Semi Automatic Digital Motion Picture Restoration System with Learning Capabilities

László Czúni[1], Gergő Császár[1], Attila Hanis[1], Levente Kovács[1], Attila Licsár[1], Tamás Szirányi[1]

[1]czuni@almos.vein.hu
University of Veszprém, Department of Image Processing and Neurocomputing, Hungary

Abstract

*We have developed a semi automatic film restoration system, DIMORF (**DI**gital **MO**tion Picture **R**estoration System for **F**ilm Archives), working with a minimum effort of parameter settings and training, with adaptive learning abilities, at relatively high speed. The system is being applied for the restoration of old films, characterized by strong color distortions, scratches, dirt, vibration, and flickering. Since we handle large amounts of visual information, video indexing and retrieval components are also very important.*

The system we present herein contains novel features very important for state-of-the-art application. Some of these are an automatic self-teaching stabilization algorithm, a neural network based learning blotch detection method, a novel automatic shot detection and indexing framework. Data regarding the restoration process, shot data, filtering data, etc. are all stored in an XML-based portable and versatile database backend, providing support for the learning algorithms, for browsing and retrieval purposes and also for logging.

The topics we will concentrate in this paper are the automatisms and the learning possibilities, which we use during the restoration process in filters and in video indexing.

1. Introduction. Goals and Problems in Film Archival

The basic purpose of our R&D project is to create a semi automatic restoration environment for film archives. Besides restoration purposes the proposed environment is capable of general video analysis tasks and due to its XML based control and data acquisition, data-mining techniques can help the work of operators. In this paper we describe some of the software components and some experiences gathered so far in the heavy duty industrial testing period of the last few months. The DIMORF [2,3] system presented herein and the filters have proved their abilities in the restoration works of the first Hungarian color film (Lúdas Matyi, 1949, director: Nádasdy Kálmán, length: 106 minutes), this summer.

The purpose of this paper is to present those novel methods which provide high level automatisms in their areas and namely the unsupervised stabilization, color and motion-based indexing and automatic deblotching with neural network learning. We will provide a short description of general filters available, then we will present the new automatic methods. We will elaborate on the usefulness of the XML data representation of filtering steps, outputs and indexing data.

The main issue in film restoration is the amount of data and organization of tasks. The operators process the same steps many times, and they hard to recognize that the similar task on a similar frame has been done once before. For this reason frame-sequences must be indexed in a proper way to find a).similar sequences and b). similar tasks. The first is the indexing and archiving issue, while the second one is the learning-phase. Meanwhile, indexing methods must help the learning steps to find the appropriate database and setting, what can be stored in a highly organized XML database.

This paper will show how to organize the tasks and the corresponding data to achieve a controllable semi-automatic film restoration procedure, where the organized data makes it possible to show learning capabilities.

In the present phase of the project we have completed the film-restoration software with several smart attributes, and the program has been deeply tested and validated in the restoration of an archive color film 'Ludas Matyi' of 120 minutes. Meantime, on the practical experiments, we have developed some new learning attributes to help the operators in the most strenuous parts of the tasks.

2. System architecture

Our system consists of several components: a high-resolution (6K) scanner, a multiple terabyte storage server, film processing workstation(s) and a laser film-printer with 6K/line resolution. While all hardware parts are made at our consortium partners, software applications are developed in our laboratory at the

University of Veszprém. In this paper we concentrate on the software architecture in detail. The Microsoft®™ Windows®™ based software restoration framework was developed with Microsoft®™ Visual C++®™ and runs independently form the scanner and film recorder on personal computers. It can also operate as a stand-alone application for general image/video processing since it is implemented with an open interface for possible future algorithms (added as filters).

