos typically include metadata, such as time and date, automatically added to the photo in a standard ‘header’ format such as Exif (Exchangeable Image File) [6] that can be used for automatic organization. If also GPS (Global Positioning System) data is available, equivalent localization-based criteria can also be used.

Although the two possibilities are described above separately, they can also be combined together in some photo consumption solutions in order to provide even richer user experiences. Therefore, these solutions are named hybrid.
2.2 Most Relevant Solutions for Consumption of Personal Photos

Since there are many solutions available for the consumption of personal photos, a brief review of this area would not be complete just by introducing a classification tree for these solutions. In order to make this review more complete and useful for the reader, some solutions will be reviewed in the following. The selection criteria for the solutions to be presented mainly regarded their performance, notably in terms of usability and user experience, and the coverage of the classes defined above, in order a representative review is obtained. In this context, first, two photo browsing solutions are presented, notably a solution based on hierarchical layouts and a solution based on album page layouts; finally, two photo presentation solutions are presented, both based on photo slideshows while differing on the temporal evolution of the spatial arrangement, notably adopting dynamic and static evolutions.

2.2.1 Hierarchical Layouts for Photo Libraries

The first system described here was developed by Kustanowitz and Shneiderman [7] and proposes a hierarchical approach for structuring photo libraries. In the proposed classification tree presented before in Section 2.1, this system fits under the photo browsing, dynamic layout and content based filling criterion branch, since it performs the photo organization based on the content itself using algorithms to dynamically define the layout.

2.2.1.1 Objectives

The main objective of this hierarchical layouts based solution is to explore the possibility of performing rapid, automatic layout with non-overlapping, 2D fixed-aspect ratio objects, such as photos. Photos may appear in a central primary region, which defines the dominant idea of the entire layout, or in secondary regions that offer lower-level detail, such as parents in the primary region, surrounded by secondary regions containing many photos of each of their children. This system addresses two-level categories, where a category represents a region to which is associated a level of detail, e.g. high or low. Furthermore, the presented real-time algorithms enable a compelling interactive display as users resize the layout, or move and resize the primary region defined as shown in Figure 4. In this context, this system has to face several challenges described in the following:

- In order to achieve a balanced view, layouts with similar quantities of photos should be able to minimize wasted space. This contributes to layout efficiency and attractiveness.
- Because users may need to find a specific photo within the entire collection, the application should allow a dynamic creation of a printable layout, e.g. a layout that shows the entire photo collection.
- When resizing regions the harmony should be kept, for instance, truncate the text (e.g. titles).
- To ensure an optimized photo resizing, the management of primary and secondary the regions should be flexible.
2.2.1.2 Architecture and Basic Approach

A compelling aspect of this system is the interactive redisplay as users change the rectangular region size and shape, add or delete secondary regions, and add or delete objects to a secondary region (see Figure 4). This engaging animated interaction is an important feature for consumer applications such as personal photo management. Thus, the algorithm requirements for this system are the following:

- **Fixed canvas size** – The algorithm cannot change the canvas size (width by height, usually the window size chosen by users). Relaxing this requirement would let the canvas size grow if the algorithm determined that a better layout could be achieved if the layout size were 5 percent wider, for example. This introduces a feedback loop in which the canvas size determines layout, which in turn determines canvas size. This requirement allows the algorithm to be deterministic, avoids nonlinearity, and maintains the user sense of control.

- **Fixed primary region size and location** – The algorithm does not modify the user defined set up in terms of the primary’s region size and location. As mentioned in the previous item, relaxing this requirement also introduces a feedback loop.

- **Uniform quanta size and aspect ratio** – Initially, all of the quanta (in the test case, photo thumbnails) must be of the same size and aspect ratio. To avoid the waste of space in each region, the algorithm removes this requirement once the layout is in place.

- **Secondary regions distributed in quadrants** – The algorithm implements a four fixed quadrant structure surrounding the primary region (see Figure 5a). Within each quadrant, several secondary regions can be placed, but no secondary region can cross a quadrant boundary. This prevents awkward L-shapes caused by using octants for secondary regions (see Figure 5b) and misalignments caused by arbitrary region placements (see Figure 5c).
- **Fixed number of thumbnails** – Users are let to decide on the fixed number of thumbnails the algorithm has to use; this prevents the undesirable feedback loop discussed previously.

- **Fixed order of regions** – To prevent regions to jump around when the canvas size is resized, regions should be laid out in the order they were added.

![Quad1 Quad2 Quad3 Quad4](image1.png)

![Rgn1 Rgn2 Rgn3](image2.png)

![Rgn7 Rgn2 Rgn3 Rgn4](image3.png)

*Figure 5 - Examples of quadrants and secondary regions set ups [7].*

Figure 6 shows the system architecture.

**Personal Photos**

- **BRQ (Bilevel Radial Quantum) layout algorithm** – The BRQ layout algorithm [8] has the objective of generating regions with largest possible thumbnail size. It is characterized as bilevel to indicate that the solution is for two-level hierarchies, as radial to indicate that the secondary regions wrap around the primary region in an ordered manner, and as quantum to indicate that the items in each region are fixed in size and shape. The BRQ layout algorithm comprises three steps:
  1. Distribute the secondary regions among the quadrants, using the *RegionSplit* algorithm.
2. Set the initial quantum width and height, which is guaranteed to be an upper bound on the possible quantum dimensions, using the \textit{InitialQuantumDim} algorithm.

3. Reduce the quantum dimensions (keeping the same aspect ratio) until there is no overflow, using the \textit{ReviseQuantumDim} algorithm; if the quantum dimensions drop below a specified minimum, the layout is handled as a special case.

- **Postprocessing** - Once the algorithm has been run, the following two processes can incrementally improve the automatically generated layout:
  
  - It is possible to increment the thumbnail size in each region until any further increase would cause overflow of the region.
  
  - A scrollbar can appear in a region if that region has substantially more thumbnails than the smallest region, or than any other region, depending on the user’s preference.

- **Dynamic behaviour** - Users have the possibility of experimenting with the algorithms’ original layout. In the dynamic perspective, it is possible to experiment with primary rectangle placement, size and, overall dimensions.

Concerning the system interface, in all cases the primary region highlights one photo, concept, or text block; the secondary regions serve the purpose of showing organized sub-collections that thematically relate to the primary region. Each layout is created by choosing several tags from the annotated collection (for example, family members Al, Shuly, Esther, Simmy, and Lani in Figure 4), and then choosing a representative photo for the primary region in the center (for example, Penina and Jack in Figure 4). The layouts are then resized to the desired dimensions, and the primary region is filled and scaled to the desired location and size. Furthermore, even a collection of this size can be resized and manipulated in real-time.

### 2.2.1.3 Algorithm Description

#### A) Bilevel Radial Quantum (BRQ) layout definition

Based on the requirements defined in the previous section, the BRQ algorithms briefly described in this section will generate regions with the largest possible thumbnail size. The BRQ layout algorithm comprises three steps in order to address the requirements previously mentioned:

1. **Distribution** - Distribute the secondary regions among the four quadrants, using the \textit{RegionSplit} tool.

2. **Initialization** - Set the initial quantum width and height, which is an upper bound on the possible dimensions, using the \textit{InitialQuantumDim} tool.

3. **Reduction** - Reduce the quantum dimensions until there is no overflow (keeping the same aspect ratio), using \textit{ReviseQuantumDim} tool.

All the steps have their own algorithm, which is briefly described in the following:

- **Distribution: RegionSplit** – This algorithm optimally divides regions among the four quadrants. One region \(n\) is chosen such that the sum of the thumbnails in all regions up to and including region \(n\) is closest to 1/2 of the total number of thumbnails in all regions. This splits the regions into two groups; following this step for each two groups, a total of four groups are generated. There is the possibility of obtaining edge layouts (non-centered rectangles); in this case, the
RegionSplit algorithm must be modified. If the primary region is moved or sized so that a quadrant doesn’t have room for any regions, edge layouts occur in which the primary region is aligned with one or more edges of the layout. To correctly handle these layouts, the algorithm must be modified by determining the number of nonempty quadrants and proceeds as follows:

- If there are four quadrants, proceed to the normal RegionSplit.
- If there are three, proceed to TriRegionSplit.
- If there are two, perform RegionSplit only once, to yield two regions.
- If there is one, allocate all regions to that quadrant.
- If there are none, do not show any regions.

Also, there is a special version of RegionSplit, named TriRegionSplit, to handle the division of a collection of $n$ discrete-sized regions as evenly as possibly in thirds (See Figure 7). This algorithm is conceptually similar to RegionSplit although it involves balanced criteria to divide secondary regions among three quadrants. To better understand how to make a bifurcating algorithm generate three regions, there are three distinct steps:

1. Divide the $n$ regions into two, as described in the first step of the RegionSplit algorithm. At this point, there are two regions closest to the RegionSplit split point (see Figure 7), one to the left and one to the right, at distances $a$ and $b$, where the distance is the number of thumbnails in the adjacent region.

2. Move one of the regions adjacent to the RegionSplit split point to a third pile, such that after the move, the average distance from each side to the absolute centre $n/2$ is as close to equal as possible. In Figure 7, $b$ would be added.

3. Repeat Step 2 until adding a region causes $|n/3 - \text{new pile’s area}|$ to be more than it was before that iteration of step 2; when this happens, backtrack one and save the result as the answer.

### Initialization: InitialQuantumDim

This algorithm sets the initial dimensions for the maximal thumbnail size. This means that the tool’s output (length, width) is defined to be an upper bound on the thumbnails’ size. In the (rare) case where every region has exactly rows $\times$ columns thumbnails, this will also be the final thumbnail size. In every other case, the thumbnail size will need to be reduced until there is no overflow (excess of thumbnails to fulfil a region).

### Reduction: ReviseQuantumDim

This algorithm takes as its input the dimensions that are an upper bound on the possible thumbnail dimensions. After, it revises those dimensions downward until there are no more columns (quadrants 2 and 4) or rows (quadrants 1 and 3) than the available space among all the quadrants. Thus, the algorithm starts with the upper bound on the thumbnail width, and reduces it by one pixel until there is no more overflow. Then, whatever extra space is redistributed at the end among all the regions in the quadrant. Using the current thumbnail size, the step is repeated for the
other quadrants. Finally, when all quadrants have been processed, the last thumbnail size will be the correct one.

Figure 7 - Dividing secondary regions among three quadrants [7].

B) Postprocessing definition

Once the BRQ layout algorithm has been run, certain actions can incrementally improve the automatically generated layout. Two are described in the following:

- **Vary thumbnail size** – Once the initial algorithm has run, it is possible to increment the thumbnail size in each region until any further increase would cause overflow of the region. This allows each region to have a minimum of wasted space, at the expense of the photos in different regions no longer lining up.

- **Add scrollbar** – For the layouts to scale properly, a scrollbar can appear in a region if that region has substantially more thumbnails (currently set at 20x) than the smallest region, or than any other region, depending on the user’s preference. Adding the scrollbar involves deciding how many thumbnails to show (some maximum per region), and then adding to that number on a per-region basis to enforce a full grid when not all photos are visible. For example, if a maximum of 40 thumbnails out of a region’s 70 are shown, and there are seven columns, the fifth row will have only five thumbnails. In this case, two thumbnails should be added to the last row so that the grid will be full unless the scrollbar is at the last position. These must be added as a post-processing step so that the fixed number of thumbnails requirement is not violated during the initial layout.

C) Dynamic behaviour

The algorithms described previously allow for interactive, dynamic behaviour (e.g. dragging and sizing actions) encouraging to experiment with primary rectangle placement, size, and overall dimensions. The typical behaviour the proposers wished to support is to allow users to move the primary rectangle from the upper-left corner into the centre, and then enlarge it to highlight its contents (see Figure 8). Other behaviours include users resizing the canvas and adding and deleting regions as well as thumbnails.
2.2.1.4 **Performance Evaluation**

The performance will be evaluated in two different ways: the dynamic behaviour of the algorithms described, and the usability of the system:

- **Dynamic behaviour** – Three trials were conducted, in which regions were incrementally added with 10, 50, and 100 thumbnails per region, until there were 100 regions added. Execution time was measured for the three algorithms in the BRQ layout algorithm, demonstrating that the algorithms were rapid enough, with no unforeseen crashes (computer freezing) even at the high end of 100 thumbnails in each of the 100 regions, for a total of 10,000 quanta being positioned. For this test case, the *RegionSplit*, *InitialQuantumDim*, and *ReviseQuantumDim* algorithms took 0.037 millisecond (ms), 0.014 ms, and 0.14 ms, respectively, on a Pentium IV 2.4-gigahertz processor with 1 Gbyte of RAM. Additional tests confirmed the linear growth of time with the number of thumbnails. The placement of the photos into the respective regions was more time consuming, spending as much as three orders of magnitude (350 ms) greater drag on performance than any of the resizing algorithms.
• **Usability study** – To gauge user responses to these BRQ hierarchical layouts, four knowledgeable users of photo library software were asked to review the interface for 30 to 40 minutes. In this modest usability study, they were shown the on-screen, but static, layouts in Figure 4 and Figure 8 in order and asked what they understood about the layout and relationship among the regions in each figure. The users understood why a particular thumbnail size was chosen, although sometimes only after carefully comparing regions to find one in an overflow situation. Several voiced interest in a feature that would relax the thumbnail size requirement to have less wasted space. They all appreciated the ability to manipulate the layout in real-time.

2.2.1.5 **Summary**

This approach is an example of a photo browsing, dynamic layout, with content based filling of a previously annotated collection. The main advantage of this system is that many applications, such as digital photo layouts, could use these algorithms to do the dynamic layout quickly and deterministically. As a matter of fact, a vital aspect of the BRQ layout algorithm is its rapid performance, which enables compelling interactive experiences as users resize the canvas, add or delete regions, and add or delete items to a region. The system’s overall performance is highly satisfactory in terms of user’s interactivity.

2.2.2. **Genetic Algorithms for Album Page Layouts**

The system described here was developed by Geigel and Loui [9] and proposes the usage of genetic algorithms for the creation of album page layouts. In terms of the classification tree proposed in Section 2.1, this system fits under the photo browsing, dynamic layout and content-based filling criterion branch.

2.2.2.1 **Objectives**

The main objective of this photo browsing solution is to generate an automated layout, producing visual pleasing results. Given a set of images in an album, a page-layout algorithm should distribute the images among a set of pages and lay out the images on each individual page. Therefore, the system that will be described here generates a set of personalized album pages for visual content. The proposers aim to produce album pages that closely resemble those in scrapbooks as opposed to a simple collection of photos (see Figure 9). However, the subjective nature of this creative page layout poses a real challenge to any layout system. Thus, the use of genetic algorithms intends to solve the problem by originating solutions that converge on layouts that closely match the user’s layout preferences.
2.2.2.2 Architecture and Basic Approach

A simplified architecture of the system described in this section is shown in Figure 10.

Figure 10 – System architecture [9].

A short walkthrough of the system is presented next:

- **Image Preprocessing** – This pre-processing module has the target to gather important information about the content and quality of each photo; after, photos may be sorted by event and evaluated for quality.

- **Album Creation** – This module consists of two sub-modules, each driven by a genetic engine. The page creation module assigns each image in the collection to a given page while the page layout module generates the layouts, this means it positions and orients the images on each individual page.

  Both modules use a genetic engine that randomly generates solutions and judges the suitability of each solution for its associated task. The determination and evaluation of a solution’s fitness is based on a set of graphic design criteria and comparison with the user preferences.

2.2.2.3 Algorithm Description

The system is based on genetic algorithms; therefore, it is important to give a brief explanation of that type of algorithms, notably regarding their general concept and procedures.

A genetic algorithm could be simply described as a class of adaptive methods that may be used to solve search and optimization problems involving large search spaces. The genetic algorithms perform the search process in an iterative way as shown in Figure 11. First, the algorithm generates an initial population of solutions. This initialization is typically done randomly, although nothing prohibits using a heuristic for this process. Once a population is established, the algorithm derives individual solutions from the population’s encodings, judged based on the fitness function, and then chooses suitable individuals for crossover and mutation. Next, the algorithm performs crossover and mutation on the chosen encodings, resulting in encodings of new individuals. A new
generation of solutions is created from these newly generated offsprings. Finally, the algorithm iterates this process until it meets a defined stopping criterion.

