Department of Computer Science
University of Fribourg, Switzerland

A Visual Signature-based Identification Method of Low-resolution Document Images and its Exploitation to Automate Indexing of Multimodal Recordings

THESIS

Submitted to the Faculty of Science
University of Fribourg, Switzerland
to obtain the degree of *Doctor Scientiarum Informaticarum*

Ardhendu Behera
from
Orissa, India

Thesis Nr. 1529

St Paul Printing, Fribourg
2006

Accepted by the Faculty of Science of the University of Fribourg (Switzerland), on the recommendation of:

- Prof. Dr. Heinz Gröflin, University of Fribourg, Switzerland (Jury President)
- Prof. Dr. Rolf Ingold, University of Fribourg, Switzerland (Thesis Director)
- Prof. Dr. Horst Bunke, University of Bern, Switzerland (External Examiner)
- Prof. Dr. Apostolos Antonacopoulos, University of Salford, UK (External Examiner)
- Dr. Denis Lalanne, University of Fribourg, Switzerland (External Examiner)

Fribourg, July 08, 2006

Thesis Director:

(Prof. Dr. Rolf Ingold)

The Dean:

(Prof. Dr. Marco Celio)

Wisdom cannot be pass'd from one having it, to another not having it,
Wisdom is of the Soul, is not susceptible of proof, is its own proof,
Applies to all stages and objects and qualities, and is content.

Song of the Open Road
Walt Whitman

Dedicated to my family

Summary

This thesis investigates methods for building an efficient application system for the document-based automatic indexing and retrieval (DocMIR) of multimedia data captured from multimodal environments such as meetings, conferences, etc. Both empirical image processing, video segmentation methods and document analysis approaches are studied to bridge the gap between temporal data and static information. The proposed system focuses on two major tasks: document-based video segmentation and low-resolution document image identification.

The captured audio-visual data of several hours should be fragmented into reasonable distinct smaller segments in order to provide useful access points. During a presentation, projected documents are often captured as a video stream and can be used as meaningful semantic pointers because they appear at specific time, remain in visual focus for a definite duration and summarize presenter's discourse at that time. The existing approaches for video segmentation are not applicable in this scenario since videos are captured from low-resolution devices, such as web-cams. In order to overcome these drawbacks, the proposed feature-based segmentation technique considers the stability rather than changes in video sequences. The technique does not require any document identification methods to confirm the change.

The identification of low-resolution documents is also required to link original electronic documents with the temporally segmented captured multimedia data. The proposed identification method uses a *Visual Signature* consisting of *Layout Signature* and *Color Signature*. This signature-based approach is considered for fast and efficient matching in order to fulfill the needs of real-time applications. It also overcomes the problem of poor resolution, noisy, complex backgrounds and varying lighting conditions of the capture environment. The visual features such as colors, their spatial distribution and layout features are extracted and structured hierarchically to form the *Color Signature* and *Layout Signature*, respectively. The matching of signature is based on both, sequential as well as multi-level linear and non-linear fusion of various visual features. The performance of the proposed technique has been compared with existing approaches using real data recorded from meetings and conferences and found to be significantly better.

The high-quality performances of the above-mentioned techniques prove the usefulness of documents as an additional modality and natural interface, to interact with multimedia data captured from multimodal environments.

Résumé

Cette thèse porte sur le développement d'un système complet pour l'indexation automatique, centrée sur le document (DocMIR), de données multimédias issues d'environnements multimodaux tels que réunions, conférences, etc. Tant des méthodes de traitement d'images que de segmentation vidéo et d'analyse de document sont utilisées pour mettre en relation les données temporelles de réunions avec les documents. Le système proposé s'articule autour de deux tâches principales : une segmentation vidéo basée sur le document et l'identification d'images de documents à basse résolution.

Plusieurs heures de données audio-visuelles doivent être fragmentées en segments de taille raisonnable afin de faciliter une navigation ultérieure. Durant une présentation, les documents projetés sont souvent capturés en tant que flux vidéo et peuvent être utilisés comme des pointeurs sémantiques pertinents, du fait qu'ils apparaissent à un instant spécifique, durent un temps déterminé et résument le discours courant de l'orateur. Les approches existantes ne sont pas applicables dans le cas où les vidéos proviennent d'appareils à basse résolution tels que des webcams. Pour remédier à ces inconvénients, la technique de segmentation proposée considère la stabilité plutôt que le changement dans les séquences vidéo et ne nécessite en outre aucune identification du document pour confirmer le changement.

Par ailleurs, une identification des documents à basse résolution est requise pour lier les documents électroniques originaux aux données multimédias segmentées. La méthode proposée utilise une signature visuelle du document composée des signatures de couleur et de mise en page. Les caractéristiques visuelles telles que les couleurs, leur distribution spatiale et la mise en page sont extraites puis structurées hiérarchiquement dans la signature. Cette approche permet une mise en correspondance rapide et efficace, afin de répondre aux besoins d'applications réelles. Elle résout par ailleurs les problèmes de faible résolution de l'image, des arrière-plans bruités et texturés ainsi que des conditions de luminosité variables de l'environnement de capture. Les méthodes de comparaison de la signature appliquent une fusion multi-niveaux séquentielle, linéaire ou non-linéaire des diverses caractéristiques visuelles. Cette nouvelle méthode d'identification a été comparée aux approches classiques au moyen de données réelles enregistrées lors de réunions et conférences, et s'est montrée significativement plus performante.

Les performances des différentes techniques développées dans cette thèse prouvent l'utilité des documents en tant que modalité additionnelle et interfaces naturelles pour interagir avec des données multimédias capturées dans des environnements multimodaux.

Acknowledgements

I would like to extend my heartfelt gratitude to Prof. Dr. Rolf Ingold, an exemplary form of perseverance; for his enriching advice, guidance, encouragements and support throughout the course of the degree program, without whom I would not have completed the project. He allowed me to run freely with his vision so that I could take it and make it my own. I could not overstate the value of his guidance, patience and support over the past four years. I am deeply indebted to my colleagues and fellow researchers for their work and contributions on this project.

I would like to extend my special gratitude to Dr. Denis Lalanne for all his efforts in co-supervising the thesis. Throughout the degree program, he has been informal to me. He taught me, not only how to be an effective researcher and author but also how to be a better person.

I also would like to thank David Bourillot at European Center for Nuclear Research (CERN) and Didier Von Rotz of Ecole d'Ingénieurs et d'Architectes de Fribourg (EIF), who have helped legitimize my research by applying it at their organizations.

I owe much to my dear parents, Mr. Jamini Kanta and Mrs. Gangamani Behera. Thanks to you for life, guidance and support. I would never have succeeded as I have without the support of my dear wife, Dr. Hema Viswambharan. Hema has supported me emotionally, technically and physically. Thank you, my love. To all my friends who stood by me who gave me strength to go on all the way of getting to the end and for all their selfless encouragements and sacrifices, I am ever grateful.

Last but not least, this work was carried out in the framework of the Swiss National Center of Competence in Research (NCCR) on Interactive Multimodal Information Management (IM2). The NCCR are managed by the Swiss National Science Foundation on behalf of the Federal Authorities. This work was also possible through financial support from the Swiss National Science Foundation for the project IM2.DI (Document Integration).

Inevitably, I would have missed some people and groups who contributed to my work and my life. I could only beg for their pardon here, claiming time pressures and sheer blockheadedness. Thank you, all of you, for making me who I am and helping to bring this work to fruition.

Table of contents

Summary i

Résumé ii

Acknowledgements iii

Table of contents iv

List of figures x

List of tables xiv

Chapter 1. Introduction 1

1.1. Context 3

1.1.1. Meeting/lecture recordings 4

1.1.2. Usefulness of documents in meeting/lecture 4

1.1.3. Type of document alignments 5

1.2. Aim of this thesis 6

1.3. Possible target applications 6

1.4. Contribution of this thesis 7

1.5. Thesis organization 8

Chapter 2. Related works and background 11

2.1. Previous works 11

2.1.1. Meeting/lectures analysis and retrieval 11

2.1.2. Application-specific classification 12

2.1.2.1. Academic applications 12

2.1.2.2. Industrial applications 13

2.1.3. Classification based on sensor-specific indexing methods 13

2.1.3.1. Whiteboard-based indexing 14

2.1.3.2. Camera-based indexing 15

2.1.3.3. Microphone-based indexing 15

2.1.3.4. Special device-based indexing 16

2.1.4. Access and replaying 17

2.1.4.1. Web-based Browser 17

2.1.4.2. Customized User Interface 17

2.1.5. Summary 18

2.2. Background information 18

2.2.1. Content-based image indexing and retrieval 18

2.2.2. Document recognition and retrieval 20

2.2.2.1. Document physical and logical structure 20

2.2.2.2. Information/content extraction and indexing 21

2.2.2.3. Document image matching 22

2.2.2.4. Summary 22

2.3. Conclusion 22

Chapter 3. Automatic document-based meeting indexing 25

3.1. Non-linear navigation aid 25

3.2. Linking electronic documents with temporal data 26

3.3. Overview of document-centric DocMIR system 27

3.3.1. Capture module 29

3.3.1.1. Capture architecture 29

3.3.1.2. Light-weight, distributed and scalable 30

3.3.1.3. Independent of capture devices 31

3.3.2. Analysis and indexing module 31

3.3.2.1. Meeting segmentation 32

3.3.2.2. Low-resolution document recognition 32

3.3.2.3. Content extraction and addition 33

3.3.2.4. Indexing performance 34

3.3.3. Retrieval module 34

3.3.3.1. Image-based retrieval 35

3.3.3.2. Keyword-based retrieval 35

3.3.3.3. Summary of retrieval performance 37

3.4. Conclusion and perspectives 38

Chapter 4. Document-based segmentation of meetings using *Slide Change Detection* 41

4.1. Motivation 41

4.2. Review of existing approaches 42

4.2.1. State-of-the-art methods 42

4.2.2. Cornell's lecture browser method 45

4.3. Proposed method 46

4.3.1. Stability detection 47

4.3.2. Slide Change Detection 48

4.4. Experiments and evaluations 50

4.4.1. Experimental setup 50

4.4.2. Criteria for evaluating the segmentation performance 51

4.4.3. Automatic evaluation 52

4.4.4. Performance comparison 53

4.4.5. Summary of segmentation performances 54

4.5. Conclusions and perspectives 57

Chapter 5. Low-resolution document identification 59

5.1. Motivation 59

5.2. Existing approaches 60

5.2.1. Layout-based document identification 61

5.2.2. Texture-based document identification 63

5.2.3. Color-based document identification 64

5.2.4. Content-based slide identification 65

5.2.5. Impediments in using the existing approaches 67

5.3. Our proposed approach: document *Visual Signature* 68

5.4. Rectification and noise removal of captured documents 70

5.4.1. Detection of the quadrangle containing projected part 72

5.4.1.1. Edge detection 74

5.4.1.2. Straight lines detection 76

5.4.1.3. Quadrangle formation 78

5.4.1.4. Evaluation of automatic quadrangle detection 80

5.4.2. Noise removal 82

5.5. Conclusions and perspectives 83

Chapter 6. Document *Layout Signature* **85**

6.1. Layout features extraction 86

6.1.1. Run-Length Smoothing Algorithm 87

6.1.2. Bounding box extraction 88

6.1.3. Bounding box labeling 89

6.1.4. Text lines and words 90

6.1.4.1. Horizontal text lines and words 90

6.1.4.2. Vertical text lines and words 91

6.1.5. Image, horizontal and vertical solid lines 93

6.1.6. Bullets 93

6.1.7. Horizontal and vertical bar with text 96

6.1.8. Summary 96

6.2. Structuring the features in the *Layout Signature* 97

6.3. Matching of *Layout Signatures* 99

6.3.1. Exhaustive search 101

6.3.2. Hierarchical search 102

6.4. Conclusion and perspectives 104

Chapter 7. Document *Color Signature* **107**

7.1. Color features extraction 108

7.1.1. Global color features 109

7.1.1.1. Normalized histogram computation 109

7.1.1.2. *Kernel Density Estimation* 110

7.1.1.3. *Equivalent Ellipse* representation 113

7.1.2. Distributed color features 114

7.1.2.1. Reduced 3-D histogram 116

7.1.2.2. Valid peaks 116

7.1.2.3. K-mean cluster initialization 117

7.1.2.4. K-mean clustering technique 117

7.1.2.5. *Equivalent Rectangle* representation 119

7.1.3. Structuring of features in Color Signature 121

7.2. Matching of document *Color Signatures* 122

7.2.1. Global color feature matching 122

7.2.2. Distributed color feature matching 123

7.3. Conclusion and perspectives 126

Chapter 8. Visual Signature evaluation: combining *Layout* and *Color Signature* 129