2.1 Data storage in XML

Throughout the restoration process all modifications made to the film (no matter that manual or automatic) are recorded for possible recovery, further analysis and learning goals. Further analysis can be useful for both the operator to review his/her work and also to increase adaptation in an automatic way by supervised or unsupervised learning. All these steps require the handling of huge amount of information and the well-defined but complex data structures should be easily understandable for the computer and the human. We have chosen XML (eXtensible Markup Language) for this purpose. Dynamic XML structures record all classes of information except for binary data where only references are stored in XML streams. XML storage enables us to implement data processing for different purposes:

- Controlling filters and saving the results with descriptors (similarly to those in MPEG-7);
- Generation of report files (in printable format): extraction of meaningful information to review system operation in time.
- Statistical analysis of artifact’s features (those are results of manual or automatic error detection) and user interaction.

Storing film, scene, shot and processing data in XML gives a unique opportunity for advanced indexing, search and retrieval. For example color and motion indexing data can be used to search for specific types of motions or frame sequences with desired color content. Shots containing specific types of errors can also be searched, correction parameters can be extracted and similar errors can automatically be corrected using the parameters obtained from the earlier correction phases.

Besides filter and film data, we also store bookmark information for shots, with the possibility to provide user-input textual data even for every frame if desired (Figure 1).

```
<bookmarks sequence_id="123_a">
<mark type="error" filename="1" date=""
creator="">beginnign of the spinnign sequence</mark>
<mark type="dust speckles" filename="123_a_0001" date=""
creator="">lots of speckles</mark>
</bookmarks>
```

Figure 1. Bookmark data in XML.

Also, storing film information in XML data structures gets us closer to the multimedia indexing and retrieval technologies emerged in recent years. Many projects and standards (e.g. MPEG-7) were and are being developed which use XML annotations for storing data about multimedia content for indexing, search and retrieval purposes. This is why we also chose XML for data storing and presentation.

Besides giving an easy to handle backend for multi-purpose data storage, we also use the XML representation for restoration filter parameter analysis and automatic machine learning. E.g. a manually, correctly deblotched sequence can provide enough data for the neural network scheme to achieve good results in blotch recognition in the next shot, this way considerably reducing human interaction.

2.2 Description of Components

In this section we outline our digital restoration system, demonstrating the key algorithms and some special circumstances. Based on our results in [2,8] we have implemented more sophisticated algorithms and we have tested them on several real life 2K archive sequences. Since great impact has been made to achieve automatic operations, film analysis is supported by automatic cut detection and video indexing based on color information and motion activity.

Every algorithm (i.e. filter) produces an XML data stream besides filtered frames, for statistical and learning purposes. Our main goal is to produce such algorithms (i.e. filters) that can learn and improve themselves by examining their previous XML results and the data produced by previous filters. Filters are also capable of low-resolution analysis producing XML data streams and then using the produced analysis results for high resolution correction.

Filters provide methods of local application (e.g. deblotching a few frames), but they also provide data for a greater goal. For example the semi-automatic deblotch data is used by the automatic neural network-based deblotch filter for learning and automatic parameter selection. Or else, indexing is useful when searching for similar scenes, but also provides means to categorize and select sequences and scenes for the automatic filters for learning and classification purposes.

2.3 Semi-automatic operations

The requirements of state-of-the-art restoration systems are related to the problem of the high costs of manual film restoration, the most relevant problem of the restoration industry [5]. That is why the proposed algorithms and software solutions support an open, expandable, and well-structured system driving towards built-in intelligence and automatisms.

That is besides the aim to be able to handle all major types of film error we would like to emphasis the following features:

- The system should enable data-mining and intelligent process control where XML data structures record all user interactions and function responses.
- The application should be user friendly and should support the work of operators with easily understandable digital report files.
- Due to long restoration times continuous operator intervention is to be avoided. The components of the system must operate parallel and multi-threaded in (semi-)automatic mode, with occasional manual interventions for fine-tuning. At the current state this requirement is partially fulfilled: deblotching and scratch removal still requires manual supervising and sometimes corrections.

To help automatic operation, film processing is organized into a hierarchical structure. The basic processing element is the filter. Filters can be chosen from the filter list that is easily expandable with new algorithms. Several filters in an ordered sequence create a job and several jobs can be organized into a task. Within one task one image sequence is processed. Naturally, different jobs can also be run on only certain selected frames of sequences, called SOIs (Sequences of Interest). To run algorithms on different image sequences several tasks can be started independently even in automatic batch processing mode (Figure 2 shows an illustration of the main program window). To organize system resources optimally, all data storage mechanisms and process control is done through XML data files.