Figure 11 - Flowchart for genetic algorithm processing [9].

When applying a genetic algorithm to a given problem, three major tasks must be performed:

- **Coding** - Genetic algorithms maintain populations of problem solutions. During implementation, these solutions are encoded using some data structure (genotype). Like DNA in living organisms, this data structure will represent the genetic code and contain all the information needed to construct a solution. Common genotype data structures used in genetic algorithms include lists, arrays, and trees. Each instance of a genotype will represent a given instance of a solution within the problem space (phenotype). Part of the coding definition is a description of the conversion from genotype to phenotype.

- **Defining genetic operators** - The algorithm creates new solutions via crossover and mutation of encodings of individuals from previous generations. Given a particular genotype structure, the user must define the means for performing these operations. During crossover, one or more children solutions are derived from two or more parents. With mutation, the algorithm generates new individuals by mutating a single encoded solution. Figure 12 gives examples of crossover and mutation operators for genotypes defined using arrays and trees. In Figure 12a, for crossover, the algorithm randomly selects a position within the array and then it swaps the array values between the two parents; for mutation, the algorithm chooses and swaps two random array elements. In Figure 12b, for crossover, the algorithm selects a random node for each parent and swaps the sub-trees from these chosen nodes; for mutation, it randomly chooses nodes from the same tree, and then swaps them.

- **Fitness determination** - The most challenging and application specific task in applying genetic algorithms to a problem domain is in defining a fitness function. The fitness function judges individual solutions and returns a score based on its evaluation. In essence, the fitness function defines the difference between a good solution to a problem and a poor one. Much care must be taken in defining the fitness function, as the genetic algorithm will converge on solutions deemed “fit” by this function.

Further details on these algorithms may be found in [9].
Figure 12 - Genetic operators for arrays (a) and for trees (b).

The main modules in this system are the Preprocessing and the Album Creation modules the last one with two sub-modules, as can be seen in Figure 10. The algorithms for those modules will be briefly described in the following:

- **Preprocessing Algorithm** – To perform this task two independent algorithms were implemented: for automated event-clustering, an algorithm using a K-Means clustering approach, with a block-based histogram correlation method has been used [10]; for image-quality screening, an algorithm to detect problematic images caused by underexposure, low contrast, and camera defocus or movement [10]. This module has also interesting capabilities in terms of photo relevance; this involves a numerical score, named emphasis value, which measures the most valuable pictures in terms of composition, colorfulness, the presence of people and faces, expressions, mood, and so on, within the group of photos resulting from an event-clustering. To sum up, as the images exit the pre-processing phase, it is assumed that each is dated and time stamped, grouped by event, and has an emphasis value assigned to it.

- **Page Creator Algorithm** - This algorithm is part of the Album Creation module and is based on genetic processing. This solution uses a tree-based structure, inspired by the space-filling algorithm proposed by Shneiderman [11]. The nodes at the second level represent individual album pages. At the third level the nodes represent visually grouping images within a given page. The module uses standard crossover and mutation operators for trees (as illustrated in Figure 12b) to genetically manipulate the structure.

- **Page Layout Algorithm** – This algorithm is part of the Album Creation module and is also based on genetic processing. As previously explained, there are three
essential tasks that must be performed when applying genetic algorithms to a problem. Therefore, this algorithm work as follows:

- **Coding** – The position and orientation of an individual image is defined using a 4 tuple \((x, y, s, \theta)\) where \((x, y)\) is the position of the image, \(s\) is the scaling factor, and \(\theta\) is the rotation. A floating-point array is used as the genotype for the page layout module (see Figure 13), where each element of the array, or gene, corresponds to an individual value of the 4 tuple. The complete genotype for a layout contains the image positioning values of all images to be placed, resulting in an array of length \(4N\) for a page with \(N\) images. Individual gene values are appropriately scaled to ensure that all elements of the array are floating point values ranging from 0 to 1. This guarantees that all genes are equally considered when performing genetic operations.

- **Defining genetic operators** – The page layout module uses standard crossover and mutation operators for arrays (see Figure 12a). Alternatively, the module uses an additional mutation operator, which chooses a random gene within the array and deviates its value by a random amount. The system carefully ensures that gene values remain between 0 and 1 after the mutation.

- **Fitness determination** – To determine the fitness degree, it is essential to define the concept of user preferences. In this module, the goal is to define a score which measures how well the solution matches the user preferences and expectations. To quantify this, a number of criteria are defined by which each solution will be judged. Each genetic module will judge solutions by comparing them to the user preferences with respect to a set of criteria. These values are called preference parameters, and assume a value of 0-low to 1-high. In addition to the preference parameters, importance parameters are introduced, indicating how much (0-low, to, 1-high) variation the system is allowed with respect to a given criterion. Also, in calculating the fitness for a given solution, each genetic module will judge a solution regarding the same set of criteria, giving a score of 0 to 1 for each criterion. The criteria set for layout includes:
  - spatial criteria (white space, overlap, and rotation),
  - balance criteria (spatial balance, rotational balance, border symmetry), and
  - emphasis.

The genetic module compares evaluation scores with the user preference. Then, the comparison is scaled by the importance parameter values for each criterion. Mathematically, it is possible to summarize this final fitness score calculation as:
where $I$ represents the set or importance parameter values, $P$ represents the set of preference parameter values, and $E$ represents the set of raw evaluation scores, in all cases one per criteria.

A detailed description of these criteria and the evaluation methods is provided in the next section.

2.2.2.4 **Performance Evaluation**

The evaluation for each criterion has a score from 0 to 1; scoring details for each criterion are given in Table 1.

<table>
<thead>
<tr>
<th>Evaluation Criteria</th>
<th>Description</th>
<th>Extreme Scores</th>
</tr>
</thead>
<tbody>
<tr>
<td>White space</td>
<td>Measure of the amount of white space on the page.</td>
<td>Layout where the images placed on the page take up the entire page area.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Indication of an empty page with no images on it.</td>
</tr>
<tr>
<td>Overlap</td>
<td>Maximum percentage of any image area covered by another image over all the images placed on the page.</td>
<td>Scores below the minimum or greater than the maximum amount of overlap allowed (unacceptable).</td>
</tr>
<tr>
<td>Rotation</td>
<td>Indication of the total amount of image rotation on an album page.</td>
<td>Average is scaled by $\theta_{\text{MAX}}$ to achieve a score between 0 and 1.</td>
</tr>
<tr>
<td>Spatial Balance</td>
<td>Comparison of the image areas in the four quadrants of the album page.</td>
<td>Spatial unbalanced layout.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Spatial balanced layout.</td>
</tr>
<tr>
<td>Rotational Balance</td>
<td>Average of all the rotation values of the images.</td>
<td>Rotationally unbalanced layout.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Rotationally balanced layout.</td>
</tr>
<tr>
<td>Border Symmetry</td>
<td>Evaluation of how closely the edges of the image on the page form a natural boundary.</td>
<td>Unfit solution.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fit solution.</td>
</tr>
<tr>
<td>Emphasis</td>
<td>Measure of the proportionality of the images’ sizes with respect to the emphasis values assigned.</td>
<td>Negative correlation between image size and emphasis values.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Strong positive correlation between image size and emphasis values.</td>
</tr>
</tbody>
</table>
A sampling of layouts produced by the page layout module for a page containing four images is given in Figure 14. For each of the examples, the page layout module runs the genetic algorithm through 300 generations, resulting in layouts with final scores ranging from 0.85 to 0.95.

In Figure 14 from (a) to (h), the effects of user preference settings are illustrated: (a) white space; (b) overlap; (c) rotation; (d) rotational balance; (e) spatial balance; (f) border symmetry; (g) emphasis; and (h) several criteria. Particularly, the module’s skillfulness at varying page layouts is demonstrated best if it is modified several preference parameters at once before running the genetic algorithm. In Figure 14h, the preferences for white space (in Figure 14, yellow space), rotation, and overlap are all modified between two layouts. This results in layouts with two entirely different feels,
the first being very traditional and rigid while the second has more of a capriciousness quality to it.

2.2.2.5 Summary
This approach is an example of a photo browsing, dynamic layout, and content filling criterion photo consumption solution. This system is interesting since it uses genetic algorithms to generate solutions with a fitness based on user preferences defined using some basic graphic design criteria. The genetic algorithms were chosen because of the artistic nature of the scrapbook layouts. Since genetic algorithms operate by creating and evaluating solutions rather than trying to mimic an existing process, there is no need to reverse engineer the artistic task that humans employ in generating creative album pages. This system generates a number of interesting and creative solutions to the layout problem with the best solution chosen using the fitness function. The main difference to the system presented in the previous section is the dynamic behavior. In this system, the changes are automatically performed by the algorithms while on the previous one the user has a more active role in terms of manually changing the layout structure.

2.2.3. Content Based Photograph Slideshow with Incidental Music
The system described in this section was developed by Hua, Lu and Zhang [12] and proposes a content based approach for a photo slideshow. In the classification tree proposed in Section 2.1, this system fits under the photo presentation, static layout with content based filling branch.

2.2.3.1 Objectives
During many years, the slideshow concept has remained without many improvements in terms of its visual approach. In order to improve the visual experience, it is possible to extract some content information from photos and musics, to re-organize and match the photos and music based on their content. Therefore, the proposers of this photo slideshow aim to present a more attractive and enjoyable style of presentation by adding music to the photos in a ‘correlated’ way. The main objective of this photo presentation solution is to achieve more reasonable and pleasing results; for this, the following principles, suggested by many users, were taken into consideration:

- Low quality photos should not be selected; here low quality photos include those blurred or under/over exposed.
- For very similar photos, only one of them should be shown on each session.
- Music should be suitably matched with the photo slideshow.

For instance, photo transitions should occur at the music beat, which can make the photo slideshow more pleasing. Based on these principles, content-based analysis on photograph and incidental music is necessary. Generally, incidental music is often "background" music, and adds atmosphere to the action; in this case, the action is the slideshow itself.
2.2.3.2 **Architecture and Basic Approach**

The architecture of the system proposed for fusing music and photos to generate a photo slideshow using content-based features is shown in Figure 15.

![Figure 15 – System architecture [12].](image)

A short walkthrough of the system is presented next:

- **Music analysis** – In order to make the photo transitions happening at the beat positions of the incidental music, it is necessary to detect the beat series in the music. After, incidental music is segmented into sub-music clips to perform later the match with the photos.

- **Photos analysis** – It is important to emphasize that not all photos are shown in this implementation. First, the low quality photos are filtered out by a photo quality detector. Then according to the proposers’ decision, the remaining photos are grouped into a three-layer structure, according to their colour and time similarity, which are named as (from high to low) layer importance, day, scene and GoS (Group of very Similar photographs). Then, one photo is selected from each GoS as the photo set to be used for generating the slideshow.

- **Slideshow composition** – Based on the previous analysis results, the fusing algorithm to combine photos and music together to get the final output is applied. Also various transition effects and transformation effects are likely to be applied to improve the user’s experience. Each photo is connected by specific transitions, which occur at the strong beats of the music based on the structure and content of the photos. Transformation effects are also available on each photo, such as grayscale, blurring, fading
in/out, rotation, thresholding, etc. According to the different users’ preferences, several editing styles are also supported by this algorithm.

All algorithms above are briefly described in the next section.

2.2.3.3 Algorithm Description

As can be observed in Figure 15, the first level in the block diagram includes two different stages, notably:

- **Music analyses** - The algorithm proposed in this module focuses on sub-music clips that are used as the basis for the photo slideshow. The approach of this sub-music segmentation algorithm is interesting since it tries to efficiently solve the beat detection problem in a simple way. The exact beat series are not detected, only onset, since beat information is not obvious, especially for that no-drum light music which is commonly selected to accompany photo presentations. Therefore, this is an interesting new approach based on user supposed preferences. Basically, there are two consecutive steps, the detection of the onset, and the sub-music segmentation. To detect the onset sequence, an octave-scale filter-bank is applied to divide the music frequency domain into several sub-bands:

\[
\begin{bmatrix}
0, \frac{a_0}{2^n}, \frac{a_0}{2^n}, \frac{a_0}{2^n}, \ldots, \frac{a_0}{2^n}, \frac{a_0}{2^n}
\end{bmatrix}
\]

where \( a_0 \) is the sampling rate and \( n \) is the number of sub-band filters.

After, and supposing the tempo of the music\(^1\) constant, the range of the sub-music length can be experimentally set as:

\[
SubMusicLen_{\text{min}} = \min\{\max\{2 \times \text{Tempo}, 2\}, 4\}
\]

\[
SubMusicLen_{\text{mix}} = SubMusicLen_{\text{min}} + 2
\]

where \( SubMusicLen_{\text{min}} \) and \( SubMusicLen_{\text{mix}} \) is the lower bound and the upper bound of the sub-music length. The music tempo can be obtained from the auto-correlation of the onset sequence.

- **Photos analysis** - The quality filtering involves the removal of low quality photos, because most photos are taken by unprofessional users; the following cases are considered: under or over exposed images, similar photos and blurred photos. The photo grouping involves a three-layer structure. Photos are first grouped into the top-layer “day” based on date information. Then, a hierarchical clustering algorithm with different thresholds is used to group the two lower layers. Colour histogram intersection is used as the similarity measure for two photos or two photo clusters. The proposers use a HSV colour cone that is quantized by a 3D Cartesian coordinate system with 20 values for X and Y, and 10 values for Z (lightness), respectively. Let \( h = \{ h_i, 0 \leq i \leq N \} \) and \( g = \{ g_i, 0 \leq i \leq N \} \) denote the quantized colour histogram, the intersection of them is defined as:

---

\(^1\) The speed at which music is or ought to be played, often indicated on written compositions by a descriptive or metronomic direction to the performer.
where \( N \) is the dimension of the histogram, equal to 4000 in this implementation. Two thresholds, \( T_{\text{scene}} \) and \( T_{\text{GoS}} \), are set to determine whether two photos are in the same scene or GoS, respectively. If the similarity of two photos or photo clusters is larger than \( T_{\text{scene}} \) or \( T_{\text{GoS}} \), they are merged into one cluster. The photo selection regards similarity, to prevent similar photos from appearing adjacently; only one photo is selected from each GoS in the final output.

Afterwards, the results of the two independent stages above are fused to generate the slideshow, as explained next:

- **Slideshow composition** - The main idea here is to connect photos by transitions at the beats of the music, together with transformation effects to get a more enjoyable output. The transformation and transitions used are provided by Microsoft Software, Movie Maker [14] and DirectX. Different styles are supported according to the users’ preferences; still there are three predefined styles, namely, music video, day by day, and old photo. In the music video style, first the music is segmented according to the tempo of the music. That is to say, if the music is fast the sub-music will be shorter, and vice versa. Then photographs and music are fused together to get the slideshow. In the day by day style, when a new day comes out, a man-made photo is added before the first photo of the day to illustrate the creating date of the photos coming next. In the old photo style, sepia tone or grayscale effect is applied to all photographs, while only “fade right” transitions are used between photos.

Figure 16 illustrates the interface of the system prototype.

![Figure 16 – Interface for the prototype of the Photo Slide Show application [14].](image)

### 2.2.3.4 Performance Evaluation

The performance of this slideshow system has been evaluated by a user study. The system has been compared with some commercial products available, notably ACDsee
and PhotoJam. The system was tested with three sets of photos and music: photos in Test1 were sightseeing analog scanned photos while photos in Test2 and Test3 were digital photos of outdoor events and indoor conferences, respectively; so, there are nine sets of results in total. All users were required to give a satisfaction score to each slideshow produce. The score of the first photo slideshow generated by ACDSee was fixed to 0.50 in order to provide the users with a benchmark; thus, the users could take it as an example to give scores to the other results. Average satisfaction values are listed in Table 2, showing that the presented system has reached a much higher level of satisfaction from the users than the considered alternatives.

Table 2 - Results of the user study.