8.1. Constructing document *Visual Signature* 129

8.2. Matching of *Visual Signatures* 130

8.3. Experiments and evaluations 131

8.3.1. Experimental setup 131

8.3.1.1. *Smart Meeting Room* data set 132

8.3.1.2. *MLMI*'04 data set 132

8.3.2. Evaluation procedure and evaluation metrics 133

8.3.3. Performance of the document Layout Signature 134

8.3.4. Rules for combining two or more features set 137

8.3.5. Performances of sequential matching 138

8.3.5.1. First experiment: effect of global color feature 138

8.3.5.2. Second experiment: effect of distributed color feature 140

8.3.5.3. Third experiment: combining all 141

8.3.6. Fusion strategies 144

8.3.7. Performance using fused layout and color features 146

8.3.8. Performance comparison 148

8.4. Conclusions and perspectives 149

Chapter 9. Conclusions and perspectives 151

9.1. Major achievements and contributions 151

9.1.1. Recording task 151

9.1.2. Document-based segmentation task 152

9.1.3. Document identification task 152

9.1.4. Retrieval and browsing task 153

9.2. Limitation of the current work and possible future research directions 153

9.3. Impact of this thesis 155

Bibliography 157

Webliography 169

Curriculum Vitae 171

List of figures

Fig. 1.1. A simple model for mass document convergence and document engineering process: (a) conversion of electronic to paper documents, (b) paper to electronic documents, (c) viewing/playback of electronic documents and (d) creating and/or editing electronic documents. 3

Fig. 2.1. A simple model of multimodal meeting/lecture capturing, indexing and annotation system. 12

Fig. 2.2. Content-based image retrieval framework [*Remco* and *Tanase*, 2000]. 20

Fig.2.3. Digitization of physical documents: (a) physical document, (b) acquisition, (c) document structure analysis, (d) processing of each region and (e) document modeling and understanding. 21

Fig. 3.1. Complete architecture of the *DocMIR* system for automatic indexing and retrieval of recorded meetings using three main modules: (a) capture modules (top bounding box), (b) analysis and indexing module (left, outmost bounding box), and (c) retrieval module (right, outmost bounding box). 28

Fig. 3.2. Snapshots of the document-centric *Smart Meeting Room*: (a) FireWire web-cam and Labtec microphone pair per participant, (b) overview from a web-cam capturing meeting, (c) another overview web-cam positioned diagonally opposite and (d) a web-cam focused on the projector's screen. 29

Fig. 3.3. Snapshot of meeting recordings played with SMIL containing multiple audio/video streams containing the close-up view 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). 31

Fig. 3.4. Content extraction from the electronic documents by comparing the *Layout Signature* of captured documents with the *Xed* output without using any *OCR* systems. 33

Fig. 3.5. FriDoc browser; cross-meeting navigator (left), keywords and/or captured image-based retrieval of documents, that are linked to the corresponding meeting segments and intra-meeting navigator (right) containing audio/video, documents, speech transcripts, control bar with sunBurst visualization. 38

Fig. 4.1. Auto-focusing modifies the lighting conditions and fading during transition. 44

Fig. 4.2. Stability detection in the video stream using a sliding window with duration of 2 seconds. 47

Fig. 4.3. Confirmation of instability by considering each frame in the sliding window. 48

Fig. 4.4. Two or more consecutive windows of the same type are merged to form stable period or unstable period. 49

Fig. 4.5. An example of SMIL file, which is the output of our slide change detection. 50

Fig. 4.6. Slideshow-wise performances of various *SCD* algorithms using *F-measure* (*F*) for tolerance of 1 (left) and 4 frames (right). 53

Fig. 4.7. Overall performances of various *SCD* algorithms using *F-measure* (*F*) with *SEM* for tolerance of 1 and 4 frames. 53

Fig. 4.8. Slideshow-wise performances of various *SCD* algorithms using *Recall* (*R*) for tolerance of 1 (left) and 4 frames (right). 55

Fig. 4.9. Overall performances of various *SCD* algorithms using *Recall* (*R*) with *SEM* for tolerance of 1 and 4 frames. 55

Fig. 4.10. Slideshow-wise performances of various *SCD* algorithms using *Precision* (*P*) for tolerance of 1 (left) and 4 frames (right). 56

Fig. 4.11. Overall performances of various *SCD* algorithms using *Precision* (*P*) with *SEM* for tolerance of 1 and 4 frames. 56

Fig. 5.1. An example of page segmentation: (a) original image of a newspaper and (b) logical structure of the newspaper with text and image zones. 61

Fig. 5.2. Three different textures with the same distribution of colors. 64

Fig. 5.3. *Global Color Histogram* (*GCH*) of the block pattern, striped pattern and checkerboard pattern (left-to-right) of the Fig. 5.2. 65

Fig. 5.4. Low-resolution document identification: a systematic procedure for feature extraction to form the document signature and its matching from the repository. 69

Fig. 5.5. Documents captured from a DV camera, a web-cam, a mobile phone and a digital camera (left-to-right). The document zone is highlighted. 70

Fig. 5.6. Transformation of a quadrilateral from an image plane to world plane of world coordinate system with z = 0. 71

Fig. 5.7. Deformation correction using perspective transformation (a) captured image loaded in application for correction, (b) after perspective correction, (c) background removal by cropping. 73

Fig. 5.8. Edge detection by derivative operator: (a) an image containing a dark strip on a light background, (b) vertical profile of a line in the image and its (c) first order derivative and (d) second order derivative [*Gonzalez* and *Woods*, 2002]. 74

Fig. 5.9. Edge image of a captured document image using Sobel operators and Canny edge detection algorithm. 76

Fig. 5.10. The Hough transform of a line in X-Y plane to a point in R-θ plane [*Jain*, 1997]. 77

Fig. 5.11. Detection of straight lines from the edge image of the captured image in Fig. 5.9a. 77

Fig. 5.12. Automatic detection of boundary of projected part: (a) an image captured using handheld digital camera, (b) detected straight lines from Hough space, (c) derived final quadrangle from merged straight lines. 79

Fig 5.13. (a) Merging of detected Hough lines of Fig. 5.11c to form longer straight lines and (b) the detected final quadrangle from these straight lines........ 79

Fig. 5.14. Non-uniform boundary added to the captured document from the capture card. 81

Fig. 5.15. Detection of quadrangle containing the projected document in the captured images from (a) capture card, (b) digital camera and (c) DV-camera. 81

Fig. 5.16. Result of the perspective correction to the captured image in the Figs. 5.15b and 5.15c. 82

Fig. 5.17 Noise removal from low-resolution documents: (a) rectified perspective distortion of a captured slide image and (b) removal of noise using low-pass *Wiener* filter. 83

Fig. 6.1. *Otsu* segmentation: (a) original slide document, (b) output of *Otsu* segmentation and (c) correction to the *Otsu*'s output using mean horizontal run length of white and black pixels. 86

Fig. 6.2. Output of the *RLSA*: (a) horizontal direction, (b) vertical direction and (c) combining both the directions, of the binary image in Fig. 6.1c. 88

Fig. 6.3. (a) One slide document and (b) its corresponding *RLSA* output where two or more text lines are in the same text block component. 90

Fig. 6.4. Horizontal projection profile of a text block to separate individual text lines. 91

Fig 6.5. Vertical projection profile of a text line for the detection of words. 91

Fig. 6.6. (a) A typical slide document image containing text and graphics, (b) visual features such as horizontal text lines, vertical text lines, words in text lines, solid lines, bars with texts, images and bullets are all parts of the *Layout Signature*. 93

Fig. 6.7. (a) original slide document, (b) segmented blocks using connected component analysis to the final output of *RLSA*, (c) captured document image using DV-camera of the same projected slide, (d) after perspective correction and noise removal (Chapter 5), (e) segmented blocks using connected component analysis to the final output of *RLSA*........ 94

Fig. 6.8. (a) An original slide from the meeting repository, (b) bounding box corresponding to the layout features in the *Layout Signature*, (c) and (d) the same for the captured image from the slideshow video. 97

Fig. 6.9. The *Layout Signature* of the original document image in Fig. 6.8a is represented using XML. 98

Fig. 6.10. Tree representation of layout features in a *Layout Signature*. 98

Fig. 6.11. Flow chart of the hierarchical matching technique........ 103

Fig. 7.1. (a) Original image, (b) *KDE* of the color distribution in the *rg*-color space, (c) its pseudo-color representation for true color (24-bits) and d) reduced color (21-bits). 112

Fig. 7.2. (a) Captured image of Fig. 7.1a from projector, (b) its *KDE* of the color distribution, and (c) pseudo-color representation for 21-bits in the *rg*-space, (d) rectified image captured using a digital camera, (e) its *KDE* of the color distribution and (f) *Equivalent Ellipse* of the density surfaces of the original and the two captured image.. 113

Fig. 7.3. *Equivalent Ellipse* representation of the estimated color densities in the reduced *rg*-space of (a) slide documents randomly picked from 5 different slideshows and (b) original and captured slide documents from a single slideshow. 115

Fig. 7.4. Histogram for peaks detection of (a) original image (Fig. 7.1a) and (b) its captured image using a digital camera (Fig. 7.2d) in reduced 6-bits RGB space.. 117

Fig. 7.5. Clusters are represented with *Equivalent Rectangle* in image plane (a) clusters from original slide (Fig. 7.1a, solid boundary) and output from projector (Fig. 7.2a) (b) clusters from original slide and captured from a digital camera (Fig. 7.2d).. 121

Fig. 7.6. *Color Signature* consisting global color features (*Equivalent Ellipse*) and distributed color features (*Equivalent Rectangle*) of the slide document of Fig. 7.1a in XML format.. 121

Fig. 7.7. Merging of *Equivalent Rectangles*: (a) and (c) original slide from two different presentation, (b) and (d) corresponding *Equivalent Rectangles*, (e) the *Equivalent Rectangles* of each image is merged to represent single rectangle. 125

Fig. 7.8. (a) Rectangles from the original slide image are merged (dotted) to imitate the rectangles in the captured image from the projector (Fig. 7.5a), (b) rectangles from both the original and the captured image from a digital camera are merged to bring them closer for the comparison. 126

Fig. 7.9. (a) Captured slide image using a handheld digital camera, (b) pre-processed captured image, (c) formation of image after clustering (4 clusters), d) original slide image, (e) formation of image after clustering (8 clusters) and (f) formation of image after merging of clusters in original image to 4 clusters.. 127

Fig. 8.1. The top-down structure of the *Visual Signature* of a document consists of layout and *Color Signature* at the intermediate level and the corresponding feature sets at the leaf level.. 130

Fig. 8.2. Derivation of weighting function W_1 and W_3 for the non-linear fusion using the matching score of feature sets, F_1 and F_3... 145

List of tables

Table 4.1. Comparison of performances of various slide change detection methods 54

Table 6.1. Thresholds used for the layout feature vectors comparison 101

Table 6.2. Weights used for the various layout features 101

Table 8.1. Results of evaluation for images captured using a webcam 135

Table 8.2. Results of evaluation for images captured using a digital still camera 136

Table 8.3. Results of sequential matching of global color and layout feature set for images captured using a digital video camera 139

Table 8.4. Results of sequential matching of distributed color and layout feature set for images captured using a digital video camera 140

Table 8.5. Evaluation of the performance using single and sequential matching of various feature set by considering whole repository 142

Table 8.6. Evaluation of the performance using single and sequential matching of various feature set by considering presentation-wise 142

Table 8.7. Results of sequential versus fused matching of distributed color and layout feature set for images captured using a digital video camera 146

Table 8.8. Evaluation of the matching of document *Visual Signature* 147

Table 8.9. Performances of various *OCR* systems on slides captured from the projector's output in the *MLMI'04* data set [*Daddaoua et al.*, 2005] 148

Table 8.10. Performance evaluation of the matching method proposed by *Chiu et al.* 149

Chapter 1

Introduction

What is a document? To answer this question, various available resources should be considered. According to the *Oxford* dictionary, a document refers to papers, forms, books, etc. giving information about some evidence or proof of something. In *Wikipedia* (encyclopedia, www.en.wikipedia.org), a document is termed as a container of information. It often refers to an actual product of writing and is usually intended to communicate or store collections of data. The online dictionary (www.dictionary.com), defines it as a written or printed paper that bears the original, official, or legal form of something and can be used to furnish decisive evidence or information.

From ancient times, paper documents have been used as the main container for archiving information. Paper documents exist from 105 A.D., when it was first invented by a courtier named *Ts'ai-Lun*, from Lei-yang in China. However, the word *paper* is derived from the name of the reedy plant papyrus, which grows abundantly along the river *Nile* in Egypt.

From the late 1980s, desktop computers are widely used and therefore, electronic documents play a major role in archiving information. Nowadays, although electronic documents are extensively used, it has to go a long way to be fully taken over by paper documents or as paper-less office.

