Available online at www.sciencedirect.com
SCIENCE DIRECT®
Forensic Science International xxx (2006) xxx–xxx
Forensic Science International
www.elsevier.com/locate/forsciint

A forensic image processing environment for investigation of surveillance video

M. Jerian [a], S. Paolino [b], F. Cervelli [b,*], S. Carrato [a], A. Mattei [b], L. Garofano [b]

[a] *DEEI, University of Trieste, v. Valerio 10, 34100 Trieste, Italy*
[b] *Raggruppamento Carabinieri Investigazioni Scientifiche, Strada Fonderie 10, 43100 Parma, Italy*

Received 9 June 2006; accepted 14 June 2006

Abstract

We present an image processing software suite, based on the Matlab environment, specifically designed to be used as a forensic tool by law enforcement laboratories in the analysis of crime scene videos and images. Our aim is to overcome some drawbacks which normally appear when using standard image processing tools for this application, i.e. mainly the lack of full control and documentation on the operations which have been performed on the images, and the absence of new, more sophisticated algorithms which can provide improved performances and "make the difference" in critical cases.

Keywords: Surveillance video; Image processing; Video processing; Open architecture

1. Introduction

Video-surveillance systems are one of the main source of information during investigations, thanks to their wide-spread and increasing presence in our countries. However, the adopted closed-circuit devices are often affected by poor quality mainly because of economical and practical problems. Although this fact let us reflect if they can be considered more a deterrent for criminal actions rather than a valid identification system, in many cases also a low quality image can give useful information both during the first phase of the investigation and in courtrooms.

As a consequence, the images and sequences coming from video-surveillance systems need to be digitalized in order to be processed by dedicated software to enhance features useful for crime analysis. Generally, this is done either to reduce the different kinds of corruptions that have been introduced in the acquisition, conversion, and storage processes of the data or to overcome the limits of the overall system.

The characteristic problems to deal with are:

- low resolution of the images, which often implies the need to increase the size of the interesting details;
- lack of contrast;
- different types of noise or disturbances;
- blurring caused by motion or lack of focus;
- jitter or misalignment of lines due to the wear of video cassette recorder (VCR) heads;
- geometric distortions, thus severely limiting the reconstruction of the dimensions of the objects inside the image (e.g. the numerical estimation of the biometric features of subjects).

Each operation performed must be logged and certified; moreover, a complete knowledge of each step of the applied algorithms is needed to obtain full objectivity and to guarantee that the same result can be obtained following that exact procedure by anyone anywhere. Consequently, in theory the use of commercial software may not be suited from a legal point of view, due to the fact that its source code is usually not available for inspection. Open-source software, such as GIMP [1], meets this need, since each single part of the process is totally visible and open to scrutiny. However, open-source programs may not be equipped with the latest operators which have been appeared in the literature for common tasks (e.g. zooming, image noise

[*] Corresponding author. Tel.: +39 0521 537732; fax: +39 0521 206396.
E-mail address: cervelli@df.unipi.it (F. Cervelli).

0379-0738/$ – see front matter © 2006 Elsevier Ireland Ltd. All rights reserved.
doi:10.1016/j.forsciint.2006.06.048

reduction), or may lack tools to treat very specific problems such as recovery of a video recorded by a defective VCR.

In this paper, we present what we have called the Modular Image Processing Environment (MIPE). This software is the result of the collaboration between the developers and the end users to face the aforementioned problems. After a brief review on the state-of-the-art systems (Section 2), we describe the adopted software architecture and the operators that have been already implemented in the system (Section 3), showing some examples of their application (Section 4). We conclude presenting the current limitation of the system and describing the future of MIPE, which is, in our aim, the prototype of a more ambitious project, namely a new forensic image processing suite.

2. State-of-the-art

Presently, several products exist on the market which are dedicated to the analysis of image-based information for forensic science applications; we can recall, in no particular order and with no intention to be complete, dTective by Avid and Ocean Systems [2], Impress by Imix [3], StarWitness Video by Signalscape [4], Video Analyst by Intergraph [5], and Video Investigator by Cognitech [6].