<table>
<thead>
<tr>
<th>System</th>
<th>Test1</th>
<th>Test2</th>
<th>Test3</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACDSee</td>
<td>0.50</td>
<td>0.48</td>
<td>0.45</td>
<td>0.48</td>
</tr>
<tr>
<td>PhotoJam</td>
<td>0.73</td>
<td>0.69</td>
<td>0.62</td>
<td>0.68</td>
</tr>
<tr>
<td>Proposed System</td>
<td>0.85</td>
<td>0.88</td>
<td>0.86</td>
<td>0.86</td>
</tr>
</tbody>
</table>

2.2.3.5 Summary

This work presents an approach to the photo consumption problem fitting in the photo presentation branch since it mainly serves to display user photos with minimum user interaction. Moreover, the proposers use content-based criteria to fill the spatial layout in a static manner (photos are showed one by one). Even though some temporal-based criteria are used at same stage, there are not enough arguments to refer to this system as hybrid (both content-based and temporal-based). The main difference of this system regarding those previously presented is its user orientation - the options made in the implementation regard user satisfaction.

2.2.4. An Audiovisual Tiling Slideshow Presentation

In [17], Chu, Chen, and Wu propose a new type of audiovisual presentation where a slideshow system explores both content and temporal criteria to select the information. Regarding the classification proposed in Section 2.1, this solution fits under the photo presentation with dynamic layout branch using both content based and temporal (hybrid) filling criteria.

2.2.4.1 Objectives

The main objective of this tiling slideshow system is to address the rather common shortages enumerated in the following:

a) Develop conventional slideshows – Sequential presentations often take much time and make users tired.

b) Quality analysis – Photos taken by amateurs often suffer from quality degradation.

c) Improve displaying method – Conventional slideshows display photos one-by-one, therefore, the visual experience is cut off.

2.2.4.2 Architecture and Basic Approach

The proposed system generates a descriptive presentation via elaborate photo arrangements. According to technical writing guidelines, a solid paragraph contains a topic sentence that identifies the main idea and several supportive sentences that
provide complementary materials. Many paragraphs are concatenated to convey the whole narration of an article. Likewise, it is deemed that a journey or an event can be reproduced by many photo paragraphs, which will be named layout. Each layout is composed of a larger-sized topic photo (topic cell) and several smaller-sized supportive photos (supportive cell). Tiling multiple photos into the same frame strengthens the atmosphere of the viewing experience. A photo presentation that is synchronous to music beats further improves the enjoyment of browsing.

As a consequence, the proposers of this system designed a system that integrates both visual and music analysis and automatically composes a vivid audiovisual presentation. The architecture of the proposed tiling slideshow application is shown in Figure 17.

![Figure 17 – System architecture [17].](image)

A short walkthrough of the system is presented next:

1. **Beat detection** – This module performs the detection of music beats and their usage to drive the progress of the presentation. Hence, usage of beat information serves as the timer for photo presentation, both in layout switching and photo displaying.
2. **Photo preprocessing** – This module regards photo manipulation at a rather low-level since it performs orientation correction and removes low-quality photos caused by blur and/or underexposure/overexposure.

3. **Photo organization** – This module automatically organizes the photo collections by using time and content characteristics; it integrates these two types of characteristics to perform a finer clustering so that photos at the same scenic spots or presenting the same event are grouped together.

4. **Composition** – This composition is made regarding two perspectives: from the temporal perspective, photo presentation and frame switching are synchronized to the music beats; from the spatial perspective, photos having similar characteristics are elaborately manipulated and arranged in the same frame.

2.2.4.3 **Algorithm Description**

A brief description of the algorithms for each module in the architecture presented in Figure 17 is provided next:

- **Beat detection** - The proposers detect music beats based on the algorithm proposed in [4]. In the following, it is described how the timing for layout switching and photo displaying are determined.
  - **Timing for layout switching** - In addition to music beats, the proposers also consider sound energy differences between adjacent audio frames for frame switching. In the example in Figure 18, if the starting time of layout 1 is \( t_1 \), the sound energy differences in the range from \( (t_1 + r_1) \) to \( (t_1 + r_2) \) are checked to detect the largest energy difference in this range. To guarantee the coordination between visual and aural media, the time stamp of the nearest beat to the largest energy difference is set as the timing for layout switching, like time stamp \( t_4 \) in Figure 18. In this implementation, \( r_1 \) and \( r_2 \) can be adjusted to control the displaying speed and meet different people’s preferences.
  - **Timing for photo display** - For each layout, the photo’s occurrence time stamp has to be determined. In this implementation, the proposers average and distribute the displaying duration (for example, from \( t_1 \) to \( t_4 \), as shown in Figure 18) for each photo. Thus, they found the time stamps of the music beats that are nearest to the average distributed points. With this elaborate design, the proposed scheme synchronizes the visual display with the music’s pace.
8 - Example of determining the timing for layout switching and photo display [17].

- **Photo preprocessing** – The proposers intend to address the photo orientation problem which derives from the inconsistency between the user’s intuition in browsing and the angles of photos. It is also intended to filter out low quality photos before the system organizes the photos. These two pre-processes prevent users from a rather tedious work.

- **Orientation correction** – This module corrects the photo orientation by checking the available EXIF information. Fortunately, more and more digital cameras are equipped with orientation sensors and simultaneously store orientation information as EXIF metadata when shooting. Thus, it is easy and reliable to use this information in correcting the photo orientations.

- **Blur detection** – This module adopts a wavelet-based method to detect blur by checking edge characteristics in different image resolutions; with this information, the system can filter out photos with severe blur degradation.

- **Underexposure and overexposure detection** – This module adopts a simple detection method to detect underexposures and overexposures based on the photo intensity characteristics; for instance, when the number of darkness (or brightness) pixels in a photo is larger than a predefined threshold, the algorithm indicates that this photo is underexposed (or overexposed).

- **Photo organization** – In order to realize the concept of a ‘photo paragraph’, the photos in the same cluster are semantically related and are displayed in the same layout. Therefore, the system organizes photos based on time and content characteristics with two different clustering approaches. Also, because the layout space is smaller than that of multiple photos, it is inevitable that some photos will need to be shrink (cropping); therefore, regions of interest (ROI) have to be detected.

- **Time-based clustering** – Photos are first sorted by their shooting time. This algorithm dynamically detects noticeable time gaps by checking the time stamps of photos in a sliding window. These time gaps are associated to
changes of the shooting pace and reveal that photos have been taken in different places.

- **Content-based clustering** – The system uses the average of the normalized dominant color and color layout distance to measure the similarity between photos [18].

- **Region of interest determination** – To determine what is the most important or attractive part in a photo, the user-attention model, presented in [19] is used to detect the region of interest (ROI).

- **Composition** – After completing the aforementioned processes, the photos from the same cluster are put in the same layout. At the composition stage, several challenges are faced sequentially:
  - **Cluster selection** – Photo clusters are sorted based on their cluster-based importance in descending order, and the first N clusters are picked for presentation if only N music segments are available. The importance of each cluster is estimated based on two features: photos per minute (PPM) and photo conformance (PC). PPM denotes the shooting frequency of photos in a cluster, while PC regards the content-based similarities between photos in the same cluster. These two features are fused to describe the importance of a given cluster.
    - **Template design and determination** – According to some publication layout guidelines [20], the proposers designed several layout templates for showing different numbers of photos. The layout templates were designed according to the following principles: Showing limited content in a limited space – The number of photos in a frame is limited to no more than 12; a region that displays one photo is called a cell.
    - **Enlarging important photos to drive visual perception** – Photos in the same layout are elaborately scaled into different sizes to show their relative importance.
    - **Designing layouts by adjusting uniform subunits** – In this implementation, the layout is divided into 12 equal-sized, basic units. Then, to construct a layout template that consists of a number of cells, the needed number of basic units is merged resulting in the desired number of cells. Figure 19 shows some of the templates for showing three, four, and five photos.
  It is also necessary to determine which template is appropriate for showing a given photo cluster; conceptually, more representative photos should be allocated within a larger space. Therefore, template- and photo-based importance values are defined to be the metrics for template determination as described in the following:
    - **Template-based importance** – As Figure 19 shows, each tiling template consists of, at least, one topic cell and several supportive cells. The ratio of the cell’s area over the whole layout determines each cell’s importance. The system then sorts the cells’ importance values in the same template in descending order.
    - **Photo-based importance** – The system calculates the photo-based importance based on the face region and attention value. The system applies a linear weighting method to combine these two features and
derive the photo-based importance; it also sorts the calculated photo-based importance in descending order.

\[
\hat{i}^* = \arg \min_{i=1,2,...,n} \arccos \left( \frac{\mathbf{PV} \cdot \mathbf{TV}_i}{||\mathbf{PV}|| ||\mathbf{TV}_i||} \right)
\]

(5)

where \( \mathbf{TV}_i \) is the corresponding template-based importance vector of template \( \mathbf{Ti} \). The minimum angle between two vectors denotes the best match between photos and templates. Because both importance vectors are sorted in descending order, this process also determines which photo should be put into which cell.

- **Spatial composition** - Once the system determines the match between photos and displaying templates, photos are elaborately resized or cropped to fit in the limited region (cell). The final task to generate a tiling layout is to put photos into the designated cells. Making the parallel with the concept of photo paragraph mentioned above, the processes of cluster selection, template determination, and spatial composition build the photo paragraphs. After determining the timing for displaying photos and switching layouts, the system concatenates photo paragraphs and generates a tiling slideshow.
2.2.4.4 Performance Evaluation

To proposers consider both objective and subjective perspectives in evaluating their methods and system. Five photo sets taken by different amateur photographers have been used for evaluation, as listed in Table 3 which also shows the final presentation's music length.

Table 3 - Data from photo set evaluation [17]

<table>
<thead>
<tr>
<th>Set</th>
<th>Type</th>
<th>Number of Photos</th>
<th>Descriptions</th>
<th>Length of Music (In minutes and Seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Travel</td>
<td>780</td>
<td>Travelling in Japan – Including people, cityscapes, and landscapes.</td>
<td>3:31</td>
</tr>
<tr>
<td>2</td>
<td>Travel</td>
<td>522</td>
<td>Travelling in Australia - Including people, cityscapes, and landscapes.</td>
<td>4:38</td>
</tr>
<tr>
<td>3</td>
<td>Wedding</td>
<td>388</td>
<td>A Chinese wedding ceremony where people are the primary targets.</td>
<td>3:49</td>
</tr>
<tr>
<td>4</td>
<td>Graduation ceremony</td>
<td>227</td>
<td>A graduation ceremony where people are the primary targets.</td>
<td>4:38</td>
</tr>
<tr>
<td>5</td>
<td>Landscape</td>
<td>133</td>
<td>Pure landscape in Taiwan, including mountains, rivers, and forests.</td>
<td>4:38</td>
</tr>
</tbody>
</table>

Concerning the objective evaluation, the following topics were considered:

- **Clustering performance** - The owners of the evaluated photos have been asked to judge whether the photos in the same frame belong to the same event or scenic spot. From the content owners’ thoughts, it was concluded that only a few frames consisted of poorly clustered photos, as shown in Table 4.

Table 4 - Clustering performance evaluation [17].

<table>
<thead>
<tr>
<th>Slideshow</th>
<th>Number of Layouts</th>
<th>Number of Photos</th>
<th>Number of Layouts with a clustering error</th>
<th>Minimum Photos in a Layout</th>
<th>Maximum Photos in a Layout</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>37</td>
<td>127</td>
<td>1</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>48</td>
<td>172</td>
<td>1</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>38</td>
<td>155</td>
<td>0</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>48</td>
<td>212</td>
<td>0</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>5</td>
<td>48</td>
<td>131</td>
<td>8</td>
<td>1</td>
<td>7</td>
</tr>
</tbody>
</table>

- **Cropping performance** - To evaluate the cropping performance corresponding to how well photos were cropped (ROI determination), each photo in layouts was checked, and judged whether it was badly cropped. The last column refers only to the topic cell of the layout, which is of major importance since displays the topic photo of the layout.

Table 5 - Cropping performance evaluation [17].

<table>
<thead>
<tr>
<th>Slideshow</th>
<th>Number of Photos</th>
<th>Total Number of Badly Cropped Photos in All Cells</th>
<th>Number of Badly Cropped Photos in the Topic Cell</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>127</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>172</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>155</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>212</td>
<td>9</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>131</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>
Concerning the subjective evaluation, the users’ satisfaction regarding the slideshows generated by the proposed system and the alternative ACDSee [15] and Photo Story [14] systems were compared. ACDSee generates conventional slideshows by sequentially displaying photos one by one. On the other hand, Photo Story generates camera motion effects on single photos and sequentially switches photos as well. Twenty-seven evaluators were invited to join the user study in order to judge different results according to their subjective perception. Generally, they found the tiling slideshows much more satisfactory than the other options.

2.2.4.5 Summary

This solution is an example of a photo presentation system, with dynamic layout, exploiting content and time features to fill the frames. This system is the most innovative presented in the presentation branch since it includes the totality of features presented for this type of system. By aggregating both content-based and temporal-based criterion to fill the layout, it corresponds to the most complete solution reviewed to address the photo presentation problem. This presentation solution mainly differs from the one previously presented in the use of dynamic layouts.
CHAPTER 3

Architecture and Functional Description

Chapter 3 intends to present a first perspective of the developed musical slideshow application. To achieve this goal, this chapter will present the system architecture as well as a functional description of each of the architecture’s modules. Chapter 4 will after present an in-depth description of the processing solutions adopted to implement each module. Under the classification tree proposed in Chapter 2, the solution developed in this Thesis should be classified as a photo presentation aggregating both content-based and temporal-based criterion to fill a dynamic layout.

3.1 System Architecture

From the systems presented in Chapter 2, the solution developed by Chu, Chen, and Wu from the Taiwan National University and presented in [17][18] has been adopted as the reference system. This system is the most attractive to be used as reference since it includes several features that may be exploited and can be improved; it is also the most innovative and complete solution reviewed to address the photo presentation problem. As a consequence, there will be many similarities between the architecture proposed by these authors and the system architecture developed and implemented in the context of this Thesis.

A major difference between the solution in [17][18] and the system developed in this Thesis regards the interface provided to the user. While Chu’s et al. system only allows three parameters to be changed by the user, in the developed application all parameters from all modules may be changed by the user if he/she so wishes although there are always backup/suggested values that may be used since he/she may not feel
comfortable to manipulate the parameters. This approach allows the user to assess the performance of all the modules of the slideshow creation process, either by changing the parameters or selecting the modules which must be applied. This kind of interaction intends to offer the user a more vivid experience by giving him/her the capability to identify his/her preferred parameters set.

The system architecture is presented in Figure 20: it shows four main core modules and its associated sub-modules. The four core modules are:

- **Photo analysis**
- **Photo organization**
- **Music analysis**
- **Composition**

These modules are considered the core modules as they are fundamental to the slideshow generation process. The input to the application is a set of photos and music. The set of photos selected by the user is first temporally sorted using their creation time. From this point on, all the manipulation, analysis and organization processes may be performed using this natural temporal order.

The main output of the developed application is a musical slideshow this means a dynamic presentation where the set of photos selected by the user is displayed in a temporal sequence and spatial layout synchronized with musical features, targeting the maximization of the user experience, eventually according to the user preferences.
Figure 20 - Architecture of the developed solution.
A brief walkthrough of the application’s architecture is presented next:

1. **Photo analysis** – The first main stage in the slideshow creation process regards the analysis of the photo collection, which starts with the temporal sorting of the photos. Next, the system filters out the photos that should not appear in the slideshow since they are considered to have some artifact or deficiency and, thus, should be corrected/improved or filtered out. This module has an important role since this photo filtering process contributes to increase the system’s overall performance by not showing photos which are not considered to have the ‘minimum quality’. This stage may include several modules associated to various artifacts or deficiencies, e.g. blurring, out of focus; each of the feature filtering processes available in this module is performed independently and, thus, it is possible to filter photos based only on one or more of the selected features. The photo collection resulting as output of this module will serve as the input for the photo organization module.