2.3.1 Color filters

It is not the purpose of this paper to discuss color restoration of films. Here we only list the most important and well-know color filters already implemented in our restoration framework: hue-saturation correction; brightness-contrast modification; color balance; gamma-expose; curves (histogram transformations specified manually or by special pre-defined functions); color levels adjustment. Besides these, special de-fading algorithms are also implemented and tested.

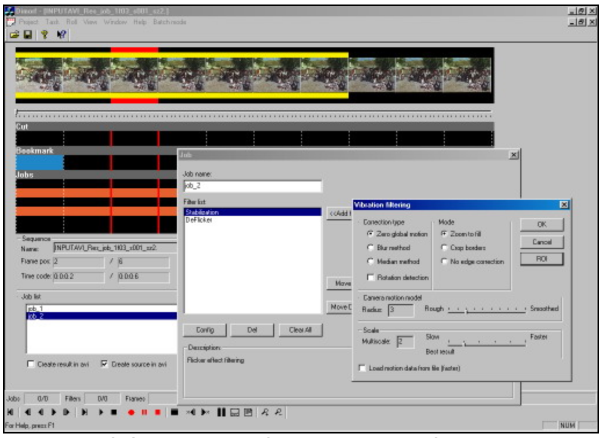

Figure 2 Screenshot of the restoration framework

2.3.2 Semi-automatic deblotching

The aim of the proposed semi-automatic deblotching algorithm is to detect and correct the so-called “one frame defects”. Our technique, similarly to the SDI detector [6], is based on two-directional motion compensation that is carried out by block matching. Those pixels where the correspondence is good between the predecessor and successor frames but bad between the center frame and its neighbors are marked as blotch candidates. The correction phase is carried out by inpainting from motion compensated neighboring frames. To do that, precise motion vectors are required over the detected artifact. According to results presented in [9] we correct the optical flow around the blotch which is called the motion vector correction phase.

To decrease the rate of false positive detection strict constraints are raised, e.g. color, size and optical flow reliability parameters.

With the help of these constraints the reliability of automatic correction can be increased significantly but human checking is still necessary. In our implementation the deblotching filter has two phases:

1. Automatic operation: the algorithm detects all suspicious areas and makes inpainting.
2. Operator checks the results of the automatic phase and makes corrections by rejecting corrections or by marking new areas as blotches.

The parameters and output of the filter are stored in XML for later learning and analysis purposes. For an example layout see Figure 3.

```
<Filter FName="Blotch Detection (SDIe)">
<Param PName="FF">0</Param>
 . . .
<Param PName="GMASK">false</Param>
</Filter>
```

Figure 3. Deblotch XML output.

2.3.3 Scratch removal

For vertical line scratch detection we use the following algorithm, which is based on first order derivative calculation and a certain definition of line scratches.

Let us define scratch points as glitches/faults in an image row otherwise continuous in intensity. That means that we treat an area part of a scratch when it has the shape on Figure 4.

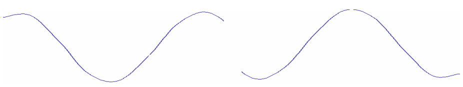

Figure 4. Illustration of shapes of image row functions where possibly a vertical scratch goes through (left:darker, right:lighter).

First we calculate first order row gradients. At possible scratch positions the row gradient will have 3 adjacent zero values, at a distance which coheres with the maximal scratch width value specified earlier. Then we generate a vertical sum for each horizontal position of the derivatives, where, if a value of the vertical sum function reaches higher than a threshold, we take the vertical position as a scratch candidate. At these specified locations search for scratch point candidates. Having marked all candidates we pass the vertical image part through a Hough transform to connect marked positions into lines.