Because of the strong correlation between the evidence source and the processing software, forensic image analysis products often come as a complete software–hardware solution. These systems perform, as a first step, the acquisition of a secure digital copy of the evidence video or images in order to prevent any possible damage of the original. The second step is dedicated to processing, and the vendors normally offer different solutions to satisfy the forensic image professional needs to analyze, enhance and edit all major image and video formats that may be constitute the evidence source.

Commercial systems offer different software capabilities (e.g. available filters, algorithms and proprietary operations, courtroom oriented functions) and hardware characteristics (e.g. installed PC RAM, graphics and acquisition cards, available media readers, additional equipments such as VCRs and printers). Here we are interested mainly in the processing software part of an ideal forensic image analysis product, and the aim of MIPE is to become an affordable application for image restoration and enhancement in forensics, using validated, bleeding edge, widely accepted and open to scrutiny image processing algorithms to extract the relevant information from the recorded sequences.

All the above mentioned programs can perform the basic image editing operations: contrast and brightness adjustment, histogram equalization and editing, zooming, mirroring and rotation of the images under analysis. Some of them offer the ability to customize filters kernel and offer proprietary filters (such as JPEG dedicated deblockers). Another example are motion deblurring filters, which allow the operator to restore the details of a moving object. Inter-frame operations can also be performed, for example, employing time information recorded by the sequence in order to gain what would be otherwise lost in just a single frame. Demultiplexing algorithms are also available, so that different camera views can be converted into video clips (see Ref. [7] for a review about video contents indexing) thus allowing the isolation of the crime scene, and deinterlacing. Finally, each case can be fully organized thanks to archive and back-up facilities, and each program creates an audit log for each case, in order to make it suitable for presentations in courtrooms.

These analysis systems offer the ability to record each step of the image processing operations on a log file, however they do not guarantee complete access to the applied algorithms. In fact, although the performed operations are recorded together with their parameters, the employed algorithms may be not public.

3. The proposed system

MIPE is the system born as an answer to the above mentioned problems. Its development follows some basic guidelines that can be resumed as follows:

- complete control and knowledge on the processing which is done on each image;
- employment of state-of-the-art algorithms;
- development in strict collaboration with the final user;
- high modularity, in order to grant easy customization of the software.

For our purpose, probably the first point is the most important: in order to grant a transparent and objective result during legal procedures, the user (and the court) needs to know all the details on the processing applied to an image. As expressed in Ref. [8] it is not possible to make a clear distinction between what is called enhancement and the manipulation of the image. The best we can do is to use techniques which are widely accepted by the scientific community and give all the details on their implementation in order to make the process completely repeatable.

A complete transparency is achieved by these means:

- all the algorithms applied to an image are listed in the right sequence on a log file automatically saved with the image;
- for each employed algorithm all the involved parameters are listed;
- for each algorithm (if not a very standard one) the reference article is provided;
- for each algorithm the source code is provided.

Using bleeding edge techniques and at the same time giving the implementation details leads not only to objective results, but also better performances if compared to traditional systems.

In order to meet the above expressed requirements, Mathworks Matlab 7 [9] has been chosen as development environment. First of all, the style of programming is very clear (almost like a pseudo-code), allowing people with even very limited programming skills to understand and manage the code. All its functions are very well documented and any customization of the code is straightforward; moreover, there is a wide number of ready-to-use functions, either built-in in the basic environment or available in one of its various toolboxes. A very useful feature is that almost all its functions (except the very basic and computationally critical ones) are visible.