Documents are fundamentally multimodal and complex. They have to be navigated as they have *state* (either physical or electronic) and could be used to access history. Each state has its own strength and weaknesses. For electronic document, copying, editing, transmitting, sharing, retrieving, archiving, etc. are its strengths. Moreover, it allows keyword searching, spell checking and instant calculation. However, the content depends on the corresponding application for editing. Similarly, a paper document has its own qualities, since it is tangible, universally accepted, portable, cheap, familiar, high resolution and easy to read. Moreover, paper documents are tactile, hands and fingers can be used to manipulate them and can be doodled on with a pen/pencil. The conversion of electronic document to paper documents is straight forward. However, the reverse requires an additional intermediate process (other than scanning) of document image analysis and recognition, which is cumbersome.

Documents in all forms are frequently found in everyday life, such as in business and accounting (*e.g.* contract, invoice, quotations, etc.), academic and scientific (*e.g.* thesis, journal/conference articles, presentations, etc.), media and marketing (*e.g.* script, brief, mock-up, etc.), legal and political (*e.g.* summons, license, gazette, etc.), technical and trade specific (*e.g.* white paper, proposal, etc.). The use of document implies document personalization and often the same documents appear in various forms (*e.g.* physical, electronic, graphics, etc.) and require a proper system for managing them. Therefore, if one could automate the management procedure, it would have a significant impact on our understanding, which increases productivity eventually.

The conversion of physical image such as photographs, printed text, or handwriting to a digital image is carried out by using a scanner. Drum scanners is the oldest scanning technology. In 1985, world's first 300-dpi black-and-white sheet-fed scanner was introduced by *Microtek* (www.microtek.com). Most scanners today are variations of desktop scanners, handheld scanners, 3D scanners, etc. Nowadays, other devices such as digital cameras are also often used for capturing physical images. In 1988, the first true digital camera was introduced by the *Fuji* DS-1P (www.fujifilm.com), which recorded images as a computerized file to a 16MB internal memory card. From 1991 onwards, digital cameras are commercially available. Since then digital cameras are available in the form of various handheld devices such as mobile phones, PDA, etc. Similarly, a computer printer is a device that produces a human-readable text and/or graphics, usually on media such as paper or transparencies, from data stored in computers. In the 19_{th} century, the world's first computer printer mechanically driven apparatus, was invented by *Charles Babbage*. Today, modern printing technology such as laser printers and inkjet printers are adequately used to print everything. Using the above-mentioned various captures and printing devices, document engineering process cycle could be easily understood and is depicted in Fig. 1.1.

An increasing number of people use mobile devices such as digital cameras, mobile phones, etc. in their daily activities. Due to the relatively small size of such devices, they can be carried anywhere, at any time to capture documents of interest in lectures, meetings, conferences, supermarkets, etc. It implies lots of new possible applications such as mobile *Optical Character Recognition* (*OCR*) for the visually impaired, real-time translation, etc. These captured documents could be queried to retrieve the original documents and the available related information. Therefore, the captured documents from such low-resolution devices must be identified from a database containing original documents, which are often linked with the respective related information.

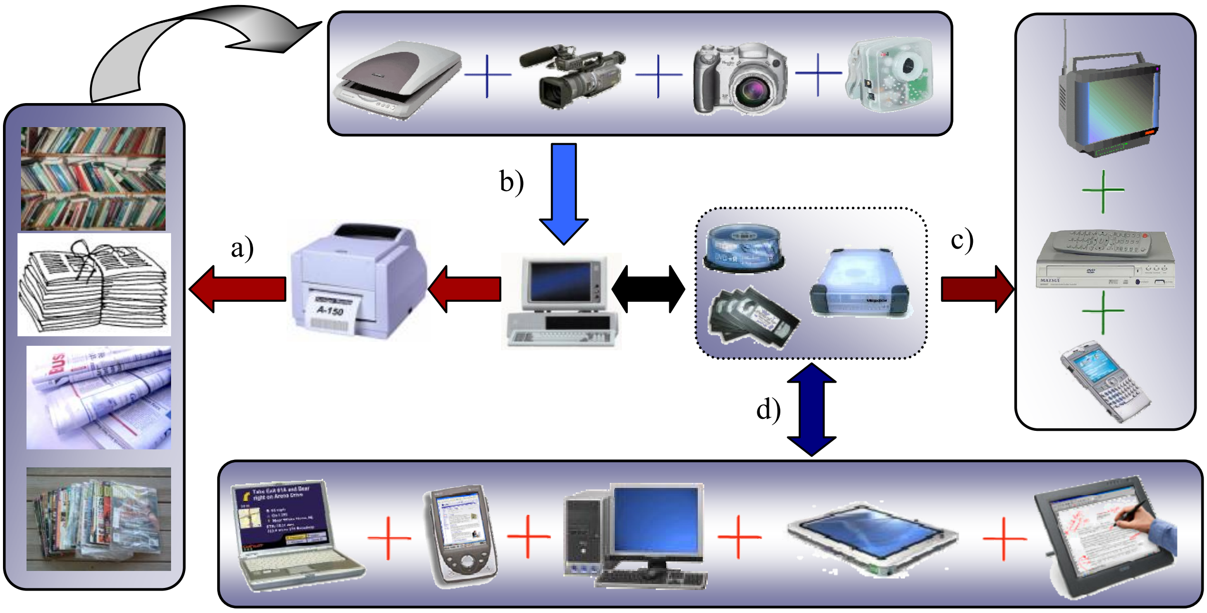


Fig. 1.1. A simple model for mass document convergence and document engineering process: (a) conversion of electronic to paper documents, (b) paper to electronic documents, (c) viewing/playback of electronic documents and (d) creating and/or editing electronic documents.

1.1. Context

As mentioned earlier, documents are important source of information and are often used by in various environments. One of such example is the meeting/lecture environment, in which documents are extensively used and are either discussed or projected. Meeting room, class room, auditorium and seminar room activities are typical examples of a multimodal environment. A typical meeting contains an important part of the design and development environment. They are used as communication and co-ordination activities of teams. Meetings are used to discuss the status, explore ideas, presenting information, resolve disagreements, making decision and enhance teamwork to achieve team goals. Therefore, meetings contain a large amount of information that is often difficult to analyze for making an accurate summary, especially if the only record we have is our own memory and perhaps supplemented with handwritten notes. However, writing down the content of meetings thoroughly, is a difficult task and could result in an inability to take notes and participate simultaneously. A common approach is the use of portable audio cassette recorder as a portable memory aid. It could be effective, but lacks the ability to capture important events, which could be helpful later such as gestures, presented documents, images of participants, body language, drawings, and so on.

Due to the advancement of technologies, the meeting room activities can be automatically captured using sensors such as cameras, microphones, etc. Therefore, participants are left free to engage actively in discussions and synthesize the happenings around them, without worrying about preserving details, tiresomely for later recall.

Meeting/lecture environment can be modeled as a *multimodal* environment, in which *human-to-human* interactions are often carried out using *human-to-computer* interactions that use different modalities such as voice, gesture, documents, typing, etc. The recorded information from a *multimodal* environment is stored as *multimedia* streams such as audio, video, images, electronic documents, etc.

1.1.1. Meeting/lecture recordings

The most common choice for recording meetings is the use of audio and video, which provides a comprehensive meeting record, which allows one to observe who were present, when a particular topic was discussed, who were giving presentations, what was discussed and the final outcome. The meeting environment is a human-to-human interaction through various modalities such as speech, presentations (slides, transparencies), black/white boards, paper documents, etc. Therefore, the meeting recordings should not only consider the audio/video of participants but also captures all the documents that are presented or discussed. This would help people to access all the necessary information without contacting the speaker for later clarifications.

Abwod et al. first introduced one of such recording system for capturing classroom happenings and it allows accessing the captured information to enhance both the learning and teaching experience [*Abwod et al.*, 1996]. Since then, many projects have discussed about the capturing and accessing of classroom and meeting information. However, the usefulness of documents for retrieval and/or browsing of captured information from such environment, have not been fully explored yet. In this thesis, a system is proposed that captures multimodal information during such events and uses documents (presented/discussed) as a way for indexing and retrieving of relevant information.

1.1.2. Usefulness of documents in meeting/lecture

In a typical meeting/lecture environment, documents are either visible and/or discussed. Such documents are available electronically, excerpt of electronic document or the document itself projected on a screen (slides, transparencies) and paper copies laid on tables. Furthermore, referred documents often appear in the speech of participants. Visible documents, such as projected and printed documents are often used in a presentation. In case of presented documents, presenters keep the key points in the projected documents and try to explain his/her ideas and thoughts by delivering a talk. Therefore, the presenters talk and discussion are mainly focused around the presented documents.

Though the projected documents do not contain all the information about what a presenter delivers, it gives an abstract description of his/her presentation. In this context, a projected document is described as a *virtual container*. Although it does not contain all information, it works as a gateway to access all other information (*e.g.* audios, videos, whiteboard data, etc.) captured during the presentation. Therefore, we define a document as a *medium* for communication, a *container* for information, or as a symbolic representation of an abstract description of ideas and expressions.

Projected documents appear at a specific meeting/lecture time and remain in visual focus for some time. Therefore, in this thesis this time information is used for structuring the meeting/lectures records temporally, the presented documents can thus be used as the storyboard of the captured meeting/lecture recordings.

1.1.3. Type of document alignments

Documents in lecture/meeting environment are often captured as images using various handheld devices. The same document can be presented in their electronic form such as PowerPoint, MS word of *Microsoft*®, OpenOffice of *Sun Microsystems*® or PDF form of *Adobe*®, printed on paper or transparencies, etc. These various forms of the same document create difficulties for aligning it with other captured media stream. In this case the alignment of documents can be described in two ways:

- *Image-based*: the captured document image (scanned or captured using handheld devices) can be matched with the original electronic documents and/or extracted key-frames from the captured video stream, so that the static electronic documents as well as the captured images can be linked with other temporal media such as audio/video streams.
- *Content-based*: the textual content of the electronic documents are matched with the speech transcriptions of presenters as well as the recognized textual content from the whiteboard to create semantic links among spoken, electronic and written texts. This is due to the fact that both the speech and whiteboard stream are temporal and allows adding temporality to electronic documents.

1.2. Aim of this thesis

The processing of captured information from a multimodal environment is often focused on audio/video analysis, speech and gesture recognition, person identification, etc. Such analysis allows structuring the raw captured information for efficient access. Considering the usefulness of documents in a multimodal environment (meetings, lectures, seminars, etc.), the captured documents using handheld devices should be aligned with other multimedia information, which are captured during meeting recordings. The problem of integrating/aligning documents with other multimedia information is a difficult task since it requires efficient identification of low-resolution documents. Moreover, the captured images exhibit perspective distortions that further hinder the document analysis and recognition. Current document analysis systems use high resolution, comparatively less noisy scanned document images and basically process the black-and-white images.

Due to the rapid growth of camera-based mobile devices and archival of color documents, the existing analysis and recognition systems should be able to process the visual information captured by such devices. Therefore, we propose in this thesis, an efficient signature-based approach for identification of low-resolution documents as an additional modality to bridge the gap between temporal data (*e.g.* audio/video, pen-stroke in whiteboard, etc.) and static information. Furthermore, documents provide natural interfaces for browsing multimedia recordings efficiently because (a) users are more familiar with documents, (b) they can be easily indexed and therefore, searched and (c) they can be easily used in web-based applications. The motivation of this thesis also includes the use of documents for browsing and retrieval of other related multimedia information.

1.3. Possible target applications

The investigation of document analysis, recognition, retrieval and its integration with other media is motivated by cutting edge applications in multimedia. Today, more and more documents along with audio and visual information are captured, archived, delivered and managed in digital forms. The wide usage of documents along with digital media files provokes many new challenges in multimodal mobile information acquisition and large multimedia database management. Some of the most prominent areas are:

- *Automatic meeting/lecture segmentation and indexing*: creates a structured, searchable view of archives of the multimodal meeting/lecture content.

- *Library digitization*: digitizes journals, magazines, newspapers and various videos using advanced imaging for digital libraries.
- *Mobile translation*: extracts and translates textual content as well as visual signs for tourist usages. For example, English documents could be translated to French, German or Asian languages using a handheld translator. For the visually impaired, the content of the documents could be translated to speech and vice-versa for the hearing impaired.
- *Automatic indexing of broadcasts*: indexes broadcast contents for the content-based retrieval.
- *Digital media asset management*: archives digital media files for efficient media management.
- *Document cataloging*: document database catalogs on the basis of content relevance.

The fundamental techniques that address the above-mentioned challenges are content-based multimedia annotations and retrieval. Most of the applications rely on offline processing due to the fact that the computational cost for the image and video processing is very high. The annotation should be structured in order to speed up real-time processing using computers. One of the important aspects of annotation is that the process should be automated and the content relevant, which would make browsing and retrieval comfortable for users. In order to obtain this, content-based indexing and retrieval tasks require the extraction of descriptive features that are relevant to the subject materials (*e.g.* video, audio, documents, etc.).

