Multimed Tools Appl (2008) 37:135–167
DOI 10.1007/s11042-007-0137-4

DocMIR: An automatic document-based indexing system for meeting retrieval

Ardhendu Behera · Denis Lalanne · Rolf Ingold

Published online: 29 June 2007

Abstract This paper describes the *DocMIR* system which captures, analyzes and indexes automatically meetings, conferences, lectures, etc. by taking advantage of the documents projected (e.g. slideshows, budget tables, figures, etc.) during the events. For instance, the system can automatically apply the above-mentioned procedures to a lecture and automatically index the event according to the presented slides and their contents. For indexing, the system requires neither specific software installed on the presenter's computer nor any conscious intervention of the speaker throughout the presentation. The only material required by the system is the electronic presentation file of the speaker. Even if not provided, the system would temporally segment the presentation and offer a simple storyboard-like browsing interface. The system runs on several capture boxes connected to cameras and microphones that records events, synchronously. Once the recording is over, indexing is automatically performed by analyzing the content of the captured video containing projected documents and detects the scene changes, identifies the documents, computes their duration and extracts their textual content. Each of the captured images is identified from a repository containing all original electronic documents, captured audio–visual data and metadata created during post-production. The identification is based on

This research is supported by grant from Swiss National Center of Competence in Research (NCCR) on Interactive Multimodal Information Management (IM2).

A. Behera (✉) · D. Lalanne · R. Ingold
Department of Informatics, University of Fribourg, Bd de Perolles 90, CH-1700 Fribourg, Switzerland
e-mail: ardhendu.behera@unifr.ch

D. Lalanne
e-mail: denis.lalanne@unifr.ch

R. Ingold
e-mail: rolf.ingold@unifr.ch

Present address:
A. Behera
School of Computing, University of Leeds, Leeds, LS2 9JT, United Kingdom
e-mail: A.Behera@leeds.ac.uk

documents' signatures, which hierarchically structure features from both layout structure and color distributions of the document images. Video segments are finally enriched with textual content of the identified original documents, which further facilitate the query and retrieval without using OCR. The signature-based indexing method proposed in this article is robust and works with low-resolution images and can be applied to several other applications including real-time document recognition, multimedia IR and augmented reality systems.

Keywords Meeting recordings · Automated meeting indexing and retrieval · Low-resolution document identification · Multimedia content extraction · Multimedia IR

1 Introduction

Nowadays, events such as meetings, seminars, lectures, conferences, etc. are often digitally captured and archived for the future access and browsing. In such highly multimodal events, documents in various forms (slideshows, scientific articles, etc.) are frequently used and are one of the major sources of information. Such documents are either discussed or projected on a screen. This observation introduces new challenges for research in document analysis for their integration within archives that consists of other temporal audio–visual data. During the recording, of the above-mentioned events, these presented documents are captured, either as video streams or images along with other audio/video streams. As more and more of such events are captured and archived, the necessity for automatic indexing and retrieval methods for later retrieval has become increasingly apparent. Attendees may not be able to retain or recall information sufficiently during meetings. Often, they take snap-shots/photos of projected documents of interest, such as slides using the handheld devices like digital cameras, mobile phones, etc. More often, presented/captured documents are shared among colleagues or are required to be summarized, creating an unavoidable necessity for attendance at such occasions. Afterwards, these captured/presented documents can be used to query the system for replaying the audio–visual streams of the related captured events to get detailed happenings during the presentations of those documents. Furthermore, a set of keywords from the captured/presented or the summarized documents could be queried for retrieval. These keywords appear either in the presented documents or speech-to-text transcriptions. In this paper, we report the design of a complete system that captures, analyzes and then indexes meetings, seminars, conferences, automatically.

Several research groups have studied the problem of integrating projected documents such as slides with the meetings or other lecturing details, which are also being captured (e.g. video, voice, whiteboard, etc.) and that would allow searching and browsing through the recordings using projected documents as an access interface [1, 11, 12, 16, 18, 20, 23, 24, 28, 42]. We categorize them into two categories according to their usability:

- *Educational use*: This is termed as e-learning and is generally made for classroom use for recording and broadcasting of lectures. Some of these projects include *Lecture-Browser* of Cornell University [30], *LectureLounge* of Fraunhofer-IPSI [28], *eClass* of Georgia Tech [16], *LectureBrowser* of DSTC, University of Queensland [25], *EmuLib* of University of Mannheim Germany [17] and *Meeting Room* of Carnegie Mellon University [33].
- *Organizational use*: This comprises recording and broadcasting of seminars and meetings for employees of organizations. This includes *Microsoft's Distributed Meeting*

Recorder [15, 24, 40], *IBM's eSeminar* [45], *FX PAL's Meeting Recorder* [12] and *Ricoh's Portable Meeting Recorder* [29].

Various commercial solutions are also available on the market. For example, *Foveal Systems AutoAuditorium*, that broadcasts live seminars to remote audiences and simultaneously records them. The system uses multiple cameras and automatically switches among them based on context. The output of the system is a single taped presentation that must be watched sequentially [9]. *Livelink's Eloquent* manually records seminars or lectures and produces presentations similar to the *LectureBrowser* [31]. However, in most of the above-mentioned systems, the temporal coherency across multiple streams has been done explicitly by capturing actions such as keystroke, handwriting, notes, or browser activities. In order to capture such activities, presenters have often to install some software and/or hardware before starting their presentation. Most often, presenters have to be aware of their actions, in order to index their presentation appropriately, which restrict their freedom of movement during the presentation. However, this explicit event-based coherency is first of all very intrusive, since speakers are obliged to install new software and it is also insufficient for building retrieval systems that allow querying of captured images from handheld devices or using keywords. In our system, there would be no necessity of any specific software on presenter's laptop that would be required for indexing. The synchronization among multiple streams is carried out using a global clock, to which various software modules that are running in the capture boxes of the system, are listening cooperatively. The only restriction of our system is that presenters are required to leave a copy of their electronic presentation files for a complete indexing of their presentations. In case they do not, the system will however segments the audio–visual streams based on the time of appearance of the documents and uses the comparatively high-resolution copy of the captured image from the projector output for browsing. In this case, the keyword-based browsing/retrieval can be carried out by manual addition of textual content or using OCR. Currently, the system extracts the textual content from PowerPoint and PDF presentation file formats. However, in case other formats are provided, these files should be converted to PDF prior to the extraction procedure.

Implicit content-based analysis is necessary in order to make retrieval systems more transparent to users, flexible and light weight. Furthermore, content-based analysis and mining would permit much richer retrieval systems. Ideally, our goal is to allow a speaker to walk into a conference, meeting or lecture hall, connect his personal laptop to the projector screen and start the presentation. All the activities during the presentation are captured as audio–visual streams without constraints on the speaker's individual action. Upon concluding the talks, the recorded audio–visual streams are analyzed and indexed as a postproduction process. The captured audio–visual streams are indexed at the granularity of visible document level and deposited in the repository for later retrieval and replay on demand. The indexing process is fully automatic and describes the points of interest by adding textual attributes to each of the video segment from the content of the projected documents, which are already identified from the repository containing the original electronic documents. The content extraction procedure is simple, does not require any computer vision-based approaches. It is done only once when the electronic documents are added to the repository. Moreover, the computer vision-based approaches are computationally expensive when the multiple instances of the same documents are captured from various handheld devices. Furthermore, they would not perform well for the current scenario's slide images, which exhibits very poor resolution and often textured and non-uniform background structure. The proposed system has been installed in the conference

room of the CERN (*European Center for Nuclear Research*) and *University of Fribourg* for the SMAC (*Smart Multimedia Archive for Conferences*) project [44].

The rest of the paper is organized as follows. In the next section, we present some related methods that link documents with the captured audio–visual streams. In Section 3, the system architecture for capturing, indexing and retrieving multimedia data from a multimodal environment is explained. Section 4 describes in detail our automated segmentation method based on document images and its performance in comparison to the existing methods. The content-based identification of the low-resolution captured documents is described in Section 5 along with an evaluation of performance. Furthermore, Section 6 presents the textual content extraction of the document without the use of any OCR, and is used for the characteristic information of the meeting segment. The relevance of the *DocMIR* sytem and its impact on other domain are presented in Section 7. Finally, we conclude our paper and propose future challenges in Section 8.

2 Related work

In this section, we discuss some of the existing capture systems, which use the projected slides for indexing of lectures or meetings and the textual content for keyword-based retrieval. Often, for better resolution, the slides are captured as screen snapshots, or from the VGA output of the presenter's laptop/workstation connected to the projector. In case of the VGA and digital outputs, a special card is needed to grab frames with the corresponding time-stamps. The content of the extracted image is matched with the original slide images, which are generated from the corresponding electronic documents (e.g. PDF, PPT, etc). However, most of these methods consider mainly the textual content of the documents without taking into account other useful features such as layout (physical and logical), color, texture, etc.

At the *FX Palo Alto Laboratory*, the conference room's activity is captured by computer controllable video cameras, video conference cameras and ceiling microphones [12]. Presentation material displayed on a screen is captured by a smart video source management component. When the rear projector displays slides running from the PC workstation, the server gets the images from snapshots of PC screen or from the video signal of the rear projector. For indexing, they make use of notes taken by meeting participants. Therefore, they have designed and built a client-server application called NoteLook. The NoteLook system allows users to incorporate images from the video sources of the room activity and presentation material into the notes [14]. Furthermore, it proposed a DCT-based image matching to link slides with multimedia data [13]. However, it runs on pen-based notebook computers and therefore, not designed for novices since it requires training to use.

In the *Classroom 2000* project at *Georgia Tech*, a single audio–stream from a lecture is recorded and slides with annotations are made available to students after class [1, 16]. The presenter would have to make a special effort to prepare the slides in a standard graphical format. The slides are displayed on a LiveBoard and note-taking is done with PDA devices (ClassPad) pre-loaded with slides. ClassPad preserves all annotations made to a series of prepared slides and creates a time-stamped log of when the user navigates between slides and when each slide is annotated with the pen. These notes are later synchronized to the audio and the slides, which have been annotated by the professor on the LiveBoard for later access on demand [11]. However, it requires an additional effort during the lecture to transform the prepared material into the desired form and requires training to use ClassPad.

The *Cornell Lecture Browser* of *Cornell University* automatically produces multimedia documents from live lectures. The synchronization of multiple audio–visual streams (overview camera, tracking camera and microphone) is carried out by generating a 1-s synchronization tone and recording it in one channel of the MPEG streams. Furthermore, they describe a method for segmenting recorded lectures using the slide duration and then matching the clipped slide images with the low-resolution video [34]. The method is based on binarizing and dilating the clipped slide images and frames first, to highlight the text regions prior to using the Hausdorff distance to compute the similarity between the text lines [39]. However, it works well only on slides that contain texts and the slide region should be accurately segmented. Furthermore, the segmentation is confirmed by the slide identification where the system encounters the corresponding clipped slide image. Moreover, the evaluation for the matching of slides is restricted slideshow-wise in the order in which it was presented, which requires an operator to sort the slideshows as they were presented.

Ricoh Innovations has developed *Portable Meeting Recorder* that records all the activities in a meeting and the directions from which the participants spoke [29]. The presentation recorder captures what is displayed on the presentation screen with the timestamps. The VGA output of the presenter's machine is converted to NTSC signal and is saved in JPEG format with a frame grabber. The output of both the meeting and the presentation recorder are synchronized by time-stamps with post-hoc clock-skew correction [18]. The grabbed images are matched using features such as OCR output, edges, projection profiles and color layout. However, the system calls for post-synchronization between presenter and meeting recorder and the use of OCR is computationally expensive and requires different system to deal with different languages and is not suitable for real-time application, as well. The slide matching method considers the images from high-resolution capture devices and from digital cameras. Furthermore, the images from digital cameras contain mostly the projected area with a rotation of less than ±5° and at least one text line.

The *e-Seminar* prototype at IBM Watson Research Center is designed to allow all IBM-Research employees access to videos and slides of talks, seminars, presentations, and other events at any IBM-Research campus worldwide [45]. The system consists of nine components/modules, which are (1) Scheduling, (2) Recording/Encoding, (3) Analysis, (4) Composition, (5) Storage, (6) Distribution, (7) Searching/Browsing, (8) Streaming and (9) Feedback/Communication. The system uses *Fovel Systems AutoAuditorium* for the multi-camera automatic production [9]. Furthermore, it also uses other analysis modules such as speech-to-text transcription, scene change detection, key frame extraction, face detection, camera motion detection and screen-shot unification during the post-production. Their future plan is to map the original presentation data with time-stamped screen-shots using OCR.

