USE OF MORPHOLOGICAL FILTERS IN DETECTION OF FLASHES AND OTHER LIGHT EVENTS IN VIDEO SEQUENCES

By Christian Ekiza & Ferran Marqués

Abstract: *In a collaboration agreement between the UPC and the Thomson Corporate Research Lab, represented by Joan Llach, the objective of this project is to detect local and global flash light events of different intensities in video sequences. Thomson has shown interest in using this kind of information to enable the application of techniques that exploit the characteristics of such events, thus expecting to improve the overall encoding efficiency. This study presents a broad definition for flash light events , and proposes the design and implementation of a flash detector in two steps; a first step of rough detection that uses morphologic filters in both spatial and temporal domains, and a second processing step that offers a much more refined result. In the first stage of the project the objectives were defined and a demo of the application of morphologic filters for the detector was presented. The second stage included a 6 months internship at the Thomson Corporate Research Lab in Princeton (NJ-USA) where the results refinement process was developed and implemented.*

- No -dijo de pronto en voz alta, en el silencio del desván-. Atreyu no renunciaría tan rápidamente, sólo porque las cosas fueran un poco difíciles. Lo que he empezado tengo que acabarlo. He ido ya demasiado lejos para volverme atrás. Solo puedo seguir adelante, pase lo que pase.

- Bastian Baltasar Bux -

[La Historia Interminable, Michael Ende (1979)]

ACKNOWLEDGEMENTS

Antes que nada, tengo que darle las gracias a Marie Line, mi madre. A ella le debo el ser la persona que soy hoy. A Claudia he de agradecerle el haber estado a mi lado siempre, tanto cuando estábamos juntos como cuando había un océano entre los dos. Gracias a ella puedo mirar hacia el futuro con ilusión. Gracias, también, al resto de mi familia por ayudarme y apoyarme cuando lo he necesitado. Quiero agradecer también a toda la gente de Distorsió, los de las revistas pasadas, los de las revistas presentes y los de las revistas futuras, pues fueron, son y serán una segunda familia. En particular, gracias a Nico por echarme un cable cuando la redacción amenazaba con estancarse y a Will por ayudarme en los tramos finales. Y hablando de segundas familias, no puedo dejar de mencionar a los cangrejos, grandes personas y grandes amigos donde los haya. Volviendo al asunto que nos trae aquí, gracias a Ferran por ofrecerme esta oportunidad única. Finalmente, agradecer a Joan, Cristina, Oscar, Rosa, Joel, Cong Xia, Ben Kao, Po Lin, Sitaram, Adarsh, Gad y toda la gente que hizo que la estancia en Princeton fuera una experiencia inolvidable. Muchas gracias a todos.

Table of Contents

1. Briefing

Digital video encoding techniques have always tried to reduce the bit-rate while conserving the highest possible quality. The main concept that has been exploited is the redundancy in both spatial and temporal domain. One of the most powerful tools to take advantage of the temporal redundancy is the prediction, or more precisely the estimation, of the content of frames yet to be encoded. Obviously, the estimations we can achieve will rarely be as good as the real frame, so an error signal is also encoded with the necessary parameters used in the prediction with the purpose to correct the obtained frame and reach the desired image quality. The better the estimation is, the lower energy will have the error image and higher compression levels will be reached.

However, there are a series of unpredictable events that will cause the energy in the error correction signal to increase. An example of these unpredictable events is, for example, when a light is turned on during a scene. The content will be considerably altered by this new source of light, the luminance of most of the scene will increase and even some areas can gain a texture that was not perceptible before that. In these situations, although a human viewer could determine it really is the same scene, in an encoder perspective, the scene will most likely be split in two shots with different characteristics. It is true that some of the information, mainly objects contour or even chrominance information, could be used to predict the new state of the scene, but the change is of such intensity and if it lasts for the frames in the rest of the scene, there is more advantage in encoding the new frame with optimal quality and using it as reference for future predictions. Otherwise, the high energy error will be appearing in successive frames as long as we keep using a reference frame from the previous scene state. In this aspect, the actual management of such events, which comes directly from the utility of dividing the sequence in different individual shots, is quite satisfactory.

But there is a particular kind of unpredictable events that, although similar to the sudden luminance increase problem, present a very significant difference, which is the number of

frames that the effect last in the sequence. While a light that is turned on will most probably remain constant, there are some events such as flash lights, thunderbolts, gun shots or moving light sources that illuminate the scene just for a short period of time. The actual encoders will manage this as, in first instance, a scene change when the luminance increases and, a few frames later, a new scene change when it returns to its original state. The problem with this line of action is not as much a matter of quality as it is an un-efficient exploitation of the scene redundancy.

There are two characteristics in this kind of events with great potential to ameliorate the encoding performance, namely:

- Improve the encoding of the frames **outside** the event. They occur **during** a scene and for a short period of time. Therefore, the content of the scene is presumed to be the same after the event. If it is possible to determine the location of these situations, we can avoid breaking the scene into three different shots, thus the frames that take place before the event could be used to improve the encoding and the compression of those who happen after this hiatus by using the content redundancy within the same scene.

- Improve the encoding of the frames **within** the event. If the events last just a few frames, the high energy error that comes from a non-optimal prediction will not be critically propagated. This should approximate the results of both the estimation approach and the treatment as a different scene **within** the scene. And here is where the location of such events also becomes critical to reach an improved encoding of the involved frames; if we can know in advance where the luminance changes happen, we can trigger advanced prediction techniques that could take profit of objects contour or chrominance redundancy. Moreover, there are some new luminance variation based prediction techniques that could directly take advantage of the fact that we know that a luminance increase is taking place in that precise frame to over-perform the traditional estimation techniques with very interesting results.

Now, knowing that there is such potential yet to be exploited in the encoder management of the described light events, the problem we are facing and trying to solve in this document is to find a satisfactory method to detect such unpredictable situations. In most of the works related with the task of detecting flash and light events, the problem is treated as one of the final or secondary steps in techniques used to detect cuts and other scene transitions, providing a lower false detection rate. These detection methods, indeed, allowed to detect many of the problematic flashes, but are limited to the light events that where falsely considered as transition detections in the initial steps of the procedure. The reason of this strategy is that the main objective was to obtain a correct temporal segmentation of the sequences, not the detection of the events that we intend to study.

In this project, we tackle the problem of detecting **all** the light events that can be considered as flashes. We are not aiming at a correct segmentation of the sequence as main objective. However, our target is not only the detection of flashes affecting a whole frame (global flashes), but also the detection of light events that may appear in local areas of the image. The reason for this approach is that the knowledge of the areas that are affected by these effects can provide useful information to improve the performance of coding systems as well as indexing or tracking techniques.

2. Flashlight Detection

In this document, we analyze the different aspects that have been studied for the implementation of a flashlight detector. There are two different approaches that may be considered:

- Implementing a flashlight detector as a part of a more general chain that may detect other types of events; typically, the various kinds of shot cuts.
- Implementing a stand-alone flashlight detector.

The first approach is based on the idea that usual shot cut detectors commonly mistake flashlights for shot cuts. Therefore, the output of a shot cut detector could be used as input for a flashlight detector, reducing in this way the amount of data to be processed by this second system.

However, since the study of the second case allows us to explore a wider view of the flashlight detection problem, in this document **we will focus on the stand-alone detector**. To do so, in ***Chapter 3*** we review the detection of light events State-of-the-Art and analyze the usual approaches to the problem. In ***Chapter 4*** we treat one of the weak points of the post shot cut detector techniques, the characterization of the flash events. In ***Chapter 5***, having reviewed the State-of-the-art and with our wide characterization of light events, we are able to define an approach based on some of their most relevant features. Then, in ***Chapter 6*** we extend the defined approach and expose the different steps of a proposed algorithm that includes a detection step and a refinement step. In ***Chapter 7*** we present and discuss some relevant results of our algorithm. The conclusions are exposed in ***Chapter 8***, in which we discuss the performance of the proposed technique. ***Chapter 9*** includes the additional features included in the project, namely, an adaptation of the algorithm to detect events with an inverted luminance pattern and the implementation of a GUI. Finally, in ***Chapter 10***, we propose some further improvements based on the obtained.

The ***Appendix A*** offers an overlook to the morphological filters that are used in the proposed technique, extending the 1D and 2D concepts to a 3D framework that includes time as a third dimension. And the ***Appendix B*** consists of a user manual for the implemented GUI.

3. Light Events Detection State-of-the-Art

The flash detection problem has been typically faced as a post-boundary detection technique used to discriminate the flashes as false detections after the process of scene change detection. Hence, the most interesting papers on this subject usually handle this problem in very close relation to the scene break detection.

Some of the basic approaches are reviewed in ***[1]*** where the **benefits of using the dc-images** are widely exposed. Since these images retain global image features, they are ideal for scene change detection and flashlight detection purposes. Moreover, algorithms performing with similar results as in the original images are substantially speeded up since full decompression is not required. Once the decision to work on these down-sampled versions of the images is made, **the authors discard motion compensation because of unreliable and unpredictable performance at this level**, and propose different metrics based on successive pixels difference, global statistics and similar methods based only on the luminance information.

Also in this paper the scene changes and flashlights are presented as a local activity in the temporal domain, and thus, an adaptive threshold is determined with a sliding window. Then, ratios between the highest distance and the second highest distance and the relation with the rest of values are used to determine both scene changes and flashlights. Nevertheless, there is a tradeoff between the detection of weaker flashes and the false detection rate.

Finally, a combined detection is proposed in which the low sensitivity to large motion effects of a histogram based approach is used to compensate some weak points of the algorithm in detriment of computational cost.

A different approach is exposed in ***[2]***, where the comparison of intensity edges in consecutive frames is used to obtain rates of exiting and entering edge pixels, thus

detecting and classifying hard cuts, fades, dissolves and wipes. However, previously, motion compensation is performed to reduce the impact of camera or objects span and a Haussdorf or partial Haussdorf distance compares the two pixels sets.

This technique presents noticeable **advantages over histogram-based methods**. Even though, it fails in detecting scene breaks in rapid changes in overall scene brightness as well as in very dark or very bright scenes. Also multiple object rapid motion causes the motion compensation to be insufficient to perform a good detection.

This study was one of the various that inspired ***[3]***, one of the few papers that, even though sharing the usual *post-shot detection philosophy*, tried a different approach closer to the flash study and indexing. While other studies are interested in the maxima of the indicators used for boundary detection, Heng and Ngan proposed to redefine the flashlight concept and study the adjacent frames in which the effects occur, instead of analyzing the frames before and after the effects, thus avoiding several situations under which traditional methods could fail. The main feature that this technique uses is that during the flashlight effect the objects remain the same in scene. Thus, in contraposition to the abrupt scene changes, a correspondence may be found by comparing the edge of the objects, taking into account possible changes in light source direction, edge boundary splitting, objects that appear and disappear and the difference of details in the same object. The comparison principle uses the number of matched edge pixels in two consecutive frames (instead of unmatched pixels rate, as seen above) adding edge direction information to reduce the amount of random pixels being matched and using matched edge elimination **(MEE)** to restrict the matching only to one-to-one pixels, thus introducing the concept of matched directed edge elimination **(MDEE)**. Some of the advantages of this technique compared with feature based detection **(FBD)** and Hausdorff distance histogram detection **(HDH)** are:

- A highly accurate thresholding
- A low uncertainty in erroneous matching for a fixed dilated radius
- And a lower sensitivity to the variation in dilation radius in matching accuracy

However, the priority in this study is set in abrupt scene change detection, thus preferring to have a flashlight event indexed as a hard cut (false detection) than to discriminate erroneously an abrupt scene change as if it was a flashlight (missed detection). Then, conclusions lead to consider this technique to significantly improve the accuracy of discriminating flashlight effects in frame pairs that fail the similarity test during shot boundary detection.

4. Characterization of a flash

As seen in the State-of-the-art, the goal we are aiming at has never been proposed as a stand-alone method, but always as part of another detection problem, thus defining the flashes with a limitation given by the previous steps of the process. Moreover, as we are aiming at the detection of many different flash events, we need to have an unbiased definition of what we are considering as a flash and why it is considered so. Then, the first step we take to face the design of a stand-alone flashlight detector is an in-depth description and characterization of what we mean by flash or light event. This definition may provide us a wide perspective of the problem in order to decide which the most relevant characteristics are and how they can be processed to extract the largest amount of information.

There are several features that can be used to characterize a flashlight. However, a flashlight may appear in the scene with high variability, presenting very different values of these features. Here, we list those features we think are the most relevant ones and, where possible, we analyze them from the viewpoint of our final application; that is, detecting flashlights to improve the temporal prediction in a coding system.

> *A **flashlight** appears in a sequence as a local or global abrupt change of the luminance in a single frame or a few consecutive ones. This luminance change usually follows a pattern involving a sudden increase of the luminance values (typically, in one frame and leading to luminance saturation in a large part of the flashed area) and a subsequent decrease, which may last for a few frames, down to recovering the original luminance values and, if the scene has not undergone large variations, approximately recovering the previous image content.*

We think that this description covers in a large extend all the actual flashlights that may appear in a scene. Therefore, we propose to base our detection algorithm in the attributes that have been listed above.

However, there are other light events that, although not being strictly flashlights, may be interpreted as such and which can be detected or not by the proposed system depending on the used descriptors.

In the sequel, we discus the various attributes that have been used in the previous description:

4.1. Spatial location (local or global)

Flashlights may affect the complete image or only a specific part of it. Thus, if the whole image, or a high percentage of it, is affected by the flash, we want to index it as a **global flash**. On the other hand, if there is a considerable area of the image that is not affected by this light event, we will consider it as a **local flash**. Flashlights affecting a very small portion of the image should not be reported. The image is initially down-sampled and flashlights are detected in this down-sampled version.

- The DC version of the image is used (one pixel per 8x8 original image block).

All the following processes are applied on this down-sampled image. Since it is interesting to separately detect which parts of the image are affected by the local flashlight(s), **images in the sequence are partitioned and the selected detector is applied to each segment in the partition**. In order to simplify the algorithm, the partition does not imply any previous analysis of the image and a fix partition into square blocks is selected.

- It seems coherent to adopt a block size related to that used in the coding scheme.

- It does not seem very relevant to classify the flashlights into other classes rather than global and local, and to provide the area of influence in the local case.

- If possible, different levels of area of influence should be provided. The scalability of the area detected as flash is related to the impact that the flash has in this image area.

4.2. Temporal span (a single frame or a few consecutive ones)

Flashlights last usually from one single frame to a few ones. In this work we assume that the duration of a flashlight is short (between 1 and 5 frames) and, if the light event lasts more than a given threshold, it will not be classified as flashlight.

- Note that, since we are locally analyzing the sequence, the fact of imposing a short temporal window does not imply that a burst of flashlights will be overlooked.

- Also, the duration of the flash detection must be adapted to the proprieties of the light events we are willing to detect considering the time and frame rate dependence

- On a side note, in the following study, we have limited the definition to the number of frames that the event lasts, regardless of the sequences frame rates. This decision has been made to simplify the study and comparison of the results for the different sequences, which had different frame rate values. By unifying the sequences with this criterion, it is much easier to determine if the detections are correct or are false alarms, no matter what sequence it belongs or what frame rate it is using. The

conversion between time and frames is considered in the further improvements section.

4.3. Content correspondence

Since short flashlights are to be studied, we can assume that the content in the scene does not suffer large changes during the event; that is, if the sequence does not undergo large variations due to motion, **roughly the same objects are present on the previous and posterior frames of the flashlight**. This content correspondence can be used to classify light events as flashlights.

- This idea of content correspondence eliminates the possibility of detecting a flashlight in a shot transition. In such cases, we have to rely on a stand-alone approach based on other descriptors.

- Objects contour and motion estimation techniques can be used to determine the content correspondence, but their high complexity must be taken into consideration.

- Also content correspondence can be found within the event if the flash only affects a delimited area of the image or if the luminance variation does not affect drastically the contour of the objects.

4.4. Luminance pattern (sudden increase and subsequent larger decrease)

The model for the luminance pattern that we are proposing corresponds to an abrupt (maximum level reached in 1 or 2 frames) increase of the luminance in the flashed area, and a subsequent decrease of the luminance (from 1 to 4 frames).

- This pattern should be simultaneously followed by collocated blocks. Neighbour blocks during a flashlight should follow similar patterns but may present different dynamics (see *Dynamics of luminance pattern*).

Although the chrominance component of the sequence also presents a particular pattern during a flash event, we consider that the information that it provides is redundant in terms of flash detection for the very source of a light event affects mainly to the luminance in the picture.

4.5. Dynamics of the luminance pattern

As previously commented, a flashlight is expected to yield saturation or very high luminance values in a large part of the flashed area. However, there may be an additional part (a crown around the saturated area) that can reach lower values, yet being part of the flashlight.

- The presence of a saturated area should trigger the search for a surrounding area where the flash is active as well.

In turn, a flashlight is expected to decay after a few frames without having any reminiscent light effect in the scene.

- The sequence, after the end of the flashlight and if it does not suffer any other strong change, should recover its original content and, therefore, its original luminance and chrominance values.

Nevertheless, we want to go further and include not only luminance events that lead to saturation, but also sudden changes in the luminance that reach lower intensities.

4.6. Shape of the flashlight: isotropic or anisotropic

Depending on the relative position of the objects in the scene and the source of the light, the flashlight may present any kind of shape. Therefore, we do not consider the flashlight shape as a discriminating feature.

5. Proposed approach for a stand-alone flashlight detector

Most of the previous works on flashlight detection have followed the approach of first applying a shot detector and then looking for flashlights among the selected frames. Usually, due to this approach, the first selection of candidates is too large and seems to make more complicated the flashlight detection. Moreover, these techniques are not using all the features that characterize flashlights to detect them. For instance, in ***[4]*** the analysis of the brightness in the flashlight area is not really exploited and in ***[3]*** only the fact that similar transitions should appear before and after the flashlight is used.

The proposed technique tries to perform an initial filtering of the sequence based on determined flashlight features, specifically, relying on the fact that flashlights;

- Last a few frames (1-5),
- Imply a rapid luminance increase that may partially saturate the image.

We propose to use Mathematic Morphology tools for the video analysis, both in space, to smooth the image, and in time, to determine the actual presence of a flashlight; we will save the more complex use of features based on contour information for posterior refinements and improvements. Such an approach is proposed in ***[2]*** and further improved in ***[3]*** where the approach is justified to detect very complicated flashlight cases: gradual flashlights, flashlights embedded in a shot cut, etcetera. On the opposite, the concept of not only analyzing the features in the frames before and after the flashlight but studying all the adjacent frames in the flashlight ***[3]*** seems interesting and we adopt it and adapt it to the concept of luminance pattern.

This way, the proposed algorithm contains the following main steps ***[Fig. 1]***:

1) Down-sample the image (8x8)
2) A Mathematical Morphology pre-processing step to exposes those events that locally behave in the way we expect the light events to perform.

3) A processing step that prepares a set of flash candidates for further in depth analysis.
4) An independent, flash resistant, shot cut detector.
5) A decision step that uses information from the shot cut detector to discriminate some candidates and uses a set of criteria to discriminate false alarms.
6) Generate output.

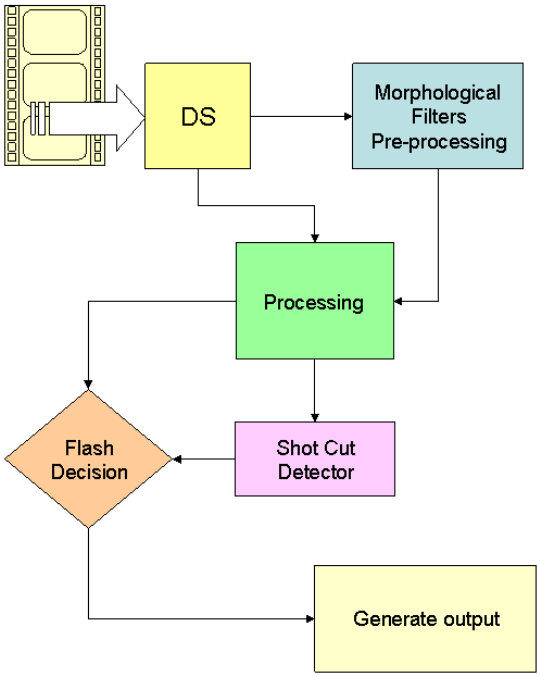


Fig.1 *Block diagram of the proposed method. After the down-sampling, a morphological pre-processing is performed, and combined with the information from the original images two parallel processes are performed; a flash detector and a shot cut detector. The results of the shot cut are used to complement and correct the flash detector results and then generate the output files.*

Some of the previous steps will not be carried on sequentially but recursively, in order to speed up the algorithm, mainly in the pre-processing and processing steps. This will end up with an algorithm unable to perform real time detection, although the technique may be used, in spite of his non-causality, in real time applications that allow a delay of a few frames.

However, since in this project we are not aiming to real time processing, we will only adjust the algorithm to the recursive structure when this means a substantial save of

resources. For example, in order to compute the temporal Morphological Filters ***[Appendix A]*** in an efficient way (recursively), and assuming that the time dimension of the Structuring Element is the maximum duration established for a flashlight (2N+1), we need to keep 3N+1 frames in the memory. Moreover, in order to analyze the flashlight pattern in a given image box, we need to have the data of the collocated blocks in the previous and the subsequent images with respect to the flashlight. Therefore, **we need to work with a temporal window of 3N+3 frames stored in memory** so we can optimize the number of times we have to access the memory to read the source frames.

Finally, since we are working with two very different kinds of flash events (global and local flashes) we will consider that when a **global flash** is detected, even though a few pixels may not experience a considerable change, the entire frame is affected by the flash.

On the other side, for the **local flashes**, since the area and shape acquires a relevant importance in its detection, we define **3 different scalability levels** ***[Fig. 2]*** for the final representation:

- The first and roughest level is defined as **'box scale'** and corresponds to a division of the images in boxes that will be marked as positive if a flash event is detected within this area.

- The second level is what we call a **'seed scale'** and presents the shape that corresponds to the core of the detected flash events that is detected using an adaptive threshold in the above mentioned boxes marked as positive detection. This level should provide the region that is more intensely affected by the detected event.