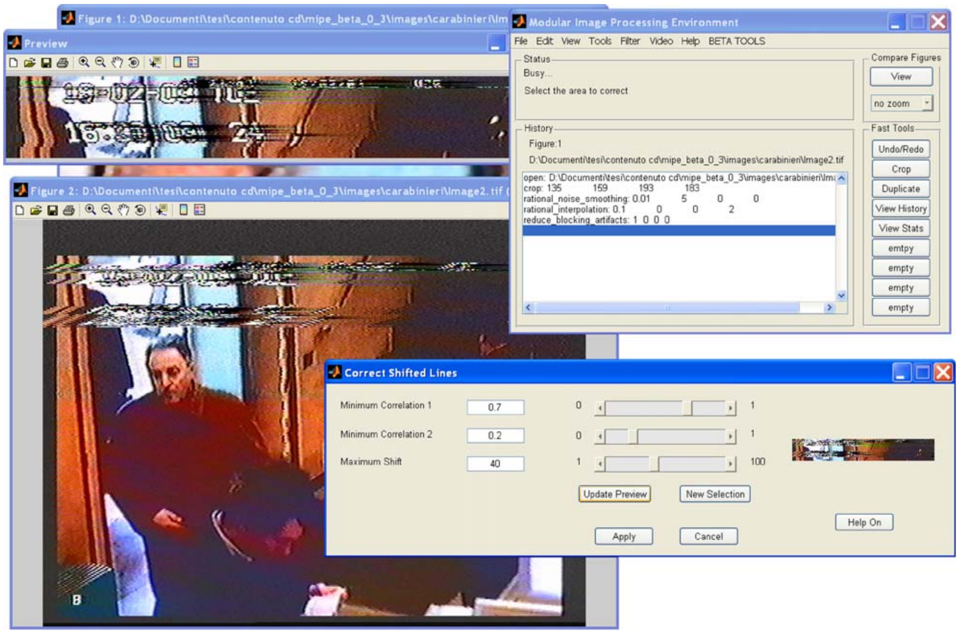
Figure 1: D:\Documenti\tesi\contenuto cd\mipe_beta_0_3\images\carabinieri\Im
Preview
Figure 2: D:\Documenti\tesi\contenuto cd\mipe_beta_0_3\images\carabinieri\Image2.tif
Modular Image Processing Environment
File Edit View Tools Filter Video Help BETA TOOLS
Status
History
Compare Figures
View
no zoom
Fast Tools
Undo/Redo
Crop
Duplicate
View History
View Stats
emtpy
empty
empty
empty
Correct Shifted Lines
Update Preview
New Selection
Help On
Apply
Cancel
B

Busy...

Select the area to correct

Figure:1

D:\Documenti\tesi\contenuto cd\mipe_beta_0_3\images\carabinieri\Image2.tif

```
open: D:\Documenti\tesi\contenuto cd\mipe_beta_0_3\images\carabinieri\Im
crop: 135          159          193          183
rational_noise_smoothing: 0.01          5          0          0
rational_interpolation: 0.1          0          0          2
reduce_blocking_artifacts: 1  0  0  0
```

Minimum Correlation 1 0.7 0 1

Minimum Correlation 2 0.2 0 1

Maximum Shift 40 1 100

Fig. 1. Screenshot of the proposed Modular Image Processing Environment.

Another interesting characteristic of the Matlab environment is the portability of the code, which can be used with few (or none) modifications on Microsoft Windows, Apple Mac OS X and Linux, being it interpreted on run-time; this latter feature also allows very fast debugging (the source can be also compiled for a specific platform, but in this way we loose the possibility to modify it on the fly). Consequently, the time needed for the development and the customization of the system is very short if compared to lower level languages. Moreover, Matlab also provides easy and powerful tools for graphical user interface (GUI) creation.

In order to give the user the possibility of customizing the software, a common processing interface has been created, shared by most of the processing functions. Following some very simple guidelines it is possible for the user to easily interface a custom-created function to the rest of the system, both from a lower level and with respect to the integration in the existing GUI. In the present version, the fast integration is allowed for functions which accept up to three numerical parameters and one optional string that can be configured for various purposes through a choice-box. For each needed parameter it is necessary to set the minimum, the maximum and the default value, while for the options box it is required to enumerate possible values for the string. Depending on the number of used parameters, the main program will draw the right number of elements (i.e. scrollbars and textboxes) in the dialog box, together with the controls for the preview image. The function will look then perfectly integrated, as if it was belonging to the original system.[1] Of course, also the user-created functions will automatically add their details