1.4. Contribution of this thesis

The focus of the thesis is to develop an efficient application system that captures, analyzes and then automatically indexes the captured multimedia information from a multimodal environment considering document as an additional modality. The following are the major contributions of this dissertation in brief:

- First, a capture system has been developed for capturing and archiving multimedia information from a multimodal environment for meeting/lecturer recordings. The architecture of the proposed system is light weight, distributed and scalable. Furthermore, the unique property of the system is that it can handle any type of capture devices used in the multimodal environment in a synchronous manner and there is no necessity of presenter/operator interaction during recording.

- Second, one of the major goals of this thesis is the automatic indexing of the captured multimedia information in order to perform efficient retrieval. The indexing was successfully implemented using the document-based segmentation of the captured audio-visual streams. The proposed segmentation technique is based on stability of the video sequence rather than change. It uses a feature-based algorithm, which out-performs the state-of-the-art techniques.
- Third, the major problem encountered at the indexing step of the proposed scheme, is the identification of the captured low-resolution document images from their original electronic documents in a repository. Therefore, the thesis focuses on structured signature-based document image matching in order to overcome the above-mentioned problem. The proposed *Visual Signature*-based identification method of low-resolution documents is one of the core findings in this thesis.
- Fourth, another finding of the indexing step is the automatic addition of the textual attribute to the multimedia segments. The attributes are assigned with keywords from the documents that are associated with the corresponding segments by natural language processing of the textual content of those documents.
- Finally, the content-based retrieval methodologies were established, which can be done by image- and/or keyword-based queries to the system.

Apart from these task-oriented specific contributions, the complete system consisting capturing, indexing and retrieving has been adopted in *Smart Multimedia Archive for Conferences* (*SMAC*) [*URL1*], in collaboration with *European Center for Nuclear Research* (*CERN*) and Ecole d'Ingénieurs et d'Architectes de Fribourg (EIA-FR). The same system has been implemented in the document-centric *Smart Meeting Room* of the *University of Fribourg*.

1.5. Thesis organization

The thesis is organized as follows:

- Chapter 2 discusses the background work related to meeting/lecture capturing and indexing problems. This includes the description of techniques that are most commonly used for indexing of captured audio-visual streams and their accessibility to users. It also explains briefly document analysis and recognition techniques that are required for the integration of documents with other media.

- Chapter 3 describes our proposed *DocMIR* system, which captures, analyzes and automatically indexes the captured multimedia stream from a multimodal environment. This is very important since much of the work proposed in this thesis is motivated by the limitations of existing systems.
- Chapter 4 first addresses the state-of-the-art techniques for the meeting/lecture temporal segmentation and their advantages and disadvantages. In this chapter, a novel segmentation technique is proposed that uses documents as pivotal aspect and extracts the time boundaries of each visible document.
- Chapter 5 explains the identification of low-resolution document images captured using handheld mobile devices. It also presents the state-of-the-art techniques for the content-based document identification and retrieval. The chapter also depicts the perspective correction and noise removal in the captured document images before extracting features for the document *Visual Signature*.
- Chapter 6 describes the extraction of features for the *Layout Signature*, which is a part of the *Visual Signature*. The chapter also presents the structuring and matching of the features in the *Layout Signature*.
- Chapter 7 describes the extraction of color features for the *Color Signature* of the *Visual Signature*. It also explains the matching of features in the *Color Signature* for document identification.
- In Chapter 8, all the evaluations of the above-mentioned matching are presented by considering various fusion and sequential strategies of the features in the *Visual Signature* as well as a comparison with the existing methods.
- The integration of these discrete tasks into practical applications including meeting/lecture segmentation and real-time document identification/retrieval is also explored in this thesis. Chapter 9 discusses various issues and performances associated with these applications.

Chapter 2

Related works and background

Multimodal meeting recordings produce huge amounts of data from various sources and contain several hours of information. The captured data do not have any structural information. As more and more of such recordings are archived, the necessity for automatic addition of structural information for later retrieval has become increasingly apparent. Various methods have been employed to structure the recorded multimodal information. A popular approach is to consider the multimedia streams, analyze the unstructured data and create semantic pointers based on certain features or pattern found in the streams.

In this chapter, we discuss some of the related and relevant works carried out in capture, analysis, indexing and access of captured multimedia meetings/lectures. Section 2.1 describes the existing approaches for analysis and retrieval of captured meetings data based on domain specific indexing procedure for targeted application. The background information for recognition and retrieval of documents often used in the multimodal environment is presented in Section 2.2.

2.1. Previous works

This section presents a literature review of most of the significant works that addressed the issues related to meeting/lecture analysis and retrieval.

2.1.1. Meeting/lectures analysis and retrieval

Several research groups have studied the problem of analyzing and indexing of the multimodal information captured during meetings/lectures (*e.g.* video, voice, whiteboard, etc.), which would allow searching and browsing at a later time point once the recording is completed. Analysis of the recorded multimodal meetings/lectures materials takes place either during or after recording. The goal of the analysis of the captured multimedia streams is to add the structural information to the captured unstructured data. The structural information provides indices to access the captured multimodal information in random manner. The robustness of user access to the captured data is highly dependent on the granularity and meaningfulness of the indexing during the analysis. Most of the multimodal meeting/lecture capturing, indexing, annotation and retrieval systems can be conceptually described by the framework illustrated in Fig. 2.1. The existing research projects, which capture and analyze meetings/lectures, can be classified into *application-specific* groups and are based on the target areas. Furthermore, the analysis and indexing approach of each application is carried out using the captured media streams, which could be further categorized as *sensor-specific* indexing.

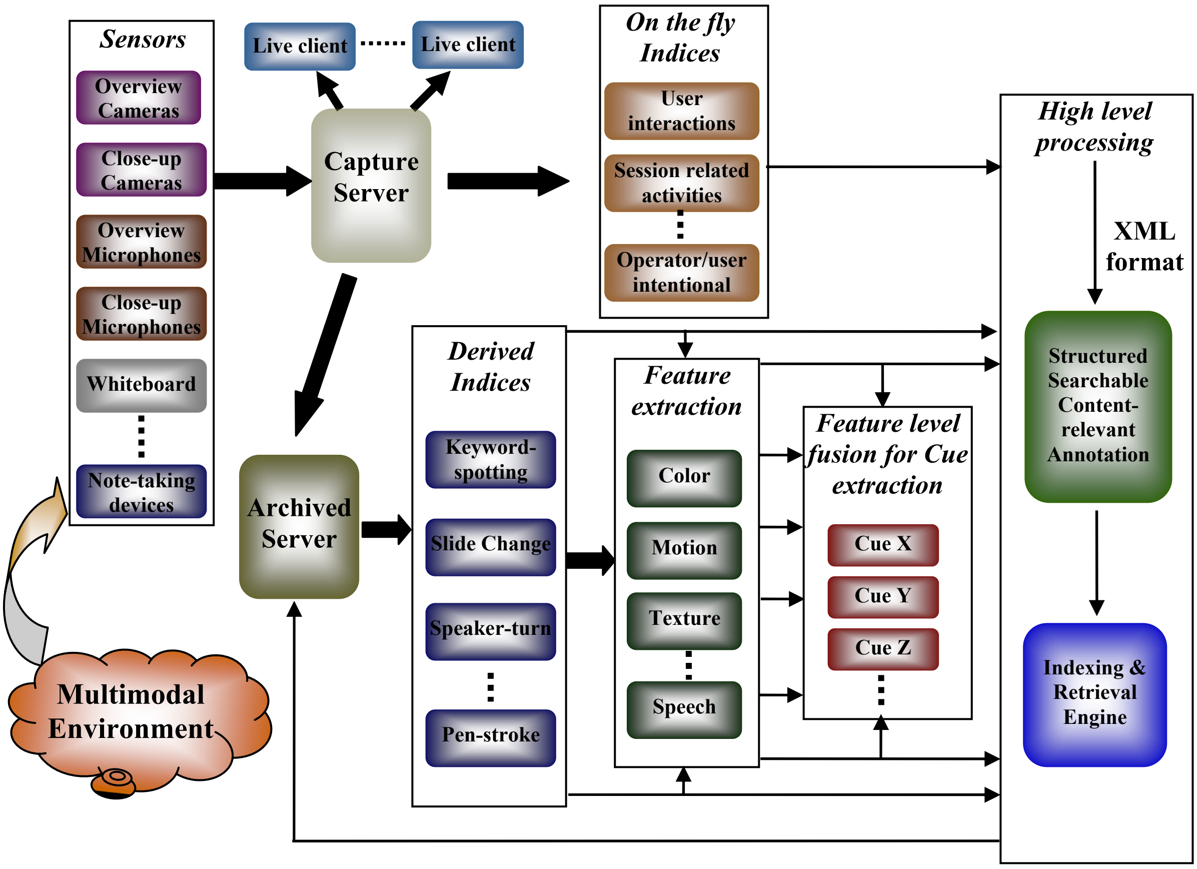


Fig. 2.1. A simple model of multimodal meeting/lecture capturing, indexing and annotation system.

2.1.2. *Application-specific classification*

The existing research projects for recording and analyzing meetings/lectures can be classified according to their target applications and can be categorized as (1) academic and (2) industrial.

2.1.2.1. Academic applications

Academic applications are termed as e-learning and are generally made for classroom use for recording and broadcasting lectures. Some of these projects include *LectureBrowser* of University of Berkeley [*URL2*], *LectureLounge* of Fraunhofer-IPSI [*URL3*], *eClass* of Georgia Tech [*Brotherton*, 2001; *URL4*], *EmuLib* of University of Mannheim Germany

[*URL5*], *CLIC* of Columbia University [*URL6*], *Digital Lecture Hall* (DLH) of Technical University of Darmstadt [*URL7*] and *Class Room Presenter* of University of Washington [*URL8*].

In such applications, the major data captured are the audio-visual streams of the instructor, projected slides and white/black board. The classroom is more instructor-dominated and time-restricted. Therefore, all the presented materials and the lecture of instructor are captured and kept in the repository, subject- and lecture-wise. Upon demand, students could then go through the captured audio-visual streams along with the presentation slides to refresh their memory on a particular lecture.

2.1.2.2. Industrial applications

These comprise of recording and broadcasting of seminars, scientific meetings, group discussions, etc. for employees of organizations. In such applications, all participants are actively involved in the discussion and are targeted to the meeting room activities, which are used to discuss the status, explore ideas, present information, making decision, etc. The existing research projects includes Microsoft's Distributed Meeting Recorder [*Cutler et al.*, 2002; *Rui et al.*, 2004], IBM's eSeminar [*Steinmetz* and *Kienzle*, 2001], FX PAL's Meeting Recorder [*Chiu et al.*, 2000a], Ricoh's Portable Meeting Recorder [*Lee et al.*, 2002], Meeting Room of Carnegie Mellon University [*URL9*], Foveal Systems AutoAuditorium [*Bianchi*, 1998; *URL10*], Meeting Recorder Project of International Computer Science Institute (ICSI), Berkeley [*Janin et al.*, 2004; *URL11*] and the Smart Meeting Project of IDIAP Martigny, Switzerland [*URL12*]. In most of the research projects above, the audio-visual stream of each participant along with the presented documents and white/black boards data are captured for meeting summarization. The indexing of the captured multimodal information uses the data captured from one or more sensors as described below.

2.1.3. Classification based on sensor-specific indexing methods

An index is a time-point, which identifies a point in the recorded multimodal meeting recording. These indices provide random-access points to the captured information, so that one can jump from one point to another without sequential scanning of the captured information. In other words, these indices add structural information to the captured unstructured multimodal information. *Minneman et al.* classify the methods for creating indices into four broad classes: *intentional annotations*, *side-effect indices*, *derived indices* and *post-hoc indices* [*Minneman et al.*, 1995]. They define *intentional indices* as indices that participants create during an activity for the purpose of marking particular time-points or segments of activity, such as sequential note-taking where the notes are time-stamped. *Side-effect indices* are indices that are created through activities whose primary purpose is not indexing. However, they provide indices because they are automatically time-stamped and logged, such as switching pages in a presentation. *Derived indices* are generated by automated signal analysis of detailed time-stream records (audio/video), such as speaker identification or scene change detection. Finally, *post-hoc indices* are produced by anyone who later accesses the activity records and is an intentional annotation. *Geyer et al.* categorize the existing indexing approaches into two major classes: *online* and *offline*. They further categorize indices of each class as *explicit* versus *derived* [*Geyer et al.*, 2005]. They defined *explicit online indices* as pointers into the meeting record that are created on the fly during the meeting. Such indices are created while users interact with the system and are based on session-related activities, user intention or interaction with artifacts. *Explicit offline indices* are created offline through the access activities of users. These indices are more like note-taking and book-marking during the activity. *Derived online indices* are generated from the captured media stream in real-time during the meeting. Some of these indices are computed on audio streams such as pause detection, speaker identification, speech recognition, etc. and comparatively consume less processing power than the video streams. *Derived offline indices* are traditionally created offline after a meeting. This is due to the high requirements of computing power for signal analysis of media streams.