- The third and last level of scalability is the **'extended seed scale'** since it is the result of applying a region growing technique to the results of the 'seed scale', extending the detected area to a more accurate shape of the real area that is

affected by the flash. At this level, we are determining the region where light affects the content with any intensity level without distinction.

Fig.2 *In the top three consecutive frames corresponding to the sequence 'Britney' where we see a local flash. In the bottom, the representation of the three different scalability levels of the flash detection for each frame. Since there are no flashes, nothing is detected in previous and posterior frames, but in the middle frame, the one where the event takes place, we can see how the box scale detects the boxed region where the light event appears, the seed scale shows the area that is the most intensely affected by this effect and, finally, the extended seed scale reconstructs the correct shape of the flash area.*

6. Proposed Algorithm

In this section we will introduce the proposed algorithm with considerations of the objective aimed at each step. Technical information about the specific morphological filters used is provided further in this document.

6.1. Down-sample the image (8x8).

The first step is an 8x8 down-sampling using the mean value of each block. The 8x8 size was chosen for compatibility with the DC component of the usual coding schemes, but can be set to other values. This step not only **fastens** considerably **the process**, but also **allows removing very small objects**.

However, if we are applying the detector to sequences with low resolution, this step may lead to work with images too small to allow an accurate performance. Thus, an option is added to allow smaller ratios in this down-sampling step.

6.2. Mathematical Morphology pre-processing.

Using only the Y planes of the down-sampled sequence, we perform an equally weighted, 1-dimensional **temporal white top-hat** (by concatenating a temporal erosion and a temporal dilation and then subtracting the result to the original luminance source) ***[Appendix A]*** in the temporal domain in order to **only preserve the short (in time) luminance peaks**. However, since this is applied in the down-sampled sequence, the luminance information that we are using corresponds to the mean value of the blocks in the original sequence. Thus, by performing the temporal white top-hat operation using the pixels that are in the same position but in the previous and posterior frames, temporal light variations with an area smaller than a block may be diluted within the corresponding block average value. The result of this is a sequence of images where the areas that locally experience a temporary increase of the luminance are emphasized, in opposition to those who do not present substantial light variations. This pre-

processing step introduces the concept of the **temporal span** that we have defined in the flash description. The **dimension** of these temporal erosions and dilations is an input parameter to the system, **typically set to 5**. This temporal span = 5 defines the length of the **vector of ones** used as structuring element. As a result, 5 is also the 2N+1 frame span that will require 3N+3 frames to be stored in order to speed up the processing time through efficient access to the video file ***[Fig.3]***. In this case, 9 frames are to be stored.

The next step aims at **discarding the small (in space) peaks of each image** (considering that each pixel corresponds to an 8x8 block) and consists of a **spatial opening on each obtained Y plane**. Although an erosion operation would perform identically in the discarding step with lower computational cost, the opening operation presents a significant advantage which is that it preserves the local influence of the detected events within the frame. The relevance of this information in further steps was decisive to choose the opening despite the required cost. The **dimension** of the spatial opening is an input parameter to the system, **typically set to 3,** which defines the size of the **square matrix of ones** used as structuring element.

Original Frames	Frame T-2 -> Y	Frame T-1 -> Z	Frame T	Frame T+1	Frame T+2	Frame T+3	Frame T+4
After Morph. Filter	Y	Z	A				
Dilation	A1	A2	A3	A4	A5		
Erosion 1			A15				
Erosion 2				A25			
Erosion 3					A35		
Erosion 4						A45	
Erosion 5						A54	A55

N		1	N	N

Read	Calculate	Result	
T	A1		
T+1	A2		
T+2	A3		
T+3	A4		
T+4	A5	A	
Frames stored in buffer to calculate A			A1, A2, A3, A4, A5, Frame T+3, Frame T+4
Total buffered frames			3N+1
Total buffered frames for pattern 3N+1+2			3N+3

Fig.3 *Schema of the frames that need to be stored in the buffer to apply the morphologic pre-processing filters.*

6.3. Processing step.

Reached this point, the mean value of the white top-hat followed by an opening (*wtho*) shows that a rough threshold could easily detect the global flash events ***[Fig. 4]***. However, the challenge lies on setting **a threshold that allows us to synthesize an efficient set of seeds for both local and global flashes detection**.

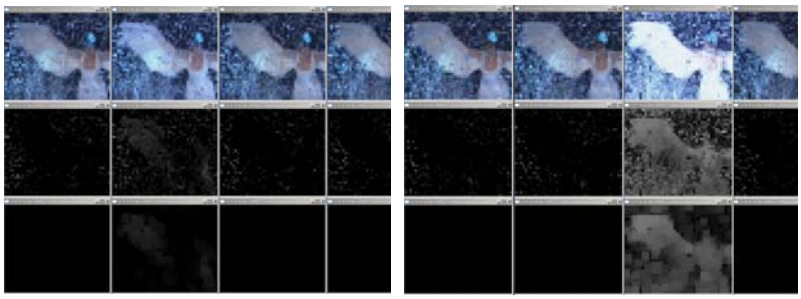


Fig.4 *In these two sets of frames we can see, from top to bottom, (1) the original down sampled sequence, (2) the temporal top hat and (3) the result after the spatial opening to that temporal top hat where the noise has been considerably reduced. In the first set we appreciate a local flash in the second frame. In the second set what we see is the effect of a global flash.*

We need an **adaptive threshold** in order to extract the desired information from the *wtho* sequence. A first approach to this problem suggests a **direct relation between the threshold and the mean luminance values**. This threshold must be within an interval fixed by:

A minimal lower threshold: to discard too weak luminance variations.

A maximal lower threshold: to include all the desired pixels when the mean value is considerably high due to a global flash or to light events of different intensities.

However, since we have yet only used some of the roughest, yet simplest, features of the flash definition (temporal span and part of the dynamics of the luminance pattern) in this pre-processing step, other events than light or flash events fit in the actual definition too. Thus, **clear objects that have a relative high and continuous displacement along consecutive frames may be**

wrongly considered as local light events. This situation causes a problem while determining a satisfactory threshold; if it is too high, some local flashes are skipped, but if it is too low, many moving objects are erroneously marked.

To circumvent this obstacle, a **motion based discrimination method** is required. The most reliable technique to approach this task is the motion estimation used to predict frames in compression methods, however, this is a very computationally expensive technique and we still are in a very early stage of the detection with too many possible candidates. This technique will be saved for later stages where the number of candidates will be considerably lower, thus optimizing the cost. So, a different approach to the motion based discrimination is adopted and a division of the frames in a **grid of 5 columns and 4 rows** grid ***(size can be modified with input parameters)*** is proposed.

The previous pixel based pre-processing techniques failed when the **clear objects** where **moving** in front of darker backgrounds, thus **marking** some parts of these objects (mainly the **boundaries**) as flash candidates. Then, the continuous motion of the objects should appear on consecutive frames in neighboring areas. Now, with the new partition, the **motion** of the objects may remain in the **same region** during some **consecutive frames**.

The study of the **temporal variation** of the **mean luminance level** of each **cell (or block)** in the 'wtho' signal allows us to evaluate the evolution of the light events that take place within its area. Thus, if in the same **block** a light event is detected through **several (>5) continuous frames**, we can interpret this detection as not being a real light event, but the **effects of the motion** of one of the above presented clear objects. This discarding process corresponds to a primary study of the luminance pattern within the defined temporal span.

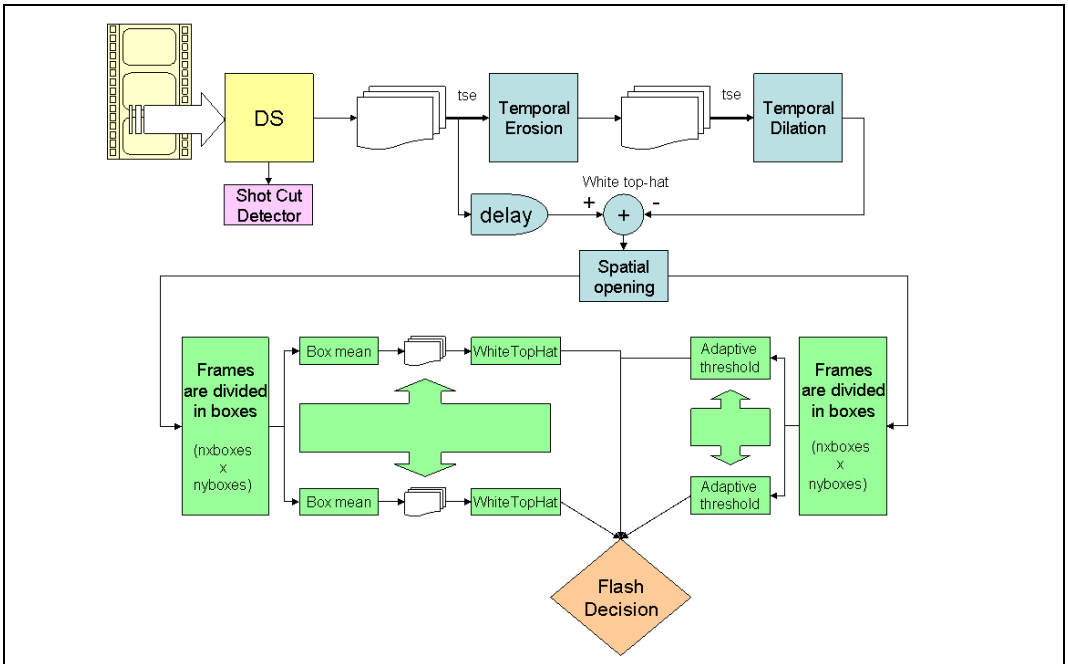


Fig.5 *After the down-sampling, the morphological filters, a temporal white top-hat and a posterior spatial opening, are performed on the sequence. Then the images are divided in boxes and an adaptive threshold is performed in each one. Also another top-hat is performed on the means of the boxes and used as a correction to decide if there is a flash in the involved boxes or not.*

Then, a new **white top-hat** is performed, in the **temporal domain** (with **dimension 5** and using the **mean Y levels of each block** in the 'wtho' sequence, the equivalent 5x4 pixels representation of the image), to remove with a new threshold the locally persistent light effects, that are presumed to be moving objects, from our candidates ***[Fig 5-9]***.

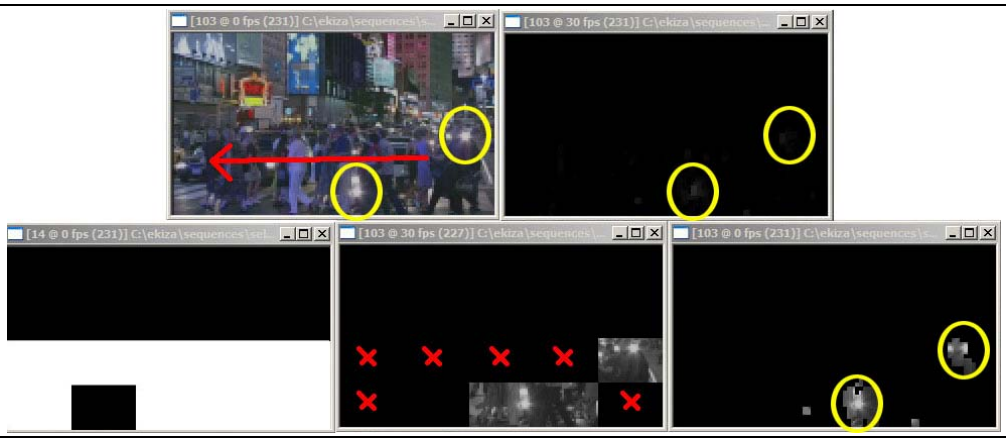


Fig.6 *Sequence: 'night' frame 103. Top left: original down-sampled frame. Top right: 'wtho' signal. Bottom left: boxes with positive flash candidates. Bottom center: boxes that remain after discrimination. Bottom right: region detected as flash.*

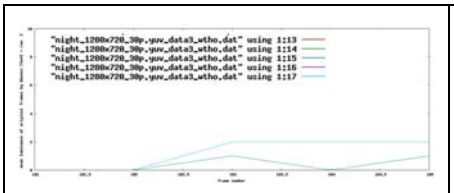


Fig.7 *Mean Y value of the boxes in the second row from the bottom in the 'wtho' signal. Only the last box (cyan line) has a pattern that can be detected with this technique (see Fig.6).*

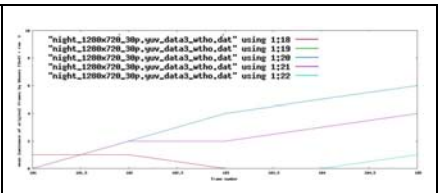


Fig.8 *Mean Y value of the boxes in the lower row in the 'wtho' signal. Two boxes (blue and purple lines) present a pattern that validates the candidate (see Fig.6).*

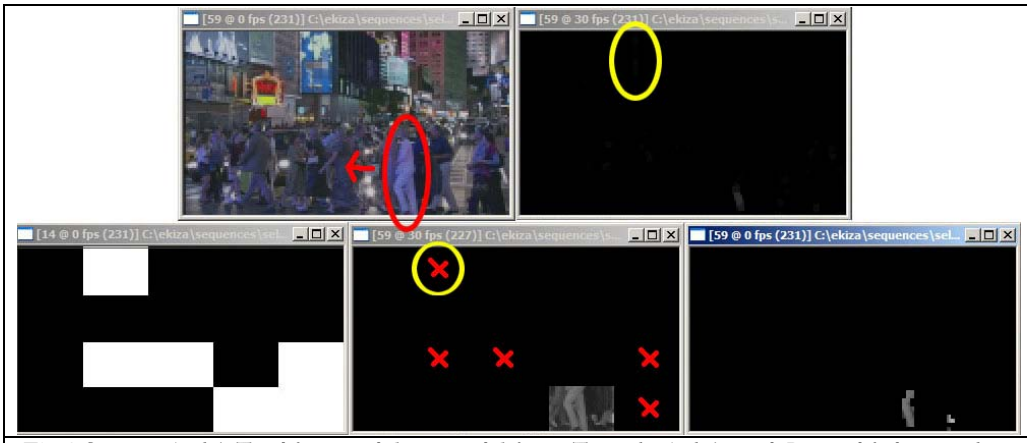


Fig.9 *Sequence: 'night'. Top left: original down-sampled frame. Top right: 'wtho' signal. Bottom left: boxes with positive flash candidates. Bottom center: boxes that remain after discrimination. Bottom right: region detected as flash.*

We can see how **the proposed method removes an important number of false candidates that have appeared due to motion**, although some extremely clear moving objects still pass as flash candidates. Also, when a flashlight affects very slightly an area that is small compared to the box size, it may disappear as a candidate as exemplified in the neon light in the upper left side of the image. Note that the light of the car in the bottom center of the image ***[Fig.6]*** is not detected since it appears through many frames and does not fit the temporal span condition of the definition.

The remaining regions are then considered as **regions of interest**. A **threshold** will be decided **for each region** using the mean value within the studied box in the *wtho* signal, and with the same criteria for minimum and maximum values as suggested before. If any **seeds** are found in these boxes, they will be marked for

reconstruction using a region growing technique. The result is a more precise binary marker for the regions considered to be flashes.

The presented technique enhances the detection by discriminating false alarms due to moving objects, but still present difficulties with very fast objects, 'oscillating' objects and with objects that move from one box to another ***[Fig. 9]***.

6.4. Shot cut detector.

In some occasions, we found that the objects that are moving in the scene just before or after a scene cut were causing the algorithm to falsely mark a flash detection. With the objective of not only solving part of the detection problem but also to provide additional information on the sequence, we designed a scene cut detector more concerned about accuracy than about precision; more precisely, the scene cut detector had to be resistant to the light events. Since in previous steps we already postponed the real time detection to further implementations and in order to simplify the algorithm, these steps are not applied immediately but after having processed the complete sequence.

A **cut detector** is executed to discard the cuts that do not have flash event in their boundaries but still have been marked as candidates ***[Fig.10]***. This process is **not aiming to all the cuts** because we do not want to remove those who are near to a light event, **but those in whose boundaries** we can have a high certainty that **there are no flash effects**.

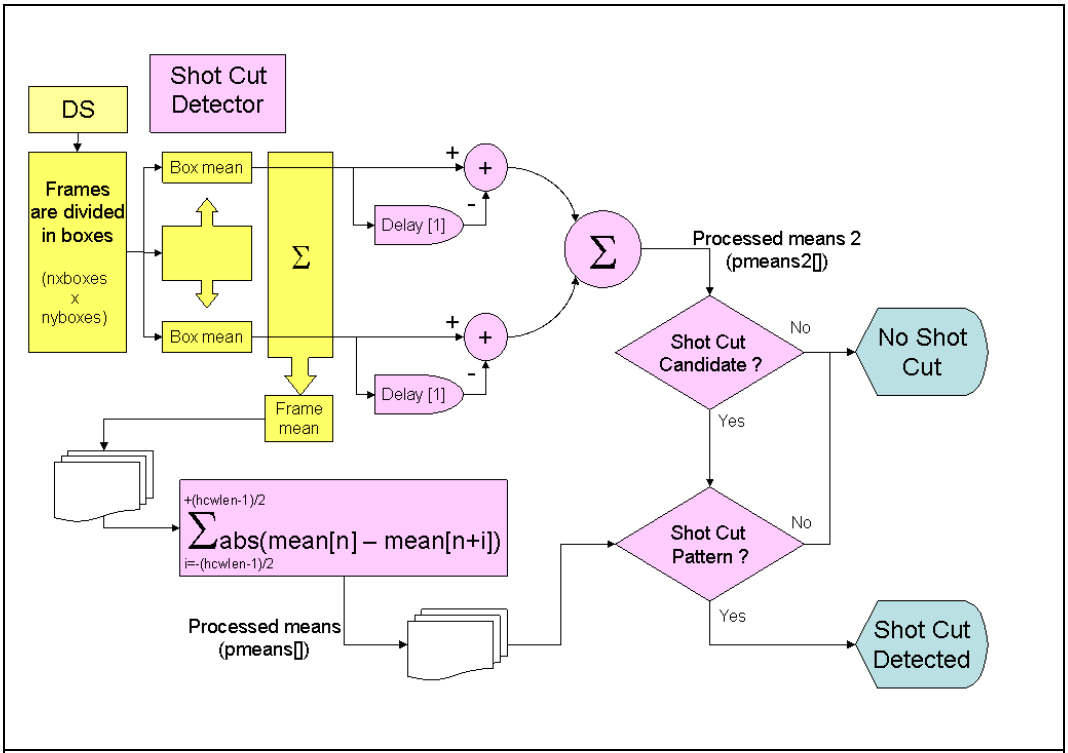


Fig.10 *For the shot cut detector, a first simple signal is calculated and used as a trigger to study a second and more complex signal that may present a particular pattern when there is a shot cut in the sequence and there are no flash events in the surrounding frames.*

Two signals are then calculated with the previously obtained box means and frame means. The first one is the sum of the **absolute difference of the blocks of one frame with the corresponding blocks of the directly preceding frame**. With this signal, and using an adaptive and a fix threshold, we determine when to activate the second cut detection step.

This second step is performed over a new signal obtained with the sum of the **absolute difference of the frame mean luminance with** those of the **frames** that are **under a rectangular window centered in the actual frame**. Thus, a window of length 7 (typically set value) will compare the frame with the previous 3 and posterior 3 frames (boundary frames). The effect is that when a cut is detected this signal reveals a **regular pyramid shape *[Fig. 11]*** with values [A:2A:3A:3A:2A:A], where 'A' is the difference between the luminance mean of one of the frames before the cut and a one frame after the cut. The benefit of this technique is that when a flashlight event occurs, the pyramidal shape does

not appear, even when it happens in the boundaries of a cut, thus preserving the light events previously detected.

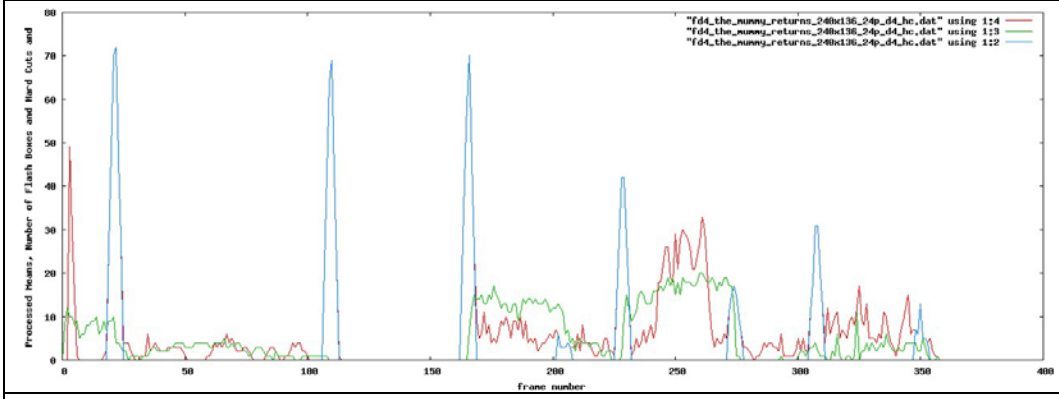


Fig.11 *In the 'The_Mummy' sequence, we see how the shot cut detector correctly detects 8 out of 9 shot cuts tracking the pyramidal pattern (blue line) in the 'processed means' signal (red line). The shot cut that fails is the one between frames 2 and 3, too close from the starting frames of the sequence.*

This particular technique has a limitation, though. If the mean luminance of the frames inside the flash remains almost constant during the whole event and it lasts for more than 6 frames, this particular event could be marked as a different shot having a scene cut when it starts and another one at the end. However, these conditions would imply an event that, although lasting several frames, does not follow the defined luminance pattern. Moreover, an event lasting for more than 6 frames is larger than the 5 frames limit we consider for this work, and could perform better if considered like a different shot while encoding the sequence. Nevertheless, if the detection of such events is further required, there is an option to **disable the use of the scene cut detector**.