Table 1
Main features implemented in MIPE

Basic editing	Crop Luminance and contrast Grayscale conversion Flip/rotate by 90°
Interpolation	Pixel replication Bilinear Bicubic Rational [13] LoWaDi2
Image restoration	Shifted lines correction Lens distortion Perspective correction Gaussian blur Motion blur
Filters	Median Usharp masking High-pass Rational sharpening [14,15] Averaging Low-pass Rational noise smoothing [14] Blocking artifacts reduction [10] Add noise (for test purpose)
Multiple images	Median Mean Simple registration Projective registration
Video	Import frame Save all frames Demultiplexing Save demultiplexed sequences
Deinterlacing	Bilinear Bicubic Rational [13] Advanced [16]

[1] Actually, another minor operation is needed to make the function to appear in the desired menu in the main window of the program. This can be easily done by Guide, the GUI design tool provided with Matlab.

on the log file. In Fig. 1 a screenshot of the proposed Modular Image Processing Environment is presented as an example.

In Table 1, most of the processing functions implemented in MIPE are presented. In some cases, different techniques are provided for performing the same tasks, although some of them may have not been deeply tested or are present just in order to compare the performances of various algorithms. Some support functions, such as the Undo/Redo functions, the possibility of easily compare two images and to display image information (size, bit depth, signal-to-noise ratio, . . .) are not mentioned in the table.

4. Experimental results and discussion

Some of the various filters and tools available in MIPE are presented here as examples, to show the potentialities of the system.

Some special algorithms have been specifically designed to cope with the rather frequent problem of the shift of subsequent lines in images due to the wear of VCR heads. They have been tested for two particular cases (one is presented in Fig. 2); however, it must be noted that it is very difficult to solve this problem in general, since the characteristics of the disturbance may vary widely from case to case.

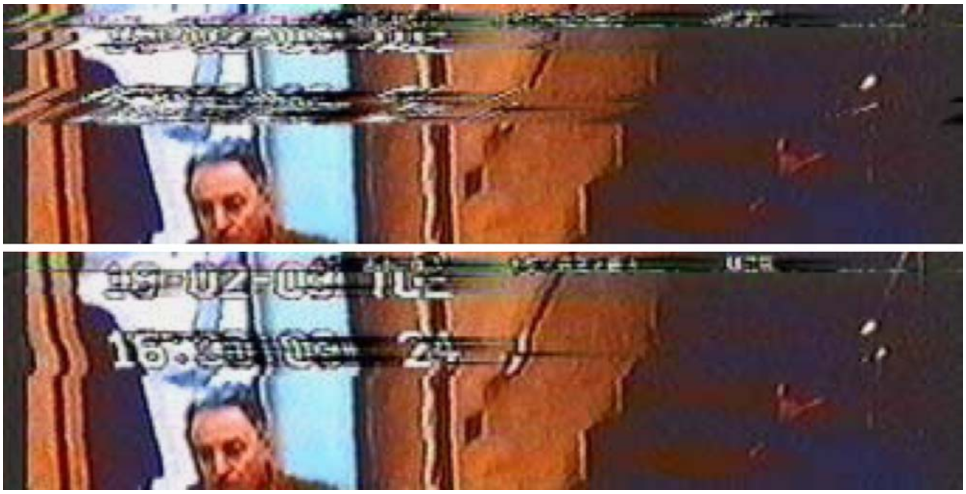

Fig. 2. Correction of shifted lines: (top) original and (bottom) processed image.