After all previous steps a scratch mask is generated that will than be used for scratch removal (local median, 1D/2D/3D median, inpainting, texture sampling, etc.). The parameters and output of the filter are also stored in XML for later learning and analysis purposes. For an example layout see Figure 5.

```
<Filter>
<Name>Filter_ScratchRemoval</Name>
<Version>1.0</Version>
<Step>yes</Step>
 <Params>
 <Param Name="AngleDeviation" type="integer">0</Param>
 <Param Name="EdgeLimit" type="integer">75</Param>
 . . .
 </Params>
<Results>
  <DataFile name="ScratchFile" type="binary"
ext="smd">DeScratch_teknoc.smd</DataFile>
</Results>
</Filter>
```

Figure 5. XML data of the scratch removal filter.

2.3.4 Deflickering

We model flicker with linear equations. Parameters are predicted by comparing intensity variance and mean of consecutive frames similarly to [9]. In case of high-resolution images local and global flicker correction is done by multi-scale computations of motion and correction parameters. Motion compensation of neighboring frames is achieved by the phase correlation technique. In our implementation the algorithm is improved to restore color images by processing each color channel independently. To make computations faster deflickering can apply motion information generated and saved by the stabilization filter ran previously. This reuse of data is possible with the help of the XML structures used for parameter and data storage (Figure 6).

```
<Job_info>
<Job Name="f_" SOI_string="0-315">
<Filters>
<Filter FName="DeFlicker">
  <Param PName="Width">16</Param>
  <Param PName="Height">12</Param>
 . . .
</Filter>
</Filters>
</Job>
</Job_info>
```

Figure. 6. XML output of deflickering.

3. Cases for automatic processing and learning capabilities

While corrections can be done by manual filtering and wholly manual corrections, these are very time consuming, very tiring and very expensive (far too many man-hours to be paid). That is the reason why we started developing semi-automatic filters (Section 2) and after an interval of testing and experimenting some fully automatic filters, some even with automatic learning capabilities based on neural networks. These automatic algorithms follow in the subsections of Section 3, which include automatic color- and motion-based indexing, automatic deblotching, a learning filter for filtering small blotches and automatic stabilization. We have also developed a Human-Computer Interaction model based on automatic self-learning hand gesture recognition based remote controlling of the restoration process [10].

3.1 Automatic filtering and indexing methods

In this section we present automatic algorithms (i.e. filters), which are color and motion based indexing, cut detection, unsupervised automatic stabilization, neural network based self-learning blotch detection. The automatic filters use previous run results stored in XML data streams to obtain learning data. After a few steps e.g. the neural network in the deblotching filter can specify the detection parameters for the whole shot.

3.1.1 Image indexing

To help operators to overview and process large amount of visual information there are different techniques to be applied. Cut detection, video skimming, content-based retrieval are the most important tools for this purpose. In this section we outline an improved version of [4].

To support the semi-automatic configuration of filters the systems builds up a knowledge database, which contains the parameters of manual user interactions (such as fine tuning). These data are correlated to the different shots of a movie and thus the system can automatically propose adaptive filter browsers for supporting the operator's decision making. For the purpose of content-based retrieval the shot representative frames are indexed.

According to our experiences [4] color artifacts can dramatically change even within few seconds of the same film roll and can cause serious problems in color indexing. For color processing we segment the HSV (H*ue*, S*aturation*, V*alue*) color space dynamically into 13 parts similarly to [1].

In the case of **indexing color signatures** the frame that has a histogram most similar (in Euclidean distance) to the histogram of the average frame of the shot will be the *r-frame* 13 color signatures are extracted from r-frames and indexed in database files. In the case of **indexing motion vector information** the selection of representative frames is different from the previous technique as *r-frames* are those having histograms less related to the average histogram of the shot. Motion information extraction is performed in the temporal neighborhood of r-frame with adjustable temporal radius (app. 10-20 frames). Motion vector estimation was performed by a Block Matching algorithm with 5x5 block size, 24 pixel range size in both direction and with a 1 pixel shift size in both direction.

We generate cumulative motion maps (MM) with the help of displacement fields so every point of this map contains the complexity measure of the surrounding motions and regular features too: angle and amplitude of the inherent motion vector. The MMs for each shot are indexed with the following three motion information: *motion intensity, dominant direction, motion complexity*.