2. **Photo organization** – The collection obtained after filtering the photo collection is next organized based on several photo features. Contrary to the analysis module, organization is not independently done for each feature since there is a sequential order to perform the organization: first, the photos are temporally clustered and, after, content-based clustering is performed inside each temporal cluster. This approach prevents computation problems, such as memory lacking and excessive execution time. Having the photos first temporally clustered prevents content comparisons between photos with a reasonably high time gap in terms of shooting time. This module produces a visible output for each of the clustering steps. The photo clusters output by this module will serve as the input for the photo composition module.

3. **Music analysis** – This module targets the analysis of the music selected by the user and is performed in parallel to both the previously described modules since they are completely independent. However, both the photos analysis/organization and the music analysis must be performed before the slideshow composition module. As music plays an important role in multimedia presentation, this module intends to take advantage of this by letting the music beats to set the pace for the slideshow and photos layout switching. Thus, the music beat information serves as the timer for the photo presentation which means the output beat information will serve as input for the photo composition module.

4. **Composition** – This module has the central function of gathering and manipulating all the output information from the previous modules in order to generate the musical slideshow. With this purpose, all the photo clusters are given an importance score according to a set of criteria, and then these clusters are sorted to easily determine which clusters to show in the final presentation. Also, the screen templates are determined and allocated to each photo cluster to finally perform the temporal composition, where the combination of the music and the photos will happen based on the determined beats. The output of this module is the final musical slideshow.
3.2 Functional Description by Module

In this section, the function of the sub-modules in the four core modules of the system architecture will be explained.

3.2.1. Photo Analysis

This is a preparatory stage in the development of the system which targets filtering the set of photos based on some criteria. Thus, some decisions had to be taken regarding the choice of the features to be used in this preprocessing. The photo analysis module has the objective of performing some preprocessing based on photo content features, to filter photos from the collection which are not considered in good conditions and could not be corrected/improved. Filtering and analysis is the first step towards achieving the final goal of a slideshow that intends to maximize the user satisfaction; using an analogy, analysis can be viewed as a brute force where important data is collected, while organization is the ‘brain’ in the process where the collected data is exploited to create an appropriate slideshow. The analysis sub-modules have, thus, the following functions:

- **Temporal sorting** – This module intends to temporally sort out the photos so that, in the next modules, they can be already analyzed exploiting their temporal proximity.

- **Similarity detection and filtering** – Commonly, photo collections have plenty of similar photos which are taken indiscriminately. Often, the same scene appears in multiple photos, i.e. people tend to take several photos of the same, or nearly the same, object or event. Therefore, to ensure diversity of presentation, very similar photos should be filtered out, preventing a boring presentation if they appear adjacently. Thus, this similarity detection sub-module intends to filter out duplicated photos in the collection. Figure 21 shows an example of similar photos filtered out from a collection.

- **Blurred detection and correction/filtering** – To maximize the user experience, not all photos will be shown in the slideshow since the low quality photos will be filtered out firstly. Blur is often caused by hand-shaking or out of focus. Blur recovery of an image without any extra information is still a very challenging and hard problem. Instead of removing blur effects from photos, this module intends to know whether the photo is blurred or not and, if yes, how much the blur extent it is. With this information, photos with severe blur degradation can be filtered out or kept for further image processing.

- **Overexposure and underexposure detection and correction/filtering** – This module intends to detect photo overexposure and underexposure which are often due to shooting with incorrect exposal camera parameters. Detection methods may be based on intensity characteristics with intensity thresholds derived from the pixel intensities statistical characteristics in underexposure and overexposure photos. When the number of the darkness (brightness) pixels in a photo is larger than a certain percentage of the total number of pixels, the photo is claimed as an overexposure/underexposure photo and excluded from the set.

- **Orientation detection and correction/filtering** – The orientation problem derives from the randomness of the user’s intuition when choosing the angle.
used for a photo. The most common case is when users take photos with a
digital camera in a vertical position, e.g. the orientation of the scene will be vertical, but the photo will by default be shown horizontally. This happens because nowadays it is supposed by default to take all photos with the camera in a horizontal position due to the display dimensions (16x9). The effect of this randomness is showed when browsing a collection that many times has photos with mixed orientations. However, more and more digital cameras are equipped with orientation sensors and simultaneously store orientation information as EXIF metadata when shooting [6]. This module intends to use this information to ‘correct’ the photos orientation.

Due to time limitations and the associated complexity, only the similarity detection and filtering sub-module was implemented in the developed application in the context of the photos analysis stage. Although further filtering could improve the set of photos, these modules are not ‘essential’ for the slideshow which may still be created although eventually with a slightly lower quality of experience. It is interesting to notice that the non implemented sub-modules did not perform appropriately in the reference system in [17], highlighting how complex it is to perform efficient artifacts detection and correction.
3.2.2. Photo Organization

After performing photo analysis, the following stage in the process regards photo organization. As mentioned before, this module is the ‘brain’ in the process as its visual results should be highly intuitive to the user, i.e. the user should get an immediate perspective of the slideshow structure. To improve the viewing experience, the photo organization module has the objective of organizing the photos so that photos in the same cluster are content and, thus, semantically related and consequently are displayed in the same layout. Therefore, the system organizes photos based on time and content characteristics, as will be detailed next:

- **Time-based clustering** – One of the most effective ways for organizing photos is to exploit the time information. From a time perspective, photos taken within a certain time period usually share the same topic and record the same semantic events, such as a party or a visit to the Zoo. Therefore, the time information automatically embedded in the photos provides a great help to their effective organization since photos within the same cluster should share a high degree of semantic correlation. In principle, the user should be able to control the granularity of the temporal clusters by defining an adequate time interval parameter to temporally group the photos.

- **Content-based clustering** – In addition to the time-based clustering, the spatial context of the photos in the same temporal cluster is exploited to perform a finer organization of the set of photos. Since the homogeneity of adjacent photos tends to enrich the viewing experience, it is expected the photos in the same layout to be visually similar to each other. Therefore, this sub-module has the target to perform content-based categorization for the photos in the same time-based cluster. In this context, the user should be able to change the content clustering criteria and the associated parameters although there will be always be backup solutions available.

- **Region Of Interest (ROI) determination** – Because the layout space is limited, it may be useful to shrink some of the photos into smaller sizes to fit in one specified part of the layout. The simplest way to directly resize photos is according to their aspect ratios. However, blindly resizing often causes significant information loss since the details of important objects may be rudely shrunk. In order to make information loss as small as possible, this sub-module targets cropping the region from the original photo which retains its most important and attractive part, the so-called region of interest (ROI).

Figure 22 illustrates time and content-based clustering for a photo collection. The time-based clusters are displayed vertically, i.e. each line represents a time cluster; for each ‘horizontal’ time-based cluster, the content-based clusters are horizontally separated by a block of information about the content cluster (in this case the dominant colors).
Due to time limitations, the ROI sub-module was not implemented in the photo organization module of the developed application. This sub-module is not ‘essential’ although the user experience performance may slightly decrease due to its absence.

### 3.2.3. Music Analysis

Music plays an important role in multimedia applications, since it is constantly present in people’s life. Therefore, the music analysis module has the objective of extracting relevant features to allow taking advantage of the possibility of coordinating the music beats with the photo switching in the slideshow since this typically improves the user experience. Music analysis includes two sub-modules that are described in the following:

- **Beat detection** – This module intends to analyze the music in order to detect beats that correspond to the sense of equally spaced temporal units. These music beats are used to drive the progress of the photos presentation.

- **Music segmentation** – This module has two main functions: the first is to determine the number of photo clusters that are possible to show within the time-limited music clip; the second is to divide the music into sub-units according to a user predefined length parameter. Thus, as photo clusters will have a sub-unit attributed, this segmentation directly affects the slideshow duration and the cadence of the displayed photos.

### 3.2.4. Composition

To finalize the slideshow creation process, the information resulting from the previous modules needs to be processed in order to generate the desired slideshow. The composition sub-modules are briefly explained next:

- **Cluster importance determination** – This sub-module has the objective of attributing importance scores to the temporal and content clusters.
according to a number of preselected features. This procedure will allow a further rearrangement of the photo collection in order to improve results since more relevant/exciting photos should get some priority.

- **Cluster selection** – This sub-module has the objective of reordering both the time-based and content-based clusters according to the importance scores resulting from the previous cluster importance determination sub-module. This cluster reordering allows deciding which clusters are best to show within the available time. In practice, this sub-module makes the final selection regarding the clusters to display.

- **Template determination** – Once the clusters to be displayed are sorted, several layout templates are available for showing a different number of photos in the presentation layout. This sub-module intends to determine which template should be allocated to each photo cluster, and for each cluster how are the photos distributed in place and size in the frame layout. This sub-module is crucial to define the visual aspect of the final application output.

- **Temporal composition** – This sub-module makes the bridge between the photos and the music since the spatial photo templates are filled with content-clusters at a pace determined by music features, in this case the music beats. Also, each photo within a cluster may enter the layout according to a metric related to the music information; that information may be crucial to define the time that each photo will stay in the slideshow.

- **Player** – The player module uses the results of the previous sub-modules to create a visual and audio output, this means the final musical slideshow presentation. In practice, the output is a sequence of spatial templates filled with photos that are displayed synchronously with the music beats.

Figure 23 shows example results of the composition module. In the top, the temporal clusters are sorted according to the cluster importance while, in the bottom, the slideshow screenshot shows the resulting layout from the composition sub-module.
Chapter 4 will provide an in-depth description of the processing algorithms used for all modules and sub-modules presented in this chapter.

Figure 23 - Composition screenshots: Cluster selection (top) Slideshow layout (bottom).
CHAPTER 4

Processing for Musical Slideshow Generation

This chapter will offer the reader a detailed description of all algorithms implemented in the context of the musical slideshow application developed. The description will be made according to their ‘appearance’ order in the system architecture and also their associated importance. Since the photo analysis and music analysis modules are hierarchically in the same position, priority has been given to the photo analysis since photos are the primary content in this application. As explained in detail in Chapter 5, the application has been implemented using the C# language with Microsoft Visual Studio 2008 (.net framework 3.5).

4.1 Photo Analysis

This section intends to motivate and detail the algorithms used in the photo analysis module.

4.1.1. Temporal Sorting

The first algorithm to be addressed in this system has the target to automatically sort the photos by their creation time. As this is the first algorithm being described in this chapter, some introductory aspects of the implementation will be mentioned, such as the data structure adopted to store the photos. This intends to give a high level overview of the options made concerning the wide universe of sources available in the software. The algorithm comprises the following steps:
• **Data structure creation** – A matrix named `datetaken` with \( N \times 6 \) entries is created, where \( N \) represents the number of photos of the collection. The 6 entries (columns) represent the year, month, day, hour, minute, and second in this specific order.

• **Date taken feature extraction** – Using the available time metadata, a property named `datetaken` [21] is accessed to retrieve the time information for each photo. The `datetaken` feature is parsed to a format `yyyy:MM:dd HH:mm:ss`, and then, for each photo, a matrix entry is filled with this information.

• **Photo temporal sorting** – The locations of the photos (the paths of the location where photos are stored in the computer) are stored in a photo array; in this chapter, all the manipulations are done according to the positions of the photos in that array. Therefore, temporal sorting re-arranges the photos’ path positions in that photo array concerning the temporal order. After all the dates are stored in the `datetaken` matrix, comparisons between the different fields of the `datetaken` matrix are made in order to sort photos temporally. This part of the algorithm works as follows:

  1. **Initialization** – If the first photo is being analyzed, a position marker is attributed to that photo; the marker will serve as a memory position to easily re-arrange photos in the photo array.

  2. **Photo array temporal sorting** – For each remaining photo, proceed as follows:

     2.1 The position marker is updated with the new photo position.

     2.2 The photo date is compared with the previous photos’ dates in the `datetaken` matrix until a stopping condition is verified. The stopping criterion is the detection of an earlier date; then the marker is updated with the position of the photo where the condition was verified.

     2.3 All the photos starting on the marked photo are dragged one position to the right in the photo array in order to open a space to the newly temporally ordered photo.

     2.4 The date of the photo is also updated in the `datetaken` matrix in order to perform further comparisons with other photos.

     2.5 The photo is finally inserted in the new position of the photo array.

     2.6 Go to step 2 above and process the next photo.

• **Photos displaying** – Photos thumbnails are, finally, displayed on the screen panel in the temporal order with the respective date and time below for user viewing.

4.1.2. **Similarity Detection and Filtering**

The idea behind this similarity detection and filtering algorithm is quite simple: photos are first divided into small regions and dissimilarities between regions are defined as the Euclidean distances of the corresponding RGB values. When the dissimilarity between two photos is significantly low, they are classified as similar. In addition, since similar photos are often taken within a short period of time, this algorithm only performs
similarity detection for the photos that are temporally adjacent to each other. The algorithm uses one control parameter:

- **Similarity detection threshold**, $\delta$: The choice of $\delta$ is given to the user in order to allow him/her to set the level of similarity defining ‘similar photos’. The similarity detection threshold is available to the user, mainly because of the various objectives when generating a slideshow; for example, some users may want a generic slideshow in which similar photos will play an irrelevant role, and so they should not be included; others may have a different approach and want similar photos to be included. This parameter sets different levels/meanings of similarity.

Similarity detection is, in this way, a customizable process as the user can change $\delta$ in order to obtain the best possible results for his/her purposes. Having described the similarity detection control parameter, a detailed description of the algorithm is presented:

1. **Photos into regions division** – First, the photos are divided into 64 equal regions according to their respective dimensions, i.e. 64 (8x8) rectangles of equal dimension with a size depending on the photo resolution. The regions are created with the following dimensions:

   \[
   \begin{cases}
   R_{h,i} = P_h/8 \\
   R_{w,i} = P_w/8
   \end{cases}
   \]  

   (6)

   where $R_{h,i}$ and $R_{w,i}$ are the $ith$ region height and width, which correspond to the vertical and horizontal dimensions, respectively, of the $ith$ region. $P_h$ and $P_w$ are the height and width, which correspond to the vertical and horizontal dimensions, respectively, of the photo being analyzed.

2. **RGB region averages computation** – For each region (rectangle) above, three averages R, G and B are computed and stored in the a 8x8 matrix with the pixel RGB average values. These averages are computed according to the formula:

   \[
   S_{R,G,B_{R,i,j}} = \sum_{l=m}^LP_{R,G,B}/R_{hij} \times R_{w,i,j}
   \]  

   (7)

   where $S_{R,G,B_{R,i,j}}$ represents the R, G or B average pixel value of the region $i,j$ to fill the corresponding entry in the 8x8 pixel average values matrixes. $PV_{R,G,B}$ represents the value of each single pixel inside the region $i,j$ obtained from the color structure available on the software [22]. The $l$ and $m$ variables serve to sweep the rectangle to extract all the pixels values. Thus, each photo will have three, R, G and B, average pixel values 8x8 matrixes.

3. **RGB averages comparison** – According to the procedure previously defined (comparison only between temporally adjacent photos), the similarity is computed as the sum of Euclidean distances of the R, G and B average values for corresponding regions (same relative position in both photos). This is performed according to the following formula:
\[
\text{similarity} = \sum_{i,j} \left( S1_{R,G,B_{R,i,j}} S2_{R,G,B_{B,i,j}} \right)^2
\] (8)

where \( S1_{R,G,B_{R,i,j}} \) and \( S2_{R,G,B_{B,i,j}} \) are the average R, G or B pixel value of the region \( i,j \) for the two photos being compared.

4. **Similar photo decision** – To decide whether a photo is similar to another or not, the result of the previous formula is compared with a pre-defined threshold which determines the level of similarity to filter:

\[
\text{if} (\text{similarity} \leq \delta), \quad (9)
\]

then the two temporally adjacent photos are considered similar.

5. **Photos filtering** – All the photos considered similar are filtered out from the collection.

The algorithm above is rather simple but effective for the purposes of the preprocessing system developed since, in the worst case, very similar photos are always removed from the collection.

### 4.2 Photo organization

This section intends to motivate and detail the algorithms used in the photo organization module. To introduce this module, it is first given an idea of how it was possible to manipulate each photo concerning the organization performed. The data structure aggregated to each photo included several variables to store features related to each sub-module of the photo organization. These variables will be explained in their corresponding sections.