The *Distributed Meetings* (DM) system at Microsoft is designed for high quality broadcasting and recording of meetings as well as browsing of archived meetings' data [15]. It uses various capture devices such as 360° camera, overview camera, whiteboard capture camera and microphone array to capture meetings. The system creates the indices for the indexing of the captured audio–visual data using various techniques such as vision-based person detection and tracking, audio-based sound source localization (SSL) for speaker segmentation and clustering. It also uses offline image analysis of high resolution whiteboard image sequences to detect the creation time of each pen stroke as well as the detection of key frames. However, the system does not include the capturing and alignment of presented materials (e.g. PowerPoint slides, PDF, etc.), which is one of the major source of information during meetings, conferences, etc.

At the *DSTC Pty Ltd*, Hunter and Little [25] investigated the mechanisms for capturing, indexing, searching and delivering digital online presentation using SMIL. They developed a set of tools to automate and index the content of lecture from both the University of Queensland and Cornel University. The system takes video footage, PowerPoint slides and timing information generated by Cornell Browser [34] and generates a log file with the output from the *PresentationLogger* application that runs in the background of the laptop or PC used for the presentation. *PresentationLogger* provides the slide numbers, timing and the textual content of the slides. However, the system needs to be installed in the presenter's laptop or PC and the lecture browser does not support keywords- and image-based query and retrieval.

The *LectureLounge* system of Fraunhofer-IPSI Darmstadt is designed for capturing, management and publication of presentations of educational work [28]. It uses a digital camera, wireless microphones and a laptop for the acquisition. It also captures slides directly from the video projector with a VGA-grabber card. During post-production, the system uses various tools such as scene-change detection, speech-to-text transcription, speaker recognition, and slide summarization to extract the metadata. The system does not mention about the linking of presented electronic documents with the captured temporal data and also does not support the image-based query and retrieval.

In all the above-mentioned systems, the projected documents are linked with the captured multimedia documents (video, voice, whiteboard, etc.) for later replaying. They are either linked explicitly by the operators/users or by some special hardware/software. However, there is no automated indexing of the captured videos, which could enhance the retrieval of the corresponding videos by querying an image captured during the presentation. Moreover, our goal is to provide a system, which indexes at the granularity of projected documents, as often the context of speakers' presentation is focused around the content of projected documents. The indexing process is fully automatic by implicitly analyzing the content of the captured audio–visual streams and then the indexed audio–visual streams are deposited in the repository for later retrieval and replay on demand.

3 System architecture

The architecture of the complete system we have developed for capturing, automatically indexing and retrieving meetings, conferences, lectures and seminars is shown in Fig. 1. The system consists mainly of three tools. First, a *capture tool* allows the raw data of meetings to be captured and archived. In this tool, the projected documents (slides) are synchronized automatically with other multiple audio–visual streams without installing any software or hardware in the presenter's computer. Secondly, the captured video streams are used by the *analysis and indexing tool* for the content-based complete automatic indexing. The last tool is an interactive *retrieval tool*, which takes advantage of keywords and/or captured documents from handheld devices to access the archived audio–visual streams.

3.1 Capture tool

In our document-centric smart meeting room, weekly meetings, student presentations and discussions are held. One camera is focused on the projector's screen to capture the projected documents and three cameras are used to capture the overview of the meeting room. One camera-microphone pair per participant is used to capture the head-and-shoulder video and speech of the participant. The capture architecture is simple, distributed, scalable

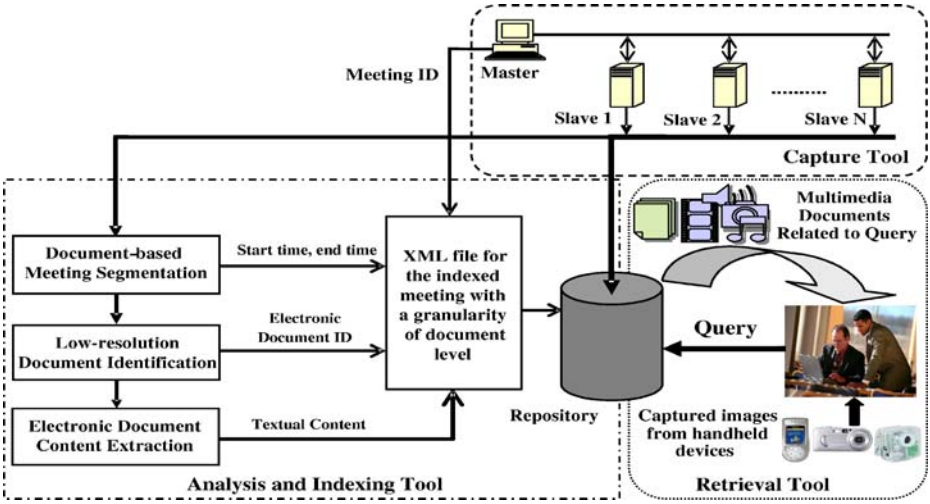

Fig. 1 Complete architecture of the system for automatic indexing of the recorded meetings (*left side*, *outmost bounding box*), and retrieval based on captured document image and/or keywords on documents (*right side*, *outmost bounding box*)

and easily adaptable for any number of capture devices as well as of different hardware variety (e.g. web-cams, DV-camera, etc.). For simplicity, we use the light-weight capture devices such as FireWire web-cams, which are not only small and inexpensive but also effortless for fixing and removing in case of shifting of the capture environment. For example, one could consider capturing at grand conferences, where multiple sessions run parallely at several smaller locations, where the capture devices are not furnished. In our capture architecture, we use a *master–slave* model. The *slave* capture boxes (PC) control the capture devices such as cameras and microphones. The total number of capture devices per *slave* is limited to three pairs of camera–microphone. This is considered to maximize the use of capture hardware without overloading, which results in dropping of frames while capturing. All *slaves* are synchronously listening to the pilot called *master* (Fig. 1). A user-friendly control interface that runs on the *master* allows selecting the devices to use (cameras, microphones, etc.), registering the participants and to select frame rate, resolution, etc. Moreover, post-processing, compression, file transfer and creation of SMIL (Synchronized Multimedia Integration Language) presentation per meeting are all automated and controllable through this interface [26, 48]. At the end of the meeting, the raw audio/video data is compressed (DivX and Real Media) and stored in a repository for later access and retrieval. All the captured audio/video streams for a particular meeting are tagged with a unique identification number called ‘meeting ID’. Figure 2 shows RealPlayer synchronously playing one of the meetings recorded in our meeting room and one of the recorded talks delivered in the international workshop on *Multimodal Interaction and Related Machine Learning Algorithms* (*MLMI 2004*) using SMIL.

3.2 Analysis and indexing tool

Once the capture is completed, the captured audio–visual streams are then used by the *analysis and indexing tool* for automatic indexing. Indexing is a central component necessary to facilitate efficient retrieval and browsing of visual information stored in the

Fig. 2 Snapshot of the RealPlayer; playing recorded meetings of multiple audio/video streams containing the head and shoulder of each participant, room overview, projected documents (*left*) and one of the presentations of the MLMI 2004 conference containing the audio/video of the presenter and projected documents (*right*)

meeting repository. The tool mainly considers the video of the projected documents and works systematically with the following three steps: (1) documents-based meeting segmentation, (2) low-resolution document identification and (3) electronic document content extraction (Fig. 1).

- *Document-based meeting segmentation*: Temporal segmentation to the meeting's audio–visual streams into semantically connected units is an important step to understand the meeting content. Moreover, it makes fast access to the meeting recordings possible. In this scenario, we use the projected documents for the temporal segmentation of meetings. During meetings, each projected document appears at a distinct time and remains visible for some time, which indicates the temporal relationship of each projected document with the meeting time. The captured meeting video containing projected documents is analyzed to extract time boundaries i.e. start and stop time of each projected document. In this step, all the detected entry points are added to the meeting annotation file (Fig. 1). These time boundaries, later serve as entry points for non-linear access, i.e. snaps directly to the desired position in the videos, without having the need to play the meeting recordings from the very beginning. Such kind of access is extremely time-saving for a user who attended the presentation and is looking for specific parts of the presentation. However, it is of little or no use if a user has not attended or seen the presentation before. Nonetheless, this method also holds true for non-attendees since they get to access the needed information from a collection of thumbnails of the projected documents. These thumbnails are already linked to the meeting's audio–visual streams with respective time boundaries. We described a novel method, which detects the above-mentioned entry points and is described in Section 4.
- *Low-resolution document identification*: In the previous step, the recorded meeting is temporally fragmented into distinct smaller segments (Fig. 1). Each meeting segment corresponds to a *stable* period of the video of the projected documents and the detection of such periods is explained in Section 4. One key-frame per *stable* period is extracted. These extracted key-frames are nothing but the captured images of the projected documents. Therefore, these key-frames must then be identified from the repository containing all the presented original electronic documents. The inclusion of the identified original documents with the meeting segments improves the visibility of the documents during browsing. Moreover, it is difficult to extract the textual content of the captured document images (key-frames) using OCR and is due to the poor quality,

low-resolution, textured and non-uniform background. In order to overcome the drawbacks above, we propose a novel approach, which is based on document's signature for the identification of the above-mentioned captured low-resolution document images and is described in detail in Section 5. At the end of this step the ID, which corresponds to each of the identified original electronic document (Electronic Document ID, Fig. 1) from the meeting repository, is added to the annotation file.

- *Electronic document content extraction*: Once the original documents are associated with their corresponding meeting segment, then the textual content of the electronic documents is extracted and added to the annotation for the keyword-based retrieval. Our research group has developed a tool that extracts the content (both texts and graphics) of the document and is described in Section 6. The extracted textual content of the electronic document, which is associated with one or more meeting segments, is included in the text attributes of those segments during the meeting annotation.

3.3 Retrieval tool

The retrieval tool generally operates on multimedia meeting archives to retrieve relevant meeting segments in response to a query of an image or a set of keywords. The retrieval performance is highly dependent on the segmentation methods used, matching performances and the quality of indexing. For image-based retrieval, the matching performances are most likely to be associated with the low-level visual contents such as color, textures, shapes, etc. This feature-based matching works efficiently with a query of similar image, but they would not perform well if the image is taken from a different angle or has a different scale [3, 37]. On the other hand, keyword-based retrieval is mainly based on the attribute information, which is associated with meeting segments in the process of annotation.

The proposed retrieval method considers both the low-level visual features (color, texture, etc.) and layout features of the document for the image-based queries. For the keyword-based, it simply searches the corresponding word in the textual attributes of the meeting segments. Once the analysis is done, the indexed XML files along with the captured audio/video streams and the projected original electronic documents are archived in the repository. The tool accepts images, which are captured from low-resolution handheld devices and/or keywords to retrieve the relevant meeting of interest to the viewer.

3.3.1 Image-based retrieval

In image-based retrieval, the captured document images from the handheld devices (digital camera, mobile phones, etc.) during the presentation are used to query the tool. Furthermore, the captured documents from the handheld devices are often compressed with the lossy compression such as JPEG. The tool looks for the original document that corresponds to the queried one. As we mentioned in the *analysis and indexing tool*, all the original documents are already associated with the respective meeting segments by identifying the extracted document image from the meeting video. Therefore, the tool delivers the time-codes i.e. the boundaries of meeting segments, which are associated with the original document corresponding to the queried document image. The captured image is processed to compute the corresponding signature, and the image of the best matched signature is picked up from the repository. However, it is not always necessary that the queried document image should be captured using handheld devices. One could also use the image of original document as often, people share or distribute their presented

documents to colleagues. The identification of the queried document image is the same as the low-resolution document identification and is described in detail in Section 5.

3.3.2 Keyword-based retrieval

In keyword-based retrieval, the given keywords are searched in all text attributes of indexed video files in the meeting repository. This is generally a full-text search engine that takes text as input and delivers time-codes when this piece of text appeared in the presented documents and/or in speech-to-text transcriptions as often, the textual content of the projected documents does also appear in the speech. To date, the results of speaker-independent speech recognition are not satisfactory to provide a closed caption, even though they are good enough to provide a base for a keyword search on the spoken text. Moreover, our first preference is the textual content of the projected documents, which are already associated with time-codes and more accurate in content extractions than the speech transcriptions. This is due to the fact that the context of the speaker's presentation would be focused around the content of the projected documents. Furthermore, since the recognized words are associated with time-codes, the keywords could again serve as entry points for non-linear access.

Figure 3 demonstrates a document-enabled interface for browsing of multimedia meeting archives and is called as **FriDoc** browser. The browser is user-friendly and helps in quick access to the desired meeting portion. First, the user gets the list of related original electronic documents using keywords and/or images as queries (Fig. 3, right). It is mentioned earlier that these documents are temporally linked with the captured multimedia documents. Once the desired document is selected, then the user is directed to the intra-meeting navigator with the focus on the desired meeting segment, in which the document was projected (Fig. 3, left). Furthermore, it allows user for non-linear access to browse that meeting using projected documents, control bar, sunBurst visualization and speech transcriptions [27].

