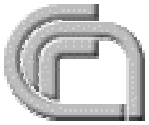
Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

An Interactive Distributed Environment for Digital Film Restoration

F Collura[2], A Machì[1], F. Nicotra[1]

RT-ICAR-PA-03-10 **dicembre 2003**



Consiglio Nazionale delle Ricerche, Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR)
– Sede di Cosenza, Via P. Bucci 41C, 87036 Rende, Italy, URL: *www.icar.cnr.it*
– Sezione di Napoli, Via P. Castellino 111, 80131 Napoli, URL: *www.na.icar.cnr.it*
– Sezione di Palermo, Viale delle Scienze, 90128 Palermo, URL: *www.pa.icar.cnr.it*

Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

An Interactive Distributed Environment for Digital Film Restoration

F Collura[2], A Machì[1], F. Nicotra[1]

Rapporto Tecnico N.: **RT-ICAR-PA-03-10**

Data: **dicembre 2003**

---

[1] Istituto di Calcolo e Reti ad Alte Prestazioni, ICAR-CNR, Sezione di Palermo
[2] Tesista Università degli Studi di Palermo Dipartimento di Ingegneria Informatica.

An Interactive Distributed Environment for Digital Film Restoration

F Collura, A Machì, F. Nicotra

ICAR Consiglio Nazionale delle Ricerche, Sezione di Palermo
Viale delle Scienze 90128 Palermo Italy
colf @ medialab.pa.icar.cnr.it, {machi, nicotra }@ pa.icar.cnr.it

Abstract. The paper presents FESR an interactive environment enabling collaborative digital film restoration over a digital network by connecting seamless the supervisor and operator's graphical frameworks, a parallel Image Processing server, a video stream server and a DBMS server. It allows a supervisor to remotely define the restoration protocol and to monitor job progress by using a metadata-driven navigation interface. The operators use a graphical desktop to interactively perform parameter steering of restoration filters and to activate and synchronize batch processing of defect detection and restoration filters on the IP server. The video stream server stages digital frames while the meta-data server stores defect masks and other film attributes by using MPEG7 formalism for video-segments representation. System prototype architecture, inspired to component technology is described.

1 Introduction

The animated documentation of a lot of historical events and of most part of cinema and TV production of XX th century is registered in movies on celluloid film ribbons. These films downgrade over time because of chemical changes due to film aging and to inappropriate ribbon handling and environment. Their recovery is essential for preservation of cultural heritage but restoration is highly costly process.
During the last ten years several studies have been published concerning (semi) automatic digital restoration of degraded video and film image sequences. Algorithms and methods have been developed to detect and correct abnormal luminance fluctuations, line or frame misalignments, to detect and restore frame areas damaged because of dust, mould or dirt, excessive heating, improper drag, or just film aging [1-3].

A number of projects have been founded by the European Commission aimed to develop software systems supporting automated restoration [4-5], and to implement parallel algorithm libraries reducing time constraints of heavy computations [6]. Other projects have concentrated their efforts in developing tools and environments for minimizing the time spent by the operator to handle the huge quantity of temporary date produced [7].
In semi-automatic approaches to digital video restoration, the restorer browses the video and bounds, around damaged frames, video sections on which to apply

appropriate sequences of spatial-temporal filters. Smoothness of change in scene properties around the damaged frames is essential for proper operation of filters. Of course, the reconstruction fails whenever scene changes abruptly or any scene component moves too quickly [8]. If a scratch spreads on more frames, the support area should be restricted inside the frame under restoration. If the defect area is large, its local neighbourhood should not be included in the support area to obtain high quality reconstruction [9-10].

Fig. 1. Blotch restoration: defected input (left), defect mask (center), frame restored (right)

Adaptive algorithms need to be tuned for proper operation .The paper presents FESR a new interactive environment enabling collaborative digital film restoration over a digital network with express support of parameter steering.
In FESR approach the supervisor browses the video, defines the restoration protocol by remotely marking sections on which to apply appropriate series of spatial-temporal filters and monitors job progress by using a metadata-driven navigation interface. The operators use a graphical desktop to interactively perform parameter steering of restoration filters and to activate batch defect detection and restoration on a parallel Image Processing server. After the automatic processing step, he browses the restored sequences and identifies unsatisfactory results. Badly reconstructed frames are then restored by in painting or reprocessed with a different series of filters.