#### 4.2.1. **Time-based Clustering**

As referred before, at this stage, the photos are already sorted by their creation time. The main target of the time-based clustering algorithm is to dynamically determine noticeable time gaps by checking the temporal context of adjacent photos. As shown in Figure 24, the final result shows the photos organized by rows, each row including the photos temporally close to each other. On the other hand, the photos in the various rows correspond to different time clusters because a noticeable time gap exists. This method dynamically determines the time clusters and, thus, the big gaps between shooting periods, revealing the changes in shooting pace.
Figure 24 - Time-based clustering example.

The algorithm needs to use a single control parameter:

- **Temporal clustering threshold,** $\beta$: The choice of $\beta$ is given to the user in order to allow the dynamic definition of the temporal gap size defining temporal clusters. The temporal clustering threshold is available to the user, mainly because of the various objectives when performing this kind of clustering; for example, as users are aware of the different time gaps, they may choose the most appropriate parameter value regarding their collection of photos. The default value for this parameter was chosen as a reasonable value to temporally structure the photos in real situations, such as holidays photos collections.

Time-based clustering is, in this way, a customizable process as the user can change $\beta$ in order to obtain the best possible results for his/her purposes. Having described the temporal clustering threshold, a detailed description of the algorithm is presented:

1. **Time clusters creation loop** – A loop continuously compares adjacent photos (temporally sorted) to check if the time gap between the current photo in the loop, and the last photo in the most recent time cluster is bigger than the temporal clustering threshold. In the positive case, a new time cluster is created and the loop is restarted beginning on the next photo. Otherwise, no new cluster is created and the loop continues to the next temporally adjacent photo. This step makes use of the `datetaken` matrix created in the temporal sorting module to perform the time comparisons. This matrix includes temporally ordered date and time information about each photo.

2. **Time cluster vector creation** – A vector named `timebased_cluster` is created in which the different positions represent clusters, and the value for each position corresponds to the number of photos inside the associated time cluster. The message box in the top right corner of Figure 24 represents an example of a `timebased_cluster` vector.

3. **Temporal clusters display** – The entire photo collection is displayed in the application panel showing a temporal cluster per row; this means that photos
inside a temporal cluster are displayed horizontally, and different clusters are displayed vertically in adjacent rows.

4.2.2. **Content-based Clustering**

The content-based clustering algorithm is performed within the time-based clusters resulting from the previous sub-module with the purpose to further structure the photos in a certain temporal period, in this case using the content similarities. In practice, the use of content information leads to a finer organization of the photos collection. The representation of the content features is performed using some relevant and meaningful content descriptors, notably in terms of color, since color is a very important similarity feature. For the application here described, the MPEG-7 dominant color (DCD) and color layout (CLD) descriptors have been selected to characterize the photos, this means the spatial context. To better understand this algorithm, both descriptors and their extraction algorithms will be described first as an introduction to this sub-module.

4.2.1.1 **Dominant Color Descriptor**

The dominant color descriptor [23] represents the statistical characteristics of an image by providing a compact description of the representative colors in an image or image region.

- **DCD structure** – The representative colors are computed from each image; this allows the color representation to be accurate and compact. The DCD is defined to be:

\[
F = \{(c_i, p_i, v_i), s\}, \quad (i = 1, 2, \ldots, N)
\]

where \(N\) is the number of dominant colors. Each dominant color value \(c_i\) is a vector of corresponding color space component values, i.e. a 3-D vector in the RGB color space. The percentage \(p_i\) (normalized value between 0 and 1) is the fraction of pixels in the image or image region corresponding to color \(c_i\), with \(\sum p_i = 1\). The optional color variance \(v_i\) expresses the variation of the color values of the pixels in a cluster around the corresponding representative color. The spatial coherency \(s\) is a single value that represents the spatial overall homogeneity of the dominant colors in an image. The number of dominant colors \(N\) can vary from image to image, and a maximum of eight dominant colors is found to be sufficient to represent an image or image region.

- **DCD extraction** – The color space used to perform the clustering is the CIE 1976 (L*, u*, v*) color space (CIELUV). The CIELUV is a two-dimensional chromaticity diagram that gives a uniformly-spaced visual representation of a three-dimensional color space. The L*, u*, and v* are non-linear relations [24]. The extraction will be explained in the following:

1. **Distortion calculation** – The distortion \(D_i\) in the \(i^{th}\) cluster is given by

\[
D_i = \sum h(n) \|x(n) - c_i\|^2, \quad x(n) \in C_i
\]

where \(c_i\) is the centroid of the cluster \(C_i\), \(x(n)\) is the color vector at pixel \(n\) and \(h(n)\) is the perceptual weight for pixel \(n\). The perceptual weights are calculated from local pixel statistics to account for the fact that the
human vision perception is more sensitive to changes in smooth regions.

The update rule for the above distortion metric can be derived to be:

\[ c_i = \frac{\sum h(n)x(n)}{\sum h(n)}, \quad x(n) \in C_i \quad (12) \]

2. **Centroid calculation** – The procedure is initialized with one cluster consisting of all pixels and one representative color computed as the centroid (center of mass) of the cluster. The algorithm then follows a sequence of centroid calculations and clustering until a stopping criterion (minimum distortion) is met.

3. **Clusters division** – The clusters with higher distortion are divided by adding perturbation vectors to the centroids until the maximum distortion falls below a predefined threshold or the maximum number of clusters is generated.

4. **Quantization of the color values** – The percentage or fraction of pixels in the range belonging to each of the quantized colors is then calculated and these resulting percentages are uniformly quantized to five bits. The color values are quantized according to the specifications of the color space and the associated color-quantization descriptors.

5. **Spatial coherence measurement** – A simple connected component analysis is performed to identify groups of pixels with the same dominant color that are spatially connected; four connectivity (the four nearest neighbors of a pixel) is assumed. The average number of connecting pixels of each dominant color is then computed. A 3x3 masking windows is used for this purpose. This is used as a measure of spatial coherency for that dominant color. The overall spatial coherence is then a liner combination of the individual spatial coherence values with the corresponding percentages \( p_i \) being the weights. The spatial coherence value is then nonuniformly quantized to 5 bits, where 31 means highest confidence and 1 means no confidence.

6. **Color variances computation and quantization** – Finally, the color variances are computed as variances of the pixel values within each cluster and nonuniformly quantized to one bit per color component.

- **DCD matching** – Each relevant region in the image is represented using the DCD as defined above. Similarity retrieval involves searching the photo collection for similar color distributions. Since the number of representative colors is small, one can first search the relevant photos set for each of the representative colors separately, and then combine the results. Searching for individual colors can be done very efficiently in the used 3-D color space.

Consider two DCDs,

\[ F_1 = \{(c_{1i}, p_{1i}, v_{1i}), s_1\}, \text{and } F_2 = \{(c_{2i}, p_{2i}, v_{2i}), s_2\}. \quad (13) \]

Ignoring the optional variance parameter and the spatial coherence, the dissimilarity \( D(F_1, F_2) \) between two DC descriptions can be computed as:

\[ D^2 \left( \sum_{i=1}^{N_1} p_{1i}^2 + \sum_{j=1}^{N_2} p_{2i}^2 + \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} 2a_{1i,2j}p_{1i}p_{2j} \right) \quad (14) \]
where the two subscripts 1 and 2 in all variables stand for descriptions $F_1$ and $F_2$, respectively, and $a_{k,l}$ is the similarity coefficient between two colors $c_k$ and $c_l$,

\[
a_{k,l} = \begin{cases} 
1 - \frac{d_{k,l}}{d_{\text{max}}} & d_{k,l} \leq T_d \\
0 & d_{k,l} \geq T_d 
\end{cases} 
\]

(15)

where $d_{k,l} = \|c_k - c_l\|$ is the Euclidean distance between two colors $c_k$ and $c_l$, $T_d$ is the maximum distance for two colors to be considered similar and $d_{\text{max}} = \infty T_d$. In particular, this means that any two dominant colors from one single description are at least $T_d$ distance apart. Recommended values for $T_d$ are between 10 and 20 in the CIELUV color space and for $\alpha$ are between 1.0 and 1.5 [23].

4.2.1.2 Color Layout Descriptor

The color layout descriptor [23] specifies a spatial distribution of colors and roughly describes the color structure of the whole image.

- **CLD structure** – The color layout descriptor (CLD) is a very compact and resolution-invariant representation of color for high-speed image comparison and retrieval. It is designed to efficiently represent the spatial distribution of colors. This feature can be used for a wide variety of similarity-based comparison as needed in the context of this Thesis. Therefore, this descriptor is used as an image-to-image matcher. A jagged array (e.g. a non-symmetric matrix) serves as the basis for the local representative colors in the YCbCr color space.

- **CLD extraction** – This descriptor is obtained by applying the DCT transformation on the previous referred jagged array. Figure 25 illustrates the extraction process of the descriptor from an image. The extraction stages will be explained in the following:

1. **Image partitioning** – An input picture is divided into 64 blocks to guarantee the resolution or scale invariance. Each of these blocks is defined with the Y, Cb, and Cr values obtained from the R, G, and B pixels values partition and average calculation.

2. **Representative color detection** – A single representative color is selected from each block. Any method to select the representative color can be applied; in this case, it is used the average of the pixel colors in a block as the corresponding representative color since it is most simple and the description accuracy is sufficient in general.

3. **DCT transformation** – Each of the tree color components is transformed using a 8x8 DCT; so three sets of 64 DCT coefficients are obtained.

4. **Nonlinear quantization of the zigzag-scanned coefficients** – The DCT coefficients are zigzag-scanned and the first few coefficients are nonlinearly quantized (using 64 and 32 levels for DC and AC coefficients, respectively). The MPEG-7 standard allows scalable representation of the feature by controlling the number of enclosed coefficients; it is used a total of 12 coefficients, 6 for luminance and 3 for each chrominance, for the images. The total bit length of the specified description (12 coefficients) is just 64 bits
including one signaling bit which specifies the extension of the number of coefficients.

It should be noted that this descriptor is one of the compact descriptors in the MPEG-7 Visual tools and it is quite suitable for applications having limitations on storage and/or bandwidth. This is important since this application tries to minimize storage in order to improve overall efficiency.

Figure 25 - The CLD extraction process.

- **CLD matching** – This descriptor is applicable both to an image as a whole and any parts of an image with arbitrary shapes. On applying to an arbitrary-shaped region, the representative color selection should be performed using only valid pixel values, and a padding process is required before the DCT transform. Representative colors of grid blocks containing no valid pixels are substituted with the average color of all valid pixels in the image. For matching two CLDs, \( \{DY, DCr, DCb\} \) and \( \{DY', DCr', DCb'\} \), the following distance measure is used:

\[
D = \sqrt{\sum_i w_{yi}(DY_i - DY'_i)^2} + \sqrt{\sum_i w_{bi}(Cb_i - Cb'_i)^2} + \sqrt{\sum_i w_{ri}(Cr_i - Cr'_i)^2} \quad (16)
\]

Here, the subscript \( i \) represents the zigzag-scanning order of the coefficients. The distances should be weighted appropriately, in this implementation it was chosen to attribute equal values to each distance.

4.2.1.3 **Content-clustering Algorithm**

Having described both the descriptors used to perform the content-based clustering, the algorithm used in the application will be detailed next. The algorithm needs to use one parameter:

- **Content-clustering threshold**, \( \gamma \): The choice of \( \gamma \) is given to the user in order to allow a customizable process, in which he/she may adjust the results of the content clustering to his/her own preferences. If a larger (smaller) value is set, the slideshow application would give preference to the spatial templates with more (less) photos displaying in the same layout. This parameter is set to 0.25 since after performing a series of tests with a variety of photo collections this value was found to be an acceptable default value.

Also, the user is let to choose the type(s) of clustering(s) to perform, time-based, content-based or both. It is also possible to choose just one color descriptor or to combine both the descriptors above; in this case, the user is presented with a slider to define the weights to attribute to each of the descriptors. The default value of this slider
is 50% for each descriptor. The choice of different percentages by the user allows an even finer tuning of the final slideshow results. Having described the content-clustering threshold, a detailed description of the algorithm is presented:

- **Descriptors extraction** – First, and after checking it is the first time that the current set of photos are being analyzed, the descriptors from each photo are extracted following the processes previously explained in the descriptors sections. The results from this extraction are stored in the variables *color* and *YCbCr* (data structure), for DCD and CLD, respectively. If the application detects that these photos have already been analyzed, than it uses the descriptions which should be already available. This method is extremely useful since the user may do all the tests he/she wants without having to wait for another descriptors extraction round (which takes approximately 2-3 seconds per photo).

- **Descriptors selection and weighting** – The user choice of the type(s) of content-clustering to perform results in attributing a weight to both DCD and CLD. If only one descriptor is chosen, it gets a weight of 100%; otherwise, the default or the user selected weights are used. These values determine the contribution of each descriptor to the content clustering process.

- **Content clustering loop** – This loop is described in the following sub-steps:
  1. **Initialization** – If the first photo is being analyzed, the first content cluster is created; naturally, this photo becomes the representative photo of that cluster.
  2. **Distances computation for content clustering** – For each remaining photo in the photo set, it is necessary to decide whether it belongs to an already existing content cluster or if it is necessary to create a new one. This decision is based on a comparison between the photo under processing and the representative photos of all content clusters already created by computing the distances between the current photo in the loop and the representative photos of the existing clusters in terms of dominant color and color layout. These distances are both normalized to the range [0,1]. The spatial context distance between two photos is then defined as the average contribution (with some descriptors weighing) of the normalized dominant color and color layout distances. Let  $P_{i}(i = 1,2,\ldots,N)$ be the $i$th photo in a time-based cluster $\Psi$; then the distance metric is:

$$D_s = \max_{P_i\neq P_j} d(P_i, P_j) \quad (17)$$

If $D_s$ is smaller than the threshold value $\gamma$ (which is set by default as 0.25), then the photo is attributed to the content cluster under study since it is similar enough to the representative photo of that cluster. This step ends when all already existing content clusters have been tested and $D_s$ is not smaller than $\gamma$ for all the cases.

3. **New content cluster creation** – If the current photo does not fit in any of the existing clusters, it is considered very dissimilar and thus a new content cluster has to be created to provide a finer content organization. This photo becomes the representative of the newly created cluster, and will serve for
the next photos as comparison to determine the similarity with the newly created content cluster.

4. **Loop advance** – Go to step 2 and process the next photo.

A clustering loop example is illustrated in Figure 26. In this case, a new content cluster was created because the analyzed photo did not fit in any of the existing content clusters.

![Diagram of clustering loop example](image)

**Figure 26 - Clustering loop example.**

- **Photo collection temporally and content clustered display** – Finally, the entire photo collection is displayed in the application panel. As photos inside a temporal cluster are displayed horizontally by rows, content clusters will be distinguished within those rows. To achieve this, a marker with cluster information relative to the photo that represents that cluster is presented before each cluster as shown in Figure 22.
4.3 Music Analysis

This section intends to motivate and detail the algorithms used in the music analysis module.

4.3.1. Beat Detection

The automatic extraction of the rhythmic pulse from musical excerpts has been a topic of active research in recent years. The goal is to design an algorithm capable of extracting a symbolic representation which corresponds to the experience of “beat” in a human listener, this means to perform beat-tracking. For the purpose of this Thesis, the beat of a piece of music is the sequence of equally spaced phenomenal impulses which define a tempo for the music.

In this application, the beat tracking algorithm proposed in [4] has been selected for the extraction of music beats. This option relies on the fact that the referred method analyzes music signals in different frequency bands and estimates beats information after envelope extraction, which is at the moment the most accurate way. As mentioned before, the beat information will serve as the timer for photo presentation. Figure 27 shows an overall view of the adopted tempo-analysis algorithm as a signal flow network; the flow is divided in the 3 algorithmic steps that will be explained below.