In the following section, the time-codes for each of the projected document are extracted from the meeting video and indexing of the captured meetings, conferences, seminars, etc. are explained in detail.

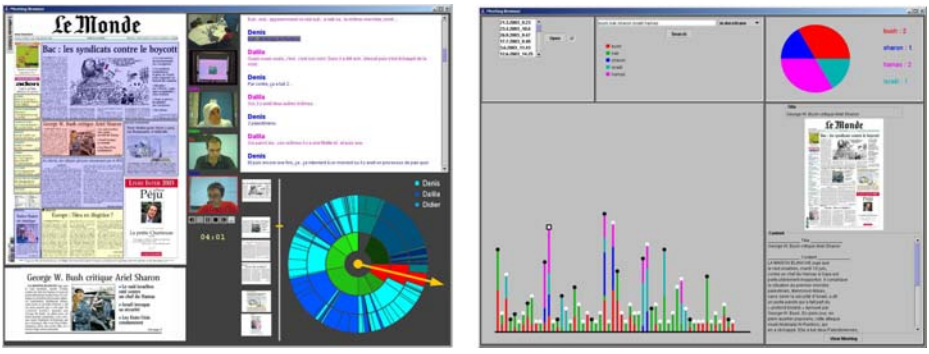

Fig. 3 FriDoc browser; intra-meeting navigator (*left*) containing audio/video, documents, speech transcripts, control bar with sunBurst visualization and cross-meeting navigator (*right*), keywords and/or captured image-based retrieval of documents, that are linked to corresponding meeting segment

4 Document-based meeting segmentation

Here, we describe the meeting segmentation i.e. the captured audio–visual streams are cut into distinct smaller segments for quick non-linear access to the meetings on demand. In our scenario, the video containing projected documents (slideshow) captured continuously from a fixed web-cam, is considered for segmentation. Web-cams have auto-focusing function, which modifies the lighting condition and adds fading during transition and thus, take nearly 0.5 s to capture stable images after a change in the projected documents (Fig. 4). Most existing segmentation algorithms look for cuts and breaks in the video, thereby dividing it into distinct scenes. If there are no cuts, like in the current scenario, these algorithms usually detect the changes such as the motion of the speakers in front of the projection screen. It is well-known that the most common approach to scene-cut detection is based on the color histogram [10, 32, 52]. *Cornell Lecture Browser* proposed a method for the segmentation of lectures by considering similar video containing projected slides captured continuously from a fixed digital video camera [34]. The method uses an assumption that slides are presented in the same order in which they appear in the respective electronic file (PPT, PDF). Moreover, they use a slide identification method in order to validate the slide change detection. The proposed segmentation technique is described below and is compared with the simplified Cornell approach (without assumptions and slide identification) and global histogram approaches.

Our segmentation technique detects the *stable* and *unstable* period rather than the changes in a sequence of frames from the video of the projected documents. The *stable* or *unstable* period detection is carried out by sliding a window of 2 s in duration over the sequence. Here, we defined those documents (slides) that stay on the screen for less than 2 s, as skipped slides. The process is mainly two-fold. First, it searches for stability of the image sequences in the current window and if found stable, then the search moves to the next window. However, if the current window is found as unstable, the second step is executed to confirm instability (Fig. 5). This two-fold process scans the whole video sequence and afterwards merges the consecutive windows of the same type to form a stable period or an unstable period. The position(s) of the dissimilar frame(s) in the unstable period is (are) extracted from the respective unstable window (s). The two-step procedure is explained below in detail.

1. In the first, the first frame, F_1 and the last frame, F_{N+1} in the window are converted to bi-level image using the Otsu segmentation method (Fig. 5) [36]. For the binarization of document images, Trier and Taxt [49] showed that the performance of Otsu method is best among other global methods. For the slide images captured using a web-cam, a qualitative evaluation has been performed on various representative slide images and it has been found that Otsu method performs better as compared to *Niblack*, *Kitller*, *Yanowitz* and *entropy-based* segmentation. The similarity distance between frame F_1 and F_{N+1} is computed as: $\Delta=(d_1+d_2)\,/\,(b_1+b_2)$, where b_1=# of black pixels in F_1, b_2=#

Fig. 4 Auto-focusing modifies the lighting conditions and fading during transition

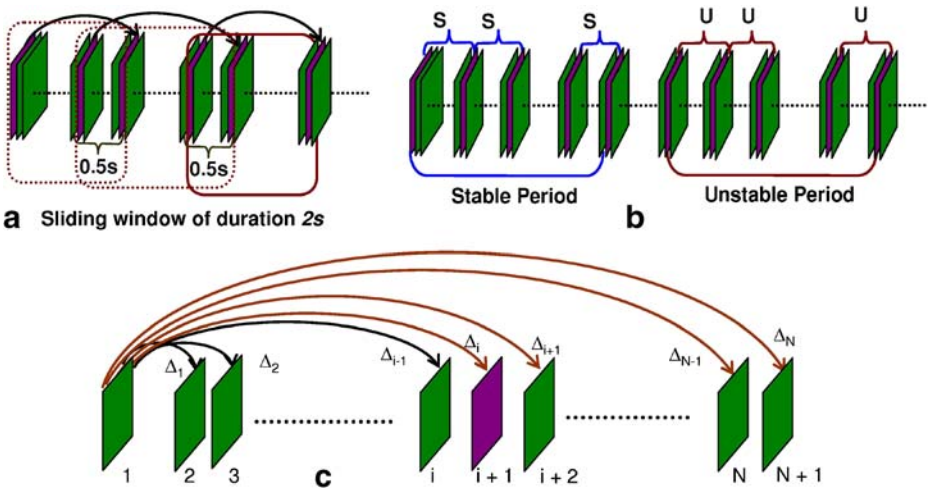

Fig. 5 Stability detection in the control video stream using a sliding window with duration of 2 s, **a** the sliding window scans the stream to check the stability; **b** two or more consecutive windows of the same type are merged to form stable period or unstable period, **c** confirmation of instability by considering each frame in the sliding window

of black pixels in F_{N+1}, d_1 is the number of black pixel of F_1 whose corresponding pixels are not black in F_{N+1}, and d_2 is the same as d_1 by reversing F_1 and F_{N+1} [34]. If $\Delta > T_1$, then the second step would be executed for the confirmation of instability; otherwise the window would be moved 1.5 s forward and the same would be continued (Fig. 5a). T_1 is set conservatively, so that no *unstable* windows go undetected. In case of some false detection, the final stability would be verified in the next step. This step helps to speed up scanning of the video stream by reducing the computational time, needed for the computation of the similarity distances for all the frames in the window. The consecutive windows of the same type i.e. either *stable* (S) or *unstable* (U), are merged together to form a *stable* or *unstable* period, respectively (Fig. 5b).

2. In the second, an individual distance Δ_i (i=1...N) is computed by comparing the frame F_1 with the rest of the N frames in the sliding window using the above-mentioned distance computation (Fig. 5c). If the ratio $R=(\Delta_m / \Delta_v) < T_2$, then the instability is confirmed. Where $\Delta_m = \frac{1}{N}\sum_{i=1}^{N} \Delta_i$, and $\Delta_v = \frac{1}{N}\sum_{i=1}^{N} (\Delta_i - \Delta_m)^2$. Normally, an *unstable* window contains two or more different kinds of frames. Therefore, in such windows the variance of distances, Δ_v would be significantly higher than those of stable ones, which contain only one kind of frames. Once, the window is confirmed as an *unstable* one and then the exact position of the first dissimilar frame with compare to all other previous frames in the window, is looked for. This dissimilar frame corresponds to the starting frame of the incoming new slide document in the video sequence. The position of this frame is computed by comparing the distance, Δ_i to the average value of the $min(\Delta_i)$ and $max(\Delta_i)$ (i=1...N) of all the distances in the window (Fig. 5c). Starting from the distance, Δ_1 and if the distance, $\Delta_i > \{min(\Delta_i)+max(\Delta_i)\}/2$ is encountered, then the frame at ith position is the incoming new slide document. The corresponding time for this new slide document is computed as t_p = (*# total frames passed till i*th *position*) / (*video frame rate*). Once this position is identified, then the sliding window is to be moved forward with the starting frame, F_1 of the window correspond to ith frame and the above-mentioned two-folded stability inspection would

```
<slidechange>
        <slide id="1" imagefile="Slide001.bmp" st="0.0000" et="5.0400" type="normal" />
        <slide id="2" imagefile="Slide002.bmp" st="5.0400" et="5.5200" type="skip" />
        <slide id="3" imagefile="Slide003.bmp" st="5.5200" et="5.9600" type="skip" />
        <slide id="4" imagefile="Slide004.bmp" st="5.9600" et="6.2800" type="skip" />
        <slide id="5" imagefile="Slide005.bmp" st="6.2800" et="7.2400" type="skip" />
        <slide id="6" imagefile="Slide006.bmp" st="7.2400" et="12.8400" type="normal" />
        <slide id="7" imagefile="Slide007.bmp" st="12.8400" et="12.9600" type="skip" />
        <slide id="8" imagefile="Slide008.bmp" st="12.9600" et="13.1200" type="skip" />
        <slide id="9" imagefile="Slide009.bmp" st="13.1200" et="13.2800" type="skip" />
        <slide id="10" imagefile="Slide010.bmp" st="13.2800" et="13.4800" type="skip" />
        <slide id="11" imagefile="Slide011.bmp" st="13.4800" et="19.2000" type="normal" />
        <state TotalSlide="11" StablePeriod="3" UnstablePeriod="2" />
</slidechange>
```

Fig. 6 An example of SMIL file, which is the output of the slide change detection

be continued until the end of the video stream. If the duration $t_p - t_{p-1}$ (time between successive change detection) is less than 2 s then the corresponding *type* attribute (Fig. 6) is updated with *skip*; otherwise *normal*.

Furthermore, it is observed that in case of animations in the presentations, the whole period is detected as an *unstable* period and often changes are detected for the frames that have content-wise dissimilarity of at least 25%. For such changes, the intermediate frames are considered as skipped slides (Fig. 6) if the appearance time is less than 2 s; otherwise considered as a new slide. In the near future, we plan for animation detection in the *unstable* period and the corresponding *type* attribute would be updated. Actually, the number of *stable* periods corresponds to the number of slides having *type* attribute of *normal* and two *stable* periods are separated with an *unstable* period (Fig. 5b), which depicts the transition of the previous slide to the current one. On the other hand, the number of *unstable* periods should correspond to the number of slide transitions i.e. if there are P numbers of slides in the presentations, and then the number of transitions would be $(P-1)$. Nevertheless, this is not always true as there is a possibility of more than one change within an *unstable* period and is due to the skipping of slides during a presentation. However, the aim is to detect and include even those slides in the annotations as this would be interesting for some of the listeners for retrieval on demand. The exact position of the new slide is looked for in the *unstable* period so that an accurate computation of start and end time of each slide is achieved and therefore, overcomes the auto-focusing, fading and poor resolution of web-cams.

4.1 Evaluations and results

The above-mentioned method has been evaluated automatically by capturing projected slideshows (PPT, PDF). However, there is a necessity of proper ground-truth to evaluate the proposed method. Producing manual ground-truth is not only time-consuming and tedious but also prone to errors while preparing it. In order to overcome this, we have developed an application, which generates the ground-truth using SMIL. Various presentations related to education, technical and non-technical contents have been collected. These are available on the web and mainly compiled from conferences and seminars in various public and private sectors. Therefore, more than three thousand slides (65 slideshows) have been accumulated, which represents different varieties of presentation styles.[1] Our aim is to capture the presentation as in the real world. Thus, the order of slides is kept as it is in the slideshows.

[1] These slideshows can be downloaded from our meeting server (http://diuf.unifr.ch/im2/data.html)

The JPEG image of each slide of the slideshow is picked up with random presentation time and if it is less than 2 s, then the *type* attribute is assigned as *skip*, otherwise *normal* (Fig. 6). One SMIL file for each slideshow is generated. The SMIL file is played in the RealPlayer of the PC/laptop connected to the projector and the web-cam focusing on the projector screen, starts capturing simultaneously. The output of video segmentation and the ground-truth SMIL are in XML. So, the matching is simply comparing the attributes (start time and end time) of individual slide in the ground-truth and in the output of video segmentation (Fig. 6). Recall (R), Precision (P) and F-measure (F) metrics are used for the performance evaluation and are defined as:

$$R = \frac{\#\text{correct changes detected}}{\#\text{total changes in ground} - \text{truth}}$$

$$P = \frac{\#\text{correct changes detected}}{\#\text{total changes detected}}$$

$$F = 2 \times \frac{R \times P}{R + P}$$

The evaluation is carried out with the tolerance of one and four frames, i.e. for the video of 15 FPS; the respective tolerances are of 66.67 and 266.67 ms. Considering the real-world presentation, the metrics are computed slideshow-wise. Furthermore, we statistically analyze the performance by considering the *Standard Error of Mean* (SEM) of n=65 slideshows and is computed as:

$$E_{\text{SEM}} = \frac{S_d}{\sqrt{n}} \text{where standard deviation } S_d = \sqrt{\frac{\sum_{i=1}^{n} \left(X_i - \overline{X}\right)^2}{n-1}} \text{ and mean } \overline{X} = \sum_{i=1}^{n} X_i$$

X_i is the performance of the ith slideshow. The performance of the proposed method excelled the Cornell and the global histogram methods [34] (Table 1 and Fig. 7). The performance of all the above-mentioned methods are presented using metrics of F-measure (Fig. 7), Recall (Fig. 8) and Precision (Fig. 9) for the respective tolerance of one and four frames along with the *standard error of mean* (SEM). From all the figures (Figs. 7, 8 and 9) it is clear that, the Cornell and the proposed method performed significantly better than the two others (gray and color histograms) while the proposed method out-performed the Cornell method.