6.5. Decision step.

Depending on the percentage (adjustable by parameter) of **boxes** that remain after *the motion based discrimination method* ***[Fig. 12]*** we can perform an initial indexing of the frames in 3 categories:

None (NONE): no flash effect is detected, but posterior processing can change this state to 'Cut' (or other events such as dissolves or fades in further improvements).

Local Flash (LOCAL): A flash effect is detected in this frame and it will be subject to further analysis to determine if: a) it is a false alarm (and should be removed from the results), b) it is a possible light event or c) it is a light event with a high degree of reliability. If it is eventually confirmed as a possible local flash, a binary map of this region will be saved in the pertinent file.

Global Flash (GLOBAL): Since it is meant to affect the whole image, no binary map will be saved in this case. However, depending on the scalability level selected, it will be calculated for further filtering.

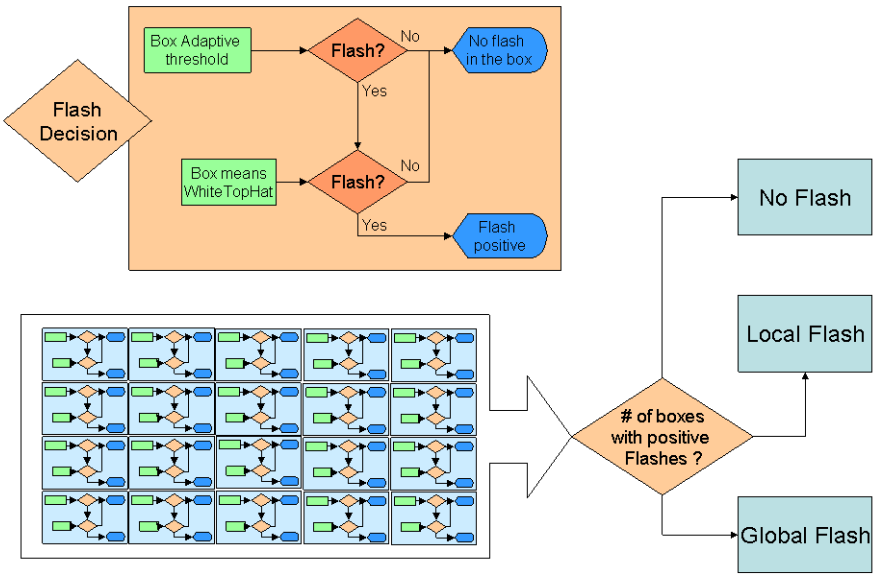


Fig.12 *The percentage of boxes in a frame that present a positive flash detection after the threshold and the correction will determine if there is a local flash, a global flash or no light events in the frame.*

If the selected scalability level is the expanded seeds bitmap, either for a local or a global flash, we have the option of setting a **minimum area condition** (typically 10%) such as to remove any candidate whose flash affected area does not reach a determined percent of the frame size. In a similar way, a **minimum global flash**

area (typically 50%) is defined in order to keep a global flash in that category; otherwise, it will be downgraded to local flash candidate, even though it may had been labeled as a global flash in previous, less precise, steps.

Once the whole sequence is processed, the previous results obtained from the scene cuts detection are included in the indexing process. Therefore, the two frames that determine each scene cut are labeled as 'Cut'. Also, due to the resistance of the cut detector to light events, the neighboring frames of the detected scene cuts are automatically discarded and indexed as if no light events were present in the involved frames (NONE) ***[Fig.13]***.

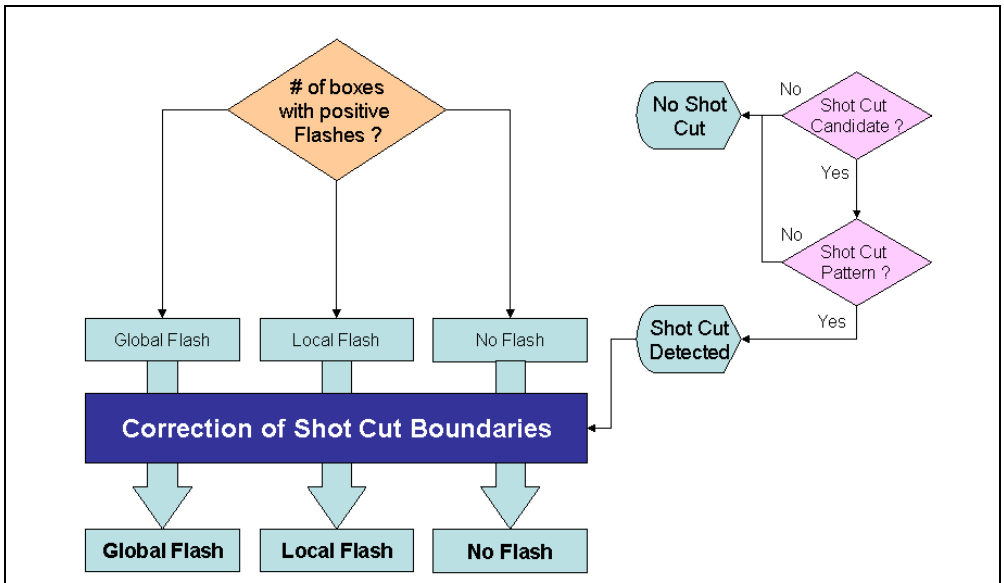


Fig.13 *When detecting a shot cut, and because of the design of the cuts detector, we assume that there are no light events in the surrounding frames. Thus, a correction is performed if necessary to remove possible false detections.*

Then, what we introduce is a **length limitation** for the detected flashes. If a light event stands for more than the desired amount of frames (typically 5 frames, as set in the earlier definition of the *temporal span*), the whole set of consecutive frames will be discarded.

Up to this point, this algorithm proved to provide a set of candidates that included almost all the light events of the sequences in which it had been tested.

Even very weak light events were clearly exposed in the resulting detection. However, in some sequences, a large amount of detections turned to be false alarms mainly due to situations with an effect that could, locally considered, fit the features we use to describe a flash. Summarizing, an extremely **high detection rate** was obtained with a very **precise shape** for the local flashes. Also an almost perfect detection was achieved for global flashes and the shot cut detector performed quite satisfactorily in terms that it fit the design conditions of flash events preservation.

The results so far, then, induce us to think that, even though the detection is also usually triggered by **moving objects** that are very clear in contrast with their background, **practically the totality of the vast range of light events are detected** in the final candidates list. We consider that, in order to reach a good flash detector for **indexing purposes**, a **very good basis** is obtained using just a few of the basic attributes of the flashes. Thus, **many paths** are still available that we can introduce **to improve the performance** of the detector.

On the other side, if we are considering the detection as a previous step to be used as an **additional input to a video encoder** aiming to improve the encoding performance, we cannot give a definitive conclusion about this level of detection without **testing the actual encoding results**. A direct evaluation of the influence of the non-flash events detected cannot be performed, since these effects that trigger the detection obviously share some local characteristics with the light events. How they may affect to the performance of the encoder can only be determined by an exhaustive testing that is not part of the scope of this study.

By this point, we have used only a few of the descriptors we defined for the events we intend to detect. There are some of them that still have to be exploited in order to improve the obtained results, namely, the luminance increase and

decrease pattern that occurs during the light events and the content correspondence.

In the following we proceed to describe ***a second step of processing*** after the first rough detection. All the decisions are ***optional procedures aiming to discriminate false alarms*** based on theoretical differences between light events and wrong detections. Also some of these procedures will help us to go further in the indexing process not only by removing false positives, but also with the introduction of the concept of **highly reliable detection** in contrast to regular detection. However, we want to emphasize the fact that, with the set of detections that we have obtained so far, the priority is to preserve the correct detections. With that objective in mind, we will tackle the refinement process considering all the detections as a legitimate positive until we can demonstrate it is not, instead of trying to prove they are indeed light events. We decide to do so because of the wide range of light events we are aiming at and the heterogeneous behavior they can present. Considering that, it is much more reliable to determine some patterns that are clearly opposing to what a flash should theoretically mean in terms of signal processing. Moreover, due to the detector's proved highly effectiveness while determining global flashes and in an attempt to avoid adding unnecessary computational time to the process, **most** of the following **refining techniques will only be applied to the local flashes**.

The next **histogram-based discrimination method** applied (only to **local events**) is related to the **luminance pattern** that is expected to appear when a light event occurs. The study of the histogram behavior during the detected events leads us to one feature that remains consistent in what we have considered to be a light event; during a flash, a group of pixels moves towards a higher luminance value range to, immediately after the event, return back to similar levels as in their original state ***[Fig.14]***. Thus, determining in the histogram a range of luminance values that represent the effects of the event will expose the

temporal evolution of the bins affected by the flash. An increase and posterior decrease of the number of pixels that we find in this 'flash range' would ratify the detection of a light event, whilst some other patterns would help us to reliably discard false alarms.

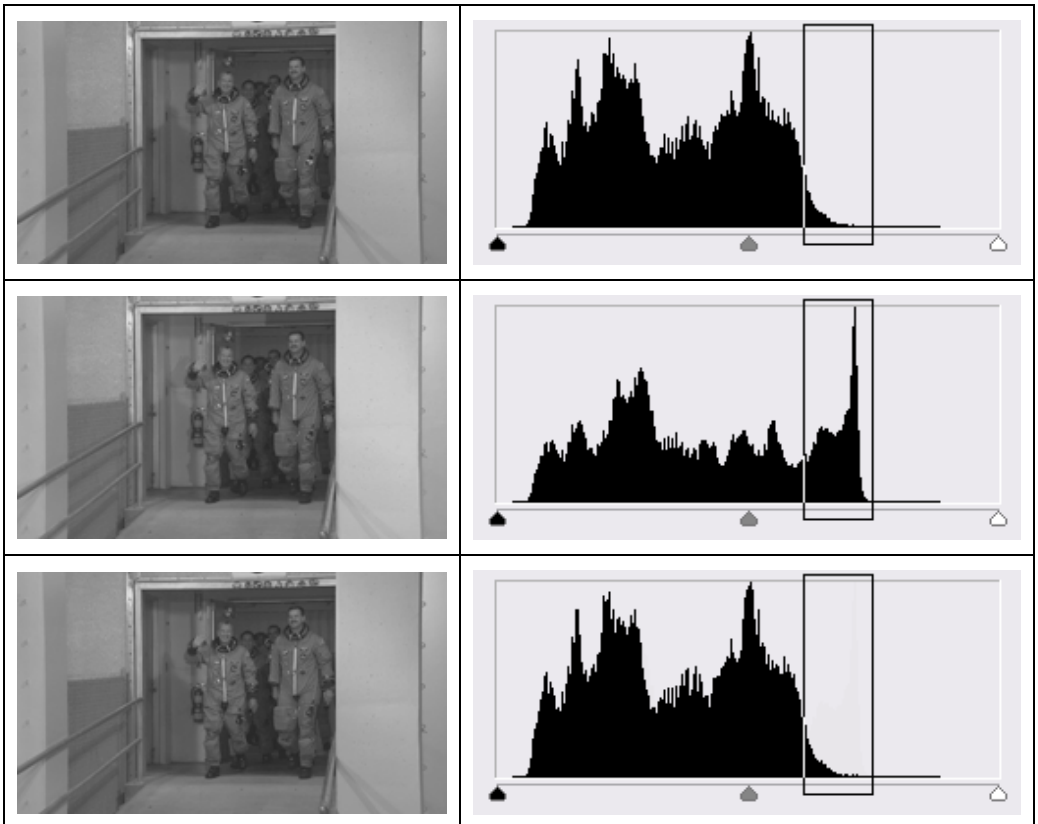


Fig.14 *Sequence 'crew' – frames 60 to 62 – local flash detected in frame 61*
We can observe in the highlighted range how the flash affects the ***histogram****.*

In the case of the local flashes that we are considering, we use the bitmaps that store the flash shape in order to extract the histogram of the corresponding flash affected areas ***[Fig.15]***. With these partial histograms, a luminance range of interest is determined and applied as a mask to the full area histograms of the studied frames, including the one before the event and the one after.

The number of pixels in each one of these biased histograms will be used as a measure that will be compared to the expected flash pattern. However, although this pattern appears very clearly in some cases, when the enlightened areas are

small or they are affected by a low intensity event, the bin range we want to determine is not that obvious with a consequent difficulty to determine if the behavior during the detected frames fits the described pattern.

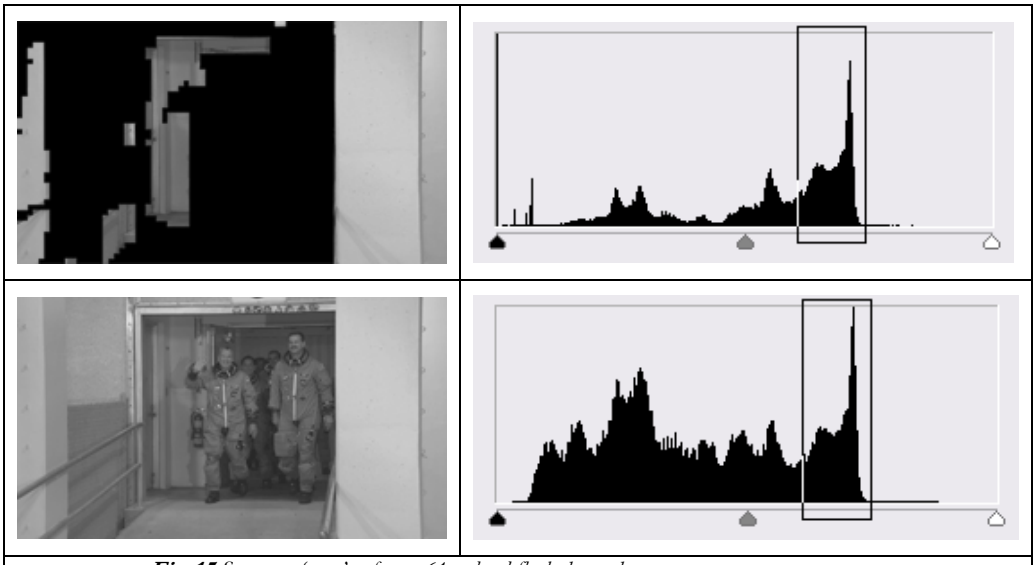
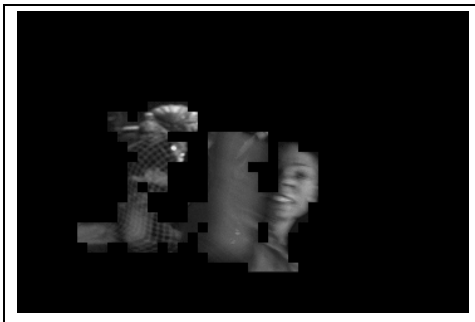


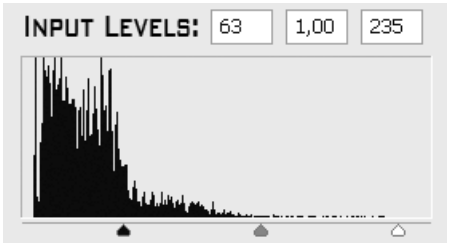
Fig.15 *Sequence 'crew' – frame 61 – local flash detected*
The histogram of the flash area helps us to determine a range that represents the flash event.

Then, some patterns are evaluated as follows:

In order to **preserve the maximum of true positives**, we will only proceed to the **removal of those events that present a pattern that is clearly not fitting the** triangular values that are **theoretically expected**. Thus, as a **first** discriminating condition, if the number of pixels in the observed range decreases when the presumed flash begins, or if this measure increases when the event arrives to its end, we consider this detection to be a false alarm and it is, therefore, discarded ***[Fig.16]***.



Flash area detected for frame 2134



From the detected flash area, the program determines that the flash luminance values range goes from 63 to 235.

Frame 2133

56567 pixels in histogram's flash range

Frame 2134 – Presumed flash

55683 pixels in histogram's flash range

Frame 2135

45419 pixels in histogram's flash range

Fig.16 *Sequence 'AvP' – frames 2133 to 2135*
A local flash is detected in frame 2134, but since the number of pixels in the detected flash values range is higher in the previous frame (56567→55683→45419), ***this frame is discarded as a light event****.*

The ***second*** method we consider to avoid some false alarms is that once we know the approximate flash affected area (or his average value if the event last several frames) and we have determined a luminance range corresponding to the

effects of that event we can use a relation between these two measures to validate or discriminate the events. We will call '***entering difference***' the difference in the number of pixels in the flash value range between the last frame before the event and the first frame of the event. Similarly, we will call '***exiting difference***' the variation in the number of pixels in the same value range between the last frame of the event and the first frame after the event. Both of these differences are presumably representing the increase of pixels in this range due to the flash and, therefore, ***should be related to the detected flash area***. If any of these two measured differences does not reach a minimum percentage of the mean flash area, then, we will consider this event as a false alarm and thus, it will be removed. However, since we do not want to be too restrictive with this step because we want to preserve as many hits as possible, a ***default*** value for this ***minimum threshold*** is set to ***5% of the average flash area of the event [Fig.17]***.

The ***third*** optional course of action to process the detected local flashes that have passed the previous steps has the objective of labeling those ***event candidates that fit the defined pattern with enough accuracy*** with a new index that implies a **high reliability** on that detection. In the following we will refer to the **higher number of pixels in the flash bin range** that we can measure **in any of the flash frames as (A)**, the **higher measure in the previous or posterior boundary frame as (B)** and the **lower measure in the other boundary frame as (C)**. To determine the level of accuracy we ask for some relations to be met by the measures of the number of pixels inside the range of interest, namely:

Frame 1014

3153 pixels in histogram's flash range

Frame 1015 – Presumed flash ***starts here***

3598 pixels in histogram's flash range

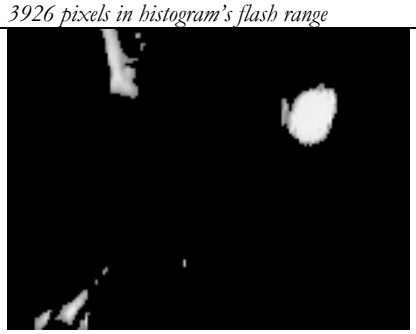
Frame 1016 – Presumed flash

Frame 1017 – Presumed flash

Frame 1018 – Presumed flash ***ends*** *here*

3926 pixels in histogram's flash range

Frame 1019



3913 pixels in histogram's flash range

Fig.17 *Sequence 'chris' – A local flash is detected from frames 1015-1018. Average flash area =* ***7376.***
The ***entering difference*** *is 3598 – 3153 = 445 (only 4.878% of the average flash area).*
The ***exiting difference*** *is 3926 – 3913 = 13 (only 0.176% of the average flash area).*

i) The number of pixels in the studied bin range of the histograms must be similar in the 2 frames surrounding the flash event. With that purpose, we set an upper threshold for the difference between both measures (B-C) (typically 20% of the maximum of the two values) ***[Form.1]***. This condition ensures that ***the scene has not substantially changed during the event*** and we can assume with enough accuracy that no other events, such as a scene cut or a dissolve, have occurred that may alter the measures ***[Fig.18-19]***.

$$(B - C) \leq 0.20 * B \quad \textbf{\textit{Form.1}}$$

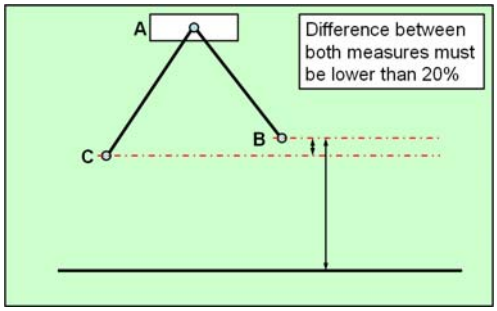


Fig.18 *Example of detected flash that would pass the test in formula* ***[Form.1]***.

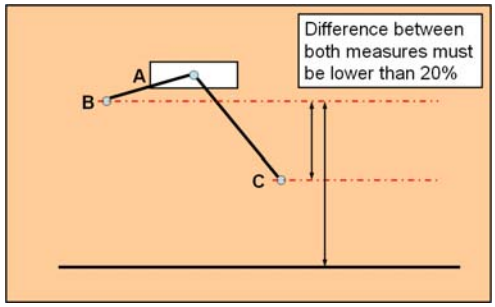


Fig.19 *Example of detected flash that would* ***not*** *pass the test in formula* ***[Form.1]***.

ii) For the second condition we define a new measure that exploits the difference between the frames that are considered to be outside the flash (surrounding it) and the frames that are inside the candidate (those that have been previously detected). We want to **register a significant increase in the number of 'flash pixels'** that ensures with no doubt that what is taking place in this scene is not a natural variation in the histogram, but a significant event that fits the description we have constructed for a light event. Thus, considering the difference between the average value of the outside frames ((B+C)/2) and the higher value reached within the flash frames (A) ***[Form.2]***, we will determine if it represents a significant increase comparing to the surrounding frames. With that purpose we will specify that it has to be higher than the difference between the outside values (B-C) (Typically at

least 10 times that difference) ***[Fig.20-21]***. However, when that difference is too small, this condition becomes insufficient and we also set a **minimum relative threshold** ***[Form.3]*** for that difference (typically an increase of at least 25% of the higher value in the surrounding frames) that proves to be enough to ensure a significant variation in these cases ***[Fig.22-23]***.

$A-\frac{(B+C)}{2}\geq 10*(B-C)$ ***Form.2***		$A-\frac{(B+C)}{2}\geq 0.25*B$ ***Form.3***

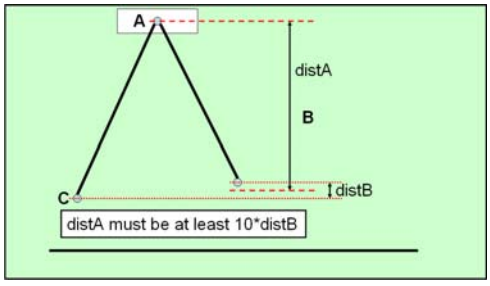


Fig.20 *Example of detected flash that would pass the test in formula* ***[Form.2]***.