In the following section, the creation of indices to the captured multimedia information is carried out by considering the data from one or more sensors used in the capture environment. The classification is based on the sensors as well as the media-specific data used for indexing. It is found that the captured signals from the following devices are commonly used for indexing.

2.1.3.1. Whiteboard-based indexing

Pen-strokes on a whiteboard are often used for indexing to access the recorded meeting/lectures. Various types of electronic whiteboards are available in market which allows capturing the content as well as the pen-strokes. Some of these are Xerox LiveWorks LiveBoard [*Elrod et al.*, 1992], SMART Board [*URL13*], interactive whiteboard of PolyVision [*URL14*], etc. *Brotherton et al.* used the pen-stroke- and world-level granularity indices, which are extracted from the captured whiteboard data for their classroom project [*Brotherton et al.*, 1998]. *He et al.* used both pen-strokes and a set of key-frames representing the written content of the whiteboard before each erasure as indices for the meeting records [*He et al.*, 2003]. *Wolf et al.* analyzed the occurrences of pen-strokes. They try to detect hot spots in a meeting and are based on the turn-taking behavior and the frequency of pen-strokes [*Wolf et al.*, 1992b]. *Pedersen et al.* described a meeting capture system called *Tivoli*, which simulates whiteboard functionality on the LiveBoard [*Pedersen et al.*, 1993]. It provides basic pen-based scribbling and editing with pen-based gesturing, wiping and indexing techniques. *Geyer et al.* explained a collaborative workspace system called *TeamSpace* [*Geyer et al.*, 2001]. The system shared a whiteboard that allows importing and annotating *PowerPoint*, *Word* and *PDF* documents during a meeting and create indices for page flips as well as for all the textual and ink annotations. Finally, for whiteboard-based indexing, handwriting analysis could be used to turn ink into text, thus making it searchable using keywords [*Liwicki* and *Bunke*, 2005].

2.1.3.2. Camera-based indexing

The indices are generated by considering the video signal captured using the camera in a capture environment. Video analysis techniques could be used to determine slide and scene detection, person detection and tracking (enter and exist), key-frame detection and extraction to create indices. Furthermore, it could be used to detect emotions, hand-raising and applauding capabilities, thus indicating where questions were asked and where decisions were made. The *Cornell Lecture Browser* creates the indices from the captured video containing projected slides by detecting the flipping of slides [*Mukhopadhyay* and *Smith*, 1999]. Microsoft's distributed meeting system uses video signal for person detection and tracking [*Cutler et al.*, 2002]. eSeminar creates video snapshots and key-frames as indices based on slide and scene changes [*Steinmetz* and *Kienzle*, 2001]. *Girgensohn et al.* produces indices using key-frames, which are extracted by considering features such as shot boundaries, slides and close-ups of human faces [*Girgensohn et al.*, 1999]. Similarly, *Portable Meeting Recorder* uses key-frames as indices, which are extracted by analyzing the activities of participants and recognizing the meeting locations [*Lee et al.*, 2002].

2.1.3.3. Microphone-based indexing

There are many different kinds of analysis that could be done with audio streams that allow extracting indices for the captured multimodal information. Such analysis include pause detection, turn detection, speaker identification, speech recognition and keyword spotting. For example, *Hindus* and *Schmandt* use pause-detection and audio analysis to detect turn-taking in a phone conversation, and thus allow indexing based on the speaker [*Hindus* and *Schmandt*, 1992]. *Multimedia Meeting Tracker* does speaker identification, speech recognition, face and action item recognition to create indices [*Bett et al.*, 2000]. *Distributed Meeting* system uses the speaker segmentation and clustering technique to generate meaningful indices [*Cutler et al.*, 2002]. *eClass* converts audio into a sequence of phonemes, which is used as indices [*Brotherton et al.*, 1998]. The search is done by translating the keywords or user phrases into their phoneme equivalent and then matching against the phoneme stream. IDIAP Smart Meeting Room detects the speaker's turn and uses speaker clustering and segmentation to extract the indices to the meeting records [*Ajmera et al.*, 2004].

2.1.3.4. Special device-based indexing

Often, portable devices are interactively used in the capture environment to create indices and add notes. Such devices are used by either an operator and/or presenters, participants and interactively communicate with the capture systems. In *Classroom 2000*, note-taking is done with PDA (Personal Digital Assistant) device called *ClassPad*, which is pre-loaded with slides. *ClassPad* preserves all annotations made to a series of prepared slides and creates a time-stamped log, when the user navigates between slides and annotated with a pen. These notes are later synchronized to the audio and the slides, which have been annotated by the instructor for later access [*Brotherton et al.*, 1998]. Similarly, *Filochat* and *Notelook* allow users to take hand-written notes that can be used to index the audio or video recordings of a meeting [*Whittaker et al.*, 1994; *Chiu et al.*, 1999]. *Notelook* also allows users to grab pages from the presentation to incorporate into those notes. *LiteMinutes* provides support for typing notes or minutes on a laptop, where each line of text acts as an index [*Chiu et al.*, 2001]. The *Coral* system creates indices based on switching slides and the handwritten and textual notes taken on a laptop computer and a whiteboard [*Minneman et al.*, 1995]. The *Audio Notebook* structures the audio recording using techniques such as *user structuring* based on note-taking activity, and *acoustic structuring* based on changes in pitch, pausing, and energy [*Stifelman et al.*, 2001]. Similarly, there exist other pen-based devices such as *Dynomite*, *Marquee*, *We-Met*, which enables users to correlate their personal notes and keywords with audio and/or video of the recordings in order to create indices [*Wilcox et al.*, 1997; *Weber* and *Poon*, 1994; *Wolf* and *Rhyne*, 1992a].

2.1.4. Access and replaying

The multimedia information of captured meetings/lectures is stored in a repository for later access and replaying. Most of the work in this area has focused on developing capture systems rather than exploring the requirements of users while browsing the captured information. However, the design of capture systems depends very much on understanding the requirements of potential users for accessing the captured information. The capture systems mainly focus on the information and indices to capture for later reviewing. For efficient access to the captured information, one should consider a user-friendly interface for replaying. It is found that mainly two types of interface are commonly used to access the captured records: (1) web-based browser and (2) customized user interface.

2.1.4.1. Web-based Browser

The web browser is used for viewing the multiple media streams of the captured multimedia information. The web-based browsing basically shows the listing of captured meetings/lectures. Users commonly use the top-down approach to access the specific part of the captured multimedia records [*Brotherton et al.*, 1998; *Mukhopadhyay* and *Smith*, 1999; *Steinmetz* and *Kienzle*, 2001; *Geyer et al.*, 2001; *URL12*]. The access points are often presented with place, date and time, image of speaker/participants, topics presented/discussed, key-frames, slides, timelines consisting speaker change, slide change, etc.

2.1.4.2. Customized User Interface

The recorded multimedia data and related metadata are presented to the user through a customized user interface that displays the metadata, the transcript, as well as views of audio and video activities in a meeting record. The design of interface considers various types of captured data and extracted metadata [*Lee et al.*, 2002; *Chiu et al.*, 2000a; *Bett et al.*, 2000; *Cutler et al.*, 2002; *Girgensohn et al.*, 1999]. The extracted metadata provides pointers to the multimedia recordings and the browsing is similar to web-browser with different appearance. The interface often provides the functionalities of the audio/video playback like the VCR metaphor. Some of the projects often use existing players such as Real One player for playing meeting/lecture recordings by using SMIL [*Hunter* and *Little*, 2001; *Mukhopadhyay* and *Smith*, 1999; *Wolf et al.*, 2004].

2.1.5. Summary

The existing approaches towards capture and access of meetings/lectures have been revisited. Different indexing approaches, which are based on the captured data from various sensors, are implied by considering the target applications. The final goal of such application is to provide effective and meaningful presentation of the captured data according to the user requirements. In the above-mentioned applications, the common way of browsing or accessing the captured multimedia streams is incorporated through semantic pointers, which are either explicitly captured during the recordings or implicitly derived during post-processing. Such pointers such as keywords, key-frames, slides, date and time, speaker turn, etc. help in browsing. However, they provide little or no help for the content-based meeting/lecture retrieval from a huge multimedia database. Furthermore, the presented/discussed documents during a meeting/lecture and their contents are highly correlated with the other captured multimedia information and often appeared as the center of discussion/presentation. The presented documents in such scenarios are often used as semantic pointers. Nevertheless, the analysis of such documents and its integration with other media has not been fully explored yet. The content-based analysis and alignment would reduce the distance between temporal media and static information that would provide a better retrieval, as well as browsing/accessing model for multimedia information. In this thesis, the usefulness of documents for meeting/lecture retrieval and/or browsing is presented. In the near future, the applications should provide the user-specific requirements *i.e. meeting/lecture-on-demand* without considering the sequential browsing of multimedia documents.

2.2. Background information

In this section, some of the state-of-the-art techniques for image and document processing and their identification are briefly explained. The thesis addresses the problem of indexing and retrieval of meetings/lectures recordings by considering the documents that are often captured using low-resolution handheld devices. Therefore, two approaches namely (a) *Content-Based Image Indexing and Retrieval* (*CBIIR*) as well as (b) document recognition and retrieval are briefly discussed.

2.2.1. Content-based Image Indexing and Retrieval

CBIIR is a technique for indexing and retrieving images on the basis of automatically-derived primitive features, which characterize image content such as color, texture, shape, spatial layout. The term *Content-Based Image Retrieval* (*CBIR*) in the literature was used by *Kato* to describe his experiments on automatic retrieval of images from a database by color and shape features [*Kato*, 1992]. *CBIR* differs from classical information retrieval in the sense that image repositories are normally unstructured, since digital images consist of arrays of pixel intensities, with no inherent meaning [*Eakins* and *Graham*, 1999]. Image databases thus differ fundamentally from text databases, where the raw material (words stored as ASCII character strings) has already been logically structured by the author [*Santini* and *Jain*, 1997]. Most of the content-based image indexing and retrieval systems can be conceptually described by the framework depicted in Fig. 2.2. The user interface (UI) of such systems generally consists of a query formulation part and a result presentation part. The retrieval of images from a database can be done in many ways. One is to browse through the database one by one. Another way is to specify the image in terms of keywords, or in terms of image features that are extracted. Another method is to provide an image or sketch from which features of the same type must be extracted as for the database images, in order to match these features. More detailed survey on image/video indexing and retrieval can be found in various literatures [*Aigrain et al.*, 1996; *Eakins* and *Graham*, 1999; *Petkovic*, 2000; *Remco* and *Tanase*, 2000]. In the case of keyword-based indexing, it has a high expressive power and can be used to describe almost any aspect of image content. However, it is a manual process and suffers from significant drawback of labor-intensive. If the procedure is automated, the efficiency of retrieval could perform better than the primitive feature-based indexing.

Videos are represented with sequence of images. Therefore, *CBIR* techniques are often adopted for video retrieval. The first step is to segment the video into individual shots and then a single key-frame called representative frame is selected by considering the visual content of image sequences for each shot. The complete set of key-frames for the video thus forms a storyboard, which can then be annotated and stored in an image database for browsing or content-based retrieval. The state-of-the-art of shot detection is described details in Chapter 4.

One of the important aspects addressed in this thesis is the identification and retrieval of captured low-resolution document images. For this purpose, color and spatial layout as two important primitive features are used in the so-called document signature, which represent document images. Spatial layout is represented as the relative position of color information in document image and is presented in Chapter 7. Furthermore, in this thesis the above-mentioned *CBIR* technique is used for the content-based retrieval of lectures/meetings recordings based on the captured visible documents.