4.2 Discussion

For the above-mentioned slideshow corpus, Cornell's average recall measure is 0.40± 0.041, average precision is only 0.21±0.029 and the combined performance F-measure is

Table 1 Comparison of performance of various segmentation methods

Metric	Proposed method		Cornell method		Color histogram		Gray histogram	
	1 frame	4 frames	1 frame	4 frames	1 frame	4 frames	1 frame	4 frames
Recall	0.84	0.93	0.40	0.80	0.07	0.13	0.18	0.27
Precision	0.82	0.91	0.21	0.51	0.04	0.09	0.12	0.17
F-measure	0.83	0.92	0.23	0.54	0.05	0.10	0.13	0.19

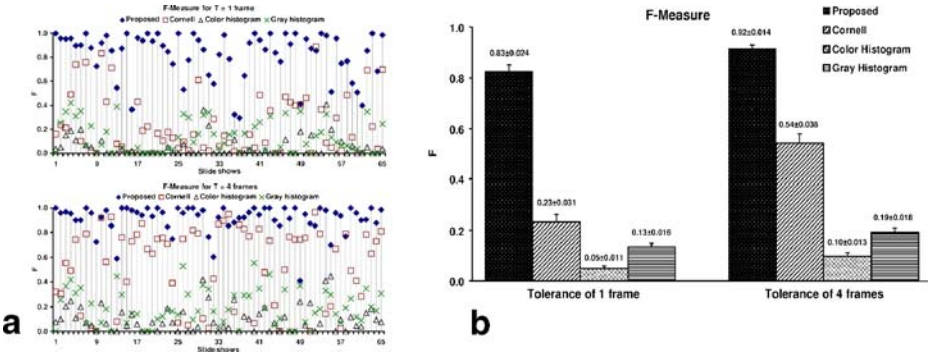

Fig. 7 **a** Slideshow-wise and **b** combined performance of various segmentation algorithms using F-measure (F) for tolerance of one and four frames

0.23±0.031 for the tolerance of one frame (Table 1). However, the method uses the slide identification mechanism based on the Hausdorff distance for confirming the slide changes [39], which should considerably increase the precision as well as the processing time for the non-existent extra slide changes. The high number of incorrect slide change detected in this method, distinctly increases the computational work. This drawback is overcome by the proposed method, which does not need to perform slide identification in order to increase the precision. In the proposed method, the average recall is 0.84±0.026, precision is 0.82± 0.024 and F-measure is 0.83±0.024. The sensitivity of each of the above-mentioned method is measured by increasing the tolerance from one frame to four frames. The proposed method is significantly less sensitive as the increment is less than 12% (R: 0.93, P: 0.91, F: 0.92), whereas in case of the Cornell, it is more than 112% with respect to the tolerance of one frame (Cornell, R: 0.80, P: 0.51, F: 0.54; Table 1, Figs. 7, 8 and 9). Though there is an improvement in Recall value (0.80) of the Cornell method for the tolerance of four frames (Fig. 8), the precision does not match up to that level (0.51, Fig. 9). This is due to a significant number of false detections, which reduces the overall performance (F: 0.54, Fig. 7). In case of the combined performance F-measure, it is observed that the SEM of the proposed method is reasonably less (0.014) for tolerance of four frames as compared to one frame (0.024). Moreover, with the Cornell method, the SEM is increased from 0.031 to 0.038 for the same (Fig. 7). This implies that in case of the Cornell, though the performance is improved however, it shows a tendency to increase

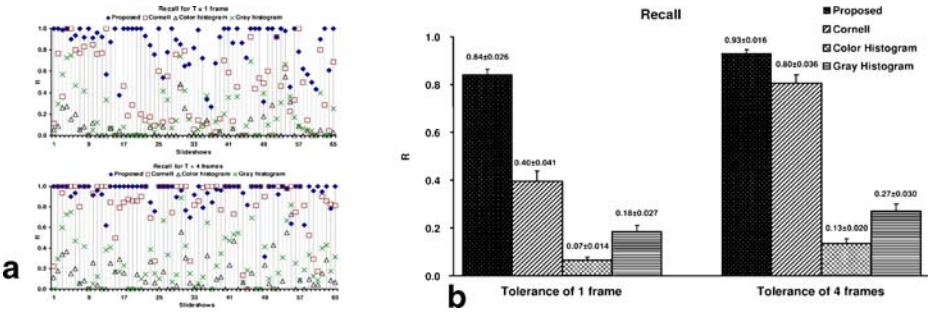

Fig. 8 **a** Slideshow-wise and **b** combined performance of various segmentation algorithms using Recall (R) for tolerance of one and four frames

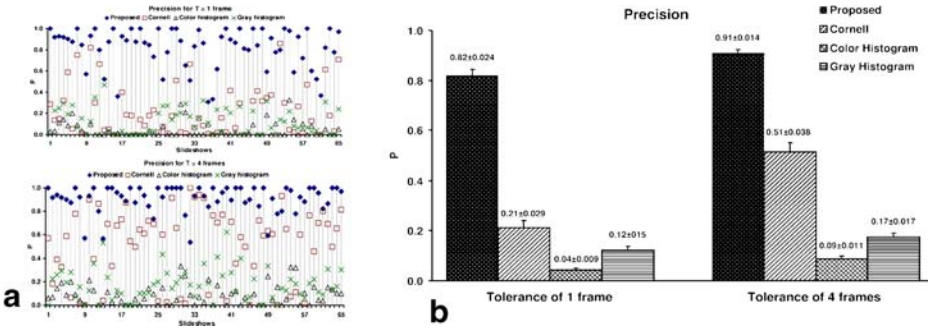

Fig. 9 **a** Slideshow-wise and **b** combined performance of various segmentation algorithms using Precision (*P*) for tolerance of one and four frames

variability with the increment of tolerance. Whereas in the proposed method, not only the error decreases but it also maintains stability in the performance, demonstrating that it is not sensitive to the variations in tolerance. Furthermore, the performance of Cornell method for tolerance of four frames is even less than that of the proposed method for the tolerance of one frame. In this evaluation, we have considered the tolerance up to four frames as it is observed that the performance of none of the above-mentioned method vary more than 10% if the tolerance is increased greater than four.

The proposed method is compared with the global color and gray-scale histogram methods. It is found that both the *Cornell* and proposed approach performed better than both the histogram approaches (Figs. 7, 8 and 9). This is due to the fact that in real world slide presentations, most of the slides in a slideshow have the same background, color and design pattern. In this case, only the textual content and layout vary. Thus, the histogram techniques are not adapted to detect such changes, especially with a low-resolution camera such as web-cams, resulting in poor contrast level.

The performance between the color and grayscale histogram is also compared. Theoretically, the color histogram method should perform better than the grayscale, because of the loss of color information in the second method. Instead, the grayscale histogram (*R*: 0.18, *P*: 0.12, *F*: 0.13 and *R*: 0.27, *P*: 17, *F*: 0.19 for tolerance of one and four frames, respectively) showed better performance than the color histogram (*R*: 0.07, *P*: 0.04, *F*: 0.05 and *R*: 0.13, *P*: 0.09, *F*: 0.10 for tolerance of one and four frames, respectively). This is mainly due to the auto-focusing nature of the web-cameras as the color histograms of all the frames in the transition period are often quite inconsistent for the same slide. This results in triggering of false slide change detection in case of the color histogram method. In the near future, we plan to evaluate the above-mentioned segmentation methods for the video captured from high-resolution devices such as DV and pan/tilt/zoom cameras. Since the proposed method out-performed the Cornell method for the videos captured using low-resolution cameras such as web-cams, we strongly believe in further improvement of performance for those captured from high-resolution capture devices.

The captured audio–visual streams have been segmented effectively using the video of the projector screen. During the segmentation, one key-frame per *stable* period is extracted. These extracted key-frames are nothing but the captured low-resolution slide images. In the following section, we present a method that robustly identifies such images from the meeting repository. The method provides a means for the inclusion of original electronic documents instead of the captured ones for the meeting annotation.

5 Low-resolution document identification

Document image matching is the kernel technology for document identification. In our case, the matching is performed using signatures, which represent the documents and are extracted by processing the document images. The image of each original electronic and captured document is processed for the extraction of their corresponding signatures. The systematic procedure for the signature extraction and matching is shown in Fig. 10. First, the low-resolution captured image is rectified for the perspective deformations and then low-pass filtered for the removal of noise in the pre-processing step (Section 5.1). Then the pre-processed image is analyzed for extraction of various features such as shallow layout structure (Section 5.2.1), color distribution in the RGB color space (Section 5.2.2) and in the document's 2-D image plane (Section 5.2.3) to form the respective signature. The document signature combines the three above-mentioned signatures. The signatures of the original electronic documents are extracted by considering its image format (JPEG) and the procedure is the same as the above, except that there is no necessity for the pre-processing step. In the following sub-sections, each of the blocks of Fig. 10 is described in detail.

5.1 Pre-processing

The documents captured from the projector's screen using any capture device not only contain the projected documents but also the surrounding background. It is, thus necessary to remove the background and to rectify the skewing of the remaining document image for identification. The capture devices are assumed to have low radial distortion. Therefore, one needs to consider the four corners of the quadrangle *ABCD* (clock-wise) of the projected part and is mapped to a rectangle of common resolution of width, *W* and height, *H*. The point *A* is mapped to the origin, *B* to (*W*, 0), *C* to (*W*, *H*), and *D* to (*H*, 0). This is done using a 2-D perspective transform, which maps an arbitrary quadrilateral into another arbitrary quadrilateral while preserving the straightness of lines. This transformation is represented by a 3×3 homogeneous coordinate matrix, M which transforms homogeneous source co-ordinates, *P* to corresponding destination co-ordinates, *Q* using transformation equation $Q = \mathrm{M} \times P$. Where $Q = \begin{pmatrix} x' \\ y' \\ w' \end{pmatrix}$, $P = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$ and the matrix, $\mathrm{M} = \begin{pmatrix} m_{00} & m_{01} & m_{02} \\ m_{10} & m_{11} & m_{12} \\ m_{20} & m_{21} & m_{22} \end{pmatrix}$ is computed by solving the above-mentioned equation using the four corners of the source quadrangle and the destination rectangle. Then the up-sampled image is computed using bilinear interpolation.

In order to detect the corners of the projected part, our system provides two different methods; (a) fully automatic and (b) manual approach using an interactive GUI as shown in Fig. 11. The manual technique is more appropriate for images captured from a fixed device. Since, all the captured images exhibit the same perspective distortion and therefore, the

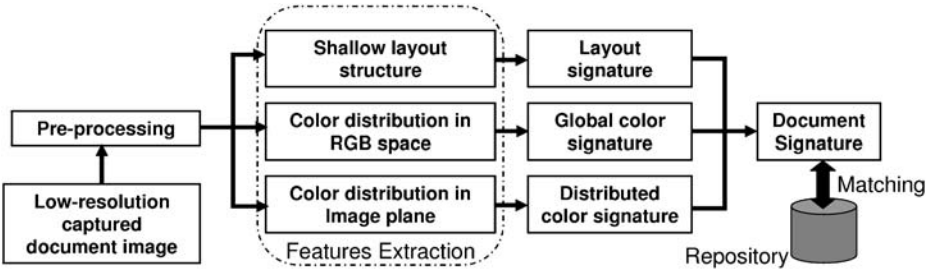

Fig. 10 Low-resolution document identification: a systematic procedure for feature extraction to form the document signature and matching

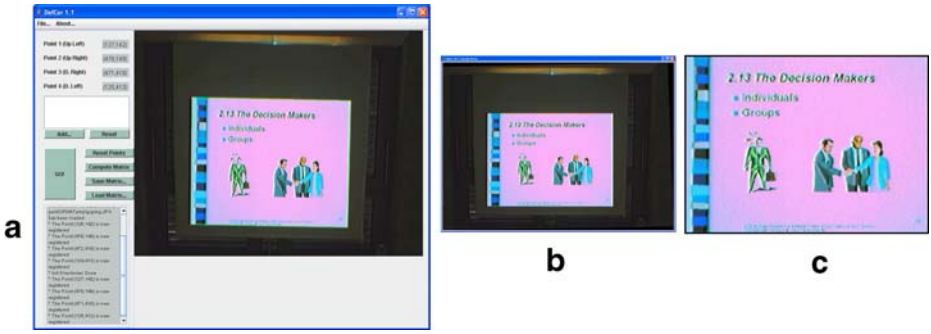

Fig. 11 Deformation correction using perspective transformation **a** captured image loaded in application for correction, **b** after perspective correction, **c** background removal by cropping

correction is done only once on one of the captured images and the same transformation matrix, M is used for the rest of images. As mentioned in the Section 4, the system captures the projected slides as a video stream and the respective slide images are extracted from the stable period for their identification. One would prefer the manual detection for the correction of such images, since it is more reliable than the automatic.