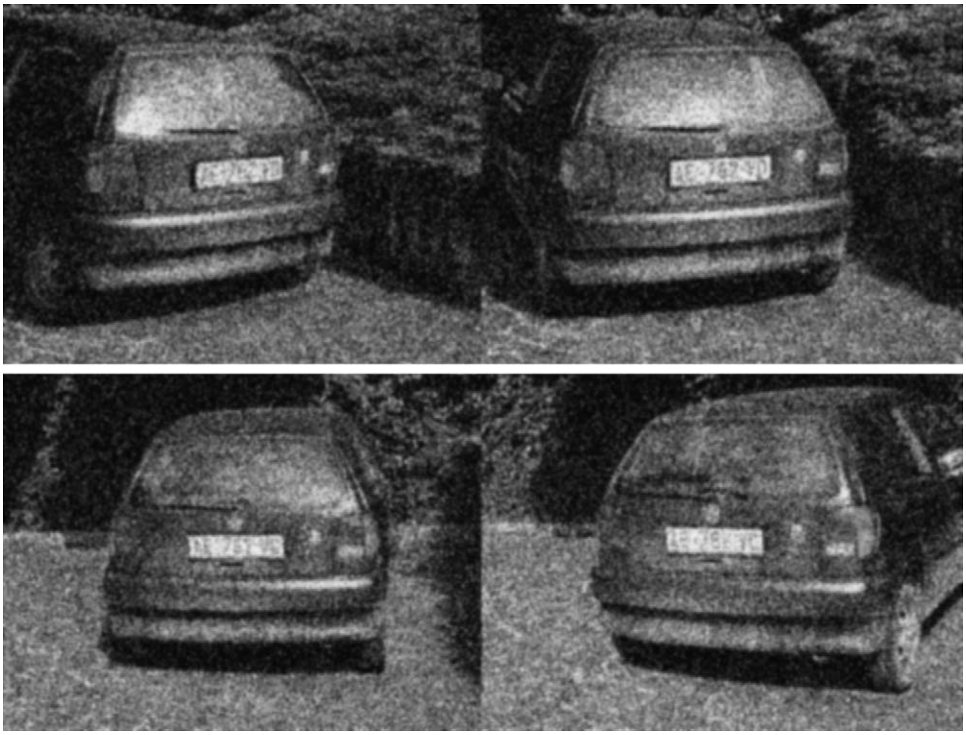

Fig. 3. Projective registration: four images from the original set.

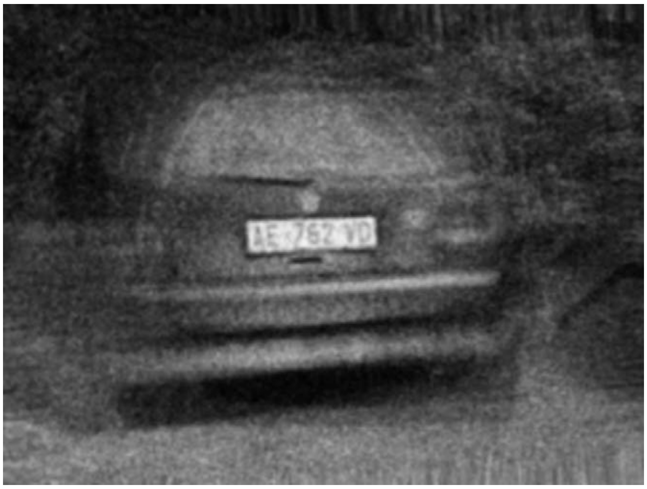

Fig. 4. Projective registration: result of frame averaging on transformed images.