1.1 Digital Film Restoration: supervised process description

Fig 2 outlines the phases composing a supervised digital motion picture restoration process. The damaged motion picture is firstly digitized from its original film tape support. Each picture frame is digitized in a high definition format (over 2k by 2k pixels) and stored in high-density digital tapes.
Four digital processing steps then follow and finally the restored version is saved on its final digital (DT, DVD) or analog (VHS tape, film) support. In the following we will use term film to refer a motion picture even in its digital form.
Digital restoration includes a preprocessing indexing step aimed to partition movie into sections homogeneous for scene temporal and semantic continuity.
Defect detection follows indexing. It is an iterative process in which a number of Image Processing filters are applied on a small number of sample frames seeking for best parameter tuning. After optimal configuration is obtained, automatic processing of an entire sequence is started.

The same steering procedure is applied before application of restoration filters and, in case of unsatisfactory reconstruction, reprocessing of difficult areas is worthwhile.

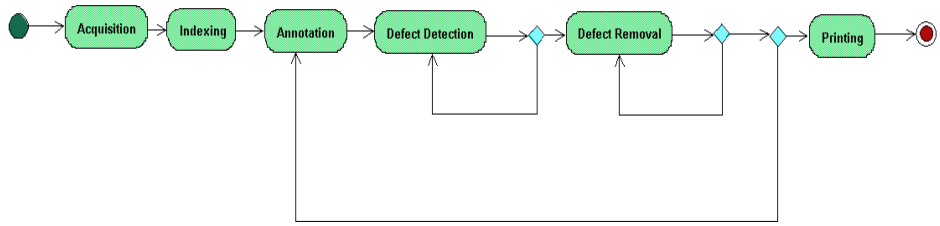


Fig. 2. Activities in the restoration process

In the FESR environment, a film is described by using an MPEG7 coding of its meta-data [11]. Its sections are represented as Video Objects whose features are labels for defect typologies, identifiers of image processing filter chains and URI references to temporary masks produced.
The supervisor draws the restoration plan for the entire film by annotating defect typology and required operations in the meta-description record of each defected section. He can also perform trial processing on short sections and annotate preliminary values for parameter settings.
Operators are responsible for contemporary processing of different sections. They can perform detection trials and steering by restricting processing to small group of frames or selected frame areas, or by locally modifying the suggested parameter set. Two processing modalities support steering: procedure calls on the local engine or high-priority request on the remote parallel server. Batch processing submission is provided for final sequence processing. Each operator maintains and updates a work-session-scoped copy of the meta-descriptor of its file section. He updates meta-description fields for found defects list and annotations, parameters settings used, final defect masks, temporary and final results of restoration.
The supervisor reviews operator's meta-descriptions of its working session and confirms or refuses results. Accepted sessions meta-data and objects replace (or are merged to) corresponding ones in sections of the official movie meta-description.

2. The interactive distributed environment

The process just described requires the environment to support patterns as event-driven programming, interactive graphics, quality-of-service processing, and session status management. FESR implements all such patterns though the coordination of several subsystems playing different roles and developed with component technology. Fig. 3 shows FESR subsystems cooperation diagram with component deployment on remote nodes interfaces and ports association. The application is developed as a three-tier client-server application with rendering interfaces (GUIs) on the user nodes, and computational, video-stream and meta-DBMS services deployed on the network.

The middle tier hosts the event subsystem, which coordinates user interaction from windows and adapters for local and remote procedure, calls (plug-ins and proxies).

It also hosts a memory engine able to store environment meta-data across procedure calls and to store and retrieve them from a DBMS service.