The statistical evaluation of irregularities in the motion field is performed as follows. In case of n orientations the interval $[0,\pi]$ is divided into n equal subintervals, and a distribution function $f \in [0,1]^n$ is obtained. The complexity of motion statistics f is measured by the normalized entropy formula:

$$e_n(f) = -\frac{1}{\ln(n)}\sum_{i=1}^{n} f(i)\ln(f(i)) \in [0,1] \quad f \in [0,1]^n \qquad (1)$$

For uniform distribution $e_n(f)=1$, and for singular distribution $e_n(f)\sim 0$ $(\lim_{n\to\infty} e_n(f_n)=0, \text{ if } \lim_{n\to\infty}\sigma(f_n)=0)$ where σ is the standard deviation function. In the case of uniform distribution all directions are present and the irregularity (complexity) is the highest. When the motion field is containing mainly parallel motions, the distribution of angles approximates to singular distribution.

In the case of camera zoom or rotating motion the distribution of angle are quite uniform for the whole frame, but in small regions distributions converge to singular. So images are divided into n*m blocks and the distributions of directions are measured during ten frames in each block. Entropy defined in Eq.(1) is calculated for each block and stored as complexity values in the first channel of MM. The obtained channel is Fourier transformed and eight most significant magnitude coefficients with low and high spatial frequency are stored as complexity index values. Here n*m=6*8.

On Figure 7a we used only the directional information for retrieval to demonstrate the possible false answer arose from inaccurate computation of dominant direction when complex motion is present. In the third column motion in the right side of the image was complex and it is failed to estimate dominant motion correctly. When complexity information is used together with direction during retrieval the results were better (Figure 7b.).

The parameters and output of the filter are also stored in XML for later learning and analysis purposes. For an example layout (cut data and motion indexing data) see Figure 8a-b.

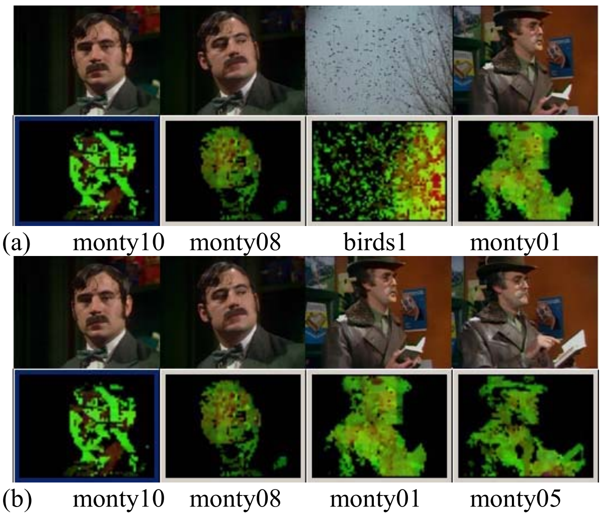

Figure 7 - (a) and (b). Query by example using dominant directions (a), and together with complexity measure (b). First column is the query, RFs are in the first row and MMs are in the second row.

```
<cuts>
 <cut type="hard" start="38" end="38" />
 <cut type="hard" start="355" end="355" />
 . . .
</cuts>
```

Figure 8a. XML representation of cuts of a scene

```
<media mediaID="Ludas01.02" title="Ludas Matyi">
<scene sceneNo="003">
<shot shotNo="0013">
 <rFrame frameNo="0123042" />
 <CameraOperation present="yes" type="pan-right" />
<RegionGlobalMotion>
 <GlobalMotion area1xValue="3" area1yValue="0" />
 . . .
</RegionGlobalMotion>
 <Faces present="yes" />
 <Captions present="yes" />
 <Subtitles present="no" />
 </shot>
</scene>
</media>
```


Figure 8b. Motion indexing data of a shot containing motion information.

3.1.2 Unsupervised film stabilization

The vibration of a degraded image sequence is usually caused by improper film transportation during the recording, copying or the digitization process. We have developed an automatic method for image stabilization consisting of two main steps: estimating vibration then correction by drifting the frame.