In case of the captured images from mobile devices (unfixed), each image often exhibits different perspective distortions. If there are a large number of such images, the use of manual detection is labour-intensive as well as time-consuming. Therefore, automatic detection is more suitable for such images. The automatic method uses an edge detection algorithm (Canny) and then uses Hough transform to detect straight lines in an image. The detected straight lines are joined to form the longer lines by using simple heuristics and finally the largest possible quadrangle is formed from the final straight lines to consider the projected part. The above-mentioned technique is explained in [8].

In all cases, the system first uses the automatic method for the rectification and displays the rectified image to the user for its validation. If not, the system displays the original captured image and requests for the manual selection of the corners. Finally, the noise in the rectified image is removed using low-pass *Weiner filter* applied to each of the RGB-channel [30].

5.2 Features extraction

Often two types of features called (1) global or high-level and (2) local or low-level are used for the matching of document image. The accuracy of the extraction of the local features (texts, texture, shape, etc.) is mostly dependent on distortion, noise and the resolution of the captured image. On the other hand, the global features (physical layout, logical layout, objects, color, etc.) are not sensitive enough to such properties but are less reliable as compared to the local features for identification. However, in many practical situations local features are correlated with the global features. Therefore, a successful document retrieval algorithm should combine both the local and global features to achieve a significantly outstanding performance.

The goal of this low-resolution document identification task is to retrieve an original electronic document from a large document repository in a fast, yet efficient manner by querying a noisy, distorted low-resolution captured document. We propose a retrieval method based on signature that considers three different *feature sets* (*FS*) respectively; (1) shallow layout features (FS_1), (2) global color features (FS_2) and (3) distributed color features (FS_3). The shallow layout feature set consists of local features and is extracted

using layout analysis. This feature set (FS_1) plays an important role in the identification of slide images in comparison to the other *feature sets* (Section 5.6). However, color features can be used to enhance the identification performance, since the slide images which needed to be identified are queried on a repository containing numerous slideshows having different design pattern as well as different color content. In the proposed method, the color features are used as global features to filter down the solution set to a reasonable number of solutions, resulting in lower computational cost and matching time. Furthermore, only the global color feature set (FS_2) contains the color (pixel value) information, whereas distributed color feature set (FS_3) uses color in order to extract the features as a preliminary step to group the pixels of similar color.

5.2.1 Shallow layout features (FS_1)

This *feature set* is mainly based on the layout information of the document image. The resolution of the captured document is very low for the extraction of the complete layout structure, i.e. both physical and logical structures. Indeed, the average size of the projected part is of 450×560 and a resolution of below 75 *dpi*. For this reason, the shallow layout feature is extracted. The shallow layout feature is based on the layout structure and close to the perception of human vision. The extraction process is a top-down approach i.e. first of all, the global information of the document is considered and then partition the document into blocks before classifying them into texts, images, solid bars, bars with text. Initial blocks are extracted from a document image by considering the bi-level document image and then pass it through *Run Length Smearing Algorithm* (RLSA) both in vertical and horizontal direction. Finally, the output of both directions are combined with an AND operator [51]. Then, each block is classified as either text or image or solid line by looking to the features like block's eccentricity, mean horizontal run length, mean vertical run length, correlation between pixels in the horizontal and vertical lines. Moreover, the text blocks are separated with individual text lines and further processed to the word level (Fig. 12). Other features like bullet and vertical text lines are also extracted. Due to the poor resolution of the captured documents, the feature extraction is restricted to the word level. This is due to the difficulties faced to go further to the character level as the adjacent characters are often overlapped. Each feature in this *feature set* has a label tag, structured according to their priority and is one of the

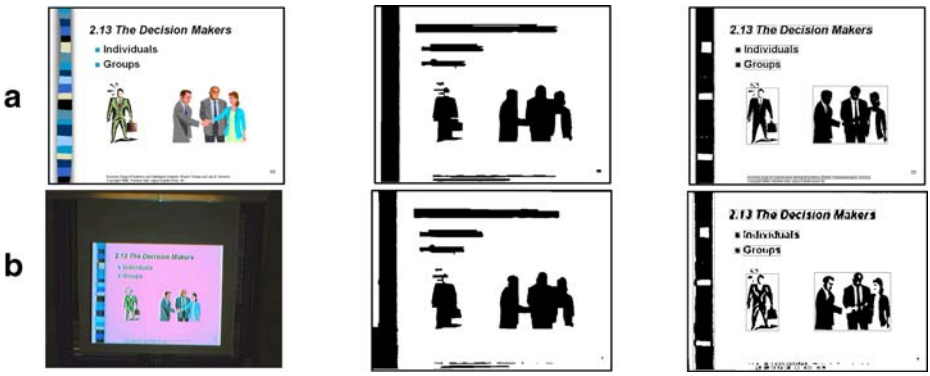

Fig. 12 **a** An original slide from the meeting repository, RLSA output, and the bounding box corresponding to the layout features (*top row*, *left-to-right*), **b** the same for the captured image from the slideshow video (*bottom-row*)

following: *horizontal text line*, *vertical text line*, *image*, *bullet*, *horizontal solid bar*, *vertical solid bar*, *horizontal bar with text* or *vertical bar with text*. The geometrical information of each feature such as its location, width, height and the bounding box pixel density are extracted and associated with the corresponding feature. Furthermore, the number of words in a text line and their respective geometrical properties are also computed. Figure 12 illustrates the bounding boxes of each of the feature, such as text lines, words, graphics, bullets, etc. of both the original and captured documents. After extracting all the features' properties, they are hierarchically organized according to their priority to form the signature called *layout signature* (Section 5.3). The detailed extraction procedures of the above-mentioned features are explained by Behera et al. [6].

5.2.2 Color distribution in rg-color space (FS_2)

This feature is considered as a global feature as it is based on the global color content of the document. The color histogram is commonly used for the color-based image retrieval. It describes the color distribution of an image in a specific color space. For a true RGB-color image, the histogram size is 2^{24}. The goal is to reduce the size of *feature set* without losing much information and the computational cost during retrieval. Considering these criteria, we decided to represent the color feature of the document with an *Equivalent Ellipse* having six parameters (center, major axis, minor axis, orientation, and density), which are computed from the kernel-based density estimation of the normalized histogram. The *Equivalent Ellipse* representation not only reduces the storage space but also speeds up the matching of features.

Normalized histogram generation A standard way of generating the RGB color histogram of an image is to consider the m higher bits of the Red, Green and Blue channels [47]. The histogram consists of 2^{3m} bins, which accumulate the number of pixels having similar color values. In order to avoid illumination, we consider the normalized $r = R/I$ and $g = G/I$, where $I = R + G + B$ is the brightness and $0 \leq R, G, B \leq 2^{m-1}$. The reduced color histogram $h(r, g)$ for an image of size $1...n_1 \times 1...n_2$ in rg-space is obtained as:

$$\begin{aligned} r = \text{int}\left(\text{M}r_{i,j}\right), g = \text{int}\left(\text{M}g_{i,j}\right), \text{M} = 2^m - 1 \\ h(r,g) = \frac{\#\text{pixelsfallinbin}r,g}{n_1 \times n_2}, 0 \leq r, g \leq \text{M} \end{aligned} \tag{1}$$

The similarity between images is often expressed as the similarity distance between respective histograms [53]. In such methods, the shape of the histogram strongly depends on the number of pixels and of the method used for lossy image representation. For smaller sized images, there would be very few points available for the histogram, which thus gives rise to erroneous results for the histogram-based comparison. In order to overcome this problem, a smooth non-parametric estimation of the color distribution is used instead of the discrete histogram representation and is based on the concept of non-parametric density estimation [41].

Color density estimation The general kernel-based estimation of a true multivariate density function $f(x)$ at any point x_0 in a d-dimensional space is given by

$$f(x_0) = \frac{1}{Nh^d} \sum_{i=1}^{N} K\left(\frac{x_i - x_0}{h}\right) \tag{2}$$

Where $i=1...N$ are the sample data points and K is the kernel function with kernel width, h. The estimation depends on the kernel function, K and the bandwidth, h. The *Epanechnikov* kernel has been shown to be robust to outliers and optimum in the sense of having minimum *mean integrated square error* (MISE) in comparison with other kernels [43].

$$K_E(x) = \begin{cases} \frac{1}{2} c_d (d+2)\left(1 - x^{\mathrm{T}} x\right) \text{ if } x^{\mathrm{T}} x < 1 \\ 0 \text{ otherwise.} \end{cases} \tag{3}$$

Where c_d is the volume of the unit d-dimensional sphere and x are the data points. In our case, we use a 2-dimensional estimation using Eqs. (2) and (3) in a reduced normalized *rg*-color space to avoid the computational expenses [7]. Figure 13 shows the density estimation of a sample of an original document (a), the captured image of the same from a projector (b) and from a handheld device (c).

Equivalent ellipse representation Each of the above-mentioned density surfaces is represented with an *Equivalent Ellipse*, which reduces the size of the features space from 2^{2m} (*rg*-space for *m*-bits/channel) to six. The parameters of the *Equivalent Ellipse* are computed from the distribution of the kernel density, K_d in the *rg*-color density surface. The center (C_r, C_g), axes (a, b) and pixel density d of the ellipse is computed as:

$$\begin{gathered} C_r = \sum_{r=1}^{M+1}\sum_{g=1}^{M+1} r K_d(r,g),\ C_g = \sum_{r=1}^{M+1}\sum_{g=1}^{M+1} g K_d(r,g),\ a^2 = \sum_{r=1}^{M+1}\sum_{g=1}^{M+1} (r - C_r)^2 K_d(r,g), \\ b^2 = \sum_{r=1}^{M+1}\sum_{g=1}^{M+1} (g - C_g)^2 K_d(r,g),\ d = \frac{\sum_{r=1}^{M+1}\sum_{g=1}^{M+1} K_d(r,g)}{\pi a b} \end{gathered} \tag{4}$$

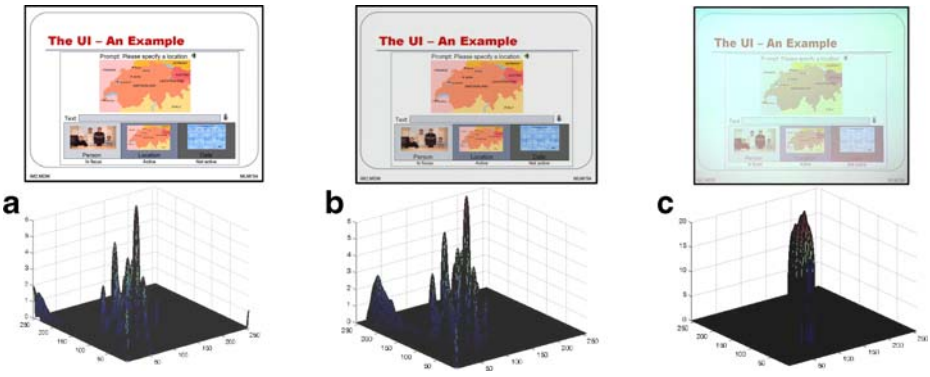

Fig. 13 Kernel density estimation of **a** an original slide document, **b** a slide image output from projector, and **c** rectified image captured from a digital camera (*left-to-right*)

The orientation of the ellipse, θ is computed using the least-squares fit of ellipse to 2-D points [19]. From Fig. 13, it is observed that the density surface of the image from the projector is very close to the original density surface but it differs noticeably for the captured image from the handheld devices. This is due to the presence of superimposed dominant color, i.e. color cast. This color cast is due to the change in lighting conditions, surface properties of the target object and even the characteristics of the capture devices. Furthermore, the resolution of handheld devices is quite low and often compressed with lossy image format such as JPEG. This creates difficulties in identification based on the matching of density surface. However, the goal is to identify a set of documents having similar color content and to reduce the number of elements in the *feature set* for fast matching. Therefore, the *Equivalent Ellipse* representation of the density surface is preferred. It is observed that most of the properties (eccentricity, orientation, etc.) of the *Equivalent Ellipse* of both the captured and the original images are preserved and that only the location is shifted (Fig. 14).