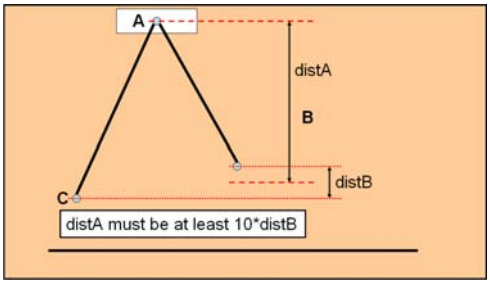


Fig.21 *Example of detected flash that would not pass the test in formula* ***[Form.2]***.

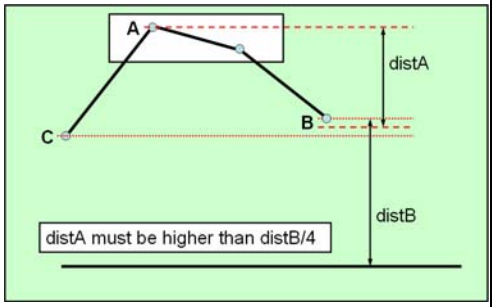


Fig.22 *Example of detected flash that would pass the test in formula* ***[Form.3]***.

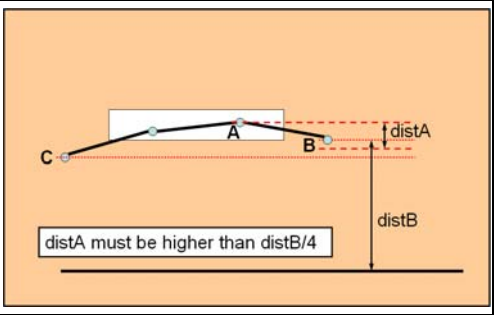


Fig.23 *Example of detected flash that would not pass the test in formula* ***[Form.3]***.

If the studied candidate fulfills the above conditions, ***we consider its detection as highly reliable*** and label it as a flash that we are very confident that is a correct detection and that would rarely be removed with future discrimination processes ***[Fig.24-25]***. This new condition will be used to skip other techniques that would only confirm the decision with a high additional computational cost. On the other side, if these tests do not ensure that the evaluated event is a flash, we will keep it as a regular flash but it may be subject to further tests.

Frame 169

C = 17297 pixels

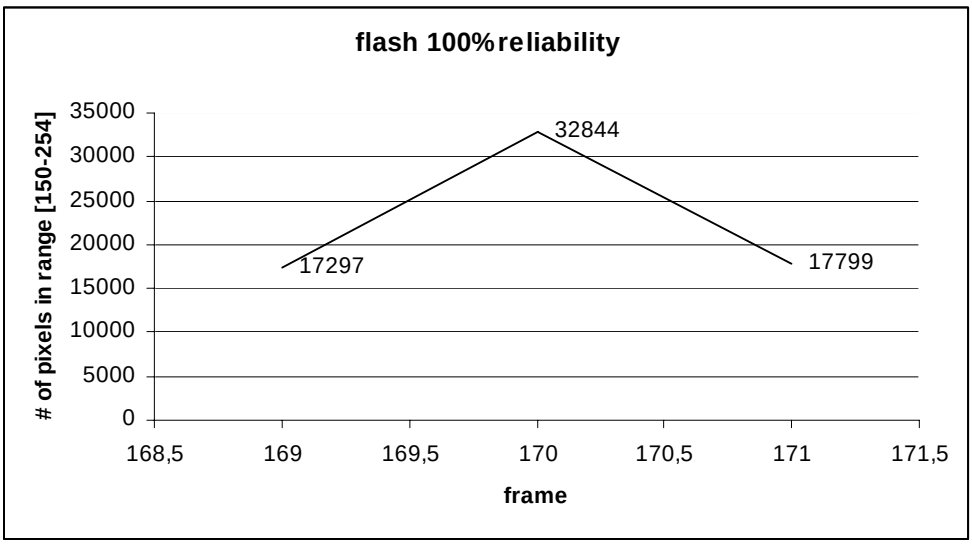
Frame 170 – Flash

A = 32844 pixels

Frame 171

B = 17799 pixels

Fig.24 *Sequence 'victoria's secret' – frames 169 to 171*
Despite the high amount of white particles moving all around the image, the local flash is detected in frame 170 within high reliability conditions. Conditions for highly reliable candidate are met.



$(B-C) \leq 0.20 * B$ ***[Form.1]***	$502 \leq 3560$
$A - \frac{(B+C)}{2} \geq 10 * (B-C)$ ***[Form.2]***	$15296 \geq 5020$
$A - \frac{(B+C)}{2} \geq 0.25 * B$ ***[Form.3]***	$15296 \geq 4450$

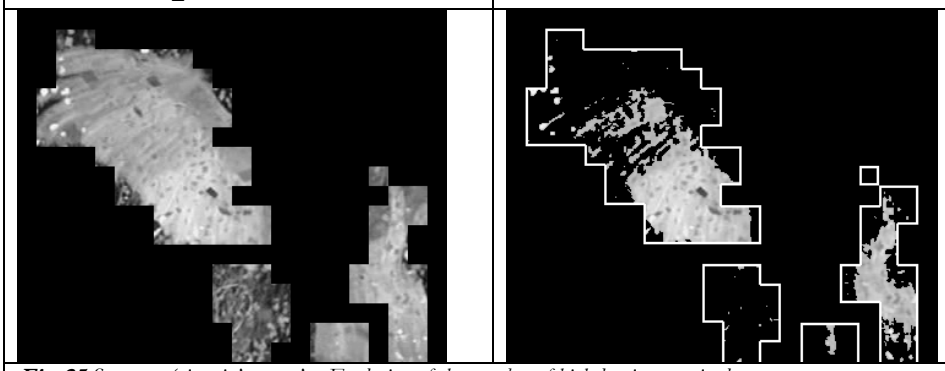


Fig.25 *Sequence 'victoria's secret' – Evolution of the number of high luminance pixels. We can appreciate in this figure how the number of pixels within the highest luminance range increases in the frame affected by the flash and decreases afterwards to normal values.*

Having introduced the concept of high reliability detections for local flashes, we have also applied this concept to the global flashes. However, since in the global flash detections the determination of the flash area shape is not always required and, by definition, the whole frame is presumed to be the affected area, we are not able to determine a range of luminance values corresponding to the flash affected area. But then, due to this same definition we are using, we can use an approximation and apply the same requirements as with the local flashes

[Form.1-3] using the frames' mean luminance value instead of the number of pixels. In this case, we are not relying in the variation of pixels in a determined value range, but in a considerable general increase and decrease of the average luminance that allows us to ***classify a global flash as highly reliable [Fig.26]***.

However, despite this additional classification, regular global flashes detected also present, as mentioned earlier, a high precision rate.

Frame – 182 Mean Luma Value = 37

Frame – 183 Mean Luma Value = 123

Frame – 184 Mean Luma Value = 42

Frame – 185 Mean Luma Value = 36

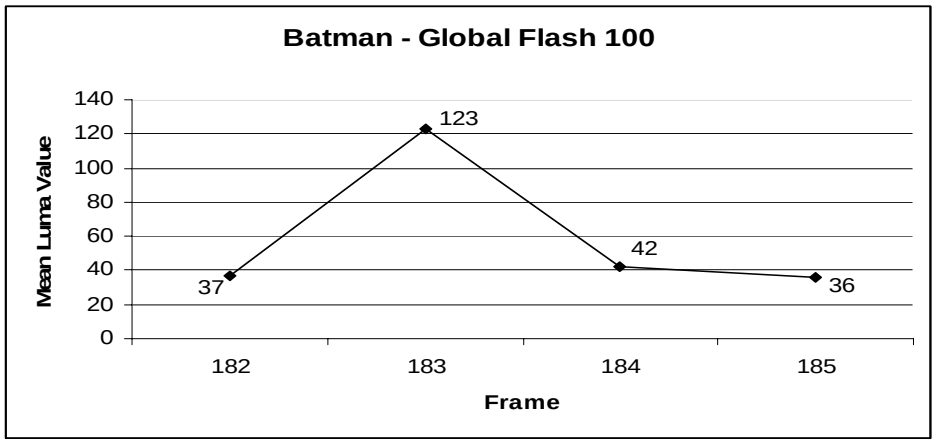


Fig.26 *Sequence 'batman2' – frames 182 to 185*
A global flash has been detected in frame 183. Also a local flash is detected in frame 184 and is merged with the previous frame and considered as part of a 2 frames long global flash. This set of frames fulfils the conditions imposed to consider it a highly reliable global flash and, consequently, it is indexed with the corresponding label.

The **fourth** and last optional refinement method that we have implemented is based on **motion estimation and prediction error** techniques and will try to exploit the difference in the content correspondence that should be found in the non-affected areas, which is presumed to be predictable with a considerable accuracy, against the flash affected areas, theoretically more unpredictable since there are no previous accurate references.

As with the histogram technique, although it could be extended to evaluate global flashes, this technique will only be applied to the local flash discrimination. This decision does not only come from the fact that the first step of the detector proved to perform much more precisely with the global flash detection, but also and more importantly, because of the high computational cost of such techniques. This complexity also leads us to perform this test at the very last steps to avoid processing the data of the already rejected local flash candidates and of those labeled with a high reliable index.

The premise behind this method is to ***reveal the particular false positive cases where the algorithm is triggered by a clear object that enters or exits the scene*** and discard them from our detected set. When evaluating a set of frames in a false positive, a different prediction error will arose when using as reference frame the one where the object is in the scene or the frame where the object does not appear in the scene (both boundary frames of the detected set of frames). Moreover, the flash area detected corresponds to that particular object that is moving into (or out of) the scene's view. The measure we use to detect that behavior is the distortion error of the predicted frame, and the procedure is as follows:

The last frame before the flash (FA) is used to predict, with motion estimation and motion compensation techniques, the first frame of the local flash (FB).

During this process, a bi-dimensional array is created that stores the distortion of this predicted frame (FB').

In an analogous way, a prediction distortion is stored for the predicted frame (FC') that results of predicting the last frame of the flash set (FC) using the first frame after the flash as reference frame (FD). In the case that the detected event lasts only for a single frame, the (FB) and (FC) frames will be the same, but this will not affect the rest of the procedure since the (FB') and (FC') frames will be obtained with different references.

Using the area mask that the detector generated for frames FB and FC, we calculate the average distortion of both predictions in the presumed flash area ***[Fig.27]***.

If either of those two average distortions is lower than a certain threshold, we will decide that a ***very accurate prediction of the scene can be made in one of the two directions*** and, therefore, that ***no substantial light variation*** is present in that frame. If this is the case, the set of frames will be considered to be a false alarm and then rejected. The value for this threshold has been ***typically set to 3*** with satisfactory results.

Also, if the two distortion measures are too different, we will consider this detection to be a false alarm, since that would mean that one of the frames had a much better reference for the flash area than the other ***[Fig.28]***. This better prediction in one direction, although not being significantly good, suggests that at least some part of the conflictive element is present in that frame and, therefore, it is probably not a light event. An optimal value for this difference is still to be determined for this customizable parameter, but a ***default value has been typically set to 0.20***.

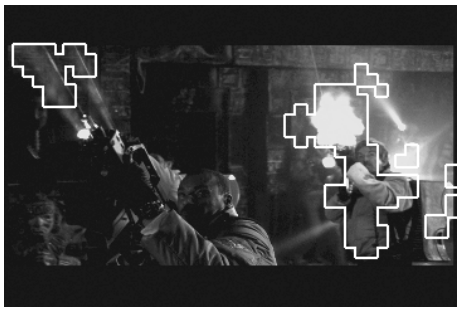
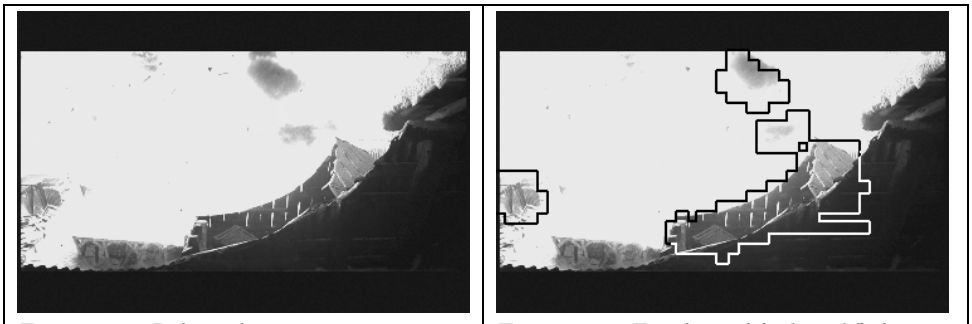
	Frame 880 (FA) is used as reference frame to predict frame **881 (FB)**.
	When we use **Motion Compensation** to predict this frame detected as flash, the resulting average distortion in the flash area is **65** when doing **forward prediction** (predict FB from FA) and **74** when doing **backward prediction** (predict FC from FD)
	Frame 882 (FD) is used as reference frame to predict frame **881 (FC)**.

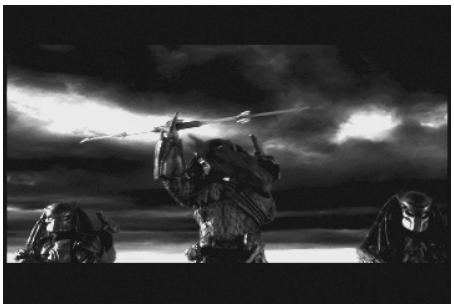
Fig.27 *Sequence 'AvP' – frames 880 to 882*
If predicting frame 881 from 880 and 882 results in similar distortion in the flash candidate area, it indicates that the content in the image after the event is similar to what was there before .



Frame 2523 – Reference frame

Frame 2524 – First frame of the detected flash

Frame 2526 – Last frame of the detected flash



Frame 2527 – Reference frame

Fig.28 *– Sequence 'AvP' – frames 2523 to 2526*
Although there is a light increase in the scene and it lasts for only a few frames, the reason for this short span is a scene change. This is reflected in an average prediction distortion of 58 when using as reference a frame of the same shot and an error of 97 when the prediction is made in the reverse direction. The content of the image has changed.

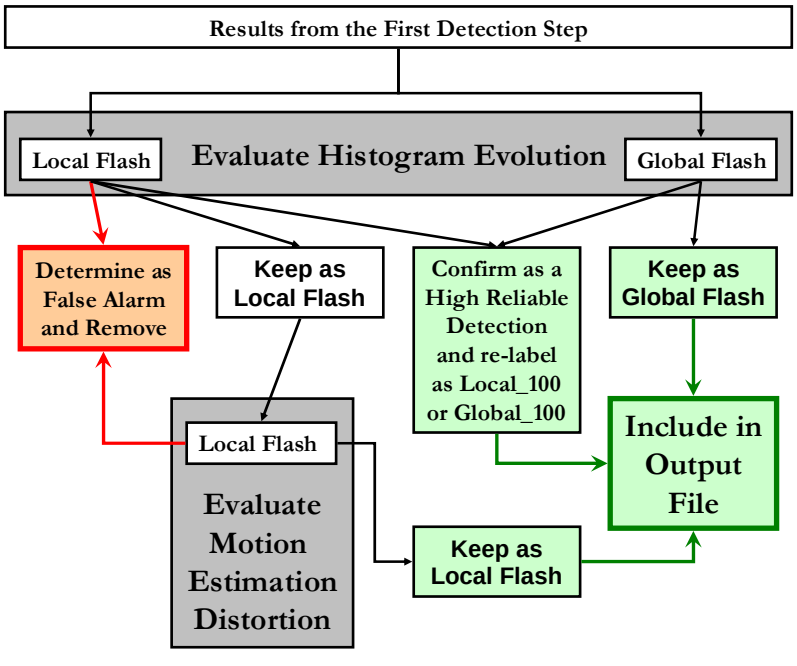


Fig.29 *– This scheme represents the discrimination process performed by the second Decision Step.*

6.6. Generate output

Then, the selected **scale** is checked to **generate the required binary maps *[Fig.30]*** and include them in the output videos used to store the markers. Also an output .scn file is generated ***[Fig.31]*** using a format compatible with the encoder that will use this information and that includes the first and last frames of the light events, the type and, if there is one, the ".yuv" file containing the binary markers.

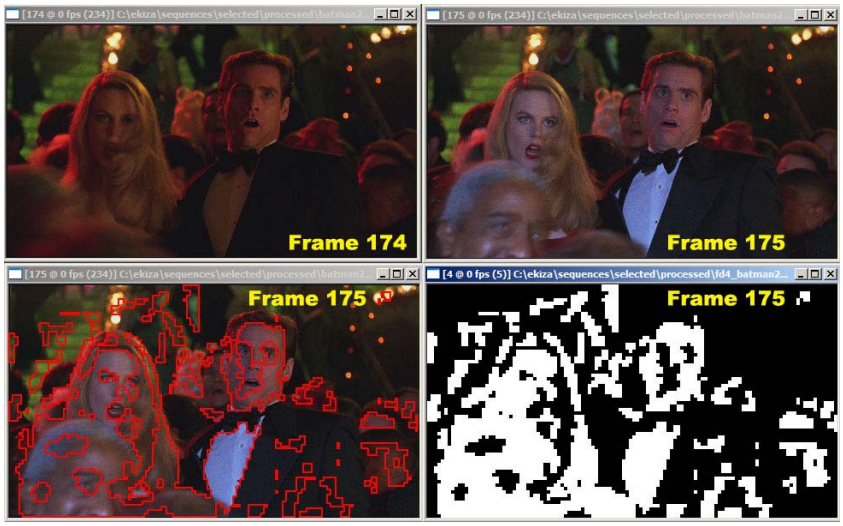


Fig.30 *On bottom right image we observe how the bitmap obtained represents the area affected by the flash.*

```
Frames = 111 - 112    Type                    =                FlashLocal
Map=fd4_hc_batman2_120x68_24p_fsh111_112.yuv
Frames = 115 - 116    Type = Cut
Frames = 119 - 119    Type                    =                FlashLocal
Map=fd4_hc_batman2_120x68_24p_fsh119_119.yuv
…
Frames = 157 - 158    Type                    =                FlashLocal
Map=fd4_hc_batman2_120x68_24p_fsh157_158.yuv
Frames = 159 - 160    Type = FlashGlobal
Frames = 164 - 167    Type                    =                FlashLocal
Map=fd4_hc_batman2_120x68_24p_fsh164_167.yuv
```

Fig.31 *Output format of the '.scn' files. When a 'global flash' or a 'shot cut' are detected no additional information is needed. In case of detecting a 'local flash' the name of the file containing the bitmaps is specified.*

7. Results

7.1. Indexing Results

Description of Selected Sequences used for testing: The following list describes the sequences used to test the performance of the designed and implemented detector in its 2 steps. The selected sequences are all in format ***YUV:420*** and the detector has been specifically designed to process input sequences in this same format.

- AlienVsPredator (avp): Action movie trailer, with very short scenes, a very intense motion through all the sequence and a wide range of transition effects (fade in, fade out, cut to black, zoom out, lateral span, dissolves, etc…). In addition to the motion, this sequence includes many flash-like artifacts such as electrical flash lights, laser beams, gunshots, explosions of light, moving reflective surfaces and many others. This is a fairly complicated sequence. [Total Frames = 2864] [Resolution = 720x480]

- Batman2: This is a clip of an action movie that includes moderate to strong motion. In the first half of the sequence many global flashes are present that cause severe problems to the scene cut detectors. In the second half of the video, there is a lot more movement and the light events consist mainly of light explosions and gunshots. [Total Frames = 234] [Resolution = 1920x1088]

- Britney: This is a music video clip that includes flashes in both steady scenes and scenes with moderate to strong motion. In the majority of the sequence, the flash sources are photographic flashes. Towards the end, though, there are many red, white and blue colored flashes due to police lights. [Total Frames = 5913] [Resolution = 640x352]

- Christina: Excerpt of a music video clip that combines long shots with few motion and very short shots with intense scene and camera motion. The transition between shots is usually a shot cut but in a few cases a dissolve effect appears between scenes. The light events in this sequence are mainly moving light sources and multiple police vehicle lights that, intermittently, enlighten the scene or produce a dazzling reflection with a wide range of intensities and sizes and, in some segments, present as a burst and/or with camera motion. [Total Frames = 3000] [Resolution = 320x240]

- Crew: This is a sequence of some people entering the scene through an open door, with low intensity motion, and with a lateral span during the last frames. Along all the sequence, multiple camera shots illuminate both characters and background with different intensities. [Total Frames = 600] [Resolution = 1280x720]

- Life2: The sequence starts with the camera turning around a person resulting in a combination of panning, zoom in and motion. During this scene, there are first indirect flashes that turn into direct flashes. Then there are some very short scenes combined with global flashes. In the last scene, there is a source of light that appears in the scene and remains there with some occasional glares. [Total Frames = 564] [Resolution = 624x352]

-Photocall: Montage of different photo-call scenes with both direct and indirect flashes lighting primarily the foreground, but also the background in some frames. There is not much movement in the sequence except for a segment with a lateral pan during which flashes keep happening. Transition between scenes is always through hard cuts. [Total Frames = 358] [Resolution = 608x320]

- Victoria´s Secret: Montage with several shots of the same scene connected with dissolves. The sequence presents low intensity motion in the background,

but has plenty of small white papers falling from the top in a random pattern that occasionally reflects light. In addition, camera flashes enlighten the whole scene (foreground and background), sometimes even during a dissolve transition. [Total Frames = 498] [Resolution = 352x288]

7.2. Results definitions

- Hit (True Positive): A perfect detection of all the frames affected by a single flash.

- Miss (False Negative): Set of frames that are considered to be a single light event and none of them appears in any of the detected events.

- False Alarm (False Positive): Set of frames automatically detected as a light event that does not contain any frame that matches any manually labeled light event.

- True Negative: All those frames that have correctly not been detected because they are not involved in any light event.

- Partial Hit: A detection of a light event that either detects some of the frames affected by a light event but not all of them, or that detects a set of frames that includes not only flash affected frames but also detects other frames that are not labeled as being part of a light event. It is considered as a correction of Hits, Miss and False Alarms according to the situation.