Earlier automatic stabilization algorithms were unsuccessful in cases when relatively large areas were covered with different optical flow fields. In these cases human interaction was necessary to achieve satisfactory results by manually selecting the base point or object (we call this area the ROF: Region of Fixation). Our algorithm [8] selects the ROF automatically; it is robust on noisy films and avoids false results for the most difficult sequences.

The main steps of automatic stabilization are:

1. The whole image is selected as one ROF to be used for vibration estimation.
2. The motion of the ROF is estimated by the phase correlation.
3. Peaks of the inverse Fourier transform of the CPS are analyzed: if a unique peak is found then we suppose that a global motion vector can characterize the ROF. Uniqueness satisfies two constraints:
   a) The height of the peak is over a predefined P_{min} threshold (here 0.023).
   b) No other local maximum can reach a given percent of H_{peak} (cc.30%), the height of the peak. Otherwise the ROF is divided into sub-regions, in a quad-tree manner, until an adequate peak is found or the ROF size reaches a lower limit. D_{max}, the maximum depth of the quad-tree (in our tests it is 3), determines this limit.

To achieve better performance and optimal resource utilization at line-resolution of 1K and over, the detection phase of the filter is performed on a downscaled version of the input frame while the correction phase is done on the original high-resolution image. For minimizing motion estimation errors we use phase correlation with sub-pixel accuracy. Figure 9 illustrates estimation results at different scale factors.

Testing on 2K and 4K sequences we found that if the detection is done on 4:1 downscaled frames the results of stabilization does not differ considerably from the ones done on the original scale (Figure 9 is a typical result of such a comparison). This amount of downscaling leads to reasonable gain in processing time: when processing a 1K resolution at 24bit color-depth image with an Intel P4@1.8GHz 4:1 downscaling reduces computation time from 7.3sec/frame to 0.9sec/frame [8].

The parameters and output of the filter are also stored in XML for learning and analysis purposes. For an example layout see Figure 10. The data the filter outputs into the XML database are then used for multiple purposes. It provides a way to correct higher resolution videos by using detection data obtained from downscaled versions of the same video. It also provided a way to generate statistics and to teach neural networks on the used parameters, providing automatic intelligence into the algorithm.

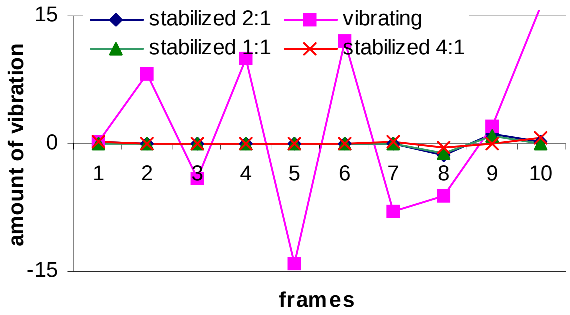

Figure 9. Results of the stabilization (of the “Camera” sequence [10]) with different downscaled frames. “Vibrating” stands for the original input, “stabilized” stands for the differently downscaled stabilized outputs.

3.1.3 Automatic deblotching with neural networks

Blotch detection via neural network can be implemented with a limited effort into the existing manual assigning/deleting/confirmation process. It can substitute the operator’s time-consuming efforts on checking small patches and greater areas for correction. If we extract and learn the average parameters of typical blotches at the beginning of the scenes of the film, we could automate the process of recognizing the rest of the blotches in that scene. We define a set of parameters measured by computer evaluation, and these parameters are fed into a NN when the learning process is controlled by the manual assigning/confirming procedure

```
<Task_info result_folder="F:\result\f07\s_s002\">
<Seq_info>
  <input_seqs_0 seq_ext_0="cin" no_frames_0="515"
. . . </input_seqs_0>
  <output_seqs_0 seq_ext_0="cin" no_frames_0="515"
. . .</output_seqs_0>
  </Seq_info>
<Job_info>
<Job Name="s_" SOI_string="0-514">
<Filters>
<Filter FName="Stabilization">
  <Param PName="Mode">0</Param>
  . . .
</Filter>
</Filters>
</Job>
</Job_info>
</Task_info>
```

Figure 10. Stabilization XML output.