![Figure 27 - Schematic view of the processing algorithm [4].](image)

The beat detection algorithm includes the following main steps:
1. **Frequency analysis and envelope extraction** - As the signal comes in, a filterbank is used to divide it into six bands. For each of these subbands, the amplitude envelope is calculated and the derivative taken.

2. **Resonators and tempo analysis** - Each of the envelope derivatives is passed on to another filterbank of tuned resonators; in each resonator filterbank, one of the resonators will phase-lock, the one for which the resonant frequency matches the rate of periodic modulation of the envelope derivative.

3. **Phase determination** - The outputs of the resonators are examined to see which ones are exhibiting phase-locked behavior, and this information is tabulated for each of the bandpass channels. These tabulations are summed across the frequency filterbank to arrive at the frequency (tempo) estimate for the signal, and reference back to the peak phase points in the phase-locked resonators determines the phase of the signal.

4.3.2. **Music Segmentation**

Music is segmented into sub-units according to a user predefined parameter, which is detailed next:

- **Music segment length, \( \delta \):** The choice of \( \delta \) is given to the user, mainly because of the different objectives when generating a slideshow; depending on user’s expectations, this parameter allows the control of the slideshow duration, providing different experiences and targeting a nowadays problem. Usually people get tired of slow processes; here the smaller the choice of music segment length is (in seconds), the faster the slideshow is.

In this application, music segmentation is a customizable process as the user can change \( \delta \) in order to obtain the best possible results for his/her purposes. Having described the music segmentation length parameter, a detailed description of the algorithm is presented:

- **Music length extraction** - Music properties are accessed through a pre-defined class named `MP3Header` [25]; this class exploits music information that is not easily accessible. The music length property is then parsed and converted to seconds.

- **Maximum number of clusters** – The music length is divided by the music segment length selected by the user defining the maximum number of clusters that the slideshow may show in the final presentation.

- **Segmentation** – Music is segmented into sub-units according to the value obtained in the previous step; these sub-units will be used in the temporal composition for coordination with beats.

This music segmentation step is important since it sets the main cadence for the slideshow layout changes based on the music segment length parameter. Also, the user will have a first perspective of the amount of clusters that the slideshow will be able to show, and may check if it find it sufficient depending on the number of photos in the selected collections and his own expectations. This information may be useful.
when perform the clustering steps of the photo organization module, since the user options can control the number of clusters generated.

4.4 Composition

This section intends to motivate and detail the algorithms used in the composition module.

4.4.1. Clusters Importance Determination

This cluster importance is measured according to two features: photos per minute (PPM) and photo conformance (PC). PPM denotes the shooting frequency of photos in a cluster, while PC denotes the content-based similarities between photos in the same cluster. PPM is important since users tend to take more photos in more relevant places and to more relevant events, and thus the shooting frequency may depict a pattern of photo similarity and reveal an important cluster within the entire collection concerning the user’s expectations. PC also addresses this concept, although it relies on photo similarity analysis itself instead of shooting frequency. The algorithm comprises the following steps:

- **PPM computation** – Given the existing content-based clusters $C_g$, $g = 1,2,...,m$, PPM is calculated by:

  \[
  PPM(C_g) = \frac{N(C_g)}{\text{Time\_Duration}(C_g)}
  \]

  where $N(C_g)$ denotes the number of photos in $C_g$, and $\text{Time\_Duration}(\cdot)$ returns the time difference between the first photo and the last photo in the content cluster, which is already temporally sorted.

- **PC computation** – PC is computed as:

  \[
  PC(C_g) = 1 - \frac{1}{n_g(n_g-1)} \sum_{P_i \in C_g} \sum_{P_j \in C_g} d(P_i, P_j)
  \]

  where $P_i, P_j$ are photos categorized in the cluster $C_g$, and $n_g$ represents the number of photos in the cluster. This may be considered the within-cluster distance which denotes the average distance between the photos in the same content-based cluster.

- **Features concatenation** – These two features are then concatenated as a feature vector:

  \[
  \vec{x} = (PPM(C_g), PC(C_g))
  \]

- **Temporal and content clusters importance (CI) computation** - The cluster importance $CI_g$, for the $g$th cluster is calculated with the formula:

  \[
  CI_g = E(\vec{x}_g) + \frac{1}{2(m-1)+ma} \sum_{g=1}^m |\vec{x}_g - E(\vec{x}_g)|
  \]

  where $E(\vec{x}_g)$ is the mean of the feature vectors for the $g$th cluster, and the parameter $\alpha > 0$ is a predefined constant. This is a nonlinear fusion scheme, which may be referred as monotone increasing function. The greater a feature is, the more greatly it affects the returned value [19]. Thus, the concatenated
feature vectors are proportional to the cluster importance in the formula above. CI is directly proportional to the difference between the feature vector of the cluster being analyzed and the mean of the feature vectors of all the clusters. The calculated value, CI, is computed for both the time-based clusters and content-based clusters. Content-cluster importance is computed as the temporal cluster importance as if there was a single content cluster in that temporal cluster, i.e. only one feature vector is taken into consideration in (22).

4.4.2. Cluster Selection

The cluster selection algorithm works as follows:

- **Number of display clusters computation** – The user-selected music clip and its smaller segments divided according to the information described in Section 4.3.2 are used in this step. A content-based cluster of photos is, therefore, displayed in each music segment. Nevertheless, it is often the case that thousands of photos (and, therefore, hundreds of photo clusters) cannot be completely displayed because a time-limited music clip (e.g. 4 minutes) is divided into just some tens of segments. This analysis sets the maximum number of clusters that stops cluster sorting, i.e. there is no point in sorting clusters that will not be shown.

- **Cluster sorting** - To be sure that the most important photos are shown in the slideshow, the time-based photo clusters are sorted based on their importance value in descending order; moreover, inside those time-clusters, content-based clusters are also sorted based on their importance. This helps in choosing the most relevant clusters for the slideshow and assures the most relevant ones are used in the slideshow.

- **Cluster choice** – Then $N_s$ clusters are picked for presentation if only $N_s$ music segments are available. The clusters are presented by decreasing order of content importance within the temporal clusters ordered by decreasing order or importance with the understanding that no clusters from the same temporal cluster are presented in sequence unless there is only a last temporal cluster from where to pick content clusters. The algorithm follows a loop inside the structure that stores the clusters; an example of this loop is illustrated in Figure 28.
Imagining a set of clusters ordered in time and content by their importance, the presentation sequence is from the most important to the least important temporal cluster picking one content cluster at a time by decreasing order of importance. This ensures the diversity of the slideshow presentation and that the most important content is presented before the least important content.

4.4.3. Template Determination

Once the clusters to be displayed as well as their sequence are determined, the problem is to select the most appropriate layouts for their presentation. Since there is more than one spatial layout template for a certain number of photos per cluster, first, the user has to determine if he/she wants the templates within each number of photos to remain always the same during the slideshow or to rotate among the available templates. Figure 29 shows the available templates for each number of photos per cluster. Rotating the templates typically leads to a more dynamic and varied presentation, improving the user experience. The proposed algorithm works as follows:

- **Template allocation** – Here a spatial template is allocated to a content-based cluster. The algorithm proceeds as follows:
  1. **Template dimension determination** – For each cluster, it is determined the corresponding number of photos. Templates with 1 to 5 photos have been defined to be allocated to each cluster depending on the corresponding number of photos. For those clusters with a large number of photos, only the first 5 photos temporally ordered are selected to fill a 5-photos template.
  2. **Flow direction definition** – This steps targets the definition of the flow direction used for the photos in the template to appear on the screen. This depends on the type of template and was decided considering design attractiveness criteria. The solution presented in the following was based on the author’s subjective evaluation of its effectiveness. If bigger cells are
aligned in the horizontal, the flow direction will be from left to right or the inverse; on the other hand, if they are aligned in the vertical, the flow direction will be from top to bottom or the inverse.

3. **Template rotation determination** – A template rotation flag is used to control whether the next cluster with the same number of photos needs to be allocated to a different template or the same one is used. The sequence of rotation is uniform, ping ponging between the two templates available for each cluster dimension. This flag is only active when the user chooses to rotate among available templates.

4. **Loop advance** – Go to step 1 and process the next cluster.

- **Photo allocation and adaptation** – The steps for the photo filling algorithm are described in the following:
  1. **Template cells photo allocation** – For each cluster, the template cells are filled with photos. The order followed corresponds to the temporal order of the photos within the cluster. Since the photos attribution criterion to fill the spaces follows that order, the representative photo that defines the content-based cluster is always attributed to one of the bigger cells. This is natural as that photo is the first in the processing line according to the temporal order, and in the implementation, the cells are filled from the bigger(s) to the smaller(s). This intends to give relevance to the photo that served as comparison reference to the others in the content-based clustering process.
  2. **Photo resizing** – Then, photos are resized in order to perfectly fill the space of the various template cells.
  3. **Loop advance** – Go to step 1 and process the next cluster.

![Slide show Templates](image-url)
4.4.4. Temporal Composition

The integration of the visual and aural media is obtained by aligning each layout with a corresponding musical sub-unit. The timing for layout/template switching is according to the music beats. Also, the timing for the different photos’ display in a specific layout is controlled with the music beats. In this implementation, the display time for each photo is rather uniformly distributed within a segment with some variations related to a metric expressing the density of the beats in the relevant period. With this elaborate design, the proposed scheme synchronizes the visual part of the musical slideshow with the pace of the music. The algorithm works as follows:

1. **Photo display units** – Depending on the number of photos in a content cluster, temporal units are defined within the previously defined music sub-units. These temporal units are averagely distributed, according to:

   \[ Ntu_n = Ng_n \quad (23) \]

   \[ Tl_n = \frac{Music\_segment\_length_n}{Ng_n} \quad (24) \]

   where \( n \) represents the active cluster, \( Ntu_n \) and \( Ng_n \) represent the number of temporal units and the number of photos, respectively, in the active cluster (for content clusters with more than 5 photos, \( Ng_n \) is taken as 5). \( Tl_n \) represents the length of each temporal unit within a music sub-unit. \( Music\_segment\_length_n \) represents the music segment length chosen by the user as described in Section 4.3.2. After \( Tl_n \) is computed, the boundaries of the temporal units will be used in the following steps.

2. **Beat density computation** – For each temporal unit, a beat density value is computed according to the formula:

   \[ S_i = \frac{Ntu_n}{Ntu\_total_n} \quad (25) \]

   where \( S_i \) represents the beat density in the temporal unit \( i \), \( Ntu_n \) the number of beats within the temporal unit \( i \) and \( Ntu\_total_n \) represents the total number of beats in the cluster \( n \).

3. **Photo beat allocation** – To fulfill the user expectations, the layout switching in the slideshow should be done as far as possible according to the related user predefined parameter, the music segment length. This means, each content cluster will remain on the screen during the period defined by that parameter. The photo beat allocation algorithm comprises two main steps:

   For each content cluster, perform:

   1. **Initialization** – Within the music sub-unit of the active cluster, each temporal units is associated with each photo of the cluster. Since the first photo being showed is the representative photo of each cluster, the beat attribution tries to match the specific relevance of each photo. Thus, to the first photo being analyzed, it is attributed for displaying the beat nearest to its respective temporal unit boundary.
2. **Photo beat allocation** – For each remaining photo in the content cluster, perform:

2.1 – Check if the photo is the second of the cluster to be shown. In this case, attribute to the photo the first beat after its respective temporal unit boundary. This tries to match the importance of the various photos referred in step 1.

2.2 – If not, attribute to the photo the closest beat to its temporal boundary but on the side, left or right, with the higher beat density. This tries to offer a more vivid experience to the user by increasing the probability of the user to see a photo switch in the more intense part of the music.

2.3 – Go to step 2 and process the next photo.

3. **Loop advance** - Go to step 1 and process the next content cluster.

Figure 30 shows an example of a temporal composition process after the templates are determined; this figure intends to provide a better understanding of the photo beat allocation step as described above.

![Figure 30 - Example of a temporal composition sequence.](image)

In the example in Figure 30, photos 1, 5, and 8 are the representative of their clusters and, therefore, have a beat attributed according to step 1 of the photo beat allocation algorithm. Photos 2 and 6 follow the rule of having attributed the first beat after its temporal unit boundary to give a privilege to the first photo (photo beat allocation step 2 (3.2.1)). Finally, photos 3, 4, and 7 have attributed the closest beat to its temporal unit boundary on the side with the highest beat density (photo beat allocation step 2 (3.2.2)). With this elaborate design, the proposed algorithm synchronizes the visual output with music’s pace while improving the user’s experience.
4.4.5. **Player**

The player sub-module generates the final output of musical slideshow application to the user, after gathering all the processing results obtained made by the previous sub-modules which provide the relevant informant. The player algorithm works as follows:

1. **Music loading** – Music is loaded with the help of a Microsoft Windows Media Player control [26].

2. **Timer start** – When everything is set (photos and music loaded), the timer is enabled. The timer controls the time that each photo enters and remains on the screen.

3. **Slideshow starts** - Making use of the potentialities of Microsoft Visual Studio 2008 (.net framework 3.5), the slideshows starts. Photos displayed are supported by a flow layout panel, which is a panel where controls may be added dynamically and consecutively in a certain flow direction (in this case, picture boxes with photos). Music is also played simultaneously [26].

Figure 31 shows examples of three consecutive slideshow layouts displayed by the player.

![Figure 31 - Example of player output layouts (3).](image)

This chapter intended to offer the reader an in-depth description of the entire process for slideshow generation, presenting all modules and algorithms developed to achieve the system’s purposes. The chapter should be able to provide a full understanding of the input photo collection manipulation evolution until the slideshow effectively starts.

Chapter 5 will describe the application implementation in detail, justifying the choice of the programming language, presenting the application’s high-level operation diagram and the frameworks and libraries used; finally, the application’s Graphic User Interface (GUI) will be described.
This chapter intends to provide the reader a description of the developed musical slideshow application. First, the choice of the programming language as well as other options, e.g. regarding frameworks and libraries used and the application’s structure, will be motivated. Next, a description of how to install the application will be given. Finally, a complete description of the application’s GUI will be provided.

### 5.1 Implementation Overview

Before describing the application, this section aims to motivate the choice of C# as the programming language and to briefly describe the frameworks and libraries used in the development of the summarization application.

#### 5.1.1 Programming Language Selection

As the developed application was intended to be a Microsoft Windows application, the use of Microsoft Visual Studio 2008 [27][28] for Internal Development Environment (IDE) was rather consensual. Microsoft Visual Studio is a software development product for computer programmers providing a development environment allowing programmers to create applications, web sites or web applications to run on platforms supported by Microsoft’s .NET framework [29][30]. Microsoft’s .NET framework provides a large number of pre-coded solutions to common program requirements, managing the execution of applications written specifically for this framework.

After choosing the IDE, the following decision was related to the programming language. Using Microsoft’s Visual Studio 2008, three languages arose as the main
candidates: C, C++ and C#. Although many multimedia applications have been developed in the referred languages in recent years, the choice was C#. Notwithstanding the author’s absolute lack of experience with this language, the main motivation for this choice was the previous successful experiences of other M.Sc. students with this programming language. As Visual Studio allows the integration of different types of programming languages, the problem regarding the reduced number of available C# multimedia libraries was minimized, as C++ multimedia libraries could and were integrated in the development of this solution.

5.1.2. Frameworks and Libraries

Almost all frameworks and libraries used in the development of this system were introduced in various sections of Chapter 4. Even so, the frameworks and libraries included in the application, without any adaptation, will be briefly described next:

1. **Windows Media Player** – The Windows Media Player (WMP) ActiveX control can be used in C# applications to add audio and video playback capabilities. The Microsoft Windows Media Player [26] Software Development Kit (SDK) provides information and tools to customize the Windows Media Player and to use the Windows Media Player ActiveX control. The SDK includes documentation and code examples that show how to use the Media Player ActiveX control from a C# application.

2. **MP3Header** – The C# MP3Header Class [25] written by Gustav "Grim Reaper" Munkby was used to extract mp3 music info, such as the music name and music length.

3. **Beat detection algorithm** – The algorithm presented in [21] and its resulting implementation was used in this application. The original code was written in C++, and a plug-in was made in order to use the code in this application.