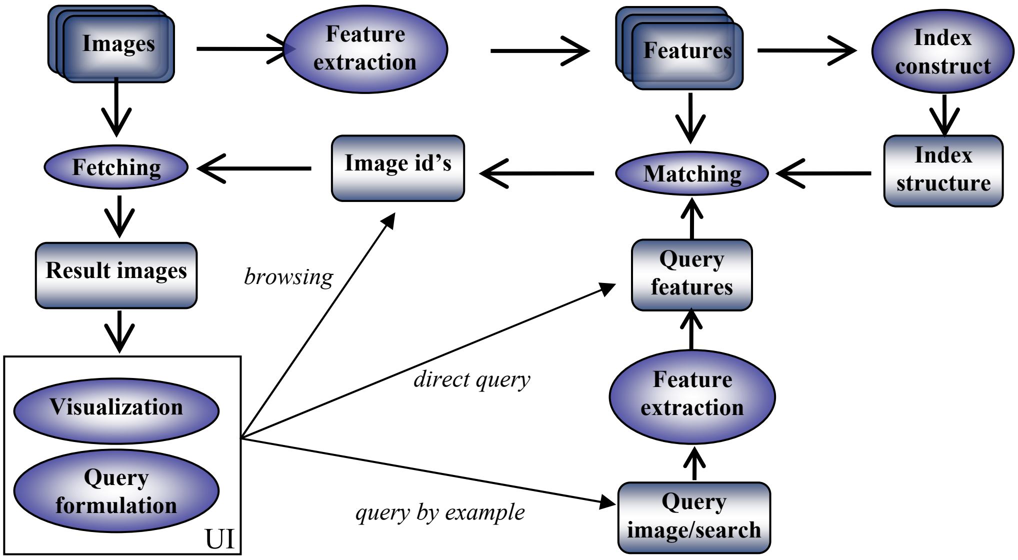


Fig. 2.2. Content-based image retrieval framework [*Remco* and *Tanase*, 2000].

2.2.2. Document recognition and retrieval

Patents on *OCR* were filed in the 19_{th} century and working models are demonstrated by 1916. In the 1950s, *OCR* was first used in business applications [*Nagy*, 2000]. Since then, there has been an extensive research on converting paper-based documents into electronic documents. Electronic documents have many advantages over paper documents, including compact and lossless storage, easy maintenance, re-usability, modifications, efficient retrieval and fast transmission. One of the major advantages of the digitization of paper documents is that it allows efficient indexing and retrieval of the information contained in documents.

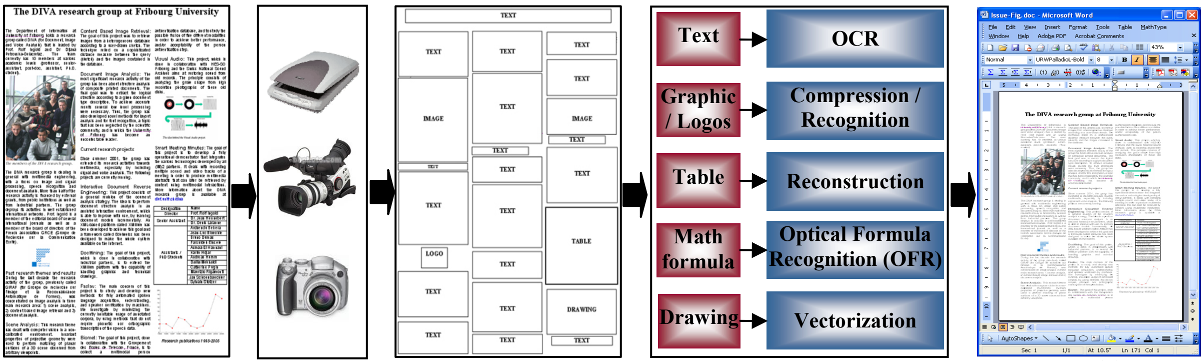
2.2.2.1. Document physical and logical structure

For the production of electronic documents from their corresponding paper documents, two major modules are essential without any knowledge of specific format:

- *Document structure*: it consists of physical and logical structures. The physical structure of a document is how the document content is laid out on the physical medium. It represents the hierarchy of physical components such as pages, columns, paragraphs, text lines, words, tables, figures, halftones, etc (Fig. 2.3c). The logical structure of a document's content is how the content is organized prior to the enforcement of a particular physical structure. It attaches logical meanings to the physical structure and determines the reading order of physical components of documents such as titles, authors, affiliations, abstracts, sections, etc [*Bunke* and *Wang*, 1997; *O´Gorman* and *Kasturi*, 1995].

- *Document understanding*: this uses *OCR*, graphics and table recognition systems that are applied to the structural component of the document image (Fig. 2.3d).

For document recognition and retrieval systems, physical and logical structure analysis of document images is crucial. Some of the existing approaches are presented in Chapter 5.


The DIVA research group at Fribourg University
TEXT
IMAGE
LOGO
TABLE
DRAWING
Text
OCR
Graphic / Logos
Compression / Recognition
Table
Reconstruction
Math formula
Optical Formula Recognition (OFR)
Drawing
Vectorization

(a) (b) (c) (d) (e)

Fig.2.3. Digitization of physical documents: (a) physical document, (b) acquisition, (c) document structure analysis, (d) processing of each region and (e) document modeling and understanding.

2.2.2.2. Information/content extraction and indexing

The content of documents should be characterized in a meaningful way for later retrieval and the process is called indexing. Once the documents are indexed, the resulting index vectors are used for retrieval by computing the distance between query and the document vector. Digitized documents contain both structured and unstructured components. The indexing approaches based on structural information without textual content consider the physical and logical structure of the documents [*Doermann*, 1998]. Such approaches rely on segmentation and often consider texture-based features. The unstructured components represent the content of documents such as text, images, logo, drawings, etc. Normally, textual content is extracted using *OCR*. For low-quality document images where the performance of *OCR* is poor, the indexing approaches are often based on *keyword spotting*, *characterizing text* and *automatic abstracting of images* [*Doermann*, 1998]. Textual content-based indexing and retrieval is presented in Chapter 3 that considers keywords. The other unstructured information used for document indexing are graphics such as drawings and maps [*Amlani* and *Kasturi*, 1988;

Syeda-Mahmood, 1995; *Gudivada* and *Raghavan*, 1993] as well as logos [*Doermann et al.*, 1996; *Suda et al.*, 1997].

2.2.2.3. Document image matching

Today's documents are clearly multimodal. Therefore, the identification of various instances of the same documents in different media is an important task. One of the approaches for identifying same documents in various graphics format is termed as document image matching. The document image matching algorithm is useful for the applications where the objective is to identify visually similar or finding the instances of a given document from a document repository. In the case of image matching, a feature-based similarity detection algorithm locates a group of N documents that are visually similar to a given image. The document image matching method is first addressed by *Hull* to solve the problem of content-based matching [*Hull*, 1994; *Hull* and *Cullen*, 1997]. It describes a method for matching documents which have the same character content but which may have been reformatted or distorted prior to re-imaging. Similarly, a signature-based matching approach is proposed by *Doermann et al.* for detecting duplicate documents in very large image databases using features, which are extracted from the image to form the document signature [*Doermann et al.*, 1997].

2.2.2.4. Summary

The *CBIR* is a famous ongoing research topic. The technique is currently used in applications such as image/video retrieval as well as video-on-demand. Some of the existing approaches and their usefulness for the image retrieval are briefly presented. The same content-based analysis is applied to document images with some additional information of layout structure. The document analysis approaches are mainly focused on the digitization of the paper-documents that helps the *Information Retrieval* research community.

2.3. Conclusion

In this chapter, the existing approaches for indexing and retrieving captured multimedia meeting/lecture recordings are revisited. The existing approaches are well-suited for browsing captured multimedia meetings rather than content-based retrieval. The goal of this thesis is to provide content-based meetings/lectures retrieval in addition to the existing browsing capability. The basic idea about the *CBIR* as well as its application in *document recognition and retrieval* is presented. These two approaches are used in this thesis for the detection and identification of documents captured during meetings/lectures. The real challenge of identification concerns the poor resolution of the captured noisy documents and is presented in Chapter 5. In the following chapter, a document-centric meeting indexing and retrieval system is presented that also handles and identifies noisy documents captured during meetings/lectures.

Chapter 3

Automatic document-based meeting indexing

In applications such as meeting or conference recordings, documents are available either in their classical printable form (physical) or in electronic form. The interaction styles between these two different forms do not resemble each other. However, the relationship of documents (physical or electronic) with other temporal media in a multimodal environment remains the same. For instance, projected documents are in fact very frequently used in oral presentations and appear either in electronic form such as slides, images, figures, etc. or in physical forms such as transparencies. These printed electronic documents are often provided to participants during the presentations. These projected documents contain the main messages the presenters would like to convey, as well as the basic facts and serve as a *virtual container* of information. Therefore, in our opinion those documents that are in visual focus during the presentation could be a better estimation of the whole presentation context. Furthermore, these documents appear at a specific time, stays for some time and therefore, they hold a temporal relationship with the other temporal media such as audio, video, etc. recorded at the same time.

In this chapter, the role of a document in a multimodal environment is described. The organization of this chapter follows the document-based automatic indexing of the multimodal meetings by integrating non-temporal documents with temporal media captured during the meeting recordings. The chapter starts with a document-based browsing interface that supports non-linear navigation to temporal data and is described in Section 3.1. Most often, the documents that appear in the meetings are captured as image or graphics and lose their electronic functions such as edit, copy, paste, etc. However, the original electronic form of these documents is preserved in a repository and is available to the user. Therefore, the temporal linking of these electronic documents is explained in Section 3.2. The major part of this chapter is focused on the proposed document centric *DocMIR* system, which is described in Section 3.3. Finally, the chapter concludes with future perspectives in Section 3.4.

3.1. Non-linear navigation aid

Multimedia recordings of meetings contain multiple streams of several hours of recordings. This captured information is archived in the form of audio, video, documents, notes, etc.

There are at least two kinds of users for accessing this captured information: attendees and others. For the former group of users, it is very difficult to retain detailed happenings during the meetings in their memory. For the latter group, a user would like to get an overview of the meeting without watching inexhaustively several hours of meetings starting from the beginning, which is a linear process. However, he/she would wish to watch some specific parts of the recorded meetings at leisure for different interests, which is the so-called non-linear access. In this scenario, the documents that appeared during the meeting recordings are considered as semantic pointers. As mentioned earlier the context of speaker's presentation is focused around the content of the projected documents. Moreover, the visible documents hold a temporal relationship with the meeting times. During a presentation, projected documents such as slides change after a specific time as well as refer to a specific topic, which is the center of discussion during the presentation of that particular slide. Moreover, attendees often use their handheld devices to capture the projected slide of interest that would help them retrieve the specific part of the recorded meetings later, which he/she wish to share or discuss with colleagues, rather than watching the whole recordings. Often, due to the time constraint during meetings, some attendees might not get enough time to discuss about some specific points, which he/she could watch later for clarification. Therefore, finding specific information requested by a user is a difficult task and requires meaningful indexing. The captured audio-visual streams during the meeting recordings are sequential and do not provide any structural information. However, it provides the time-line for linear and non-linear access. Multimedia records of meetings would only be generally useful if the existing tools or technologies that help users avoid replaying much of what has been recorded [*Ginsberg* and *Ahuja*, 1995]. This is exactly what documents allow for quick accessing and browsing meeting recordings. This kind of directed browsing, searching, and visualization of meeting records requires meaningful indices that act as semantic pointers into the meeting record [*Geyer et al.*, 2005].

Therefore, by considering the relevance of the projected documents, it is one of the best meaningful indices to the recorded meetings for non-linear access and the core of our document centric *DocMIR* system (Section 3.3).

3.2. Linking electronic documents with temporal data

Slides projected during meetings are often recorded as either images (snapshots) or video stream [*Mukhopadhyay* and *Smith*, 1999; *Chiu et al.*, 2000a; *Erol et al.*, 2003; *Steinmetz* and *Kienzle*, 2001]. Projected slides are basically in electronic form *i.e.* most of them either

Microsoft$^{®}$ *PowerPoint* (*PPT*) or *Adobe*$^{®}$ *PDF*, which is the current document format pivot. When they are captured as images, only the image is preserved without the characteristics of electronic form such as edit, copy, paste, etc. are lost. Since, the context of the speaker's presentation is focused around the projected documents, the content of the projected documents can be very much helpful for meeting summarization and indexing, which later provides a fast and efficient retrieval mechanism of recorded meetings. As the projected documents are captured as images, the only way to extract the content is by using *OCR* systems, which is time consuming and requires different systems to handle different languages. Moreover, the quality of the captured document images is far below than the scanned images that makes the task of content extraction more difficult. The non-uniform color and textured background of the captured document images also denies the efficient extraction of textual content.

Since the electronic slide documents are already available, these documents would provide the best means of extracting the textual content from the existing available electronic documents rather than using *OCR* systems in the low-resolution captured document images. For that, the available electronic documents must be linked with the captured multimedia meeting recordings. This could be achieved by identifying the captured document images (snapshots or extracted images from video) from the meeting repository containing the original electronic documents. Since the captured document images are already aligned with the meeting minutes, therefore after identification, the original documents would replace the captured documents. Moreover, this not only helps in annotating recorded multimedia meetings and keyword-based access to the archived information but also allows image-based access. Hence, the user could query the system using a set of keywords which is present in the projected documents and/or querying a captured image of projected document or original documents. The identification of the low-resolution captured documents is described in detail in Chapter 5. Once the original electronic document is identified, the content extraction and annotation is straight forward as explained in subsection 3.3.2.