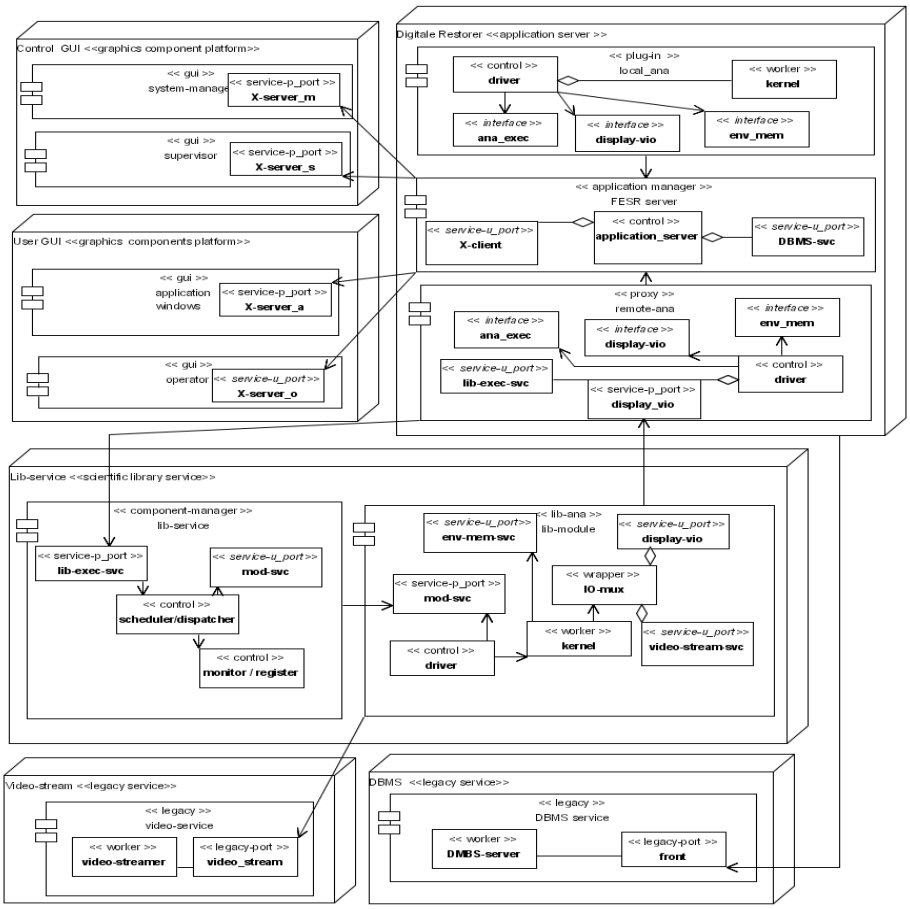


Fig. 3. FESR components cooperation diagram.

3.The event driven application server subsystem

This subsystem is composed by the graphics framework and by the restoration server
The windows of the interactive subsystem (GUI) may be made remote by using the intrinsic facilities of the underlying graphical X system.
Four windows arrangements are provided: *supervisor, operator, display and list*.
The *supervisor* window allows navigation of movie structure. Browsing and editing of Video Objects of type *Shot*, *Clip* and *Task* is supported.

The *operator* window manages parameter selection and activation of local or remote processing on a movie section. Three modalities are supported: *test*, *merge* results with previous run or replace session meta-data. The *display* windows may be connected to output streams from processing modules. Facilities for panning, zooming and snake-ROI selection are provided. Finally, the *list* window allows movie selection from movie database.

.**Fig. 4.** Snapshot of a supervisor window showing movie structure, one frame and annotations.

The GUI subsystem is implemented by using the Trolltech Qt [13] Graphics Toolkit over the X-Window graphic system.

Graphics events generated by user interaction with a window through the mouse and keyboard are managed as signals, registered by an *event-engine* and routed to other platform components registered for receiving them.

An analysis engine manages processing. Adapters for local processing of restoration modules (plug-ins) or remote procedure-calls on the processing servers (proxies) may be loaded at run time as dynamically linked libraries.

At loading time, the analysis engine reads a description of graphics requirements for window control and a description of the module activation signature through the adapter introspection interface. At activation time, the adapter reads relevant data from execution environment, calls methods of the controlled module and updates environment, according to the test/merge/replace policy selected by the user.

For remote processing the environment memory is reflected on the network through NFS. The analysis module gets direct access to video data streams from the Video-server.

An environment-engine manages persistence of meta-data throughout a session and their storage or retrieval from the DMBS server.