5.1.3. Application Structure

This section aims to provide the reader with a rather very high level structure of the implementation adopted for the musical slideshow application. To do so, Figure 32 shows a high-level operation diagram of the application.
The diagram in Figure 32 represents the application operational structure divided in three hierarchies. Blue boxes represent the operations; inside each box one or more operation may be performed. As it is possible to conclude by inspecting the figure, every box of operations has a button associated. A brief walkthrough of the application operations will be provided next:

- **Input** – The first grade corresponds to the functions that internally prepare the photos and the music to be operated by the next grades.

- **Processing** – This grade directly implements the modules described in Chapter 3; each module has its functions grouped, and running starts by clicking a button named “RUN”.

- **Output** – The last grade is the bridge between the application results and the user viewing. Also, some browsing operations are available.

This intends to present the reader with a conceptual idea of how this kind of application was developed. By aggregating the different functions to an execution button, it is possible to differentially analyze the various results obtained.
5.2 Installation

The application’s installation process is straightforward and quite simple. The system has a single requirement, this means the availability of Microsoft’s Visual Studio 2008 which may be downloaded from the following link, http://msdn.microsoft.com/en-us/evalcenter/bb655861.aspx [29] and has a straightforward installation.

After installing Microsoft’s Visual Studio 2008, the user has to copy the folder SlideshowData with the application to (C:) and run the Slideshow application from the root program folder.

5.3 Graphical User Interface

This section aims to provide a detailed description of the application’s Graphical User Interface (GUI). The GUI has a major impact in the final usability of the application, notably considering this application is targeting users without any expertise requirements.

The application is composed by a single Windows Form, which is divided into 5 main areas, as shown in Figure 33.

![Figure 33 - Application’s GUI.](image)

Figure 33 represents the state of the application after selecting the folder with the photos for the slideshow; the 5 areas will be described next. In the buttons and tab colors, red is related to the content selection buttons, white to the central panel, blue to the main tab control, yellow to the slideshow and browsing buttons, and green to the application information. The 5 areas highlighted in Figure 33 have the following main functions:

1. **Selection buttons** – To select the audio and visual content (photos and music).
2. **Central panel** – Area to present the outputs of the application, notably to display the final slideshow.

3. **Main tab control** – Area for user control and parameters input to the various processing modules.

4. **Slideshow and browsing control buttons** – To start/stop the photo browsing and the slideshows. For comparison purposes, there are two types of musical slideshows available: 1) a basic slideshow which displays the photos one by one with music in the background but without any mutual synchronization; 2) the developed musical slideshow which displays a selection of the photos, synchronized with the selected music.

5. **Application information** – Areas for output information related to the application current status of the slideshow generation process.

The top menu strip in Figure 33 provides the basic functionalities of the application. They are briefly described next, starting from the top menu strip:

4. **File > Help** – Shows the user the help menu.

5. **File > Credits** – Shows the user a window containing information about the application and the authors.

6. **File > Exit** – Allows the user to exit the application.

### 5.3.1. Selection Buttons

Figure 34 represents the content selection buttons.

![Figure 34 - Selection buttons.](image)

There are two content selection buttons:

1. **Select photos** – Allows the user to open a folder browser dialog to select a folder with photos to include in the slideshow.

2. **Select music** – Allows the user to open a file browser dialog to select the music to play during the slideshow.

### 5.3.2. Central Panel

This is the area where the photos (thumbnails in most cases) are displayed and the slideshow is played. There is no user interaction in this area.
5.3.3. **Main Tab Control**

The main tab control is the 'brain control' area of this application. Each tab allows the user to control one of the core processing modules, notably in terms of its parameters; they are coherent with the architectural modules presented in Chapter 3. The main tab control is formed by three tabs that will be detailed next:

- **Analysis tab** – The analysis tab provides the user interaction for the various modules in the Analysis stage described in Sections 4.1 and 4.3 of this Thesis. Figure 35 shows the analysis tab interface: the photo analysis part of this tab is only enabled when a folder is already selected; similarly, the music analysis of this tab is only enabled when a music is selected.

![Figure 35 - Analysis main tab.](image)
The Analysis tab includes the following areas:

1. **Photo analysis** – Allows the user to select if the similarity detection should be performed by checking the respective box.

2. **Music analysis**
   a. **Sub-modules to process** – Allows the user to select which actions should be performed by checking the respective boxes.
   b. **Music segment length** – Allows the user to assign different values to the parameter (in seconds).

3. **Analysis tab execution** – This ‘RUN’ button allows the user to execute all the actions selected in this (analysis) tab.

- **Organization tab** – The organization tab provides the user interaction for the various modules in the Organization stage described in Section 4.2 of this Thesis. Figure 36 shows the organization tab. Photo organization is only enabled when a folder with photos is already selected.

![Figure 36 - Organization main tab.](image)
The Organization tab includes the following areas:

1. **Time-based clustering** – Allows the user to determine whether to perform time-based clustering or not by checking the associated box. Also, the temporal gap box allows the user to control this module's parameter by assigning the desired value (maximum: 23 (h-hours) and 59 (m-minutes)).

2. **Content-based clustering** – Allows the user to determine whether to perform content-based clustering or not by checking the associated box.
   a. **Threshold value** – This box allows the user to assign the desired value to the control threshold (from 0.1 to 1); a default value of 0.25 is available.
   b. **Descriptors selection buttons** – These buttons allow the user to choose what descriptors to use for the content-based.
   c. **Descriptors Weighting Slide bar** – If both descriptors are selected, this bar allows the user to control the weights of each descriptor for the clustering.

3. **Organization tab execution** – This button allows the user to execute all the actions selected in this (organization) tab.

- **Composition tab** – The composition tab provides the user interaction for various modules in the Composition stage described in Section 4.4 of this Thesis. Figure 37 shows the composition tab. The photo composition functionalities may only be executed after performing the essential functionalities of the previous tabs.

The Composition tab includes the following areas:

1. **Cluster importance determination and cluster selection** – These sub-modules are grouped together because their checkboxes are mutually locked, i.e. checking one means checking both. This happens because cluster selection needs information from the cluster importance determination to be performed. Also, there is a slide bar available where the user may determine the weighting for the contribution of the PC and PPM importance metrics for the final cluster importance determination.

2. **Template determination and allocation** – Allows the user to determine whether to perform template determination and allocation or not by checking the box. This possibility is given to the user since it is possible to perform several cluster importance determination and cluster selection combinations; therefore, the template determination and allocation do not have to be performed all those times; this increases the application performance. The same approach applies to the next sub-module, temporal composition.
   a. **Template radio buttons** – This radio buttons allow the user to determine whether to rotate among the available templates for each number of photos in the panel or not.
   b. **Show available templates** – This button allow the user to see which templates are available for each number of photos in the panel; this is just informative, thus without any direct impact in the application's result.
3. **Temporal composition** – Allows the user to determine whether to perform temporal composition or not by checking the box.

4. **Composition tab execution** – Allows the user to execute all the actions chosen in this tab. Also, this button provides an informative message box detailing the actions that still need to be performed before the start slideshow button may be unlocked.

![Figure 37 - Composition main tab.](image)

5.3.4. **Slideshow and Browsing Control Buttons**

Figure 38 represents the visual aspect of the buttons addressed in this section.
2. Start basic slideshow button – This button starts/stops the basic slideshow. The button is only enabled when a folder of photos and a music is loaded.

3. Show all photos button – This button shows on the screen thumbnails of all photos in the current selected collection.

4. Next button – This button shows a single photo on the screen; by clicking on this button, successive photos are showed in temporal order.

5. Previous button – This button shows a single photo on the screen; by clicking on this button successive photos are showed in reverse temporal order.

5.3.5. Application Information

This application information area intends to provide a rapid way for the user to check vital information about the current status of the slideshow generation process, and the results for the actions that have already been performed. There are six informational outputs in this area:

- **Number of photos** – After the user selects a folder of photos, this label is updated with number of photos; also, after performing the filtering preprocesses, this number is updated.
- **Music length** – After the user selects a music, this label is updated with the music length in the format mm:ss.
- **Number of music beats** – After the user performs beat detection, this label is updated with the number of beats in the selected music.
- **Maximum number of clusters to show** – After performing music segmentation, this label is updated with the maximum number of clusters that is possible to show in the slideshow.
- **Number of temporal clusters** – After performing time-based clustering, this label is updated with the number of temporal clusters created.
- **Number of content clusters** – After performing content-based clustering, this label is updated with the number of content clusters created.
- **Slideshow slide bar** – Displays the progress of the slideshow processing using a visual scale.
Chapter 5 has introduced the application, presenting an overview about its implementation with a brief description of the frameworks and libraries used as well as its high-level architecture. In order to allow a proper use of the application, it is explained how to perform the installation; finally, a detailed description of the GUI aiming to explain all the options given to the user was provided.

Chapter 6 will present the performance results obtained when evaluating the multimedia system developed in this Thesis.
Since the development of any multimedia application is not finished without a serious and meaningful evaluation of its performance regarding the defined objectives, this chapter aims to present and analyze the results obtained by a subjective evaluation study which was carried out to evaluate the developed application’s performance.

Considering the type of system developed, it is believed that the adequate performance evaluation methodology should follow a subjective and not objective approach. Furthermore, it would be hard or even impossible to perform a meaningful objective evaluation of the proposed system since it is difficult to define efficient metrics which may adequately access the user’s quality of experience. Thus, it was decided to design an adequate subjective evaluation methodology and conduct a user evaluation study to assess how the developed application performs in view of the initially defined objectives, this means to improve the user experience while browsing a photo collection. This chapter will present the test objectives, its methodology, the subjective scores and, finally, the analysis of the obtained results.

6.1 Test Objectives

The tests designed intended to evaluate the overall performance of the developed application and, implicitly, of the selected signal processing algorithms. With this purpose in mind, a subjective evaluation study was designed with four main objectives:

- **Quality of experience** – The first objective is to evaluate how good and improved is the user experience provided by the generated musical slideshows. To achieve this, the musical slideshow generated by the developed application is compared with an alternative but rather simpler slideshow similar to the one generated by the commercial software ACDSee [15]; this simpler application will
be named *Basic Slideshow* in the following. In [15], a more ‘conventional’
slideshow is generated by simply sequentially and automatically switching photos
one-by-one; moreover, for better and fairer benchmarking but differently from
ACDSee which has no ability to accompany the slideshow with music, the *Basic
Slideshow* used here as benchmarking will play a music in parallel but without
any synchronization with the photos. The basic slideshow is also implemented in
the developed application to be used as reference in the performance evaluation
process. In this context, the *Basic Slideshow* will provide the benchmarking to
evaluate how much the photos and music processing included in the developed
slideshow has improved the user quality of experience, this means how much is
the added value of the adopted processing technology. This first performance
evaluation objective regards the overall performance evaluation of the slideshow.

- **Photos Processing** – The second objective is to evaluate the added value of
displaying the photos in a more sophisticated way, notably exploiting their
similarities, and not simply one after the other. This objective intends to evaluate
the added value of the photos processing tools and the user experience
improvements in terms of photos displaying, e.g. regarding the visual coherence
of each layout.

- **Music Adding and Processing** – The third objective is to evaluate the added
value of adding synchronized music to the photos display. This objective intends
to evaluate how much the user experience improves with the inclusion of
synchronized music, more precisely exploiting the beat information to control the
progress of the slideshow.

- **Acceptance** – The fourth and last objective is to evaluate the overall level of
acceptance of the developed application, this means its impact among possible
users. This objective intends to know if the application has the potential to have
commercial success.

The next section presents the test methodology designed in this Thesis to achieve
these four objectives.

### 6.2 Test Methodology

In order for the test performed to be credible, it has to be well defined enough to be
reproducible by other experts with results and conclusions which are statistically similar.

The test methodology is now presented, starting with the test questions proposed to
the test subjects.

1. **Test Questions** – The test questions defined for this test to address the
objectives above defined are:

- Question 1 – How did you enjoy the *Musical Slideshow* compared with the *Basic
  Slideshow*?
  
  a) Much worse; b) Worse; c) Same; d) Better; e) Much better

- Question 2 – How did you enjoy the spatial and temporal play sequence of the
  photos in the *Musical Slideshow* compared with the *Basic Slideshow*?
  
  a) Much worse; b) Worse; c) Same; d) Better; e) Much better
• Question 3 – How did you enjoy the music synchronization for the photos displaying in the Musical Slideshow compared with the Basic Slideshow?
  a) Much worse; b) Worse; c) Same; d) Better; e) Much better
• Question 4 – If you had this tool, would you use it to generate your own slideshows?
  a) Never; b) Seldom; c) Sometimes; d) Many times; e) Always

In terms of performance evaluation, the scores above are converted to values from 1 to 5 when going from score a) to e). The higher will be the score, the larger is the quality of experience improvement with the Musical Slideshow.

2. Test Material – The test set was constituted by 2 photo sets and 2 associated musics; each pair of photos set and music forms what is named a musical photo set. Based on these 2 musical photo sets, 2 slideshows were produced, using the developed Musical Slideshow application.

The 2 musical photo sets correspond to:
• Wedding – This set of photos contains scenes from a wedding, where the primary targets are people; the accompanying music is an 80’s music, with a constant well defined beat pattern.
• Travelling in Azores – This set of photos contains scenes with landscapes and people in leisure activities in the Azores; the accompanying music is a rock song, with clearly defined strong beats in some parts.

More detailed information about the musical photos sets is presented in Table 6.

Table 6 – Test material.

<table>
<thead>
<tr>
<th>Collection of photos features</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Set</strong></td>
<td><strong>Set name</strong></td>
</tr>
<tr>
<td>1</td>
<td>Wedding</td>
</tr>
<tr>
<td>2</td>
<td>Travelling in Azores</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Music features</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Set</strong></td>
<td><strong>Set name</strong></td>
</tr>
<tr>
<td>1</td>
<td>Wedding</td>
</tr>
<tr>
<td>2</td>
<td>Travelling in Azores</td>
</tr>
</tbody>
</table>

3. Application control – The slideshows for each of the content sets were produced using the parameters in Table 7 for the processing algorithms.
Table 7 – Algorithms’ parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Musical Photo Set 1</th>
<th>Musical Photo Set 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Photo Analysis</strong></td>
<td>Similarity detection activated</td>
<td>Similarity detection activated</td>
</tr>
<tr>
<td><strong>Music Analysis</strong></td>
<td>Music segment length: 2 sec</td>
<td>Music segment length: 2 sec</td>
</tr>
<tr>
<td><strong>Photo Organization</strong></td>
<td>Time-based clustering step: 00 hours 24 Minutes</td>
<td>Time-based clustering step: 10 hours 15 Minutes</td>
</tr>
<tr>
<td></td>
<td>Content-based clustering descriptors: DCD – 50% CLD – 50%</td>
<td>Content-based clustering descriptors: DCD – 50% CLD – 50%</td>
</tr>
<tr>
<td></td>
<td>Threshold value – 0.25</td>
<td>Threshold value – 0.25</td>
</tr>
<tr>
<td><strong>Photo Composition</strong></td>
<td>Cluster importance: PPM – 50% PC – 50%</td>
<td>Cluster importance: PPM – 50% PC – 50%</td>
</tr>
</tbody>
</table>

The parameters chosen differed for the musical photo sets on the time-based clustering step. This decision relies on the specific content of each set: while set 1 regards an event where the cadence (shooting frequency) of photos is high, and thus a small step was chosen, set 2 regards a trip, and thus a large step was chosen trying to group together photos taken in the same days.

4. Sequence of Testing – A group of volunteers were asked to operate the application and to give their subjective assessment to the questions above, for each of the slideshows created, following the sequence of steps defined next:

a. Click on the “Select Folder” button and open the “Evaluation Data” folder located in the desktop. Then open and visualize the photos, starting from “Wedding”. Also, for this set of photos, click on the “Select Music” button and open the same “Evaluation Data” folder to select the music “Fire Inc. - Nowhere Fast”.

b. Press the “Start Basic Slideshow” button and visualize. When the basic slideshow ends, press the “Start Musical Slideshow” button and visualize again. Note: The generation of the second slideshow may take some time due to the necessary processing.

c. Answer to questions 1 to 4 above, marking with a cross (X), in the evaluation table, the desired classification score. If it is not easy to decide on a score, it is possible to go back to step b) to visualize again any of the slideshows. The scoring may be done during the visualization of the slideshows.

d. Go back to a), and load the “Traveling in Azores” photos set and the associated “Blur - Song 2”. Proceed after to the next bullets.