- Precision: #TruePositives / (#TruePositives + #FalsePositives)

- Recall (a.k.a. Sensitivity): #TruePositives / (#TruePositives + #FalseNegatives)

- Specificity: #TrueNegatives / (#TrueNegatives + #FalsePositives)

- Accuracy: (#TruePositives + #TrueNegatives)/#TotalFrames

- ROC curves: These curves represent **1-Specificity** (False Positives Rate, a.k.a. FPR) in the X axis against the **Sensitivity** (True Positives Rate, a.k.a. TPR) in the Y axis. The better results are those that combine a low FPR with a high TPR.

7.3. Results obtained before false alarm removal

- Parameter configurations:

We ran the first step of the detector with 6 different parameter configurations. The **'a'**, standing for area, determined the minimum percentage of the frame that must be affected by the detected flash to be taken into consideration, while the other parameter, **'b'**, affected the threshold that determines if an event is intense enough to be considered a flash or not.

	a = 8	**a = 10**	**a = 15**
b = 4	A	B	C
b = 5	D	E	F

After some executions, it was stated that with the same targeted area, a **b=5** was working equal or better than b=4 for almost all situations. As for the area, it was obvious that the higher the requested area was, the more flashes were rejected by this restriction. We, then decided to use an area of **a=5** for we wanted to prove our detector as a reliable performer against local flashes while maintaining the accuracy.

- Avp:

TOTAL FRAMES	2864			
HIT (TP)	122		Sensitivity:	88,40%
MISS (FN)	16		1-Specificity:	3,67%
FALSE ALARMS (FP)	100		Precision:	54,95%
TRUE NEGATIVES (TN)	2626		Accuracy:	95,95%

- Batman2:

TOTAL FRAMES	234		
HIT (TP)	19	Sensitivity:	82,61%
MISS (FN)	4	1-Specificity:	3,79%
FALSE ALARMS (FP)	8	Precision:	70,37%
TRUE NEGATIVES (TN)	203	Accuracy:	94,87%

- Britney:

TOTAL FRAMES	5913		
HIT (TP)	229	Sensitivity:	90,16%
MISS (FN)	25	1-Specificity:	1,75%
FALSE ALARMS (FP)	99	Precision:	69,82%
TRUE NEGATIVES (TN)	5560	Accuracy:	98,90%

- Christina:

TOTAL FRAMES	3000		
HIT (TP)	59	Sensitivity:	80,82%
MISS (FN)	14	1-Specificity:	1,61%
FALSE ALARMS (FP)	47	Precision:	55,66%
TRUE NEGATIVES (TN)	2880	Accuracy:	97,97%

- Crew:

TOTAL FRAMES	600		
HIT (TP)	76	Sensitivity:	58,91%
MISS (FN)	53	1-Specificity:	0,00%
FALSE ALARMS (FP)	0	Precision:	100,00%
TRUE NEGATIVES (TN)	471	Accuracy:	91,17%

- Life2:

TOTAL FRAMES	564			
HIT (TP)	7		Sensitivity:	87,50%
MISS (FN)	1		1-Specificity:	0,00%
FALSE ALARMS (FP)	0		Precision:	100,00%
TRUE NEGATIVES (TN)	556		Accuracy:	99,82%

- Photocall:

TOTAL FRAMES	358			
HIT (TP)	40		Sensitivity:	81,63%
MISS (FN)	9		1-Specificity:	0,65%
FALSE ALARMS (FP)	2		Precision:	95,24%
TRUE NEGATIVES (TN)	307		Accuracy:	96,93%

- Victoria´s Secret:

TOTAL FRAMES	498			
HIT (TP)	20		Sensitivity:	95,24%
MISS (FN)	1		1-Specificity:	0,84%
FALSE ALARMS (FP)	4		Precision:	83,33%
TRUE NEGATIVES (TN)	473		Accuracy:	99,00%

It is expected that the high number of false alarms obtained in some of the sequences will be considerably reduced after the second processing step with a potential increase in the precision. Also, a side effect of the second step may be a slight increase of the amount of missed frames.

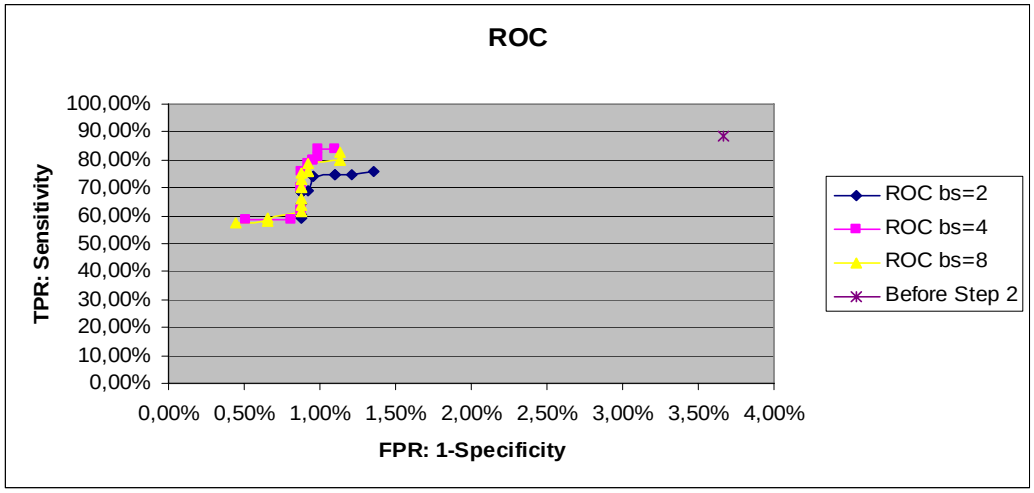
7.4. Results obtained after false alarm removal

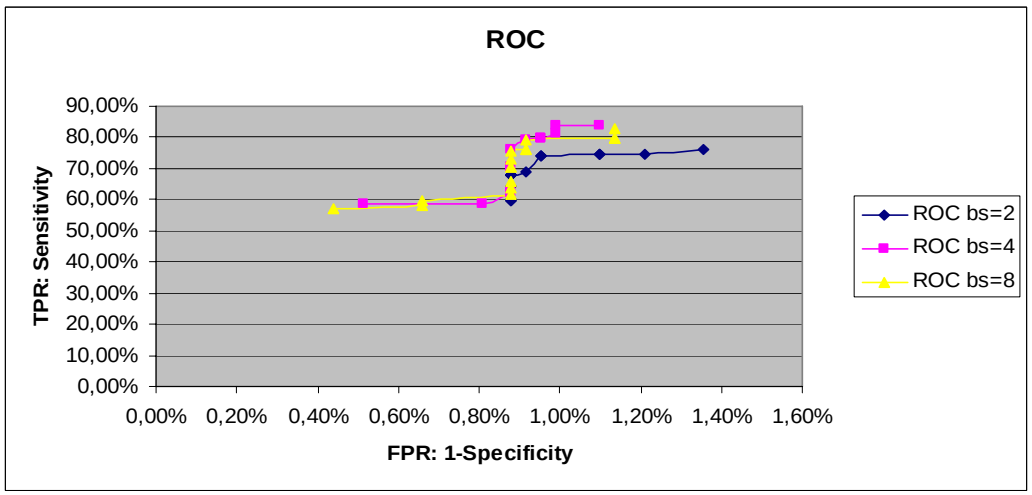
- Parameter configurations: For the second part of the decoder, the one that aims to an exhaustive false alarm removal, we will be testing how the detector

performs while modifying both parameters **'bs'**, which determines the block size used in the motion estimation step, and **'t'**, which defines a threshold to discriminate false alarms. We have run, based on our previous results that used b=5 a=5 in the first step, a set of executions with parameters **bs** = [2, 4 and 8] and **t** = [0'10 ... 0'40] with a step value of 0'02.

In many cases, different parameter pairs provide the same results. If multiple combinations return the best detection rates, we will present the results prioritizing the combination with the highest **bs** value and then the one with the lowest **t** value.

- AvP:





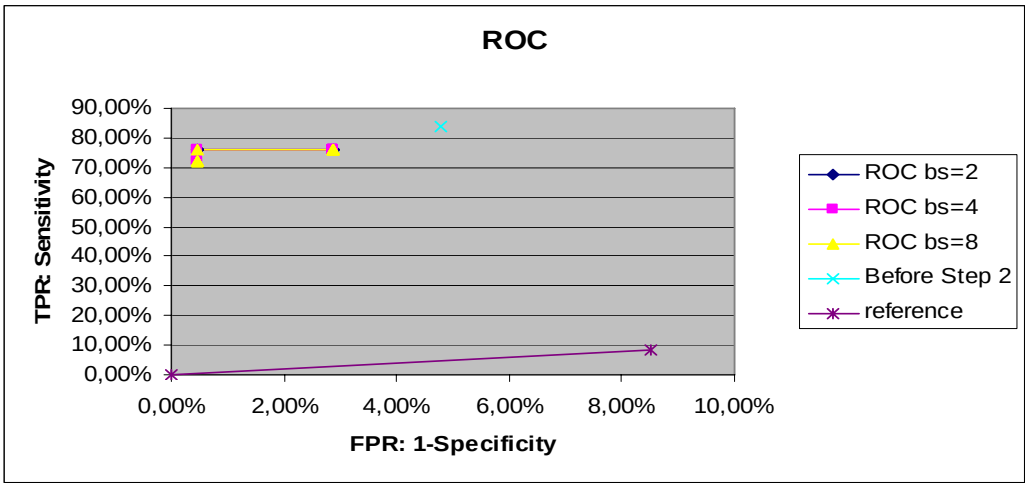
Best result : bs=4 t=0,38

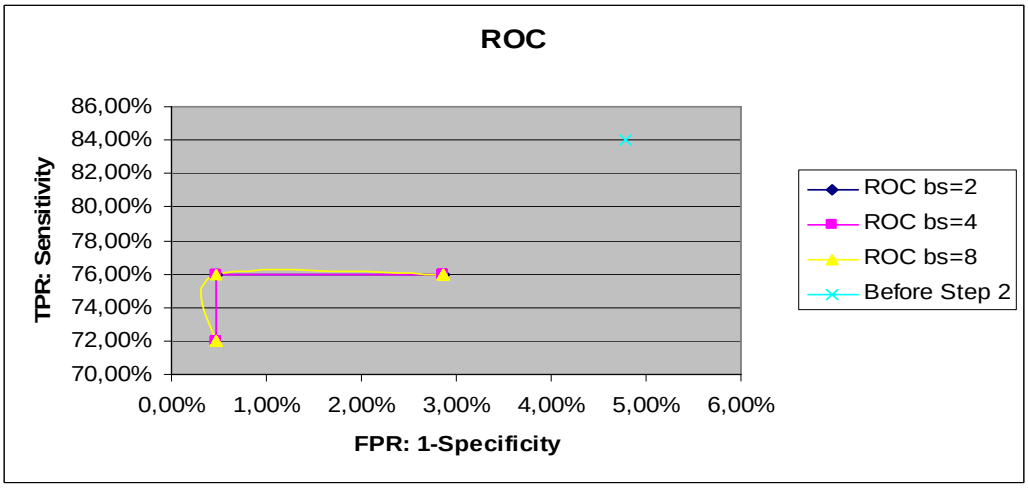
TOTAL FRAMES	2864			
HIT (TP)	116		Sensitivity:	84,05%
MISS (FN)	22		1-Specificity:	0,99%
FALSE ALARMS (FP)	27		Precision:	81,12%
TRUE NEGATIVES (TN)	2699		Accuracy:	98,29%

Sensitivity (Recall): Has decreased 4,35 points (trade off)
False Positives Rate: Has decreased 2,68 points (improvement)
Precision: Has increased 26,17 points (improvement)
Accuracy: Has increased 2,34 points to a total of 98,3% (improvement)

The second step reduces the amount of false positives in almost 75% (73 false positives removed) with an additional loss of less than 4,5% of true positives (6). This contributes to a substantial increase in the precision and boosts the already high accuracy to an even better value, considering the considerable number of frames in this sequence. **92,4% of the detections removed during the second step were false positives.**

- Batman2





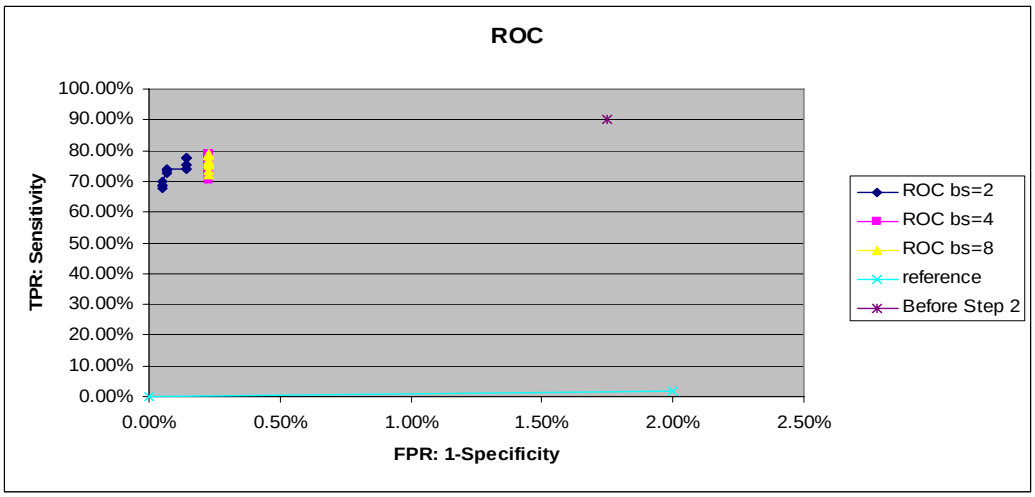
Best result : bs=4 t=0,22

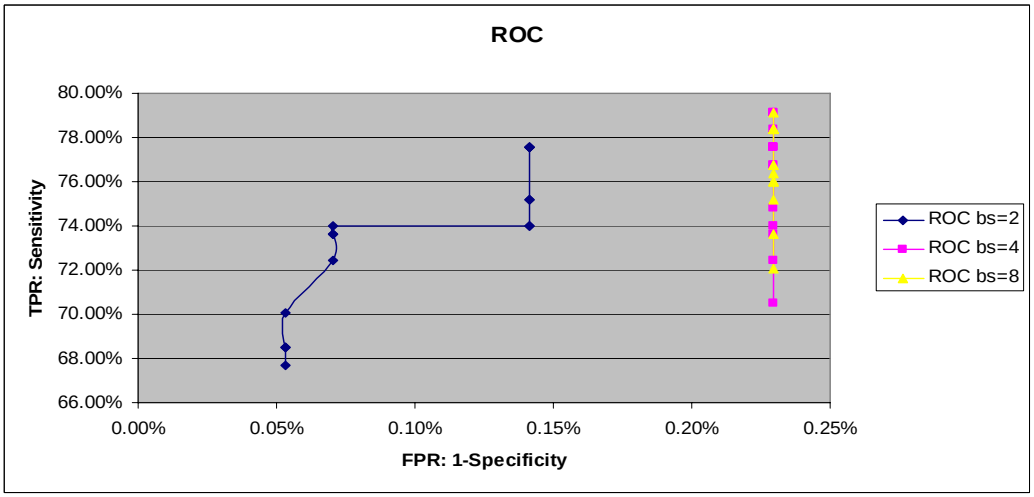
TOTAL FRAMES	234			
HIT (TP)	19		Sensitivity:	76,00%
MISS (FN)	6		1-Specificity:	0,48%
FALSE ALARMS (FP)	1		Precision:	95,00%
TRUE NEGATIVES (TN)	208		Accuracy:	97,01%

Sensitivity (Recall): Has decreased 6,61 points (trade off)
False Positives Rate: Has decreased 3,31 points (improvement)
Precision: Has increased 24,63 points (improvement)
Accuracy: Has increased 2,14 points (improvement)

The second step reduced the amount of false positives in 87,5% (7) with an additional loss of 8% true positives (2). This contributes to a substantial increase in the precision and boosts the already high accuracy to an even better value. **77,78% of the positives removed during the second step were false positives.**

- Britney





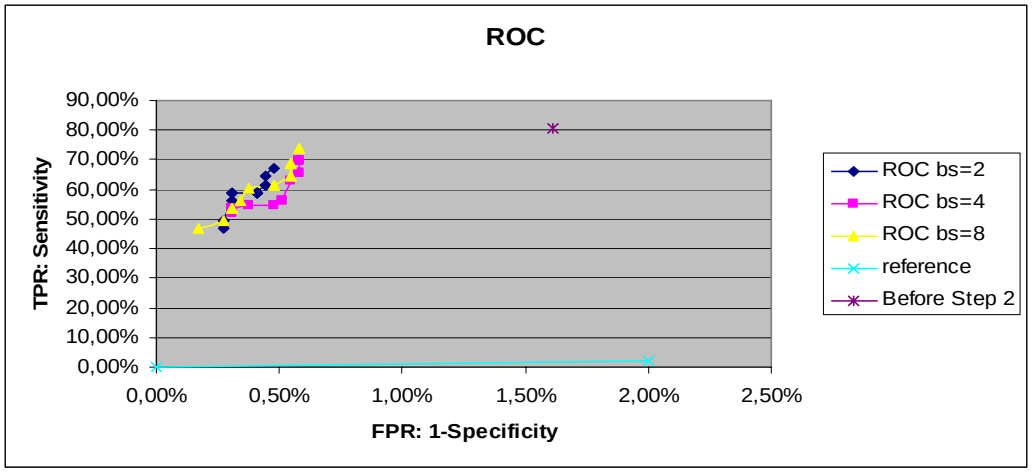
Best result : bs=8 t=0,40

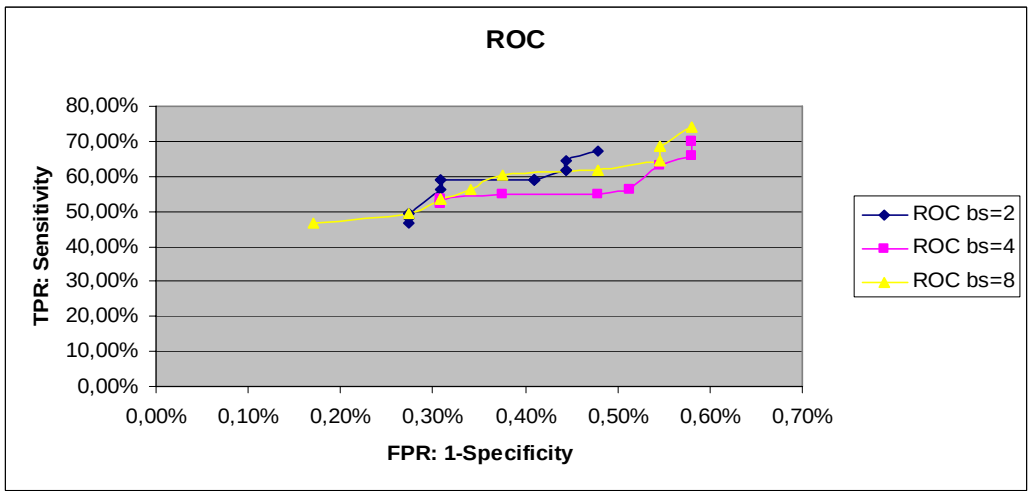
TOTAL FRAMES	5913			
HIT (TP)	201		Sensitivity:	79,13%
MISS (FN)	53		1-Specificity:	0,23%
FALSE ALARMS (FP)	13		Precision:	93,93%
TRUE NEGATIVES (TN)	5646		Accuracy:	98,88%

Sensitivity (Recall): Has decreased 11,03 points (trade off)
False Positives Rate: Has decreased 1,52 points (improvement)
Precision: Has increased 24,11 points (improvement)
Accuracy: Has increased 0,98 points (improvement)

The second step reduced the amount of false positives in 86,87% (86) with an additional loss of 11% true positives (28). This contributes to a substantial increase in the precision and boosts the already high accuracy to an even better value. **75,44% of the positives removed during the second step were false positives.**

- Christina





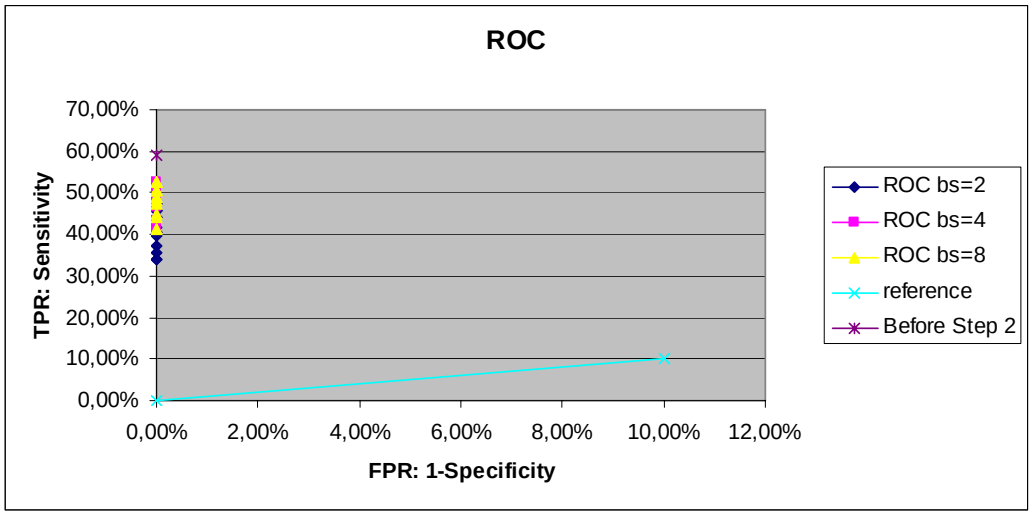
Best result : bs=8 t=0,38

TOTAL FRAMES	3000			
HIT (TP)	54		Sensitivity:	73,97%
MISS (FN)	19		1-Specificity:	0,58%
FALSE ALARMS (FP)	17		Precision:	76,06%
TRUE NEGATIVES (TN)	2910		Accuracy:	98,80%