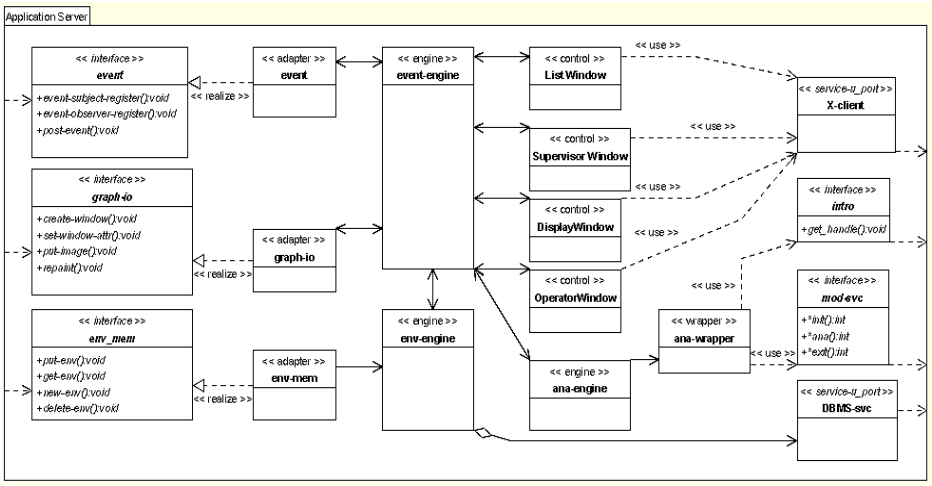


Fig. 5. Architecture of the application server.

4. The Image Processing parallel server

The parallel server offers Image Processing services according to two specific Quality of Service modalities: shortest-time and minimum-cost.
Shortest-time mode supports steering. Testing on short movie sections for parameter tuning needs near-real-time response and graphical rendering. Minimum-cost mode supports global optimization of the queue of long queue batch jobs during production.
The parallel server is implemented as a container of parallel components. Each parallel component is an instance of a data-parallel or stream parallel version of a detection or restoration module and is identified by its URI. Modules are implemented using skeleton *map* and *ordering-farm* templates [12,14]. A Directed Acyclic Graph represents workflow among Virtual Processors implementing the skeleton. Actual mapping of DAG to physical nodes at run-time realizes skeleton configuration [15].

The server manages a set of 2*n virtual nodes. Half of them are devoted to execute tasks according to the minimum-cost contract (batch), the other half to serve tasks according to minimum-time contract (interactive). Each physical node hosts one interactive and one batch process. When active, the interactive process freezes its bat partner. A scheduler manages proper partitioning of virtual nodes among tasks for optimal DAG mapping.
Fig. 6 sketches the architecture of the Parallel server composed by a command *acceptor*, a task *scheduler*, a job *dispatcher*, a *control* engine and *configurable skeleton* modules. Service requests are accepted by the *acceptor* and inserted in two queues. The low-priority queue accepts requests for minimum-cost tasks, the high-priority one minimum-time tasks.
The *scheduler* honors service requests according to a FIFO policy. It assigns to a ready-to-run task half of its interactive available nodes if it requires minimum-cost

performance, otherwise an optimal number of batch virtual nodes. Optimality is based on evaluation of performance based on a cost model for the skeleton implementation. The *dispatcher* performs actual mapping of skeleton DAG on the node set assigned by the *scheduler* and the *control engine* manages component life cycle on the server-distributed platform.
Parallel version of restoration algorithms implemented through the data-parallel *map* skeleton are normally activated for interactive processing, parallel versions implemented through the stream-parallel *ordering-farm* skeleton are normally activated for batch processing

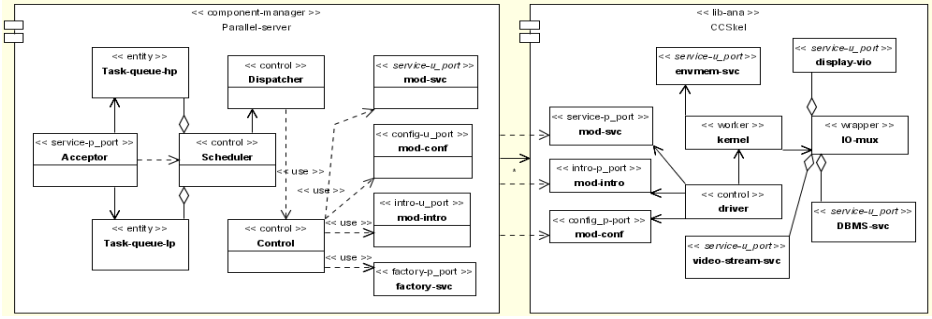


Fig. 6.Architecture of the Image Processing parallel server

5. The Video-stream and the metadata servers

The Video-stream server provides a simple video-on-demand service optimized for video-frame streaming according to a GET and PLAY modes. A GET command provides a single frame or part of it selected, at original resolution or sub sampled. A play command provides a continuous video stream of a movie section with flow control. The server is implemented with a multi-tread architecture. An acceptor thread queues client requests and routes them to worker threads.
The MPEG7 meta-data server stores and retrieves movie Video Objects