The next section presents the obtained results and their analysis.

6.3 Results and Analysis

This section will present the obtained performance scores in this user evaluation study and their in-depth analysis. A precise analysis of the results obtained for each
question will be made, after the presentation of the results. The following tables, namely, Table 8, Table 9, Table 10, and Table 11 contain the average results for both musical photo sets, obtained for each question of this user evaluation study. The scores were provided by the 14 volunteer subjects involved, ten males and four females. These volunteers were chosen trying to have a wide range of people in terms of technical knowledge, age, and occupation.

- **Results and Analysis for Question 1**

  *Table 8 – Evaluation results for Question 1.*

<table>
<thead>
<tr>
<th>Question 1 – How did you enjoy the Musical Slideshow compared with the Basic Slideshow?</th>
</tr>
</thead>
<tbody>
<tr>
<td>a) Much Worse</td>
</tr>
<tr>
<td>Musical Photo set 1, Wedding</td>
</tr>
<tr>
<td>Musical Photo set 2, Travelling in Azores</td>
</tr>
</tbody>
</table>

Table 8 presents the results for question 1. Regarding question 1, the results show that 35,70% and 64,26% for set 1, and 21,42% and 78,54% for set 2, of the inquired subjects considered that the musical slideshow was ‘Better’ or ‘Much Better’ than the basic slideshow. None of the inquired subjects scored as the same or even worse the musical slideshow. Analyzing the results, it is immediately recognized that the musical slideshow has effectively captured people’s satisfaction. The slightly better results obtained for set 2 may be due to the fact that it has more diversity of layouts and thus the result impresses the user even more. Intuitively, people prefer a slideshow where photos are not shown one by one, with some synchronized music; the musical slideshow results in an innovate experience provided to the user with 100% of the subjects expressing their preference for the Musical Slideshow.

- **Results and Analysis for Question 2**

  *Table 9 – Evaluation results for Question 2.*

<table>
<thead>
<tr>
<th>Question 2 – How did you enjoy the spatial and temporal play sequence of the photos in the Musical Slideshow compared with the Basic Slideshow?</th>
</tr>
</thead>
<tbody>
<tr>
<td>a) Much Worse</td>
</tr>
<tr>
<td>Musical Photo set 1, Wedding</td>
</tr>
<tr>
<td>Musical Photo set 2, Travelling in Azores</td>
</tr>
</tbody>
</table>

Table 9 presents the results for question 2. Regarding question 2, the results show that 42,84% and 57,12% for set 1, and 21,42% and 78,54% of the inquired subjects considered that the Musical Slideshow was ‘Better’ or ‘Much Better’ than the Basic
Slideshow concerning the spatial and temporal play sequence of the photos. None of the inquired subjects scored this aspect of the musical slideshow with ‘Same’ or even ‘Worse’. These results were expected and are somehow consistent with the results for the previous question. People enjoyed the disposal of the photos on the screen, and the gathering of similar photos in the same content clusters. Then, grouping photos together is highly appellative to the user. The slightly better results for the content set 2 may be related to the fact that this set is more diverse, thus intensifying the added value of cleverly clustering the photos.

- **Results and Analysis for Question 3**

  *Table 10 – Evaluation results for Question 3.*

| Question 3 – How did you enjoy the music synchronization for the photos displaying in the **Musical Slideshow** compared with the **Basic Slideshow**? |
|---|---|---|---|---|---|
| | a) Much Worse | b) Worse | c) Same | d) Better | e) Much better |
| Musical Photo set 1, Wedding | 0,00% | 0,00% | 7,14% | 49,98% | 42,84% |
| Musical Photo set 2, Travelling in Azores | 0,00% | 0,00% | 28,56% | 35,70% | 35,70% |

Table 10 presents the results for question 4. Regarding question 4, the results show that 49,98% and 42,84% for set 1, and 35,70% and 35,70% for set 2 of the inquired subjects considered that the Musical Slideshow was ‘Better’ or ‘Much Better’ than the Basic Slideshow concerning the music synchronization for the photos displaying. In this question, 7,14% and 28,56%, for set 1 and 2, respectively, of the inquired subjects considered that both slideshows behaved in the same way. None of the inquired subjects enjoyed the ‘Same’ or even ‘Worse’ the Musical Slideshow regarding this question. The results may be considered positive despite the absence of performance difference, for a certain percentage of subjects, between the Musical Slideshow and the Basic Slideshow. This may be related to the fact that the beat estimation is not good enough and the synchronization between visual and audio media does not always results as people expected.

- **Results and Analysis for Question 4**

  *Table 11 – Evaluation results for Question 4.*

| Question 4 – If you had this tool, would you use it to generate your own slideshows? |
|---|---|---|---|---|---|
| | a) Never | b) Seldom | c) Sometimes | d) Many Times | e) Always |
| Musical Photo set 1, Casual night party | 0,00% | 0,00% | 14,29% | 57,12% | 28,56% |
| Musical Photo set 2, Travelling in Azores | 0,00% | 7,14% | 35,70% | 42,84% | 14,29% |
Table 11 presents the results for question 4. Regarding question 4, the results show that 85.56% and 56.31%, of the inquired subjects would use the musical slideshow application to generate their own slideshows ‘Many Times’ and ‘Always’. This may be considered highly positive since people are usually accustomed to use familiar tools and it is hard to accept a change. The worse results verified in the content set 2 may derive from the fact that the execution time of this set is larger because it has more photos to process. As people’s time is getting more precious and scarce, there is reduced acceptance due to the processing time. The optimization of the involved algorithms would solve this minor problem when compared to the whole system performance evaluation and the general satisfaction of the subjects when exposed to the final result of the application.

This chapter presented the performance evaluation of the developed system which was obtained by conducting an user evaluation study aiming to evaluate how good was the developed musical slideshow when compared to a more basic slideshow. Also, based on the results and the users overall satisfaction it is possible to predict that the application would possibly have commercial success.
Chapter 7 finalizes this Thesis by presenting the reader a brief summary of the solution developed to address the photo presentation problem as well as some conclusions; after, some prospects for future work are proposed.

7.1 Summary and Conclusions

Chapter 1 introduced the problem addressed in this Thesis, highlighting that the recent boom of digital photos production as well as its common use in people’s everyday life, justifies the development of a system capable of fulfilling people expectations when viewing their photo collections.

Chapter 2 structured the problem at hand, photo consumption, by dividing it into two main areas: solutions targeting photo presentation, and solutions targeting photo browsing. As the solution to be developed intended to fulfill the user expectations in a more passive way, a photo presentation approach was chosen for this Thesis.

Chapter 3 presented the architecture of the developed solution as well as a functional description of each of its modules. The proposed musical slideshow system automatically generates a composite audiovisual presentation. Photo collections are first clustered according to temporal and spatial contexts, providing the units for the agglomerative presentation. For a cluster of photos, the cluster importance is estimated providing the metrics for cluster selection and smart photo manipulation. Accompanying with incidental music, the tiling slideshow not only switches the layout with the music pace, but also displays each photo according to the beat information.

Chapter 4 described in detail the processing algorithms developed to perform the slideshow generation. Four main steps were adopted to produce the final summary. The first step has to process the photo collection; photos that would not be shown on the slideshow are filtered out here. The second step is related to the photo organization with
the goal to cluster the photos according to time and content features. The third step is
the music analysis and extracts the information needed to synchronize the music with
the photos. The final step, composition, is where all the information from the previous
step is gathered in order to generate the final slideshow.

Chapter 5 intended to offer all the information needed by the user to make a proper
usage of the developed application, explaining briefly its high-level software structure,
providing an installation guide, and bringing an in-depth explanation of the application’s
GUI.

Chapter 6 presented the user evaluation study conducted to evaluate the solution’s
performance, describing the test’s objectives, its methodology and, obviously, a deep
analysis of the obtained results.

Resuming, the main objective of this work was to develop an application capable of
generating a slideshow for any kind of audiovisual content by applying complex
processing solutions to produce a final output to increase the viewer’s satisfaction during
the visualization.

Based on the results obtained in Chapter 6, it is possible to conclude that, for the
majority of the cases, the application developed allows reaching good results, being able
to produce exciting/attractive slideshows. The results and the feedback given by the test
subjects were taken as promising, indicating that these innovative slideshows are
considered useful in today’s multimedia world. Despite the positive results, some
improvements may be made in the developed application. These improvements were left
for future work and are discussed in the next section.

7.2 Future Work

The encouraging results obtained in the user evaluation study, let know that this
Thesis has definitely made some strides in creating a better photo presentation
slideshow for consumers. However, on the basis of the proposed idea and from the
feedback received, various issues and extensions may be investigated. Therefore, the
solution developed still leaves room for improvement. In the following paragraphs, it is
detailed some of this discussion from different perspectives:

- **Photo preprocessing** – The sub-modules that were not implemented due to time
  constraints such as the ROI determination sub-module may be easily integrated in
  the future by adding a function since the architecture and interface are already
  prepared and structured to receive those models.

- **Photo organization** – On the basis of photo characteristics, one may want to have
different clustering requirements at the organization stage. If the targeted data are
photos in daily life or are simply landscapes, the system should apply content-based
clustering first to collect photos with similar appearances, and then perform time-
based clustering inside those content clusters. In this case, more elaborate features
can be used for the content-based analysis. Also, including a ROI detection algorithm
would provide new features to the application such as face detection that would
make possible to crop photos around faces.

- **Beat detection** – As users expect a perfect and harmonious synchronization
  between the photos and the music, the accuracy of the music beat detection is one
direction for improving the audiovisual synchronization.
• **Influence of user intervention** – One of the major factors affecting the subjective satisfaction is the users’ preferences. For instance, one person might always prefer to put a specific person’s photo in the bigger cells of the templates. In addition, some photos might be significantly more important to someone even if they are rather similar. The system did not take subjective preferences or specific considerations into account in the process. Manually selecting some photos for a certain action would solve this problem although making the application less ‘automatic’ and ‘passive’.

• **Application extensions** – This musical slideshow application can be adopted to build many interesting applications. For example, given a traveler’s schedule and his or her corresponding photos, the traveler can make a photo based tour. In this case, the application’s GUI would have to be adapted in order to display in small devices such as smart phones. Similar ideas can also be applied to generate customized auto guidance or electronic lecturing. In auto guidance, instead of adopting beat detection to control the slideshow, a GPS system would control the photos appearance cadence (photos indicating the path to next location) coordinated with the user’s movements. In electronic lecturing, a timer predefined by the lecturer would set the photo appearance cadence that would be better to support the topic in discussion.

• **Adding a textual dimension** - Moreover, many research issues still exist if one takes textual information into account. For example, it is possible to allocate some space for text and coordinate more diverse media (such as text, photos, and music).

• **Presentation diversity** - Finally, although this slideshow is currently presented as a continuous clip, it is possible to come up with different kinds of visualizations in the future. For instance, it is possible to just output a text-based script that describes the involved photos and the time stamps and locations for presentation. This script could then be read by a multimedia player such as Adobe Flash and then it could be visualized.

In conclusion, there is still a lot of work to do in the field of enjoyable photo presentation; the possibilities are almost infinite depending on the imagination ...
Appendix A

User Evaluation Study
Instructions

This annex includes the instructions sheet given to the test subjects while performing the test. It defines first the objective of the test and after the sequence of steps to be performed by the test subjects.

**Musical Slideshow application evaluation test**

1. **Test objective**

   This test is destined to evaluate the final application of a project developed with the goal of automatically producing a musical slideshow from a set of photos and an incidental music.

   The test will take, approximately, 15 minutes and consists on the visualization of 2 sets of photos with a music aggregated (1 per folder of photos), for two different types of slideshows: i) a ‘conventional’ slideshow where photos are displayed one-by-one; and ii) the slideshow generated by the developed application. After visualizing both slideshows, the evaluator should score the “quality” of the Musical Slideshow, accordingly to the 4 questions formulated below.
2. Useful definitions for this test

The test set was constituted by 2 photo sets and 2 associated musics; each pair of photos set and music forms what is named a musical photo set. Based on these 2 musical photo sets, 2 slideshows were produced, using the developed Musical Slideshow application.

The 2 musical photo sets correspond to:

- **Wedding** – This set of photos contains scenes from a wedding, the primary targets are people; the accompanying music is an 80’s music, with a constant defined pattern of beats.

- **Travelling in Azores** – This set of photos contains scenes with landscapes and people in leisure activities in the Azores; the accompanying music is rock song, with clearly defined strong beats in some parts.

3. Test methodology

For each set of photos, do:

e. Click on the “Select Folder” button and open the “Evaluation Data” folder located in the desktop. Then open and visualize the photos, starting from “Wedding”. Also, for this set of photos, click on the “Select Music” button and open the same “Evaluation Data” folder to select the music “Fire Inc. - Nowhere Fast”.

f. Press “Start Basic Slideshow” button and visualize. When the basic slideshow ends, press “Start Musical Slideshow” and visualize. Note: The generation of the second slideshow may take some time due to the necessary processing.

g. Answer to questions 1 to 4 above, marking with a cross (X), in the evaluation table, the desired classification score. If it is not easy to decide on a score, it is possible to go back to step b) to visualize again any of the slideshows. The scoring may be done during the visualization of the slideshows.

h. Go back to a), and load the “Travelling in Azores” photos set and the associated ”Blur - Song 2”. Proceed after to the next bullets.
# Performance Evaluation Table

**Name:**

**Question 1** - How did you enjoy the *Musical Slideshow* compared with the *Basic Slideshow*?

<table>
<thead>
<tr>
<th>Photo Set</th>
<th>a) Much Worse</th>
<th>b) Worse</th>
<th>c) Same</th>
<th>d) Better</th>
<th>e) Much better</th>
</tr>
</thead>
<tbody>
<tr>
<td>Musical Photo set 1, Wedding</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Musical Photo set 2, Travelling in Azores</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Question 2** - How did you enjoy the spatial and temporal play sequence of the photos in the *Musical Slideshow* compared with the *Basic Slideshow*?

<table>
<thead>
<tr>
<th>Photo Set</th>
<th>a) Much Worse</th>
<th>b) Worse</th>
<th>c) Same</th>
<th>d) Better</th>
<th>e) Much better</th>
</tr>
</thead>
<tbody>
<tr>
<td>Musical Photo set 1, Wedding</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Musical Photo set 2, Travelling in Azores</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Question 3** - How did you enjoy the music synchronization for the photos displaying in the *Musical Slideshow* compared with the *Basic Slideshow*?

<table>
<thead>
<tr>
<th>Photo Set</th>
<th>a) Much Worse</th>
<th>b) Worse</th>
<th>c) Same</th>
<th>d) Better</th>
<th>e) Much better</th>
</tr>
</thead>
<tbody>
<tr>
<td>Musical Photo set 1, Wedding</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Musical Photo set 2, Travelling in Azores</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Question 4** - If you had this tool, would you use it to generate your own slideshows?

<table>
<thead>
<tr>
<th>Photo Set</th>
<th>a) Never</th>
<th>b) Seldom</th>
<th>c) Sometimes</th>
<th>d) Many Times</th>
<th>e) Always</th>
</tr>
</thead>
<tbody>
<tr>
<td>Musical Photo set 1, Wedding</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Musical Photo set 2, Travelling in Azores</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Thank you for your cooperation,

Bruno Tomás
References


[16] PhotoJam; http://www.shockwave.com/sw/content/photojam.
[21] Bitmap Metadata Date Taken Property; http://msdn.microsoft.com/
[22] Color structure; http://social.msdn.microsoft.com/
[25] MP3 Header class; http://floach.pimpin.net/grd/
[26] Windows Media Player 11 SDK;
[27] Microsoft Visual Studio;