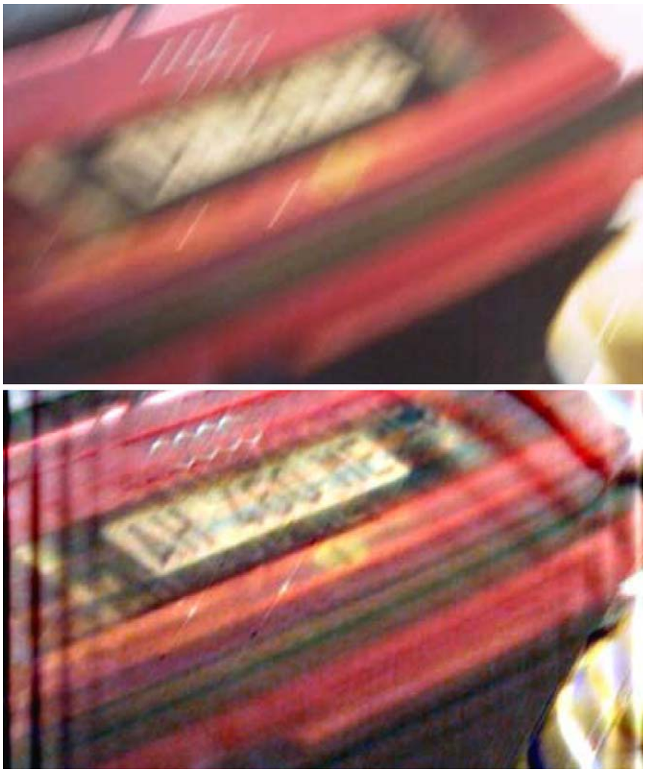

Fig. 5. Motion deblurring: original (top) and processed (bottom).

Another useful tool realized for MIPE let us perform frame averaging of an object which is present in different frames and captured under different point of view. Commonly available techniques allow to frame-average objects translated parallel to the image plane in consecutive frames, while more advanced algorithms also allow image rotation and rescaling. Our approach is more general and the projective registration tool lets us frame-average the object of interest (for example a car license plate) even if subject to different perspectives. The example in Fig. 3 shows how the combination of only few frames can lead to good results. Indeed, in Fig. 4 the license plate characters are easily readable.

A more common feature, that sometimes offers really good results also in the worst cases, is the implementation of motion deblurring through one of the deconvolution functions of the Matlab Image Processing Toolbox. It is easy to note in Fig. 5 that the license plate has been recovered from an apparently useless image.

Very frequently, images and sequences are stored in a format that implies a lossy compression, which causes loss of details and the introduction of artifacts on the image. MIPE offers a filter, based on Ref. [10], which is able to efficiently reduce the blocking artifacts caused by many common block-based compression tools, e.g. JPEG. An example is presented in Fig. 6.

In order to apply geometric transformations to an image with the minimum possible loss of detail, some new interpolators have been studied. In Fig. 7 our algorithm, applied to an 8× enlargement, is compared to the classic bicubic interpolation. The employed technique is an evolution of the WaDi algorithm presented in Ref. [11]. Other examples taken from the features of MIPE are presented in Ref. [12].

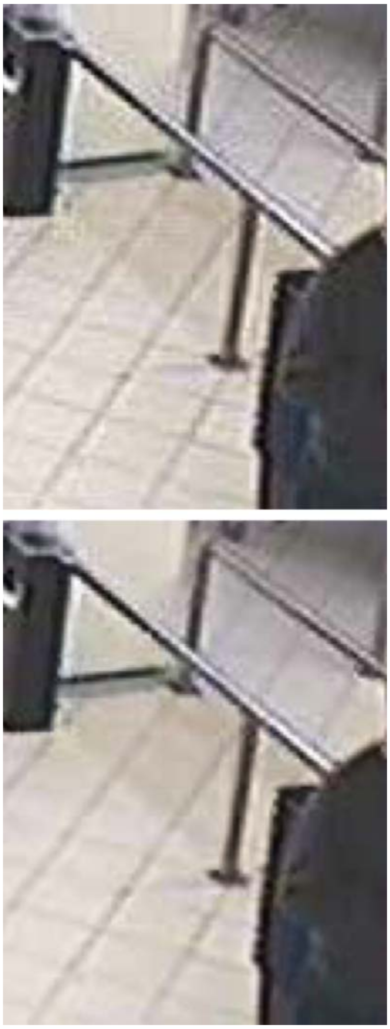

Fig. 6. Compression artifacts reduction: original (top) and processed (bottom).