Sensitivity (Recall): Has decreased 6,85 points (trade off)
False Positives Rate: Has decreased 1,03 points (improvement)
Precision: Has increased 20,4 points (improvement)
Accuracy: Has increased 0,83 points to a total of 98,73% (improvement)

The second step reduced the amount of false positives in almost 64% (30) with an additional loss of 6,85% true positives (5). This contributes to a substantial increase in the precision and boosts the already high accuracy to an even better value, considering the considerable number of frames in this sequence. **85,7% of the detections removed during the second step were false positives.**

- Crew:



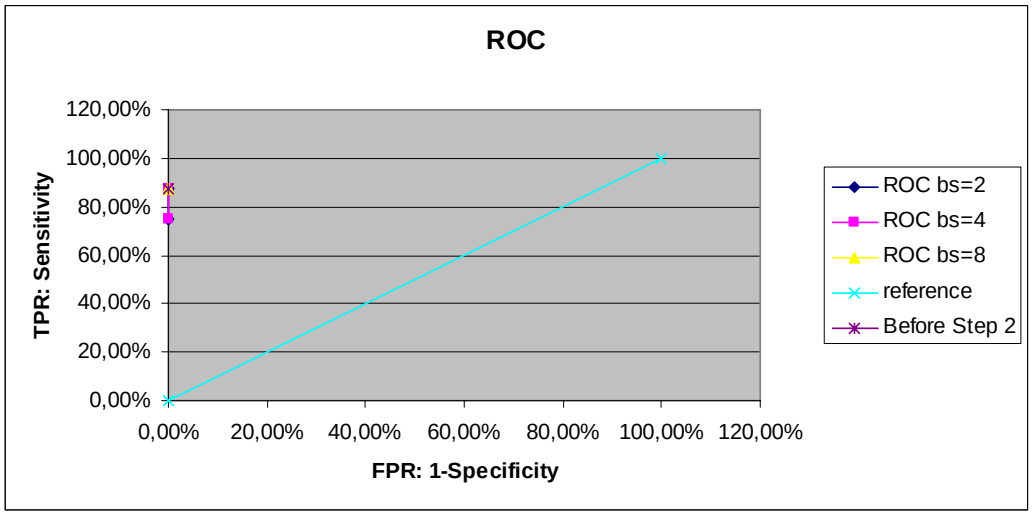
Best result : bs=8 t=0,34

TOTAL FRAMES	600			
HIT (TP)	68		Sensitivity:	52,71%
MISS (FN)	61		1-Specificity:	0,00%
FALSE ALARMS (FP)	0		Precision:	100,00%
TRUE NEGATIVES (TN)	471		Accuracy:	89,83%

Sensitivity (Recall): Has decreased 6,2 points (loss)
False Positives Rate: Remains constant at 0,00%
Precision: Remains constant at 100%
Accuracy: Has decreased 1,34 points (loss)

A sequence that already presented 0 false alarms after the first processing step can never be improved by the removal of false alarms, which is the essence of the second step. . 8 true positives were lost in this process (6,2%). Nevertheless, **both the sensitivity and accuracy only suffer of controlled variations**

- Life:



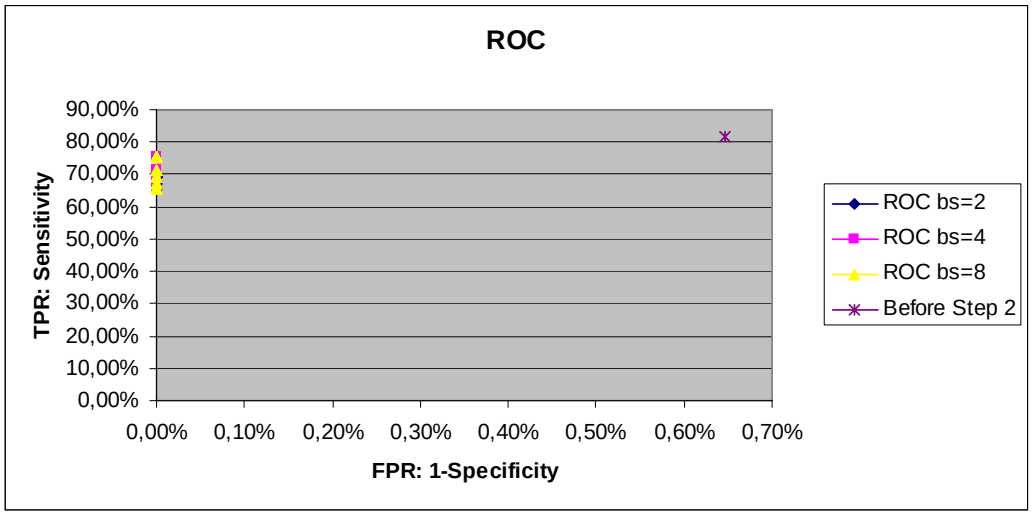
Best result : bs=8 t=0,10

TOTAL FRAMES	564			
HIT (TP)	7		Sensitivity:	87,50%
MISS (FN)	1		1-Specificity:	0,00%
FALSE ALARMS (FP)	0		Precision:	100,00%
TRUE NEGATIVES (TN)	556		Accuracy:	99,82%

Sensitivity (Recall): Remains constant at 87,50%
False Positives Rate: Remains constant at 0,00%
Precision: Remains constant at 100%
Accuracy: Remains constant at 99,82%

A sequence that already presented 0 false alarms after the first processing step can never be improved by the removal of false alarms, which is the essence of the second step. Nevertheless, for most of the parameter combinations used in the second step, hits and missed flashes remain constant with no variation in the result. **There has not been any negative impact caused by the second processing step.**

- Photocall:



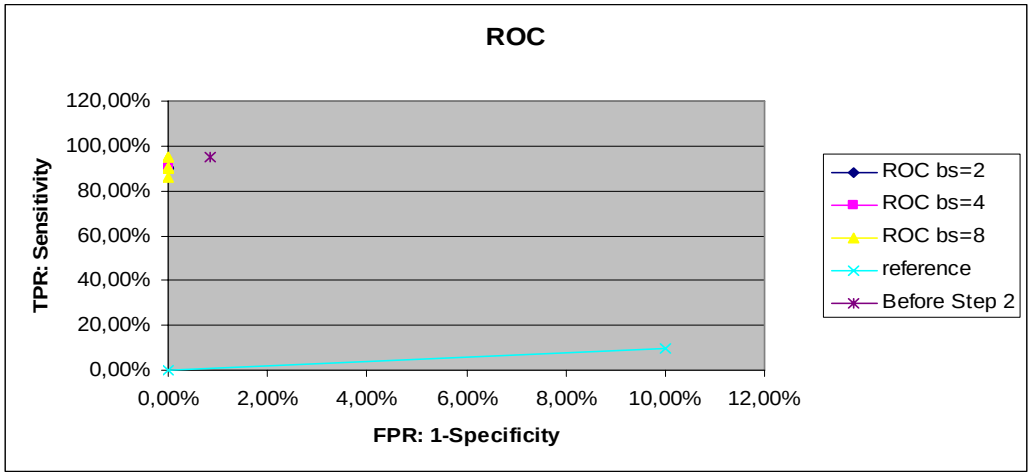
Best result : bs=8 t=0,28

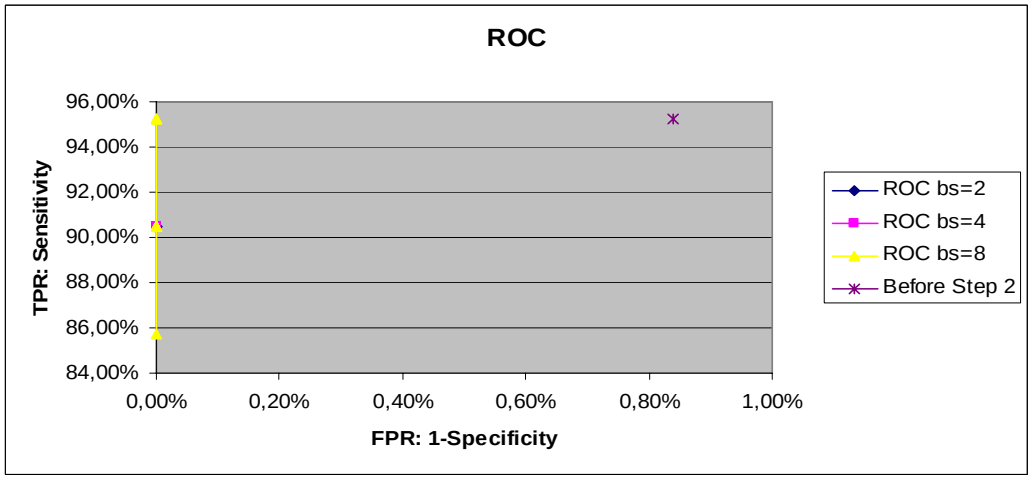
TOTAL FRAMES	358			
HIT (TP)	37		Sensitivity:	75,51%
MISS (FN)	12		1-Specificity:	0,00%
FALSE ALARMS (FP)	0		Precision:	100,00%
TRUE NEGATIVES (TN)	309		Accuracy:	96,65%

Sensitivity (Recall): Has decreased 6,12 points (trade off)
False Positives Rate: Has decreased 0,65 points to a total of 0,00% (improvement)
Precision: Has increased 4,76 points to a total of 100,00% (improvement)
Accuracy: Has decreased 0,28 points (trade off)

The second step removes a 100% of the false positives (2) with an additional loss of only 6,12% true positives (3). This contributes to reach a perfect specificity and precision rates and even increase an excellent accuracy value with a minimal loss in the accuracy. For a sequence that presented a considerably high accuracy after the first processing step, there is an acceptable loss in sensitivity and accuracy to reach a 0 false alarms detection after the second step. **Both the sensitivity and accuracy only suffer of controlled variations.**

- Victoria´s Secret:





Best result : bs=8 t=0,36

TOTAL FRAMES	498			
HIT (TP)	20		Sensitivity:	95,24%
MISS (FN)	1		1-Specificity:	0,00%
FALSE ALARMS (FP)	0		Precision:	100,00%
TRUE NEGATIVES (TN)	477		Accuracy:	99,80%

Sensitivity (Recall): Remains constant
False Positives Rate: Has decreased 0,84 points down to 0 false alarms (improvement)
Precision: Has increased 16,66 points for a total of 100% (improvement)
Accuracy: Has increased 0,80 points for a total of 98,80% (improvement)

The second step removes a 100% of the false positives (4) while it keeps all of the true positives. This contributes to reach a perfect precision rate and even increase an excellent accuracy value. **100% of the positives removed during the second step were false positives.**

7.5. Average results and visual evaluation

After 1st Step (before false alarm removal)

TOTAL FRAMES	14031			
HIT (TP)	572		Sensitivity:	82,30%
MISS (FN)	123		1-Specificity:	1,95%
FALSE ALARMS (FP)	260		Precision:	68,76%
TRUE NEGATIVES (TN)	13076		Accuracy:	97,27%

After 2nd Step (after false alarm removal)

TOTAL FRAMES	14031			
HIT (TP)	522		Sensitivity:	74,89%
MISS (FN)	175		1-Specificity:	0,43%
FALSE ALARMS (FP)	58		Precision:	90,00%
TRUE NEGATIVES (TN)	13276		Accuracy:	98,34%

Sensitivity (Recall): Has decreased 7,41 points (trade off)

False Positives Rate: Has decreased 1,51 points to a total of 0,43% (improvement)

Precision: Has increased 21,25 points to a total of 90% (improvement)

Accuracy: Has increased 1,07 points for a total of 98,34% (improvement)

The second step **reduced the amount of false positives in 77,69%** (202) with an additional loss of 7,41% true positives (50). This contributes to a substantial increase in the precision and boosts the accuracy to 98,34%. **79,53% of the positives removed during the second step were false positives.**

7.6. Enconding Results: Theoretical Improvements

The knowledge of the location of a light event in a video sequence should open the path for many processing methods designed to improve the efficiency of the encoding process in different ways.

The first and more obvious improvement is that, if we are absolutely certain of the existence of a flash in the frames that a shot cut detector has found a change of scene (or even some consecutive changes during a short number of frames), we can correct this in order to obtain an accurate segmentation of the sequence in scenes.

This correct determination of the boundaries will allow the selection of a more efficient set of GOP's that may take more profit of the redundancy within the shot. For example, a small set of frames that suffer a temporal alteration, that steps back returning to content similar to the found in the previous frames may indicate that using the last frame before the event as a reference frame for the posterior predictions has the potential to improve the encoding results.

As well, the results of this study can help to identify frames that, since they present an unexpected and punctual change in the scene, should not be used as reference frame or, if convenient, used only to predict other frames also labeled as flashes during the same shot.

Moreover, determine where (in time and space) a light variation takes place, should allow the activation of more complex techniques that can try to perform an enough accurate prediction of the frames affected by the flash with a lower encoding cost than the regular prediction and error coding. Those specific prediction techniques should even, in extreme cases with high saturation and/or content distortion, prevent encoding the corresponding frame in 'intra' mode with the corresponding increase of the bit rate.

As for the possible false alarms, since they would rarely match a shot cut detector false alarm, they should only be of concern in terms of prediction. The mentioned prediction techniques for light variation scenarios are not indiscriminately used instead of the traditional ones, but performed additionally to the regular prediction method used in the encoding process. Then, comparing the results obtained with both methods, the encoder will decide whether to still consider the frame as a particular event that requires this

particular prediction or decide instead that the usual technique performs good enough. With this behavior we ensure that, even with false alarms, we only consider the frames as flashes if it will improve the encoding. Thus, if the encoding of a wrong detection does not improve if it is treated like a flash, the only price paid is in computation time but, in the other hand, it may enable the use of these light events oriented techniques in frames that would take profit of them and that, otherwise, would not have been processed in that way.

8. Conclusions

The method we have proposed in this document introduces a direct flash detection algorithm, in contrast with the usual techniques that are applied after a shot cut detector. This approach allows a deeper study of **local light events,** those who do not affect the whole frames, while also detecting what we define as **global flashes**.

The data obtained experimenting with the test sequences shows that using a down-sampled image to detect such events usually brings up the better results when using an 8 pixels block size. This is quite convenient in the first step, where most of the frames are discarded from the detection, since it allows us to apply this technique to encoded sequences just using the dc-image during this first approximation. At this point, we have obtained an average 97,27% accuracy, with up to an 82,30% sensitivity. Considering the wide range and diversity of the effects we were aiming to detect, and the many adverse artifacts that usually take place when they happen (intense motion, short scenes with many abrupt changes, multiple light sources with different chromatic energy, etc…) this is an encouraging result.

The results of the refinement process that takes place in the second step cannot, by design, offer a better sensitivity. Instead, we focus on the resulting trade-off; a loss of 7,41% sensitivity that offers improvements in the false positives rate, the accuracy and the precision, being this last one the most representative of the benefits increasing 21,25%. Although the overall sensitivity has decreased to a 75%, all the other performance measures reach much appropriate values, starting with a 90% precision, a 0,43% false positives rate and a 98,34% precision.

Additionally, while the refinement step can take much more processing that slows down the process, the first step is very fast, even more if the down-sampling operations are replaced by a direct pull of the dc-image from an encoded file. This low complexity combined with the short frame span needed for the detection of short flash events make

the first detection step a candidate for on-the-fly flash detection that only introduces a few frames delay. This fast method, although presenting a higher false positives rate, should not be underappreciated; the frames falsely detected as flashes may not present a proper flash event, but the same features that have triggered the detection could also benefit of the posterior encoding techniques defined for the optimization of frames with flashes.

There is still much work to do in this field, and much other information to exploit. But we have proved that a morphological approach to the flashlights problem offers a solid basis for the detection of such events. Grand part of the complexity of the problem comes from the variety of forms, colors, duration and intensity, in which flashes are manifested. The use of 3-Dimensional morphological filters in different planes (Y, U, V, R, G, B, …) notably enhances the data extraction of all these characteristic features opening the doors to a broader interpretation of flash light events detection.

9. Additional Features

9.1. Anti-Flash Detector

The morphological filters that we use in the pre-processing steps to determine the frames and regions to be studied have all an opposite filter in formula and concept. This dichotomy can be used to, by following the same procedure on a sequence with the complementary filters, detect the global and local, short in time, light events that perform as an opposite of the defined flash. This events that we call anti-flash events can appear through many sources, for example, an element suddenly obstructing the light source that illuminates the scene (a person walking on front of the lights or, depending of the conditions, even a cloud that casts a shadow upon the scene) or a flickering light that is starting to fail (a very used resource in horror movies). All these effects are mainly related to the appearance of shadow areas for a very short period of time.

Although the modifications performed on the algorithm already allow the option of detecting these anti-flash light events, the parameters that we use are taken from our experience with the flash-like light events and thus are not optimized to this particular task. The advantages that this knowledge could provide to the encoder and to provide more accurate scene change detection suggest that a further discussion on this subject and an intensive testing to adjust the parameters to the new problem must be performed.

9.2. GUI

The use of the detector, with the many different parameters that allow an extent customization of the characteristics of the events we intend to detect, has become very complex. To approach the use of the detector to users who are not very familiar with the encoder procedures with a more intuitive interaction, and to provide a much powerful and comfortable tool to the users who know how to exploit the parameters to adjust the detector to their particular needs, we have

implemented a GUI that allows full control over the different variables that the implementation allows to modify.

Also, the GUI provides a previous checking of some parameters to ensure that the commands are neither contradictory nor invalid (such as valid down-sampling values) and provides a simple way to name the output files in concordance with the format used in the ".yuv" reproduction tool provided by Thomson.

The last option implemented to this GUI is the possibility to generate a batch file that will execute sequentially a series of detections, which is very useful to run exhaustive tests or to detect light events in a series of sequences.

A more in depth explanation on the GUI utilization can be found in Appendix B.

10. Further Improvements

Since this is a preliminary version of a direct approach to flash detection using morphological filters, many aspects should be revised and improved in further studies.

One of the first aspects that rise as susceptible of improvement is the **adaptation of the algorithm to the basic sequence characteristics**. The **temporal span**, manually determined, must be reconsidered for each kind of flash we are aiming to detect, but should also be related to the **frame rate of the sequence**. Thus, a **frames/seconds** temporal span should be defined and, consequently, the input options of the program should be adapted to allow both formats. Then, the program should calculate the value of the complementary variable using the sequence's frame rate. Similarly, the determination of the **number of boxes** in which we divide the frames will directly influence in the proper discrimination of motion effects from light events. This suggests that, even though visually the same number of boxes will perform similarly in the same sequence at **different resolutions**, a considerable difference may appear due to the different local relevance of the events at these image sizes.

The flash resistant scene cut detector is used in our implementation in order to remove a particular case of false alarms. However, the flash detector results, if meant to be the input for a video encoder, can not only be used as a light events management input. The detection of these events has a great potential as complementary information to other input data, mainly the traditional scene cut detection, which has many problems due precisely to light variations. An **appropriate combination of the light events information with additional scene transition data**, obtained through other algorithms, should allow discarding possible transition events that have been misidentified as light events.

Continuing with the **flash resistant scene cut detector**, as stated in ***Section 6.4***, the actual implementation uses a **window with a fixed value of 7 frames**. For the present

study we have determined that this value is adequate in terms of expected temporal span of the flashes. However, **the need or utility of detecting longer events has to be determined in further discussions**; depending on the length of these events, the possible benefits of treating the event as a different, independent scene need to be checked against the benefits of considering it to be a light event. If the detection of such events is eventually justified, the luminance pattern of these same events may attenuate the problem caused by this fixed value; if, on the contrary, this does not improves the final result, the **option of skipping the scene cut detection** step while processing the sequence has been implemented. Nevertheless, **a length adaptive version of the flash resistant scene cut detector** should be implemented to expand the possibilities of the application.

In order to provide additional information to the algorithms meant to merge the set of scene transitions with the output of the flash detector, a proper **measure that reflects different degrees of reliability for each detected flash** is to be obtained. This probabilistic measure would be used to decide what index shall prevail when there is a conflict between a detected flash and any kind of detected transition. The merger shall include some intelligence to evaluate the information from both inputs and decide on the final output.

Being the **detection of local flashes** the most complex aspect of the described process, if we aim at a more accurate detection, the topic that must be extended is the **discrimination of non-flash events** that are erroneously detected. In ***Section 6.3*** of the algorithm, while discussing the "motion based discrimination method" possibilities, we defined a division of the scene in several boxes to locate and remove false positives that we can relate to motion while avoiding the cost of motion estimation. The idea behind this technique was to **remove locally persistent light effects**; therefore, a compromise has to be reached to determine the optimal number of rows and columns. The more boxes/cells we use, the better we can appreciate the local effect of the detected events; but if we are trying to locate and eliminate clear objects in motion within a box area, it is

much easier that the elements swap to an adjacent box since the area covered by each box is smaller. On the other hand, if we use a small number of boxes, we will be able to find and reject moving objects that move within a greater area covered by the boxes, but also the effect of smaller events will be diluted within the whole box area and may go unnoticed. To improve this compromise, the idea of tracking these events not only in adjacent frames but into adjacent blocks (within the adjacent frames) too seems to be an interesting option. **Overlapped boxes** in the division of the image could be a possibility to offer some extra information to allow a rudimentary tracking method and thus improve the discrimination of motion related false alarms. If this approach proves to deliver better results, more complex tracking techniques should be evaluated.

Despite down-sampling the original signal to increase the speed, the **actual detector uses some computationally expensive operations**. Some of them could be substituted by techniques that perform with similar results and that may considerably fasten the process while offering a satisfactory result. For example, the actual **region growing** operation should be tested against a **controlled or limited growing** method, or even against a **threshold based connected region matching** algorithm to determine which one offers a higher speed/accuracy ratio. Also, as the algorithm is defined so far, in local flash candidates, **motion compensation estimation** is calculated for the whole involved frames, but only information for the relevant area is considered. Performing this operation is extremely expensive and should be **restricted to the relevant areas** using the local flash candidate bitmap as an input for the prediction.