5.2.3 Color distribution in the image plane (FS_3)

In this method, the *feature set* is extracted by projecting the global color histogram in document's 2-D image plane. The *feature set* computed with the assimilation of the two different features such as the color feature and the geometrical layout feature. Even though the colors of the captured images are distorted due to changes in the lighting environment or even of capture devices, nonetheless the geometrical distributions of the color in the image plane remain constant. The feature extraction procedure starts with grouping of the pixels of similar color in the reduced RGB color space to generate two or more clusters. Then each cluster's center and radius in *X–Y* direction are computed in the 2-D image plane rather than in the 3-D RGB color space of document image.

Pixel clustering K-mean clustering is used and the number of clusters K corresponds to the number of predominant peaks in the 3-D color histogram of the image in reduced RGB color space. The color histogram is generated and smoothened using a *Gaussian* window and then the predominant peaks are selected [5]. Furthermore, the clusters' centroids are initialized with the average RGB values of the surrounding pixels of the selected peaks in the histogram and thus, the processing time is extremely less in comparison with random seeding or adaptive clustering [46].

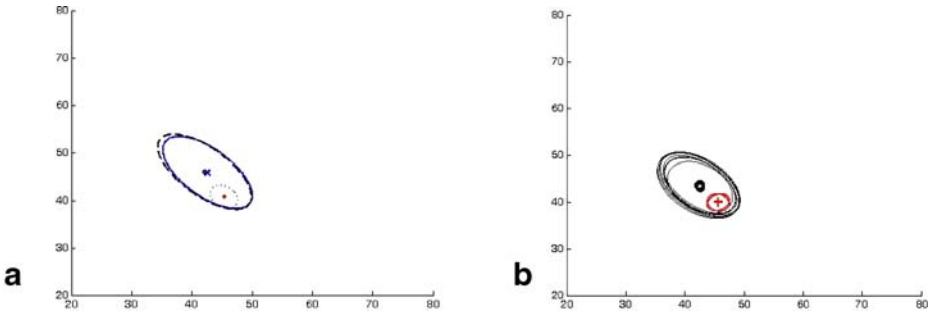

Fig. 14 *Equivalent Ellipses* represent the density surface of **a** original slide, output from projector and captured image from a digital camera, **b** original and captured slides from a slideshow

Geometrical distribution of each cluster Once the clustering is done, geometrical locations of pixels in each of the cluster are looked for as we are interested in their *X*–*Y* locations in the image plane rather than their values. For each cluster $i = 1...K$, the center ($C_{x,i}$, $C_{y,i}$) and the radius ($R_{x,i}$, $R_{y,i}$) are computed as:

$$C_{x,i} = \frac{1}{N_i}\sum_{\forall p \in i} X(p) \text{ and } C_{y,i} = \frac{1}{N_i}\sum_{\forall p \in i} Y(p)$$
$$R_{x,i} = \frac{1}{N_i}\sum_{\forall p \in i} \left(X(p) - C_{x,i}\right)^2 \text{ and } R_{y,i} = \frac{1}{N_i}\sum_{\forall p \in i} \left(Y(p) - C_{y,i}\right)^2 \quad (5)$$

Where N_i is the number of pixels in cluster *i*. The center and radius of each cluster is considered for the *feature set* along with the number of pixels per cluster. Figure 15 represents such features with the rectangle having a center representing the mean and the sides of the rectangle representing the variance of the geometrical locations of the pixels in the clusters. The clusters with solid boundaries are derived from the original image and dotted boundaries, from the captured images. It is observed that the clusters from the captured image from the projector are closer to the clusters of the original image (Fig. 15a) than those captured from DV camera (Fig. 15b). This is due to color deformations and low-resolution of the captured image, as explained earlier.

5.3 Documents' signature

After extraction of different *feature set* (*FS*) as described in the previous section, they are then structured in order to form the corresponding signatures. The *feature set*, FS_1 is the only local feature and the corresponding signature is called *layout signature*, S_1 (Table 2). The main idea of structuring the signature is to speed up the matching of signatures by giving more importance to the high-level features, which narrows down the search path. The organization of the features in the signature is based on the feature's priority; higher-level features appear first in the hierarchy and lower level features stand at the leaves (Fig. 16a). The hierarchy of the features in the signature is according to their extraction process. Features requiring less processing are first extracted and are more reliable than those that need more. Furthermore, the features that occur more frequently in the document are given higher priority than others.

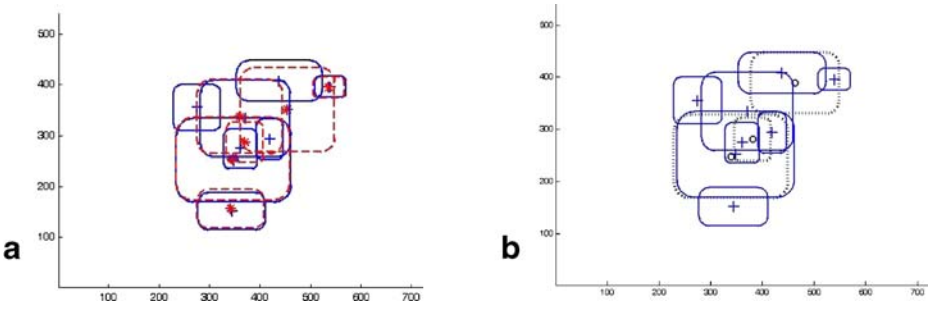

Fig. 15 Clusters are represented with rectangle in image plane **a** clusters from original slide (*solid boundary*) and output from projector (*left*) **b** clusters from original slide and captured from a digital camera (*right*) of Fig. 13

Table 2 Possible signatures using various feature sets

Signatures	FS_1	FS_2	FS_3
Layout Signature (S_1)	Yes	No	No
Global Color Signature (S_2)	No	Yes	No
Distributed Color Signature (S_3)	No	No	Yes
Document Signature (S_4)	Yes	Yes	Yes

The signature which corresponds to *feature set*, FS_2 is called the *global color signature*, S_2 (Table 2). There is no need of structuring it since it is constant for all documents and has only six parameters. The first node (GlobalColor) of Fig. 16b represents this signature.

The *distributed color signature*, S_3 contains the *feature set*, FS_3 (Table 2). The elements of the set vary with the number of major color contents in the documents. This is structured by keeping the clusters' properties in descending order of cluster density since the cluster having the highest number of pixels cover more area in the image than the others. The second node (DistributedColor) of the XML hierarchy of Fig. 16b represents such signature.

5.4 Matching of signatures

The signature of the captured document image is matched with the signatures of all the original electronic documents in the repository for identification. In this section, we discuss the technique and strategies used to match features in the corresponding structured signatures.

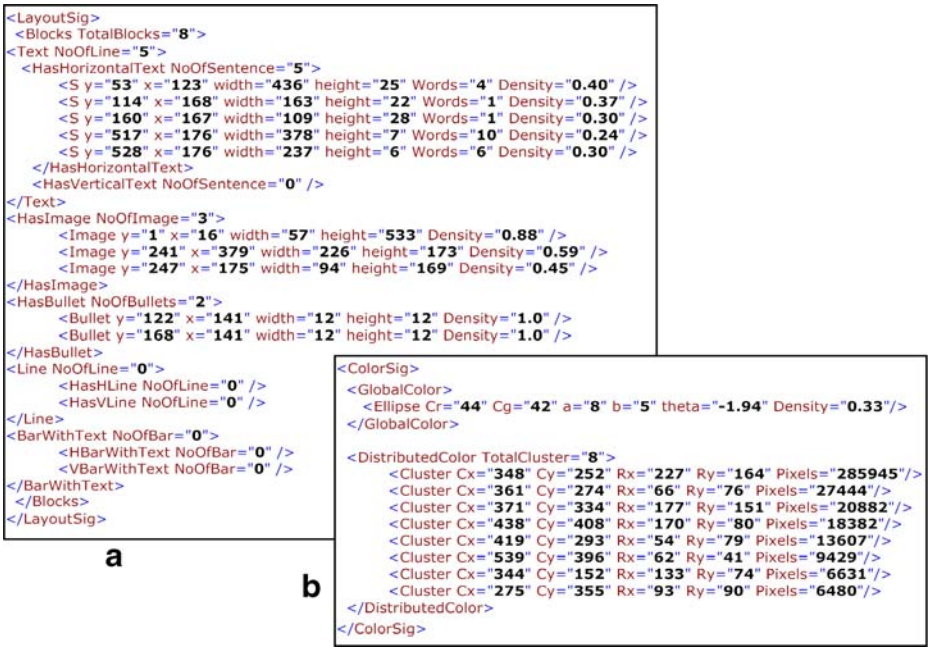

Fig. 16 **a** Layout signature from the *feature set* FS_1, **b** color signature from the *feature set* FS_2 (*GlobalColor* node) and FS_3 (*DistributedColor* node) of the slide document of Fig. 11 in XML format

5.4.1 Features matching

For the *layout signature* (S_1), the matching technique follows its hierarchy and uses simple heuristics. The matching score, f_i at each feature node, i is computed by considering the ratio of the number of matched element upon the total number of elements in that particular feature. For any element in the queried signature, the matching decision is taken by comparing the differences in the geometrical properties (co-ordinates, width and height) and bounding box pixel density of the elements in the same feature node of the target signature. Additionally, if the element belongs to the text node, then the difference in the number of words is also compared. The total score corresponds to the weighted sum of scores at each node. The weight, w_i is assigned adaptively considering both the number of elements in the node and the position of the node i in the hierarchical tree [6]. Then the total score per signature is computed as $s=\sum f_i w_i, \forall_i$ and after comparing the signatures within the whole repository the signature having the highest score, s is picked up for the solution.

The *global color signature* (S_2) has six constant features for any document image. The matching is performed by comparing the absolute distance between the corresponding features of the queried and target signature to a certain threshold, T_G.

For the matching of the *distributed color signature* (S_3), the properties of the clusters in the signature are compared. Due to the presence of color cast, often the number of clusters in the captured image is different than that of the original. The color cast provokes more convergence in the color histogram, i.e. adjacent colors are often brought closer (Fig. 13c). The idea is to imitate the geometrical distribution of the clusters of the captured image as in the original image, by merging the clusters in the original signature and *vice versa* (in case of divergence). This helps to bring the centroids of the resulting clusters in both images closer. On the other hand, separation of the clusters rather than merging is not feasible, since the locations of each pixel are not in the *feature set*. The matching follows the *top-down* approach. The properties of each cluster of the queried signature are compared using *one-to-one* followed by *one-to-many* mappings with the clusters in the target signature. In *one-to-many*, two or more clusters in the target signature are merged and then the resulting cluster's properties are compared. Let's say pth and qth clusters are to be merged and compared with the cluster i, then the resulting cluster's (j) density, center and radius are computed as:

$$\begin{aligned} &d_j = \left(N_p + N_q\right)/N, 1 \le p, q \le M,\ N = \#\text{total pixels} \\ &C_{x,j} = \left(N_p C_{x,p} + N_q C_{x,q}\right)/\left(N_p + N_q\right) \text{ and } C_{y,j} = \left(N_p C_{y,p} + N_q C_{y,q}\right)/\left(N_p + N_q\right) \\ &R_{x,j} = \left(N_p R_{x,p} + N_q R_{x,q}\right)/\left(N_p + N_q\right) \text{ and } R_{y,j} = \left(N_p R_{y,p} + N_q R_{y,q}\right)/\left(N_p + N_q\right) \end{aligned} \quad (6)$$

The resulting cluster j is considered for the set only if $||d_i - d_j|| < T_D$ and $||C_{x,i} - C_{x,j}|| + ||C_{y,i} - C_{y,j}|| < T_C$. If the set contains more than one cluster, then the cluster having properties such as center, radius and density is closer to the cluster i (minimum distance) is finally considered. If no match is found, then the same procedure is carried out with mapping of *many-to-one* followed by *many-to-many*. In the process of merging, the maximum number of clusters for merging is restricted to half of the total number of clusters. Figure 17 is an example of such merging of clusters. Before merging, there are eight clusters in the original image and only six in the captured image from the output of a projector. The clusters in the original image are then merged to six clusters for the comparison with the captured images. Similarly, the clusters in the original image are merged to three to bring them closer to the clusters in the captured image from the DV camera. It is easily understood that the clusters are brought closer after merging.