At every frame, when the operator analyzes the blotches by hand, he decides to accept or decline the blotches found by the SDI filter. After several good and bad learned examples, the neural network is capable to decide the validity of the blotches. So before the filter draws the next frame, the image will be reanalyzed and some of the blotches will be removed as the result of the neural network decision. The following parameters were used for the teaching and deciding process: blotch area, inner intensity and variance of the blotch, outer intensity and variance around the blotch, slope of the blotch edge.

For later use, every measured parameter of the blotches will be automatically saved to a properly formed XML stream (Figure 13). After some frames with teaching we will have a trained neural network, which will remove most of the invalid blotches. This database can also come from another scene or other parts of the film. Naturally new blotch information will be added to the XML database in that case.

```
<database>
<blotchdata date="12:35:43 Saturday, April 03, 2004">
<blotch area="692" maxcol="221" mincol="0" contrast="221"
innerR="128" innerG="130" innerB="111" outerR="72"
outerG="68" outerB="51" diameter="0" shapefrm="0.72112"
validblotch="0" />
</blotchdata>
</database>
```

Figure 13. Automatic deblotching filter output.

3.1.4 Filtering small blotches

Unlike in the case of normal or large blotches, motion based detectors cannot be used for very small intensive dirt. Blotches of only a few pixel in size (2 – 10 pixels in diameter) are too small for traditional block-based motion estimation on 2K images. Therefore we developed a less computationally intensive method based on image morphology and neural network which works on a single image without any temporally information such as motion vectors.

In the case of small blotches the following problems arise. Small blotches can be present in a very large number (even hundreds per frame) therefore manual selection cannot be a viable option. Secondly, on some occasions even the operator cannot tell whether a blotch is a real one and should be filtered or just a natural shining or detail of the image.

Due to the versatility of small dirt (featured by its form, shape, average intensity, variation, slope, etc.) and the complexity of the background (buttons, shining of the eyes, shiny textures, etc.) many parameters are required for the right selection, but an operator cannot easily set 6-10 detection parameters reasonably easy and quickly. Instead we decided to use the following scheme:

Two phases are applied: 1. teaching a neural network (NN) through right and false parameter settings to select real blotch examples and false ones; 2. detection and filtering by the recognition of the trained neural network.

In the first phase a morphological step uses only 3 parameters, which can directly be set by the operator: median radius, threshold, blotch size threshold. The main operations of the morphological preprocessing step are as follows: take the original image, generate a median filtered version of it, and obtain their difference and threshold this difference. With the help of different adjustments of this filter we can get good and false blotch candidates, which will be the training input of the NN.

In the second phase the operator runs the morphological algorithm with a setting to minimize the false negative rate then applies the NN to eliminate false positive candidates. The parameters used to teach the neural network are as follows: the maximal horizontal and vertical intensity change inside the blotch area, local internal intensity contrast inside the blotch area, local internal mean and variance of the blotch area, and 3 parameters from the pixels neighboring the blotch candidate: external local contrast, mean and variance.

The performance of the NN greatly depends on the right selection of the positive and negative samples as a function of the experience of the operator. In the initial case, as a result of the morphological blotch detection, the rate of false positive selections is generally high but there are only a very few false negative cases. The purpose of the NN recognition is to decrease the false positive rate drastically while the false negative rate should be kept as low as possible. How does it sound in numbers ? In a general example (where there was 50-150 real blotches per frame, see Figure 14a-b for a part of a 2K image) the rates are the following:

1. Results of the initial morphological detection:
   a. False negative: 2-10/frame
   b. False positive: 100-300/frame
2. Results after applying the NN filtering:
   a. False negative 4-15/frame
   b. False positive 10-50/frame.

Figure 14a. Input image for Figure 14b.