Due to the non-causal nature of the detection algorithm, the **upper limitation** for it would be to perform an **on-the-fly processing with a real time stationary throughput and a minimal frames delay**. For this, the detector should be redesigned so the first (rough detection) and second (tune up) processing steps run in parallel threads. Also, the real time detection functionality would most probably benefit from techniques that allow massive parallel processing, such as GPGPU, to take care of most expensive frame processing operations (region growing, motion estimation).

As stated when talking about the **theoretical encoding improvements**, the encoder does not exclude traditional encoding techniques when the use of light event oriented techniques is suggested by the detector. The fact that it compares the results obtained with both techniques to decide which one performs better makes this same **encoding process the fittest element to determine** if all the false positives that arose in the first detection steps can take **advantage of the new techniques** or if their discrimination do not represent a loss of useful information. However, the balance between additional computational cost and possible compression gain requires an exhaustive testing that does not take place in this study.

About the **additional events detection**, detecting other scene changes such as **dissolves** and **fades** is an interesting objective; even more since we noticed that one of the signals used to determine the shot cuts position presents an interesting reaction to these events ***[Fig.32-33]***. Exploiting this information to expand the transitions detection to other events, or considering the implementation of other known detectors should be studied as an improvement for the detector.

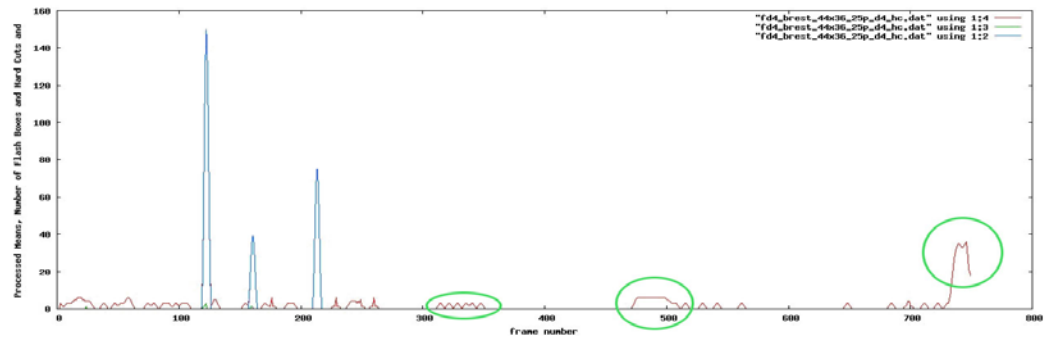


Fig.32 *Sequence 'Brest'. Correct detection of shot cuts (blue line) at 122-123, 160-161 and 213-214 (100% detection). The sequence also presents some* ***dissolve*** *effects at 313-351, at 474-511 and 735-749.*

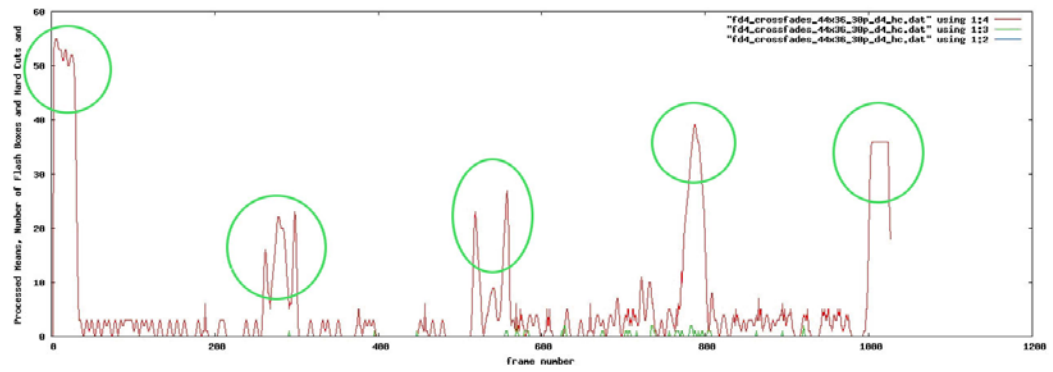


Fig.33 *Sequence 'Crossfades'. There are no shot cuts in this sequence, but we can find a* ***fade in*** *from 0 to 27,* ***dissolves*** *at 263-297, 520-558, 771-804 and a* ***fade out*** *at 1001-1027. This other effects present some particular shapes in the 'processed means' signal (red).*

A basic **modeling** with simple shapes of the region affected by the flash and the intensity of the same could be helpful to generate a more efficient prediction in both local and global events in an **encoding context**. In local flashes, the processing cost could be reduced by limiting the specific processing to the area corresponding to the event and also indirectly reducing the negative impact of the false alarms. Moreover, since most of the area is similarly affected when a flash takes place, a single value could be used to represent the whole figure with a clear benefit for the encoding of global flash and possibly of local events information.

The whole study has been focused on the **processing of the luminance signal (Y)**. However, although proving quite reliable, this has also been a limitation in certain situations. The luminance is a signal that contains a weighted combination of the Red, Green and Blue signals that is much related to grayscale images ***(Y = 0.2126 R + 0.7152 G + 0.0722 B)***. This results in a signal where we are best suited for detecting events that occur in the Y direction; flash events that come from a source of white light. However, **if the source of the light is not white**, but colored, the amount of **flash event energy** that ends up **in the luminance image is reduced** in a different ratio depending of the color. Depending on how much energy is left out, we might end up missing the detection of these particular events. This case has revealed to be particularly noticeable in the scenes involving ***police car flash lights, that alternate red light with blue light***. The participation of the red component in the luminance signal is around 1 fifth of the total

red energy, which makes low intensity red colored light events hard to detect. This is even more of a problem in the case of blue lights, which contribute even less to the luminance and are almost not represented, resulting in many light events missed by the detector. To resolve this issue in further studies, an option to select the desired signal to use (Y, U, V, R, G, B …), or even the parameters for a **weighted combination of signals**, could **tune the detector to be more receptive to events of a certain color**. Expanding this option, the output of the detections using different parameters could be used to cross-reference the events and add insight into the reliability of the detection, or even **add color information to enhance the model of the event**.

11. Appendix A: Morphological Filters

11.1. Erosion & Dilation

Erosion and dilation are the two basic operators in the area of mathematical morphology and here we will work on a grayscale version. The structuring element ***[Fig.34]*** describes the filter that will define size of the area around the pixel affected by the operator and the visual effect that will be performed on the image.

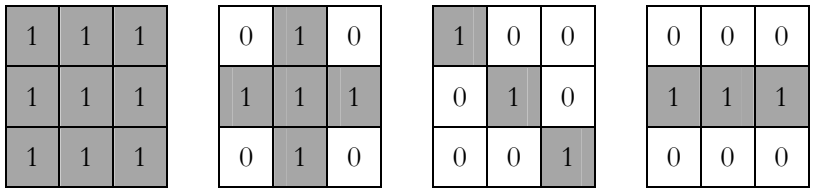


Fig.34 *Here are some examples of widely used structuring elements.*

The basic effect of the **erosion operator** on a grayscale image is to locally erode away the boundaries of regions of higher level pixels (***i.e.*** clearer pixels in the neighboring area). Thus, if we understand the images as a topological surface, these areas of 'hill' pixels shrink in size, and 'valleys' within those areas become larger (but not necessarily deeper) and, depending on the size and shape of the structuring element, may eventually disappear ***[Fig.35]***.

On the other side, the **dilation operator** expands the 'hills' while the 'darker valleys' between the clearer areas become narrower and, as seen in the erosion operator, may also disappear depending on the size and shape of the structuring element ***[Fig.36]***.

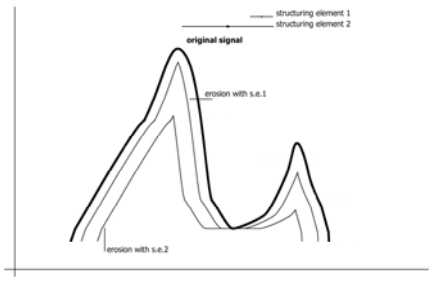
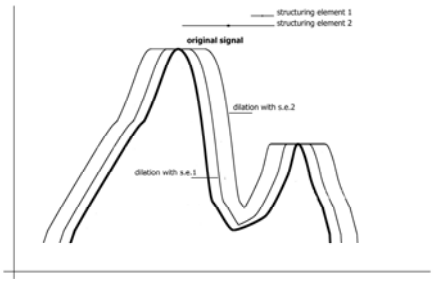
	

Fig.35 *Erosion on a 1D function.*	***Fig.36*** *Dilation on a 1D function.*

In the grayscale domain, these two operators are mainly used to smooth the image by 'filling' small holes (remove small darker areas) or eroding small peaks (remove small clearer areas). Also the structuring element is not unique, thus existing many different filters that may offer the possibility of removing 'objects' with a determined shape or orientation.

Note that removing dark or clear objects with this technique may also cause a slight deformation of the rest of the picture due to a flattening effect in remaining areas ***[Fig.37]***.

FORMULAS (to use this formulas we define the structuring element with vales [-∞,0] instead of [0,1])	
Supremum:	$z = x \vee y \quad \Rightarrow \quad z[n] = \text{Max}\{x[n], y[n]\}, \quad \forall n$
Infimum:	$z = x \wedge y \quad \Rightarrow \quad z[n] = \text{Min}\{x[n], y[n]\}, \quad \forall n$
Erosion:	$\varepsilon_b\{x[n]\} = \wedge(x[k] - b[k - n]),\ k = -\infty .. +\infty = x[n] \ominus b[n]$
Dilation:	$\delta_b\{x[n]\} = \vee(x[k] + b[n - k]), k = -\infty .. +\infty = x[n] \oplus b[n]$

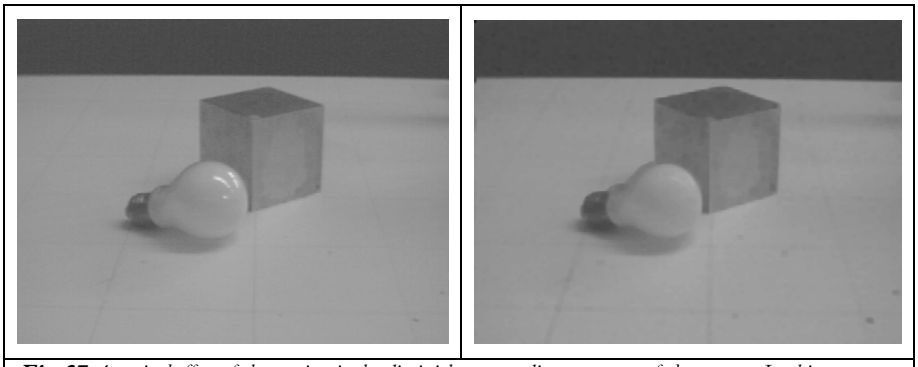


Fig.37 *A typical effect of the erosion is the diminishment or disappearance of clear areas. In this case we appreciate how the light reflections on the bulb in the original image (left) disappear in the eroded image (right). We can also notice that, since the cube is darker than what is surrounding it, his size has increased.*

11.2. Opening & Closing

The utility of the erosion and dilation operators is extended with the combination of both of them. Thus, the concepts of **opening** and **closing** arise as the solution for a better object removing technique.

An **opening** is defined as an erosion operation followed by a dilation ***using the same structuring element for both operations (assuming that a symmetrical structuring element is used) [Fig.38]***.

Then, the dual operation, the **closing**, is a dilation followed by an erosion operation, also ***using the same structuring element for both operations (assuming that a symmetrical structuring element is used) [Fig.39]***.

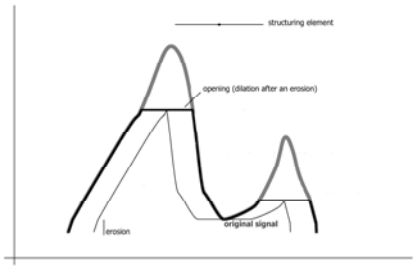


Fig.38 *Opening operation on a 1D function.*

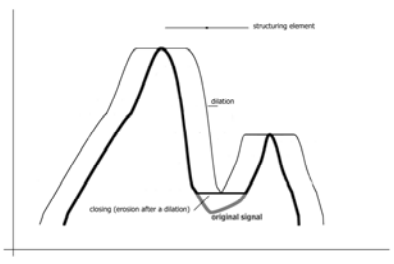


Fig.39 *Closing operation on a 1D function.*

Thus, the first step, during an opening/closing operation, removes the small clear/dark objects with no possible recovering, whilst the second one returns the

remaining objects to their original size and shape, though losing some texture information.

The obtained effect, then, is that with an **opening**, the clear objects that are smaller than the structuring element are removed and the signal is smoothed, losing some texture information ***[Fig.40]***. The opposite operation, the **closing**, is used to remove the dark objects, with same effects than the opening.



Original Image	*Eroded Image*	*Opened Image*

Fig.40 *With the opening operation we are able to remove small clear elements (salt noise in this case) while maintaining the size of the darker objects. Note, though, some loss of texture in the background, beard or hair, for example.*

11.3. Top-Hat

Since now we have seen applications of this operation aiming to the removal of small elements, but this characteristic can be easily used to obtain a method to remove all except these elements. The process is quite basic; a simple subtraction between the original and the opened or closed image will emphasize the removed information, and then, with a simple threshold, the previously removed objects can be separated from the texture information if so needed. The two versions of the top-hat operator are:

White Top-Hat: Original image – Opened image. (Detect clearer elements) ***[Fig.41][Fig.43]***

Black Top-Hat: Closed image – Original image. (Detect darker elements) ***[Fig.42]***

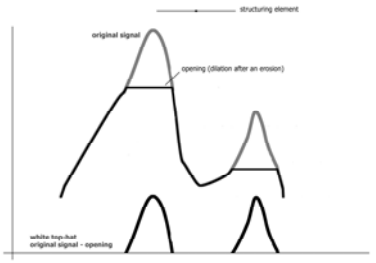


Fig.41 *White top-hat on a 1D function.*

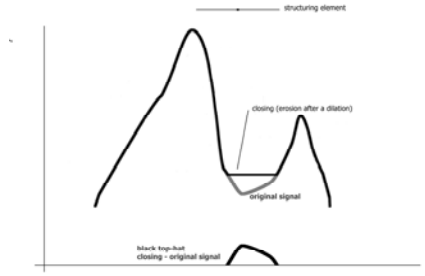
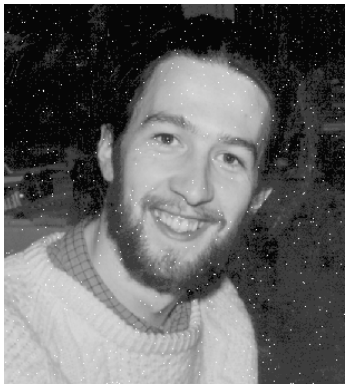
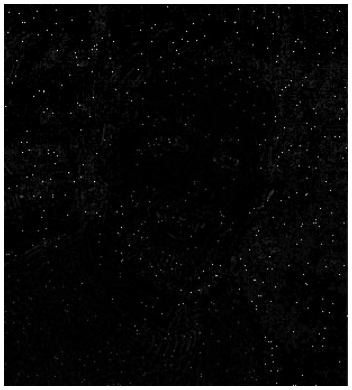


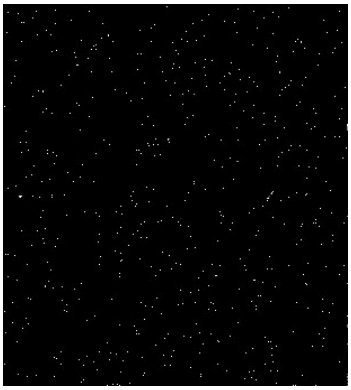
Fig.42 *Black top-hat operation on a 1D function.*



Original Image



White top-hat Image



White top-hat Image binarized

Fig.43 *With the top-hat operator we can recover the elements that are removed with the opening or with the closing operators.*

11.4. Temporal Processing

The erosion/dilation, opening/closing and Top-Hat techniques have been exposed in a 2-Dimensional example, but the same concept can be easily extended to the 3-D domain, either if we consider the [x,y,z] representation for 3-D objects or the [x,y,t] for a sequences of images. We can now consider structuring elements that include pixels in neighboring frames, thus, the concept of 'small dark/clear objects' is transformed in **'dark/clear objects that appear in a determined region for a short period of time'** ***[Fig.44]***.

When considering structuring elements for 3-Dimensional morphology processing we can choose either to use volumetric structuring elements by expanding a bi-dimensional in one dimension or use the usual bi-dimensional and linear structuring elements but in a different orientation that includes the third newly added dimension. Thus, considering the space [x,y,t] with a volumetric

element we will compare one pixel not only with some neighbors but also with pixels in the surrounding frames. In the case that concerns the present document, the temporal filtering will use a linear structuring element alongside the "t" axis, by performing the top-hat operation in only one direction and comparing a pixel with those that are in the same position in the previous and posterior frames.

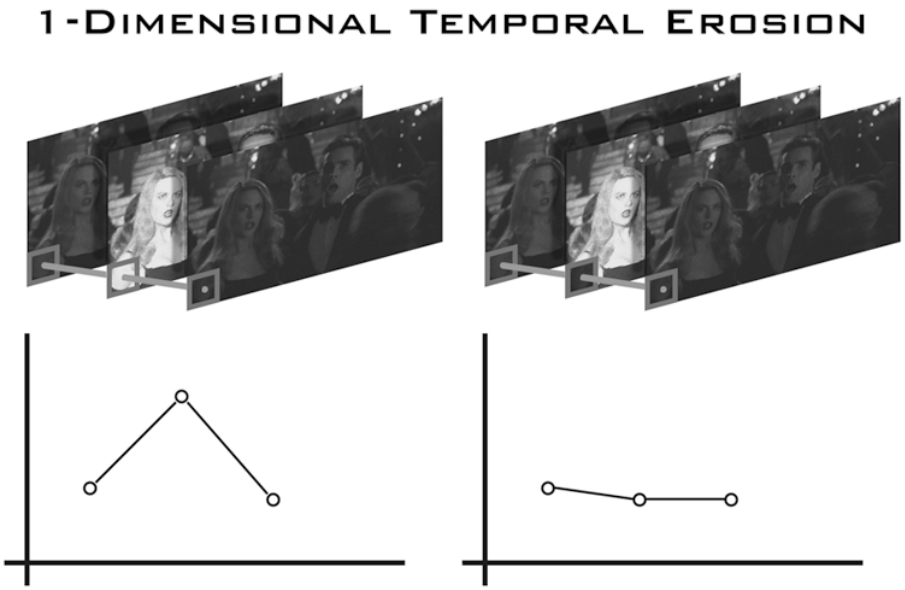
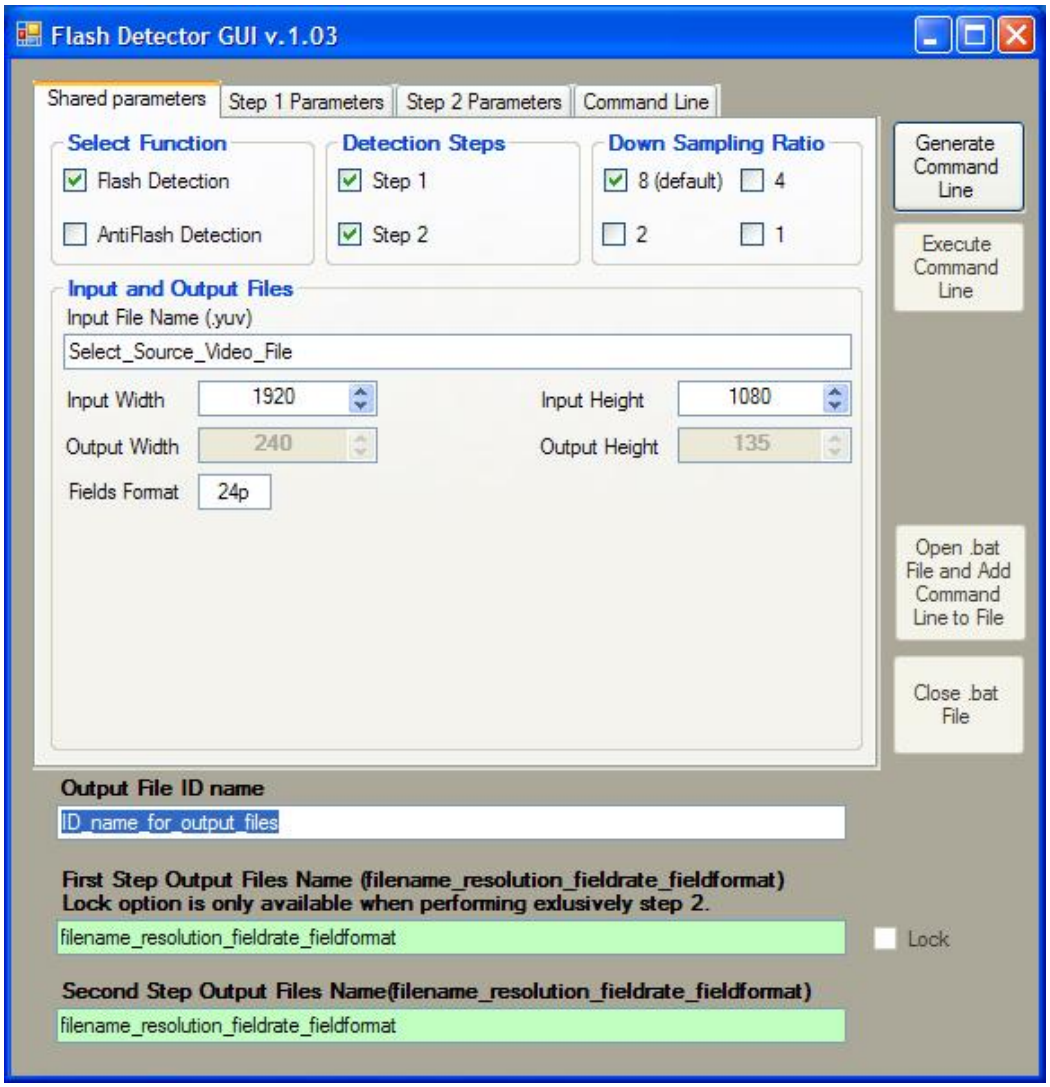


Fig.44 *Linear morphologic filtering in the temporal domain can be used to detect clear objects that appear in a determined region for a short period of time*

12. Appendix B: GUI - User Manual

Both source video file and detector executable file must be located in the same folder as the GUI. This folder will also contain all the resulting files of running the application.