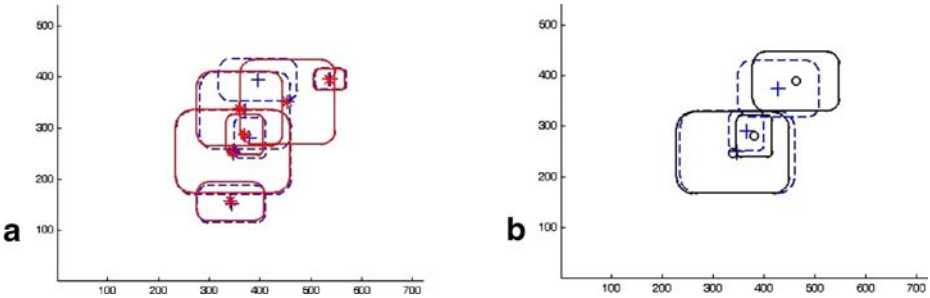

Fig. 17 Clusters of the original slides of Fig. 15 are merged (*dotted*) to imitate the clusters in the captured image **a** output from projector (*left*) **b** from a DV camera (*right*)

After computing the block distances between the clusters of the queried and all original signature, the signature having the minimum sum of all the block distances is considered for the final one. If there is more than one signature having the same minimum total distance, then the lowest distance between the radii of the clusters, which are not merged is considered.

5.4.2 Strategies for signature matching

The local features are more powerful than the global ones for the identification of low-resolution captured documents. Often, the projected documents have the same global features (FS_2 and FS_3). Therefore, the global features are first used to filter down the candidates to a reasonable smaller set and finally in combination with the local features are used for the final identification. Signature, S_3 containing *features set*, FS_3 would be more trustworthy than S_2 containing *feature set*, FS_2 as it dictates the distribution of the similar pixels in the image X–Y plane, whereas S_2 contains the global color content. Therefore, the matching of documents using only S_2 is not feasible as most of the documents have similar color content and is often distorted due to the presence of color cast. The matching of the signatures is carried out individually using S_1, S_2, S_3 and combination of S_1, S_2 and S_3 ($S_2 \rightarrow S_1$, $S_3 \rightarrow S_1$ and $S_2 \rightarrow S_3$, Table 3). In case of the combined matching, the first signature is used to filter down the solution set and combined with the following one for the final identification, which follows the rules of global to local feature matching. S_4 is the document signature which includes all the three signatures. It also follows the rules of global to local feature matching i.e. features in S_2 is first used to filter candidates followed

Table 3 Evaluation of the matching of signatures from the whole repository

Matching methods	Projector output			Camera output		
	I	*R*	Time	*I*	*R*	Time
S_1	0.93	0.002	1.57	0.73	0.074	1.54
S_2	0.16	0.000	0.33	0.01	0.000	0.34
S_3	0.86	0.005	0.42	0.54	0.008	0.37
S_2 then S_1	0.96	0.000	1.43	0.73	0.000	1.33
S_3 then S_1	0.96	0.000	0.43	0.78	0.050	0.48
S_2 then S_3	0.87	0.000	0.39	0.62	0.030	0.36
S_4	0.97	0.020	1.03	0.83	0.014	0.95

by features of S_3 for further filtering and finally the combined features score of S_1, S_2 and S_3 is used for identification.

5.5 Evaluations and results

In this evaluation, the recording of the conference on *Multimodal Interaction and Related Machine Learning Algorithms* (MLMI 2004) has been used. There were a total of 32 presentations and out of which the original documents of 30 presentations are available. The total number of projected documents is 684 and out of which 634 have been captured from the output of a projector (MP8749) having a dimension of 1,036×776 pixels with the resolution of 91.2 *dpi*. The output from the projector is connected to a capture card (*Datapath VGA capture card*) that captures presentation slides at native VGA resolutions and independent of presenter's laptop/PC hardware as well as presentation software. The projected documents are extracted from the conference video filmed on the projector screen using ParkerVision pan-tilt-zoom camera and then compressed with DivX (www.divx.com). The dimension of the projected part in the video is 750×570 pixels. A total of 674 documents have been first extracted from the *stable* period by using our document-based segmentation method (Section 4). Since the document videos from the conference are available presentation-wise, one could identify the extracted document from the video likewise. However, for the document image-based retrieval the queried document image should be compared with all the documents presented during the conference. For this purpose, we evaluated the proposed method by matching the captured document with the whole repository (Table 3) and presentation-wise (Table 4) without sorting the documents, as they are presented. All the original electronic and captured documents have then been processed to extract their corresponding signatures and kept in the repository after which the captured documents are queried for identification. For the evaluation, the metrics of recognition rate (I) and rejection rate (R) are used.

$$I = \frac{\#\text{correct documents recognized}}{\#\text{total documents queried}} \text{ and } R = \frac{\#\text{documents rejected}}{\#\text{total documents queried}}$$

The consideration of metrics of recognition and rejection rate rather than the *Recall* and *Precision* is due to the fact that we evaluated by querying the documents, which are already in the repository and only one solution is considered (true or false) rather than the top N solutions. The above-mentioned evaluation has been performed on a 1.7 GHz, 512 MB RAM, Pentium 4 PC.

Table 4 Results of the presentation-wise matching of signatures

Matching methods	Projector output			Camera output		
	I	R	Time	I	R	Time
S_1	0.94	0.003	0.15	0.78	0.133	0.15
S_2	0.25	0.000	0.06	0.08	0.000	0.06
S_3	0.89	0.002	0.10	0.68	0.004	0.09
S_2 then S_1	0.96	0.000	0.12	0.79	0.000	0.14
S_3 then S_1	0.96	0.000	0.08	0.82	0.020	0.11
S_2 then S_3	0.92	0.000	0.07	0.75	0.001	0.08
S_4	0.98	0.020	0.14	0.87	0.050	0.13

5.6 Discussion

As mentioned before, the signatures S_2 and S_3 consist of the global features. The performance using these signatures alone is not efficient (Second and third rows of Tables 3 and 4). When signatures, S_2 and S_3 are combined with the signature, S_1 which contains the local *feature set* the respective increment in performance of 80 and 10% for the images from projector of Table 3 (Fourth and fifth rows of Tables 3 and 4). This is due to the fact that the slides in a presentation often have the same color but different layout structure. The standalone performance of the global color signature S_2 is quite inferior to that of S_3. It shows that the global color content of the slides in a slideshow does not vary significantly, whereas the distribution of the similar pixel (color) in the 2-D image plane does. The performance using signature S_1 is lower as compared to the performance of S_1 combined with S_2 and S_3 (Fourth and fifth rows of Tables 3 and 4) since the slides in some of the slideshows have either non-uniform (gradient variation) and/or complex background (textured), which creates intricacy in extraction of local features. In most of the cases, the whole slide of such background is considered as a single image feature of the *feature set* of S_1. Therefore, in this case by combining the local and global features, not only increases the identification rate but also reduces the signature matching time, in seconds (last row of Tables 3 and 4) as compared to a single *feature set*. The performance of the images captured from the projector is much better than that of the captured image from the video camera and is obviously due to the poor quality of the latter. Most of the extracted documents from the presentation videos have non-uniform lighting, i.e. the center of the captured image is much brighter than the boundary (Fig. 13c). This introduces errors during the extraction of the features. Furthermore, this property mainly affects our shallow layout feature, which is considered as the local features and is more reliable than the other features. In the near future, we believe that a proper combination of all the above-mentioned signatures for exact identification rather than using global features to filter down the solutions and final identification based on local features would improve the current excellent result. In this evaluation, ideally the *rejection rate* (R) should be zero as the queried documents are already present in the meeting repository (Tables 3 and 4). Therefore, the tool should result in a solution, which could be either correct or false. Moreover, in some cases it is observed that the tool returns null as it is unable to take the decision since the captured document is too noisy to extract the features for identification. This occurs rarely as in this evaluation; the *rejection rate* is below 2% for the document images from projector's output and is inferior to 5% in the case of the documents captured from the video camera.

Both in the current and the previous section, we have finished the two major steps within the *analysis and indexing tool* of the system. At the end of the second step, the original electronic documents, which correspond to the captured low-resolution documents, are identified. In the following section, which is the final step in the *analysis and indexing tool*, we describe the extraction of the textual content from those identified electronic documents without using any standard OCR systems.

6 Electronic document content extraction

After having identified the corresponding original electronic documents, the textual contents should be extracted and added to the document signature in order to search using keywords. This is done using *Xed* (*eXtracting Hidden Structures from Electronic Documents*), a tool developed by our *DIVA* (Document Image and Voice Analysis) research group of the *University of Fribourg* [23]. The tool extracts the hidden layout structure of the PDF

documents and their contents (textual, graphical, etc.). Both the *layout signature* and the output from the *Xed* are in XML. The two XML files are matched in order to extract the textual content from the original document by considering the corresponding bounding box of the text feature. The procedure mentioned above is simple and no OCR technique is required. The use of OCR is time-consuming and generally requires various systems to deal with different languages. We could have extracted the textual content from the original PDF or PPT file by using the 'save as' option to RTF or HTML file. Nevertheless, in this case, one would get the textual content and not the geometrical and layout information, which implies using this option; one could not perform a reverse-engineering, i.e. to reproduce the original logical structure, whereas it is possible with Xed. Furthermore, the layout structure would help in the case of pointers or laser beams used during presentations in order to emphasize certain contents and could easily be annotated. Moreover, the layout information enhances the interactive browsing as it is shown in Fig. 3. For example, by clicking on different sections of a journal article in the browser, one would be able to access the audio/video clips at the time when it was discussed. The speech transcription at the same time and the corresponding projected document is displayed at this time. Finally, after this final third step, the meeting annotation file at the granularity of documents appears as in Fig. 18 for the document in Fig. 12.

7 Relevance and impact on other domains

Document analysis, recognition and retrieval systems play a major role in cutting-edge applications of multimedia technologies. To date, more and more audio–visual documents are captured and archived for future access. The current challenge is to deliver a system that can handle low-resolution documents, which are often captured using handheld devices.

The low-resolution document identification method we propose in this article can be generalized as a novel technique towards an efficient management of documents captured from handheld devices and we believe it has high impacts on prominent areas such as:

- Augmented reality systems and 3D meeting environment [4, 22, 35, 38] in which manipulated documents could be easily enhanced with meta-information, using our proposed document's signature, that structure a document according to its layout and color. Furthermore, it could enable interaction with documents, for instance to identify pointed parts;
- Life logging, personal information management or collective memory systems [2, 21, 50] that could be enriched with information and annotations on conferences, lectures and meetings, enabling the creation of knowledge maps of a person of a group, or the evolution of interests, etc.;

Fig. 18 Video annotations with meeting ID, slide ID, start and end time for slide along with their textual content

```
<VideoAnnotation MeetingID="2.5.2003_14.6" >
  <Document NoOfVisDocs="70" />
  <Doc StartTime="458.827" EndTime="464.307"/>
    <File Image="Slide68.JPG" eDoc= "Slide68.PDF" DocSig= "Slide68.xml"/>
    <Content>
     <Text>
      <TxtLine="2.13 The Decision Makers" />
      <TxtLine="Individuals" />
      <TxtLine="Groups" />
      <TxtLine="68">
      <TxtLine="Decision Support Systems and intelligent Systems, Efraim Turban and Jay E. Aronson" />
      <TxtLine="Copyright 1998, Prentice Hall, Upper Saddle River, NJ" />
     </Text>
    </Content>
</Doc>
<Doc>
…
</Doc>
…
</VideoAnnotation>
```

- Real-time document recognizers, in which our system could be adapted to extract and translate textual low-resolution document contents and signs, for instance for tourism (translation) or visually impaired people;
- And more generally, digital multimedia libraries in which the bridge between static documents such as articles, slideshows or books and temporal data, such as audio and video is often missing. Finally, and as a last example, our signature-based matching method could help detect multiple instances of a same document.

8 Conclusions and future works

In this paper, we have presented a fully automatic system (*DocMIR*) that supports document-centric meeting capture, indexing and retrieval. It consists of three major tools, which are (a) a capture tool, (b) an analysis and indexing tool and (c) a retrieval tool. Each tool has been described in details and our presentation mainly focused on the analysis and indexing tool, which analyzes and indexes meetings, lectures, seminars, etc. automatically using the projected documents. The automatic indexing process consists of mainly three steps: (1) the meeting videos segmentation based on projected documents, (2) the identification of low-resolution documents and finally (3) the extraction of the documents' textual content. The segmentation process looks for document stability rather than changes in the video of the projected documents. The document identification method uses a signature-based matching of documents. The extracted signatures consist of both layout as local feature and color as global feature for a robust and fast identification. The textual content of the identified document is extracted using its original electronic version and is added to the video annotation file along with the time-coded speech transcripts, which enable keyword-based search and retrieval. Then the retrieval tool allows users for linear and non-linear access to the captured and archived audio–visual streams by querying captured documents, original documents and/or keywords through the *FriDoc* browser. In conclusion, our *DocMIR* system imposes a novel and complete architecture, following a document-centric approach, for managing multimedia documents often captured from handheld devices. Additionally, the relevance of the system and its impact on other domains has been presented.