3.3. Overview of document-centric DocMIR system

In this section, the architecture of the proposed *DocMIR* system is described. The system is used for capturing events such as meetings, conferences, lectures, etc. Since such events could extend to durations of more than a few hours, these captured events should be segmented into reasonable smaller units and indexed for efficient retrieval. The captured events are automatically indexed by analyzing the content of the captured audio-visual streams. The *DocMIR* system captures audio-video streams of the ongoing events. The system uses the video stream from the camera that is focused on the projector screen for the segmentation of the captured events. Furthermore, it adds the textual content of the projected documents during the event to the respective meeting segments for search and retrieval.

The key characteristics of the *DocMIR* system are: (a) the possibility of integrating different capture devices and (b) the distributed architecture that allows the handling of various devices without interrupting the task of capturing. Moreover, the system considers the presented documents during the events as a way for automatic indexing. The architecture of the complete *DocMIR* system consists of mainly three modules and is shown in Fig. 3.1. The three modules are: (1) the *capture* module that allows the raw data of meetings to be captured and archived; (2) The captured audio-visual streams are used by the *analysis* and *indexing* module for automatic content-based indexing; (3) the final one is an interactive *retrieval module*, which takes advantage of keywords and/or captured documents from handheld devices to access the archived audio-visual streams of the captured events.

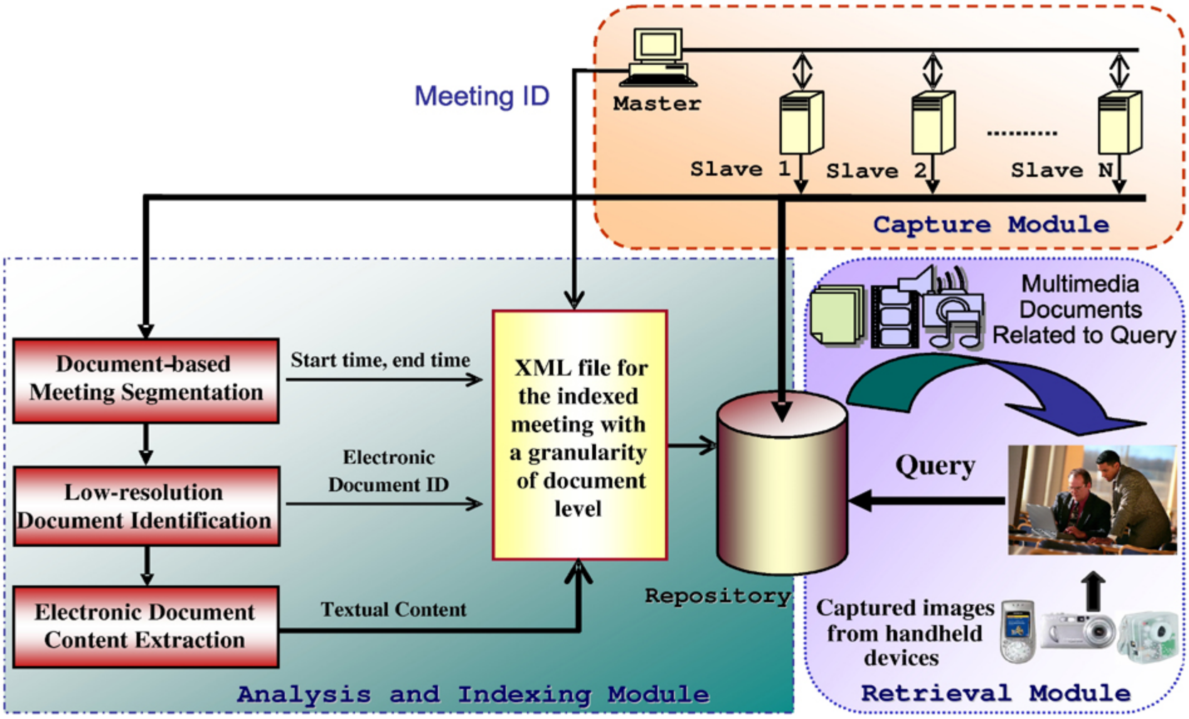

Meeting ID
Master
Slave 1
Slave 2
....
Slave N
Capture Module
Document-based Meeting Segmentation
Low-resolution Document Identification
Electronic Document Content Extraction
Start time, end time
Electronic Document ID
Textual Content
XML file for the indexed meeting with a granularity of document level
Repository
Analysis and Indexing Module
Multimedia Documents Related to Query
Query
Captured images from handheld devices
Retrieval Module

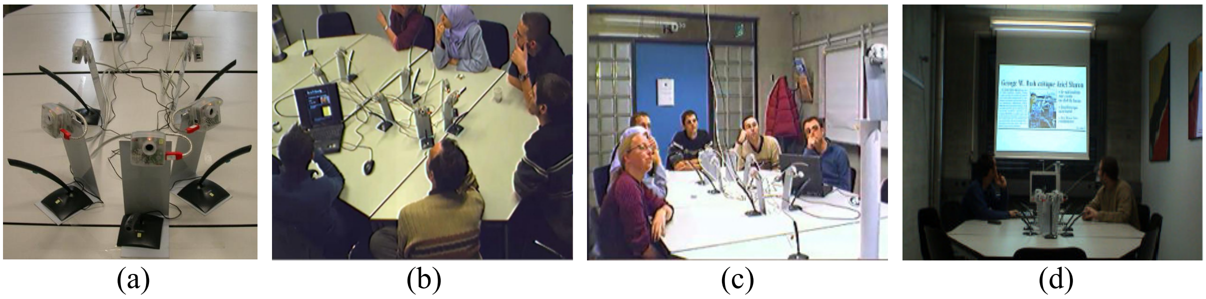
Fig. 3.1. Complete architecture of the *DocMIR* system for automatic indexing and retrieval of recorded meetings using three main modules: (a) capture modules (top bounding box), (b) analysis and indexing module (left, outmost bounding box), and (c) retrieval module (right, outmost bounding box).

3.3.1. Capture module

The capture module synchronously captures the audio-visual raw data from the meeting using various capture devices connected to it. The raw data is then compressed using various compression formats and archived in the meeting repository for streaming purposes. The DivX (www.divx.com) compression is used for the storage since the compression is reasonably high as compared to other compression formats such as MPEG1, MPEG2, etc. without much degradation of visual information. In the capture module, the projected documents (slides) are synchronized automatically with other multiple audio-visual streams without installing any additional software or hardware to the presenter's computer.

3.3.1.1. Capture architecture

Weekly meetings, student presentations and discussions are held in the document-centric *Smart Meeting Room* of the *University of Fribourg*. One camera is focused on the projector's screen to capture the projected documents and four cameras are used to capture the overview of the meeting room. One camera-microphone pair is used per participant to capture the head-and-shoulder video and speech of each participant. Fig. 3.2 illustrates the various capture devices used in the document-centric *Smart Meeting Room*.



(a) (b) (c) (d)

Fig. 3.2. Snapshots of the document-centric *Smart Meeting Room*: (a) FireWire web-cam and Labtec microphone pair per participant, (b) overview from a web-cam capturing meeting, (c) another overview web-cam positioned diagonally opposite and (d) a web-cam focused on the projector's screen.

The capture architecture is simple, distributed, scalable 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 used 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 synchronized through the master (Fig. 3.1). 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. At the end of the meeting, the raw audio/video streams are 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’. Moreover, post-processing, compression, file transfer and creation of Synchronized Multimedia Integration Language (SMIL) presentation per meeting are all automated and controllable through the above-mentioned interface [*Lalanne et al.*, 2004; *URL15*]. Fig. 3.3 shows RealPlayer playing synchronously one of the student 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. The capture architecture is simple and has following characteristics:

3.3.1.2. Light-weight, distributed and scalable

Undoubtedly, a rich amount of multimedia data is captured during the meeting recordings and is dependent on the number of capture sources *i.e.* sensors as well as the duration of capture. In this system, the data are mostly captured using camera and microphones. The number of capture devices depends on the number of participants in the meeting as well as the context of recording. For example, for the recording of lectures and conferences, one pair of camera microphone for the presenter, one camera to capture the projected documents and 2-3 overview camera/microphone pairs are enough for the audience. According to the size of conference/lecture room and number of attendees, the number of overview cameras could be decided. However, in case of meetings and group discussions, where the number of participants is limited and participants present information to others and discuss with each other, each participant’s action should be closely captured as well as the overview of the meetings for later analysis and retrieval. Therefore, the number of capture devices in such scenario is dependent on the number of participants. To overcome the capturing of huge amount of data from different devices, the capture devices are distributed in several capture boxes (PC). It has been mentioned earlier that the architecture follows a *master-slave* model.

The number of slave capture boxes could be increased according to the requirement. Furthermore, the number of devices connected to each capture box is limited to its bandwidth without overloading. Therefore, the system could handle as many capture devices as desired without any problem that explains the lightweight and scalability of the proposed architecture.

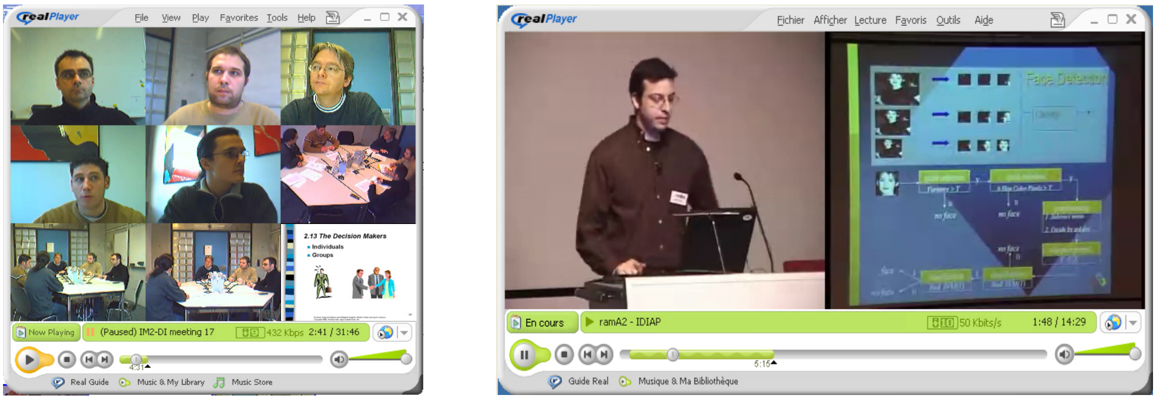


Fig. 3.3. Snapshot of meeting recordings played with SMIL containing multiple audio/video streams containing the close-up view of each participant, room overview, projected documents (left) and one of the presentations of the *MLMI* 2004 conference containing the audio/video of the presenter and projected documents (right).

3.3.1.3. Independent of capture devices

The *DocMIR* system's capture architecture could accommodate a wide range of capture devices *i.e.* from cheap and low-resolution web-cams to high resolution pan-tilt-zoom camera. The system provides a plug-and-capture functionality. For simplicity, the FireWire web-cams and Labtec's PC desktop microphone are currently used [*URL16*]. However, other capture devices could be integrated without any trouble. Once the device is registered to the system, it could be used at any time. The device should be plugged to the system before commencing recordings. The synchronization of the multiple audio-visual streams from various capture devices is handled by defining a global clock in the *master* PC to which all the capture boxes handling the capture devices communicate co-operatively through the LAN connection.

3.3.2. Analysis and indexing module

Once the capture is completed, the captured audio-visual streams are used by the analysis and indexing tool for automatically indexing the meeting/conference. Indexing is a central component necessary to facilitate efficient retrieval and browsing of visual information stored in the meeting repository. The tool mainly considers the video of the projected documents and works systematically with the following three steps: (1) document-based meeting segmentation, (2) low-resolution document identification and (3) electronic document content extraction (Fig. 3.1).