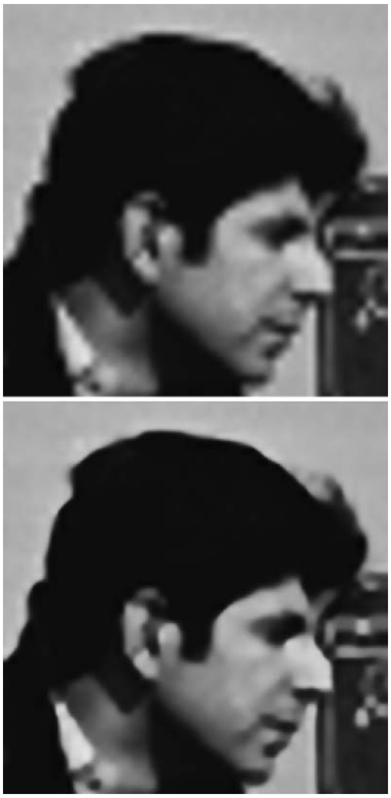

Fig. 7. Interpolation for 8× zooming: bicubic (top), MIPE algorithm (bottom).

5. Conclusions

In this paper, a general introduction on forensic image processing and its main issues has been exposed, and a new environment for the enhancement of images coming from video-surveillance devices has been proposed.

The principles which have guided our work and their practical applications in the attempt to meet, as close as possible, the forensic image professional requirements have been depicted. Matlab has been chosen as the development platform thanks to its advanced features, the ease of programming and the possibility to work directly on interpreted functions, making the code visible and easily modifiable, thus obtaining an open to scrutiny software which can better fit legal needs. Some general features of the Modular Image Processing Environment, the software born from this work, have been described and some parts of it have been presented as examples.

Being developed in Matlab, MIPE suffers from the drawbacks of this environment, i.e. mainly the speed of execution and possible problems with memory when working with large images. In order to build a better system for the end user, we have recently started to develop a novel image processing tool written as a native application, and not as a Matlab interpreted software, while using the Matlab-based MIPE as a prototyping and testing environment. It will offer the already available algorithms a wide range of new features with respect to traditional image processing software: full control on editing the processing history, seamless integration of different kind of data (still images, groups of images, video sequences), and automatic report creation.[2]

References

[1] GIMP, http://www.gimp.org.

[2] dTective, http://www.avid.com/forensic.

[3] Impress, http://www.imix.nl/.

[4] SignalScape, http://www.signalscape.com/.

[5] Video Analyst, http://solutions.intergraph.com/hardware/vas/.

[6] Video Investigator, http://www.cognitech.com/.

[7] C.G.M. Snoek, M. Worring, Multimodal video indexing: a review of the state-of-the-art, Multimedia Tools Appl. 25 (1) (2005) 5–35.

[8] House of Lords, Digital images as evidence, in: House of Lords, Science and Technology, Fifth Report, http://www.parliament.the-stationery-office.co.uk/pa/ld199798/ldselect/ldsctech/064v/st0503.htm.

[9] Matlab, http://www.mathworks.com.

[10] R. Castagno, S. Marsi, G. Ramponi, A simple algorithm for the reduction of blocking artifacts in images and its implementation, IEEE Trans. Consumer Electron. 44 (August (3)) (1998) 1062–1070.

[11] G. Ramponi, Warped distance for space-variant linear image interpolation, IEEE Trans. Image Process. 8 (May) (1999) 629–639.

[12] M. Jerian, S. Carrato, S. Paolino, F. Cervelli, A. Mattei, L. Garofano, Development of an image processing open-source software for video-surveillance devices, in: International Symposium on Imaging for Crime Detection and Prevention, London, United Kingdom, June, (2005) 49–54.

[13] G. Ramponi, S. Carrato, in: Proceedings of ICIP'97, Santa Barbara, California, October, vol. I, (1997) 389–392

[14] G. Ramponi, S. Marsi, S. Carrato, Image processing using rational functions, in: C. Kotropoulos, I. Pitas (Eds.), Nonlinear Model-Based Image/Video Processing and Analysis, John Wiley & Sons, New York, (2001) 52–92

[15] G. Ramponi, A. Polesel, A rational unsharp masking technique, J. Electron. Imaging 7 (August (2)) (1998) 333–338.

[16] R. Simonetti, A. Polo Filisan, S. Carrato, G. Ramponi, G. Sicuranza, A deinterlacer for IQTV receivers and multimedia applications, IEEE Trans. Consumer Electron. 39 (August (3)) (1993) 234–240.

[2] All trademarks and copyrights are the property of their respective owners.