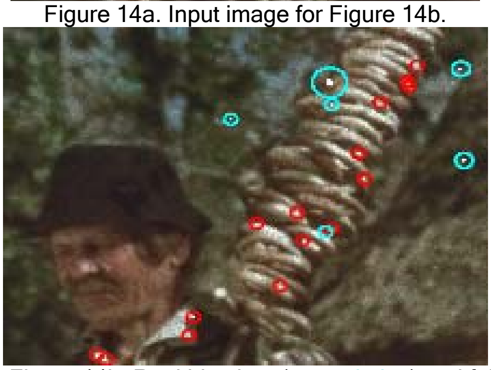

Figure 14b. Real blotches (cyan circles) and false candidates (red circles) classified by the neural-network.

Thus we could reduce the amount of risky information loss with filtering (removing image details) to appr. 10-16% while we ignore only less than 10% of hardly visible small dirt.

4. Experiences and future work

At the time of writing this paper the software system is after a real life industrial probing phase, when we are just finishing the restoration process of the first Hungarian color film (Lúdas Matyi, as mentioned in the introduction). More than 100 minutes of archive film are already restored at 2K and hundreds of tests are carried out at lower resolution. The algorithms run at about 10-15sec/frame on a usual 2GHz Intel P4 PC at 2K 48bit frames in batch mode. Since no user interactions are needed for long hours when running these filters, human operators can concentrate on other more difficult tasks requiring manual operations. Possible problems already encountered require further enhancements:

- Deflickering: the removal of very serious color flickering can cause de-saturated images since it tries to balance color variation in time leading to an average gray color. To avoid this problem the amount of color modification should be limited with a gain parameter.
- Scratch removal: manual frame-by-frame checking/overview are required to avoid the unwanted correction of image content.
- Color indexing and retrieval: the effect of fading and other color artifacts should be analyzed.

For better visual presentation of results please visit [11].

Acknowledgements

We would like to thank Dávid Piller for his contributions in blotch detection. This paper is based on research supported by NKFP-2/049/2001 of the Ministry of Education, Hungary.

References

[1] D. Androutsos, K. N. Plataniotis, A. N. Venetsanopoulos: A Novel Vector-Based Approach to Color Image Retrieval Using a Vector Angular Based Distance Measure, *Computer Vision and Image Understanding*, 1999, vol. 75, 1-2: pp 46-58.

[2] M. Bölecz, L. Czúni, B. Gál, A. Hanis, L. Kovács, B. Kránicz, A. Licsár, T. Szirányi, I. Kas, Gy. Kovács, S. Manno:: DIgital MOtion Picture Restoration System for Film Archives (DIMORF), A complex solution for film scanning, processing and recording, Proc. of the IBC, Amsterdam, 2003, pp. 509-517.

[3] L. Czúni, A. Hanis, L. Kovács, B. Kránicz, A. Licsár, T. Szirányi, I. Kas, Gy. Kovács, S. Manno: Digital Motion Picture Restoration System for Film Archives (DIMORF), *SMPTE Motion Imaging Journal*, May/June 2004, pp. 170-178.

[4] A. Hanis, T. Szirányi: Measuring the motion similarity in video indexing, In Proceedings of 4th EURASIP, Vol. 2. pp.507-512, Zagreb, Croatia, 2-5 July 2003.

[5] A. Kokaram, et al.: Robust and Automatic Digital Restoration Systems: Coping with Reality, Proc. of IBC, 2002.

[6] A. Kokaram: Motion Picture Restoration, Springer-Verlag, 1998.

[7] C. D. Kuglin, D. C. Hines: The phase correlation image alignment method, Proc. of International Conference on Cybernetics and Society, IEEE, 1975, pp. 163-165.

[8] A. Licsár, L. Czúni, T. Szirányi: "Adaptive Stabilization of Vibration on Archive Films", LNCS, CAIP 2003, Vol. LNCS 2756, pp. 230 – 237, 2003.

[9] P.M.B. Van Roosmalen, Restoration of Archived Film and Video, PhD Thesis, Universal Press, 1999.

[10] A. Licsár, T. Szirányi: "Hand Gesture Recognition in Camera-Projector System", International Workshop on Human-Computer Interaction, Lecture Notes on Computer Science, Vol. LNCS 3058, pp.83-93, 2004

[11]http://www.knt.vein.hu/~icpr04