12.1. Tab 1 – Shared Parameters



In this tab we can see the options that are shared between all the different applications of the program.

Select Function: First of all we must select if we want to use the program in his flash detection or his anti-flash detection version.

Detection Steps: Step 1 stands for the basic detection, while Step 2 refers to the refinement methods applied to the first set of candidate frames.

Down-Sampling Ratio: These are the accepted values to down-sample the original source file into a lower resolution version that is used by the detector.

Input File: The complete source file name must be introduced. Also his resolution values have to be indicated and the GUI will determine if with the current down-sampling ratio the output file resolution is acceptable; if it the resolutions are a multiple of the down-sampling ratio, the edit box will have a green background, otherwise it will be a light red background. The sequence's field format is the last of the sequence attributes that we must set, according to the format used in the files nomenclature.

Output File ID name: Introducing here an output file name for the identification of the generated files, the GUI will generate an output file name formatted as follows: filename_widthxheight_fields that will be used for the detectors output files.

First Step Output Files Name: If we intend to perform several detections with different configurations of the refinement phase (step 2) but the same parameters used in the first phase, it is recommended in order to save time to perform the refinement procedures over the results of a first single stepped detection. However, since in the command line for this execution, it is necessary to have the filename used for the first detection step, this GUI allows the use of a "Lock" checkbox to save this name to be used in further command lines. The use of this function would be as follows:

A Step 1 only configuration is set with a "filename1" and the command line saved or executed.

The "Lock" box is then checked, saving the name of the output files for this step. Different only Step 2 parameter configurations are set, creating a new filename for each one, and the new command lines are saved or executed. Since the output files name for the first step has been stored, these files will be used as additional input in these refinement steps, thus avoiding repeating the time consuming first step.

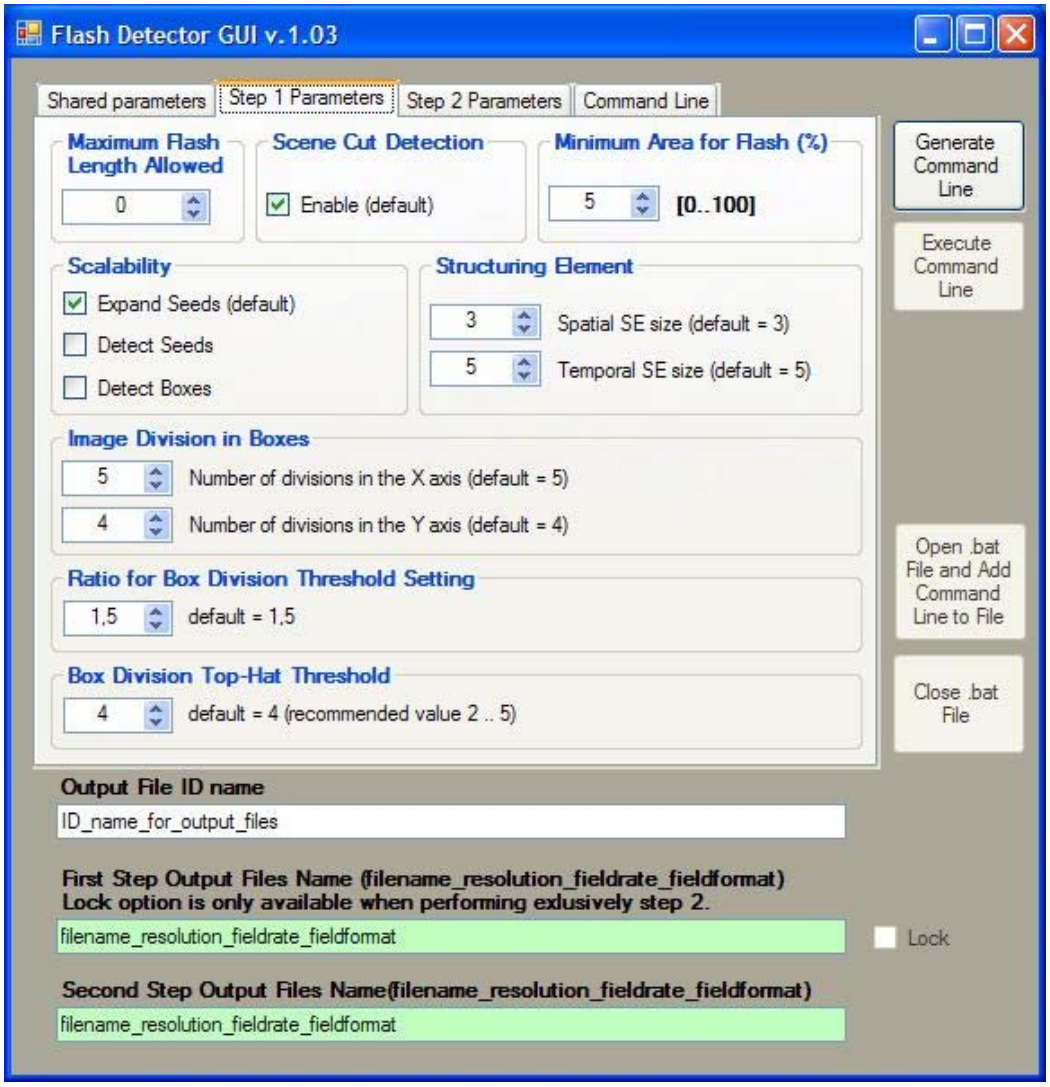
The "Lock" option will only be available when only "step 2" is checked. Once locked, the "Detection Steps" checkboxes will be disabled until the option is unlocked.

Second Step Output Files Name: This field will show the final output file name after Step 2 is performed. If Step 2 is not performed, or the "Lock" box is unchecked, it will be the same as the "First Step Output Files Name". Else, when only Step 2 configuration is selected and once the "Lock" has been applied to the Step 1 output file, this field will still show all the modifications applied to "Output File ID Name" appended with the selected resolution and field format. This is intended to allow different filenames for several executions of the second processing step without repeating the first step.

Side Buttons: This buttons are used to generate and use the command lines that we will call to run the detector or generate a batch file. Their use is explained in section e) of this appendix.

Note that both Output File and Side Buttons are located in the outer frame of the GUI. This location has been assigned to them so they can be directly accessed from any of the tabs, which makes the use of the GUI friendlier.

12.2. Tab 2 – Step 1 Parameters



This set of options and parameters refers to the configuration of the first detection step.

Maximum Flash Length Allowed: Detected flashes that last for more frames than the value indicated here will be automatically rejected. If the value is 0, no limit will be applied.

Scene Cut Detection: If enabled, a rudimentary, flash resilient scene-cut detector will be used to discriminate flash candidates.

Minimum Area for Flash (%): All frames where a flash area is detected that affects less than the percentage of the frame set by this parameters will considered as if there were no flash activity in them.

Scalability: One of three scalability levels is to be selected.

Expand Seeds: A very accurate shape of the whole area affected by the flash.

Detect Seeds: Detection of the core area of the flash, where it is much more intense. This precision level is substantially faster than the "expanded seeds" level.

Detect Boxes: A very rough detection of rectangular areas where flash activity has been detected.

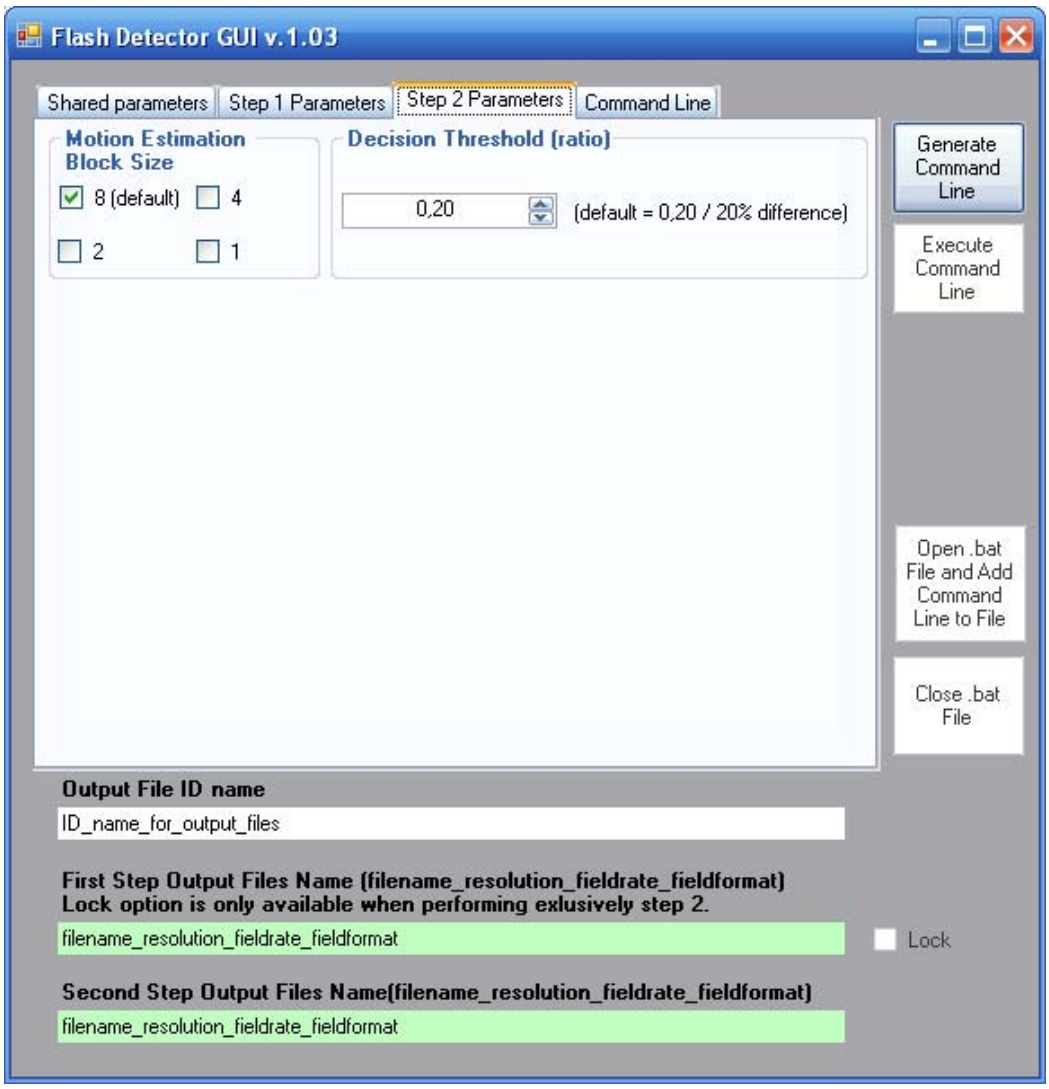
Structuring Element: Define the size of both spatial and temporal structuring elements used in the morphology steps of the detection.

Image Division in Boxes: In some steps of the detection method the image is divided in several box shaped areas. These parameters determine how many rows and columns will be used to perform that division.

Ratio for Box Division Threshold Settings: Ratio used to obtain the threshold applied to each box in order to detect possible flash activity. After morphologic filtering is performed, the mean value of each box is multiplied by this value to obtain the threshold.

Box Division Top-Hat Threshold: When the 'wtho' sequence is divided in boxes, a temporal white top-hat is applied to the mean luminance of each of these. If the new mean is lower than this threshold, the box is removed from the flash candidate and its content, because it is a locally persistent positive, is assumed to be a moving object.

12.3. Tab 3 – Step 2 Parameters



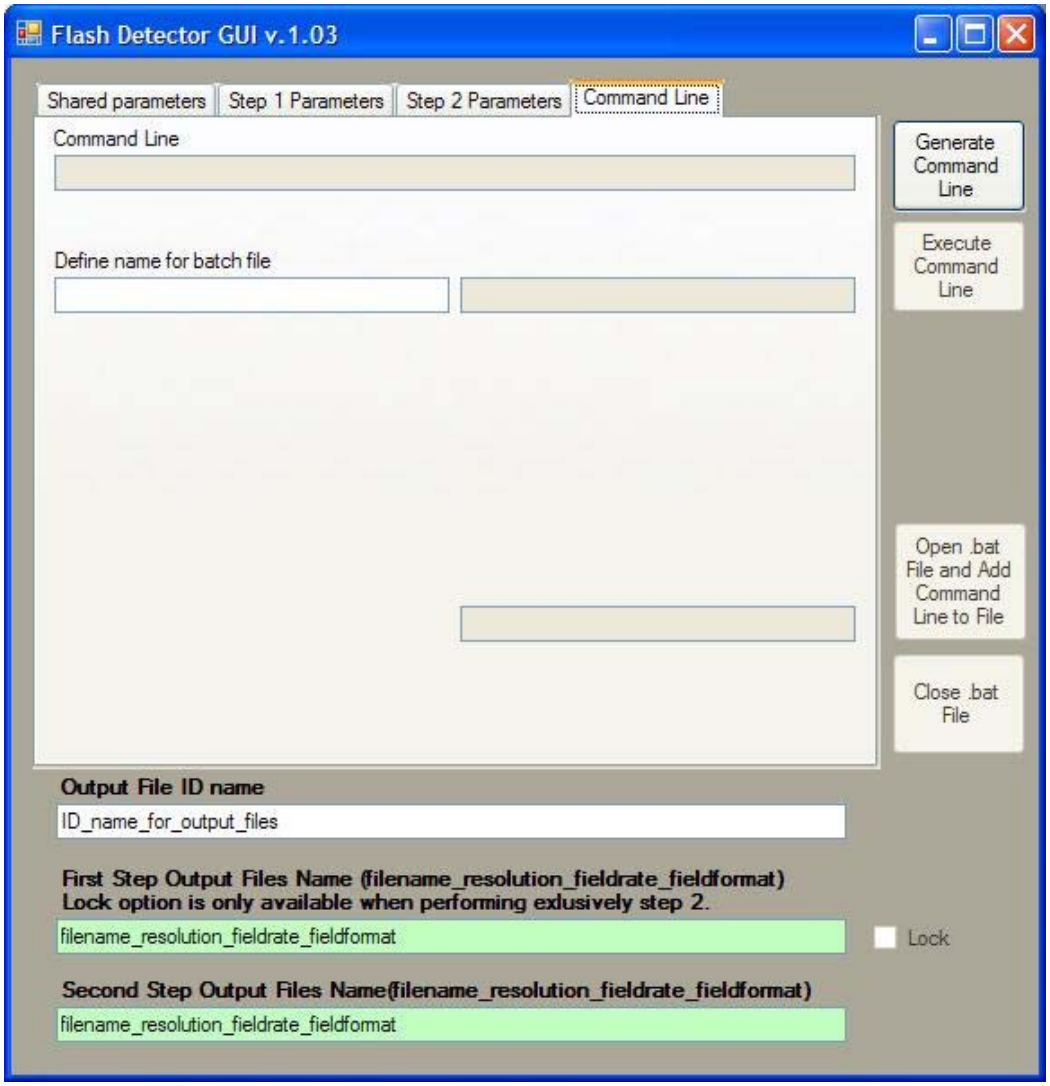
This set of options and parameters refers to the configuration of the second step, the detection refinement process.

Motion Estimation Block Size: This is the size of the blocks used in the motion estimation algorithm. It is required for this value to be equal or lower than the down-sampling ratio.

Decision Threshold: This ratio is used to evaluate how similar are the predictions of the flash frames using forward and backward prediction for a

candidate to local flash. If the two measures differ more than this ratio, the flash candidate is discarded under the assumption that the scene (or at least it's content) has changed during the event.

12.4. Tab 4 – Command Liine



Command Line: In this space a command line is generated that will execute the application with all the defined parameters.

Define name for batch file: Typing in this space will generate a filename for a batch file that will store the generated command lines for further execution. Note that while a batch file is open for writing, the filename will be stored and modifications in the "Define name for batch file" will stay on hold until the actual batch file is closed.

Generate Command Line: This button will update the Command Line text with the command line corresponding to the actual parameter's configuration.

Execute Command Line: If both the flash detector application and the source video (and the additional files if executing only step 2) are present in the same directory as the GUI, clicking this button will run execute the present command line and run the flash detection application.

Open .bat File and Add Command Line to File: This button will only be enabled if a command line is set and a batch file name is defined. Clicking on it will provoke the respective batch file to be:
Create it if it does not exist and add the actual command line to it.
Append the actual command line to it if it exists and is already open by the application.
Open the file and append the actual command line to it if it exists but it is not yet open for the application to write in it.

Close .bat File: Clicking this button will cause the actual batch file to be closed. It also will validate modifications to the "defined batch file name" occurred while a file was being edited. This modifications will not affect the file that is being closed, they are just intended for new batch files.

12.5. Intended way of use

Here we will see an example of a standard way to use this GUI:

First of all, we make sure the flash detector application, the GUI and the yuv file that is going to be used as input are in the same directory before executing the GUI. Now, in the "Shared Parameters" tab, we introduce the file name, the resolution values and the field format. We select Flash Detection, we disable "step 2" and we set the down-sampling ratio to 8. After introducing an "Output File ID Name" (ex, fd_output_step1) we can observe how an output name is generated with a valid format.

Now it's the turn to prepare the settings in "Step 1 Parameters" tab. We set the maximum flash length to 5, we make sure that scene cut detection is checked and define a minimum flash area of 20%. We want "Expand Seeds" to be checked and the default values of 3 and 5 for the spatial and temporal structuring elements are ok for this case. The rest of parameters can also be used with their default values.

In the "Command Line" tab now, we define a batch file name (batchfile1), we click the "Generate Command Line" button and we "Open .bat File and Add Command Line to File". By this point, the .bat file is already created and we have introduced our first command line that should execute the detection step of the application.

Back to the "Shared Parameters" tab, we uncheck the "Step 1" box and "Step 2" will be automatically selected. It is now time to use the "Lock" checkbox next to the output file name to preserve the filename we used for the first step.

Now we will try different combinations of the "Step 2 Parameters" tab. For each combination, we will modify the output file name (example:

fd_output_step2_bs8_d020), then click on the "Generate Command Line" button and finally click on the "Add Command Line to File" button. Note that it is not necessary to go to the "Command Line" tab to do that, and as the first batch file we used is not yet closed, all lines will be added at the bottom of the file. Once we have prepared all the command lines we need for this execution, we click the "Close .bat File" button and we are now ready to run the batch file we have just generated in the same directory as the rest of the files.

References

[1] B.L. Yeo, B. Liu, Rapid scene analysis on compressed video, IEEE Circuits Systems Video Technol. 5 (6) (December 1995) 533-543.

[2] R.Zabih, J. Miller, K.Mai, A feature-based algorithm for detecting and classifying scene breaks, Proceeding, ACM Conference on Multimedia, San Francisco, November 1995.

[3] W.J. Heng, K.N. Ngan, High accuracy flashlight scene determination for shot boundary detection, Signal Processing: Image Communication 18 (2003) 203-219.

[4] Y. Nakajima, K. Ujihara, A. Yoneyama, Universal scene change detection on MPEG-coded data domain, Proc. SPIE Visual Commun. Image Process. 3024 (2) (1997) 992–1003.

[5] A. Hanjalic, Shot-Boundary Detection: Unraveled and Resolved?, IEEE Circuits Systems Video Technol. Vol. 12, No. 2, (February 2002)

[6] X. Ubiergo, S, Bhattacharjee, Shot Detection Tools In Digital Video (1998)

[7] M. K. Mandal, F. Idris, S. Panchanathan, Image and Video Indexing in the Compressed Domain: A Critical Review, Image and Vision Computing Journal, Vol. 17, No. 7, (May 1999)

[8] R. Lienhart, Comparison of Automatic Shot Boundary Detection Algorithms, Storage and Retrieval for Image and Video Databases, No. SPIE 3656. (January 1999), pp. 290-301.

[9] C. Huang, and B. Liao, A Robust Scene-Change Detection Method for Video Segmentation, IEEE Circuits Systems Video Technol. Vol. 11, No. 12, (December 2001)

[10] S. Porter, M. Mirmehdi, B. Thomas, Detection and Classification of Shot Transitions, In: Proceedings of the 12th British machine vision conference, BMVA Press, pp 73-82. (2001)

[11] A. Miene, A. Dammeyer, Th. Hermes, O. Herzog, Advanced and Adaptive Shot Boundary Detection, Proc. of ECDL WS Generalized Documents, pp. 39-43. (2001)

[12] Y.L. Geng, D. Xu, A solution to illumination variation problem in shot detection. In: ICME, 2004. 81—84 (2004)

[13] J. Zheng, F. Zou, M. Shi, An Efficient Algorithm for Video Shot Boundary Detection, Proc. Intelligent Multimedia, Video an Speech Processing, (October 2004)

[14] R. Garcia, T. Nicosevici, X. Cufi, On the Way to Solve Lighting Problems in Underwater Imaging, IEEE OCEANS Conference, pp. 1018-1024, (2002)

[15] J. Green, T.P. Pridmore, S. Benford, A. Ghali, Location and recognition of flashlight projections for visual interfaces, In: Pattern Recognition 2004, Vol. 4, pp. 949—952 (2004)

[16] Y. Oiket, M. Ikedat, K. Asadat, A Pixel-Level Color Image Sensor With Efficient Ambient Light Suppression Using Modulated RGB Flashlight and Application to TOF Range Finding, IEICE Trans. on Electronics, Vol. E87-C, No. 12, pp. 1651-1658, (December 2004)

[17] W.J. Heng, K.N. Ngan, Post Shot Boundary Detection Technique: Flashlight Scene Determination, Proc. ISSPA'99, Vol. 1, pp. 447-450 (1999)

- Jin-Soo Kwon: Daniel... No more flash?

- Daniel Faraday: No. No more flash. The record is spinning again.

[Lost – Season 5 Episode 08 - LaFleur]