In the near future, we plan to improve the identification performance by considering the proper fusion strategies of the various layout and color features in the signatures rather than comparing them, sequentially. Additionally, the *DocMIR* system would be extended to consider the identification of partial and occluded projected documents. Finally, we intend to analyze the documents laid on the meeting table as a supplementary means to index meetings.

Acknowledgments We would like to thank the *International Workshop on Multimodal Interaction and Related Machine Learning Algorithms* (*MLMI 2004*) for providing the conference data consisting of the presented original electronic documents (PPT, PDF), the video of the projector screen and images captured from the output of projector. We would also like to thank the *University of Applied Sciences of Fribourg* and *Didier Von Rotz* for helping us set up the meeting capture environment.

References

1. Abowd GD, Atkeson CG, Feinstein CHA, Hmelo C, Kooper R, Long S, Sawhney NN, Tani M (1996) Teaching and learning as multimedia authoring: the classroom 2000 project. In: Proc. ACM multimedia, Boston MA, pp 187–198, November 1996

2. Adar E, Kargar D, Stein LA (1999) Haystack: per-user information environments. In: Proc. of 8th int'l conf. on information and knowledge management (CKIM), Kansas City, USA, pp 413–422
3. Aigrain P, Zhang H, Petkovic D (1996) Content-based representation and retrieval of visual media: a state-of-the-art review. Multimedia Tools and Applications 3(3):179–202
4. Barakonyi I, Fahmy T, Schmalstieg D (2004) Remote collaboration using augmented reality videoconferencing. In: Proc. of ACM graphics interface, Ontario, Canada, pp 89–96
5. Behera A, Lalanne D, Ingold R (2005) Enhancement of layout-based identification of low-resolution documents using geometrical color distribution. In: Proc. int. conf. on document analysis and recognition (ICDAR), Seoul, Korea, August–September, pp 468–472
6. Behera A, Lalanne D, Ingold R (2004) Visual signature based identification of low-resolution document images. In: Proc. ACM symposium on document engineering, Milwaukee, Wisconsin, pp 178–187
7. Behera A, Lalanne D, Ingold R (2005) Combining Color and Layout Features for the Identification of Low-resolution Documents. Int. Journal of Signal Processing (IJSP), ISSN: 1304–4478 2(1):7–14
8. Behera A (2006) A visual signature-based identification method of low-resolution document images and its exploitation to automate indexing of multimodal recordings. PhD Thesis
9. Bianchi MH AutoAuditorium: a Fully Automatic, Multi-Camera System to Televise Auditorium Presentations. In: Joint DARPA/NIST smart spaces workshop, Gaithersburg, MD, July 1998 http://www.autoauditorium.com/nist/autoaud.html
10. Boreczky JS, Rowe LA (1996) Comparison of Video Shot Boundary Detection Techniques. In: Proc. storage and retrieval for still image and video databases IV, IS&T/SPIE int. symposium on electronic imaging: science and technology, San Jose, CA, 2670:170–179
11. Brotherton JA, Bhalodia JR, Abowd GD (1998) Automated Capture, Integration, and Visualization of Multiple Media Streams. In: Proc. IEEE Int. Conf. on Multimedia Computing and Systems, Austin, TX, pp 54–63
12. Chiu P, Kapuskar A, Reitmeier S, Wilcox L (2000) Room with a rear view: Meeting capture in a multimedia conference room. IEEE Multimed 7(4):48–54
13. Chiu P, Foote J, Girgensohn A, Boreczky J (2000) Automatically linking multimedia meeting documents by image matching. In: Proc. ACM hypertext, San Antonio, TX, pp 244–245
14. Chiu P, Kapuskar A, Reitmeier S, Wilcox L (1999) NoteLook: Taking notes in meetings with digital video and ink. In: Proc. ACM multimedia, New York, pp 149–158
15. Cutler R, Rui Y, Gupta A, Cadiz JJ et al (2002) Distributed meetings: a meeting capture and broadcasting system. In: Proc. of ACM multimedia, Juan-les-Pins, France, pp 503–512
16. eClass, Georgia Institute of Technology, Atlanta, USA, http://www.cc.gatech.edu/fce/eclass/
17. Educational multimedia library project (EmuLib), University of Mannheim, Germany, http://www.informatik.uni-mannheim.de/informatik/pi4/projects/emulib/
18. Erol B, Hull JJ, Lee DS (2003) Linking multimedia presentations with their symbolic source documents: algorithm and applications. In: Proc. of ACM multimedia, Berkeley, CA, pp 498–507
19. Fitzgibbon AW, Pilu M, Fisher RB (1999) Direct least squares fitting of ellipses. IEEE Trans Pattern Anal Mach Intell 21(5):476–480
20. Girgensohn A, Boreczky J, Wilcox L, Foote J (1999) Facilitating video access by visualizing automatic analysis. In: Proc. of human–computer interaction INTERACT '99, IOS Press, pp 205–212
21. Gemmell J, Bell G, Lueder R, Drucker S, Wong C (2002) MyLifeBits: fulfilling the Memex Vision. In: Proc. ACM multimedia, Juan-les-Pins, France, pp 235–238
22. Geyer W, Richter H, Fuchs L, Frauenhofer T, Daijavad S, Poltrock S (2001) A team collaboration space supporting capture and access of virtual meetings. In: Proc. ACM supporting group work, Colorado, pp 188–196
23. Hadjar K, Rigamonti M, Lalanne D, Ingold R (2004) Xed: a new tool for eXtracting hidden structures from electronic documents. In: Proc. int. workshop on document image analysis for libraries (DIAL), Palo Alto, pp 212–224
24. He L, Sanocki E, Gupta A, Grudin J (1999) Auto-summarization of audio-video presentations. In: Proc. ACM Multimedia, New York, pp 489–498
25. Hunter J, Little S (2001) Building and indexing a distributed multimedia presentation archive using SMIL. In: Proc. of the 5th European conference on research and advanced technology for digital libraries, Darmstadt, Germany, pp 415–428
26. Lalanne D, Ingold R, Rotz DV, Behera A, Mekhaldi D (2004) Using static documents as structured and thematic interfaces to multimedia meeting archives. In: Proc. int. workshop on multimodal interaction and related machine learning algorithms (MLMI), Martigny, Switzerland, LNCS 3361:87–100
27. Lalanne D, Lisowska A, Bruno E, Flynn M et al (2005) The IM2 multimodal meeting browser family. Interactive Multimodal Information Management Tech. Report, Margtiny, Switzerland
28. LectureLounge, Fraunhofer-IPSI, Darmstadt, Germany, http://lecturelounge.ipsi.fraunhofer.de
29. Lee DS, Erol B, Graham J, Hull JJ, Murata N (2002) Portable Meeting Recorder In: Proc. ACM Multimedia, Juan-les-Pins, France, pp 493–502

30. Lim JS (1990) Two-dimensional signal and image processing, Englewood Cliffs, NJ: Prentice Hall
31. Livelink eloquent media server http://www.opentext.com/products/livelink/eloquent-media-server/
32. Lu T, Suganthan PN (2004) An Accumulation Algorithm for Video Shot Boundary Detection. Multimedia Tools and Applications 22(1):89–106
33. Meeting room, Carnegie Mellon University, http://penance.is.cs.cmu.edu/meeting_room/
34. Mukhopadhyay S, Smith B (1999) Passive Capture and Structuring of Lectures. In: Proc. ACM Multimedia, Orlando, FL, pp 477–487
35. Nakanishi H, Yoshida C, Nishimura T, Ishida T (1999) FreeWalk: A 3D virtual space for casual meetings. IEEE Multimedia 6(2):20–28
36. Otsu N (1979) A threshold selection method from gray-level histograms. IEEE Trans Syst Man Cybern 9(1):62–66
37. Petkovic M (2000) Content-based video retrieval. In: Proc. int. conf. on extending database technology, Konstanz, Germany, pp 74–77
38. Prince S, Cheok AD, Farbiz F, Williamson T, Johnson N, Billinghurst M, Kato H (2002) 3-D live: real time interaction for mixed reality. In: Proc. of the ACM computer supported cooperative work (CSCW), New Orleans, USA, pp 364–371
39. Rucklidge WJ (1997) Efficiently locating objects using the Hausdorff distance. Int J Comput Vis 24(3):251–270
40. Rui Y, Gupta A, Grudin J, He L (2004) Automating lecture capture and broadcast: technology and videography. ACM Multimedia Systems 10:3–15
41. Scott DW (1992) Multivariate density estimation. New York: Wiley
42. Shirmohammadi S, Ding L, Georganas N (2003) An approach for recording multimedia collaborative sessions: design and implementation. Multimedia Tools and Applications 19(2):135–154
43. Silverman BW (1986) Density estimation for statistic and data analysis. New York: Chapman and Hall
44. Smart multimedia archive for conferences project (SMAC), University of Fribourg, Switzerland, http://www.eif.ch/projets/smac/
45. Steinmetz A, Kienzle M (2001) The e-seminar lecture recording and distribution system. In: Proc. of SPIE multimedia computing and networking (MMCN), San Jose, CA 4312:25–36
46. Sural S, Qian G, Paramanik S (2002) Segmentation and histogram generation using the HSV color space for image retrieval. In: Proc. IEEE Intl. Conf. of Image Processing, Rochester, NY, pp 589–592
47. Swain M, Ballard D (1991) Color Indexing. Int J Comput Vis 7(1):11–32
48. Synchronized multimedia integration language (SMIL 2.1) specification, W3C recommendation, February 2005. http://www.w3.org/TR/SMIL2/
49. Trier ØD, Taxt T (1995) Evaluation of Binarization Methods for Document Images. IEEE Trans Pattern Anal Mach Intell 17(3):312–315
50. Wactlar H, Christel M, Hauptmann A, Gong Y (1999) Informedia experience-on-demand: capturing, integrating and communicating experiences across people, time and space. ACM Comput Surv (CSUR) 31(9)
51. Wong KY, Casey RG, Wahl FM (1982) Document analysis system. IBM J Res Develop 26:647–656
52. Zaho W, Wang J, Bhat D, Sakiewiez K, Nandhakumar N (1999) Improving color based video shot detection. In: Proc. IEEE Int. Conf. on multimedia computing and systems, Florence, Italy, pp 752–756
53. Zhang D, Lu G (2003) Evaluation of similarity measurement for image retrieval. In: Proc. IEEE int. conf. on neural network and signal processing, Nanjing, China, pp 928–931

Ardhendu Behera received his B.Eng degree (1st class honours) in Electrical Engineering from the National Institute of Technology, Allahabad, India in 1999, and M.Eng degree (1st class) in System Science and Automation from the Electrical Engineering Department of Indian Institute of Science, Bangalore, in 2001. He received the Ph.D. degree in Computer Science from the Department of Informatics, University of

Fribourg, Switzerland in 2006. He is currently working as a Research Fellow in the Computer Vision research group of School of Computing, University of Leeds, UK. He has been employed as a Member of Technical Staff (MTS) in the Multimedia group of Sun Microsystems, India before commencing his Ph.D. programme. His research interests are Document Image Processing, Image and Video Analysis, Documents Analysis and Recognition, Computer Vision and Pattern Recognition, Multimodal Systems, Cognitive Vision, Machine Learning and Information Retrieval.

Denis Lalanne is a senior researcher in the Department of Informatics of the University of Fribourg, Switzerland. He received his B.S. degree in Computer Science from Grenoble, France in 1993, M.S. degree in Cognitive Science from INPG (Institut National Polytechnique Grenoble), France in 1994, and PhD in Computer Science from the Swiss Federal Institute of Technology Lausanne (EPFL) in 1998. Dr. Lalanne has worked as a research member in the USER group (User System Ergonomics Research) of IBM Almaden Research Center, California, as a usability officer in Iconomic systems, a startup based in Switzerland, and as a research/teaching assistant in the University of Avignon (France). His major areas of expertise are Human Computer Interaction, Information Visualization, Artificial Intelligence, Multimedia, and Multimodal Content Management.

Rolf Ingold is a professor in the Department of Informatics, University of Fribourg, Switzerland, as well as director of the DIVA group (Document, Image and Voice Analysis) and head of IM2.DI, an individual project of the National Center of Competence in Research IM2. The DIVA research group covers several topics from the following areas: image processing and analysis, pattern recognition, document analysis and recognition, speech processing. He received his PhD degree in Computer Science from the Swiss Federal Institute of Technology Lausanne (EPFL) in 1989, Switzerland. Prof. Ingold is a member of the editorial board of several international journals as well as a member of the board of directors of the French association GRCE (Groupe de Recherche sur la Communication Ecrite). Current research themes are concentrated on image analysis, content-based image retrieval, as well as multimodal document alignment. His most significant achieved results cover font recognition, structure analysis of composite documents, document modeling, in which the University of Fribourg has become an incontestable leader.