6. Discussion

A process model for digital restoration has been discussed with reference to parameter steering. Support from to the FESR environment to interactivity has been described. An event driven platform supports user generation of workflow control in interactive execution and batch submission modalities from local and remote servers and application coordination. Support for meta-data storage is provided at three levels: temporary in processing modules heap during steering, session-persistent in the application server's memory component, and definitely persistent on a DBMS server. The parallel server provides support for single-frame high-priority data-parallel

processing during steering and stream-parallel processing during batch task-queue optimization.
The FESR environment has been developed for supporting. digital film restoration but it is suited for supporting other applications with similar activity diagram as multimedia authoring or biomedical dynamic imaging. Full exploitation of the MPEG7 Video-Objects description format and adaptation of the distributed environment to the Virtual Organization model over a computational grid is in course.

Acknowledgements

The precious and skilled contribution of ideas to design and assistance in testing of FESR prototype from M. Tripiciano, G. L. Alfieri is acknowledged.

References

[1] P.Schallauer, A.Pinz, W.Haas, Automatic Restoration Algorithms for 35mm Film*, Videre: Journal of Computer Vision Research*, *1*(3), 1999,60-85,The MIT Press.
[2] A.C. Kokaram, *Motion Picture Restoration* London: Springer Verlag, 1998.
[3] L. Maddalena "Image Sequences Reconstruction of blue scratches" IEEE ICIAP11 Palermo Italy 2001,547-552.
[4] LIMELIGHT Project (1994-1997):
http//iis.joanneum.ac.at/iis/html_eng/projects/limelight/limelight.html
[5] AURORA Project: Automated Restoration of Original film and video Archives (1995-1998)
http://www.inafr/Recherche/Aurora/index.en.html
http://www.infowin.org/ACTS/RUS/PROJECTS/ac072.htm
[6] W. Plaschzug, G. Hejc "ESPRIT Project FRAME Public Final Report" http://www.hpcn-ttn.org/newActivityDetail.cfm? Activities_ID =1.,
http://www.vcpc.univie.ac.at/activities/projects/FRAME
[7] P. Shallauer, G. Thakllinger, M.J.Addis„DIAMANT Digital Film Manipulation Platform ",http://www.joanneum.at/cms_img/img509.pdf
[8].A. Machì, M Tripiciano "Video Shot Detection and Characterisation in Semi-automatic Digital Video Restoration" IEEE Proceedings 15th ICPR , Barcelona 2000, pp 855-859.
[9] Alberto Machi, Fabio Collura, Filippo Nicotra: "Detection of Irregular Linear Scratches in Aged Motion Picture Frames and Restoration using Adaptive Masks": IASTED SIP02,Kawai USA 2002
[10] Alberto Machi, Fabio Collura "Accurate Spatio-temporal Restoration of Compact Single Frame Defects in Aged Motion Pictures" IAPR Proceedings 12th International Conference on Image Analysis and Processing ICIAP12,Mantova Italy 2003 pp.454-459 IEEE 2003
[11]ISO/IEC JTC1/SC29/WG11 Coding Of Moving Pictures and Audio http://www.w3.org/2001/05/mpeg7/w4032.doc
[12] M. Vanneschi: The programming model of ASSIST, an environment for parallel and distributed portable applications. Parallel Computing 28(12): 1709-1732 (2002)
[13] The Qt graphics framework http://trolltech.com/poducts/qt.
[14] G. Sardisco, A. Machì "Development of parallel templates for semi-automatic digital film restoration algorithms. ParCo2001 Naples Italy 2001
[15]A. Machì, F. Collura "Skeleton di componenti paralleli riconfigurabili su griglia computazionale map e farm ".TR ICAR-PA-12-03 - Dec 2003.