3.3.2.1. Meeting segmentation

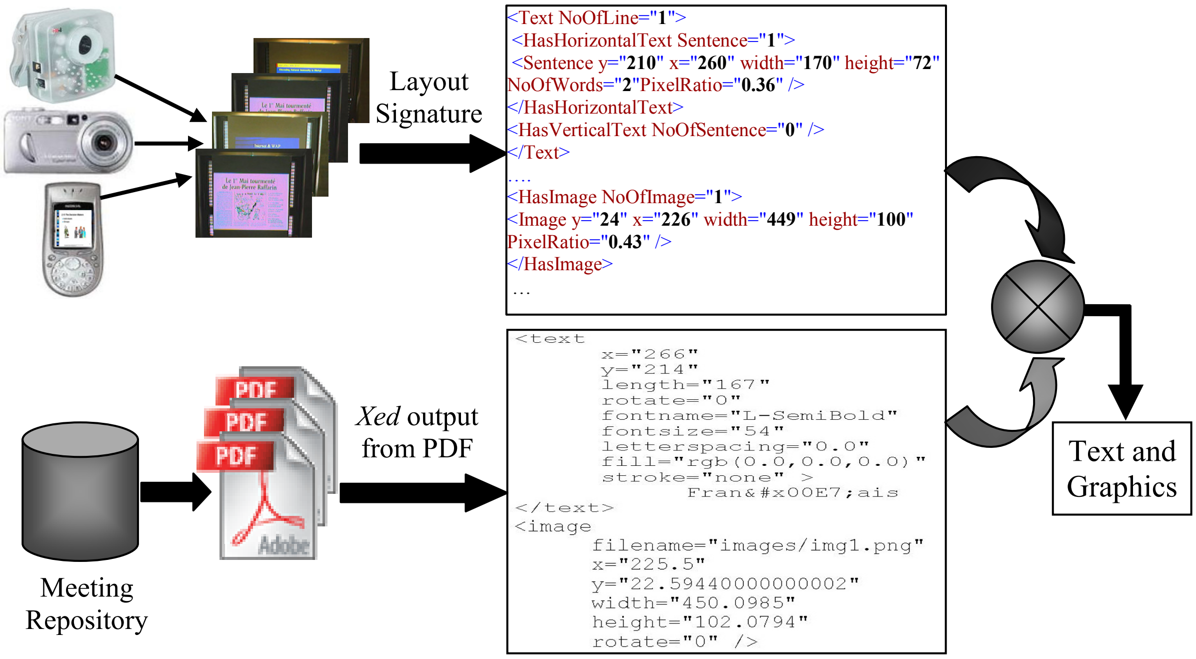
Temporal segmentation of the meeting's audio-visual streams into semantically connected units is an important step to understand the meeting content [*Sundaram* and *Chang*, 2003]. Moreover, it makes fast access to the meeting recordings possible. In this scenario, projected documents are used for the 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. 3.1). These time boundaries, later serve as entry points for non-linear access to meeting records, *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. 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. A novel method is proposed in Chapter 4 that detects the above-mentioned entry points.

3.3.2.2. Low-resolution document recognition

In the previous step, the recorded meeting is temporally fragmented into distinct smaller segments (Fig. 3.1). Each meeting segment corresponds to a stable period of the video of the projected documents and the detection of such periods is explained in Chapter 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 meeting 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 as well as it helps accessing the content of electronic documents. Moreover, it is difficult to extract the textual content of the captured document images (key-frames) using *OCR* due to the poor quality of the captured image, low-resolution as well as the textured and non-uniform background. In order to overcome the drawbacks above, a novel approach is proposed and is based on document's *Visual Signature*, which is described in detail in Chapter 5. At the end of this step the ID, which corresponds to an identified original electronic document including page/slide number (Electronic Document ID, Fig. 3.1) from the meeting repository, is added to the annotation file.

3.3.2.3. Content extraction and addition

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 meeting annotation for keyword-based retrieval. The *Document, Image and Voice Analysis* (*DIVA*) research group has developed a tool called *eXtracting Hidden Structures from Electronic Documents* (*Xed*), which extracts the content (both texts and graphics) from PDF document [*Karim et al.*, 2004]. The tool extracts the hidden layout structure of the PDF documents and their contents (textual, graphical, etc.). Both the proposed *Layout Signature* (Chapter 6) 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 (Fig. 3.4).



```
<Text NoOfLine="1">
 <HasHorizontalText Sentence="1">
 <Sentence y="210" x="260" width="170" height="72"
NoOfWords="2"PixelRatio="0.36" />
</HasHorizontalText>
<HasVerticalText NoOfSentence="0" />
</Text>
....
<HasImage NoOfImage="1">
<Image y="24" x="226" width="449" height="100"
PixelRatio="0.43" />
</HasImage>
 ...
```

```
<text
            x="266"
            y="214"
            length="167"
            rotate="0"
            fontname="L-SemiBold"
            fontsize="54"
            letterspacing="0.0"
            fill="rgb(0.0,0.0,0.0)"
            stroke="none" >
                  Fran&#x00E7;ais
</text>
<image
           filename="images/img1.png"
           x="225.5"
           y="22.594400000000002"
           width="450.0985"
           height="102.0794"
           rotate="0" />
```

Fig. 3.4. Content extraction from the electronic documents by comparing the *Layout Signature* of captured documents with the *Xed* output without using any *OCR* systems.

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. One 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 that by 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. 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 and the corresponding projected document for that particular time point, is also displayed. Upon content extraction, the extracted textual content of the electronic document is included in the text attributes of one or more corresponding meeting segments.

3.3.2.4. Indexing performance

The indexing performance of the proposed system is the combined performance of detection of meaningful semantic pointers, which is the *Slide Change Detection* (*SCD*) for this scenario, identification of captured documents and finally the content extraction from the electronic documents. Therefore, the goal is to measure the individual performances and finally to combine them for the evaluation of the system. The detection of semantic pointers *i.e.* the entry points of the projected documents is evaluated in Chapter 4. Then the identification of low-resolution captured documents is presented in Chapter 8.

3.3.3. Retrieval module

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 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 [*Aigrain et al.*, 1996; *Petkovic*, 2000]. 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.

3.3.3.1. Image-based retrieval

In image-based retrieval, captured document images from handheld devices (digital camera, mobile phones, etc.) are used to queries to retrieve the original documents. 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. One could also use the image of original document as users often 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 Chapter 5.

3.3.3.2. 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 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. Following two steps are used for the keyword-based retrieval.

3.3.3.2.1. Keywords extraction:

Once the textual content is extracted, it must then be processed for the extraction of keywords that represent each projected document. The processing step takes the input of textual raw data from *Xed* output of each slide document and the output would be a set of terms from the dictionary (list of specific terms appearing in the documents) that represents the corresponding documents. The processing step is composed of three steps: *pre-processing*, *stopping* (stop-words removal) and *stemming*. In the pre-processing step, all the non-alphabetic characters such as punctuation marks, parenthesis, etc. are removed. In the stopping step, all the *stop-words* (*e.g.* prepositions, verbs of common use such as *to be*, *to have*, articles), which are generally required to make a sentence grammatically correct, are removed. *Stemming* replaces all of the morphological variants of the same word with their stem. For example, the words '*participation*', '*participating*' and '*participated'* are replaced with their corresponding stem '*participate*'. After all the above-mentioned procedures, the original documents are converted into streams of terms, which are nothing but the keywords in the document. The processing step above is carried out on each original electronic document in the repository and therefore, the process is referred as *offline*.

3.3.3.2.2. Keywords matching:

During the keyword-based retrieval, the general queries are expressed in natural language. These queries are processed in a similar way as explained before for the extraction of relevant terms (keywords). In this case, the extraction of terms from queries is done after receiving the queries and therefore, the process is referred as *online*. These terms are matched with those that are already extracted from the original slide documents and are added to the document *Visual Signature* (Chapter 5). Documents which are relevant to the queried terms are considered as the required solutions. Normally, one query, q would contain one or more term, t. In order to retrieve the relevant slide documents, the similarity score between the queried terms and the terms in the documents are considered. The matching score is computed using the commonly used similarity metric in *Apache Lucene* text-search engine (http://lucene.apache.org/java/docs/index.html) and is defined as:

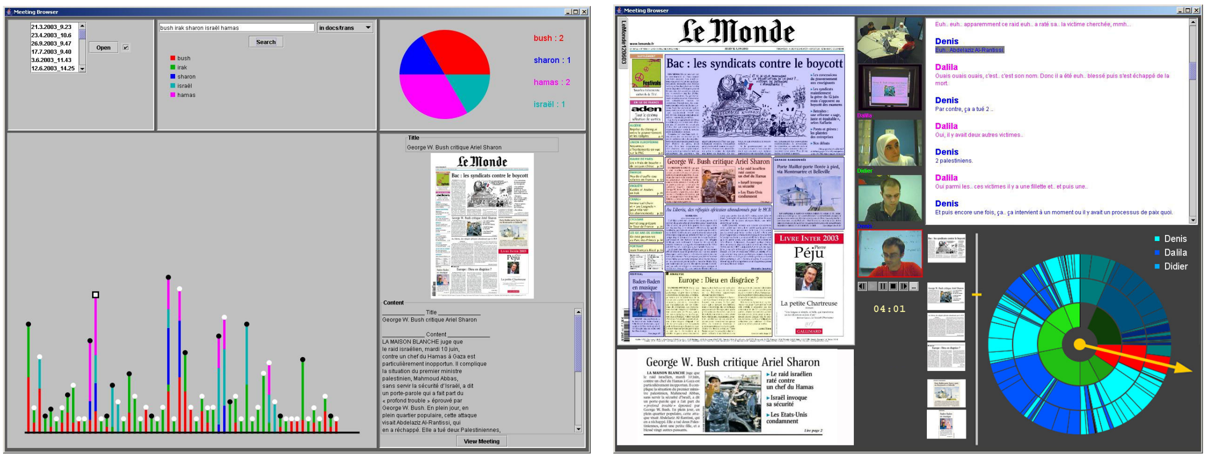
$$score(q,d) = \sum_{t \in q} tf(t,d) \times idf(t) \times NDL(t,d) \times frac(t) \tag{3.1}$$

where *tf*(*t*, *d*) is the score factor, which is based on the number of times the term, *t* appears in the document, *d* (term frequency). The score factor, *idf* refers to the *inverse document frequency*, which is computed using the ratio of the total number of slide documents in the repository upon the number of slide documents containing term, *t*. *NDL*(*t*, *d*) is the normalized document length *i.e.* the ratio of the length of the document, *d* and the average document length in the repository. The score factor, *frac*(*t*) is based on the fraction of all query terms that the document, *d* contains *i.e.* the ratio of the matched terms upon the total queried terms. The slide documents are ranked according to the matching score of the queried terms. The score would be zero when the term, *t* does not appear in the document, *d* and would be higher when *q* and *d* share more terms. The score factor, *tf*(*t*, *d*) gives more weight to the term appearing frequently in a document. Therefore, they are considered as a better representative of the document. The inverse document frequency (*idf*) makes the contribution of terms appearing in few documents higher *i.e.* more discriminative. The score factor, *frac*(*t*) represents the presence of a large portion of the queried terms. For long documents, the probability of sharing terms with a query is higher and it results in higher scores, which is the main limitation of the above approach. However, this effect is smoothened by the presence of NDL, which reduces the contribution of terms that belong to longer texts. Moreover, the targeted documents are presentation slides that often contain limited text as compared to other documents like journal articles, newspapers, magazines, etc. Therefore, the main contribution of *tf*(*t*, *d*), inverse document frequency (*idf*) and score factor (*frac*(*t*)) are considered for the computation of the keyword matching score.

3.3.3.3. Summary of retrieval performance

The retrieval performance of the system is dependent on indexing performances as described before. It also depends on the delivery of meaningful information to the user query. The query could be either image-based or keyword-based or both. The performance for the image-based query is discussed in Chapter 8. For the keyword-based retrieval, the general queries are expressed in natural language and they are searched to find all of the documents that contain one or more queried keywords. It is done by simply comparing the bag of words that is queried by a user. The keywords are searched for in the textual content of the document image or the speech transcription of the presenter or participants. The evaluation of keyword-based retrieval depends on the usefulness of keywords typed by user.

Fig. 3.5 demonstrates *FriDoc*, a document-centric interface for browsing multimedia meeting archives. 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. 3.5, left). 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. 3.5, right). Furthermore, it allows users to non-linearly access and browse meetings using documents, control bar, sunBurst visualization and speech transcriptions [*Lalanne et al.*, 2005].



Title
George W. Bush critique Ariel Sharon

Content

__________ Title __________
George W. Bush critique Ariel Sharon

__________ Content __________
LA MAISON BLANCHE juge que le raid israélien, mardi 10 juin, contre un chef du Hamas à Gaza est particulièrement inopportun. Il complique la situation du premier ministre palestinien, Mahmoud Abbas, sans servir la sécurité d'Israël, a dit un porte-parole qui a fait part du « profond trouble » éprouvé par George W. Bush. En plein jour, en plein quartier populaire, cette attaque visait Abdelaziz Al-Rantissi, qui en a réchappé. Elle a tué deux Palestiniennes,

Euh.. euh.. apparemment ce raid euh.. a raté sa.. la victime cherchée, mmh...

Denis
Euh.. Abdelaziz Al-Rantissi.

Dalila
Ouais ouais ouais, c'est.. c'est son nom. Donc il a été euh.. blessé puis s'est échappé de la mort.

Denis
Par contre, ça a tué 2 ..

Dalila
Oui, il y avait deux autres victimes..

Denis
2 palestiniens.

Dalila
Oui parmi les.. ces victimes il y a une fillette et.. et puis une..

Denis
Et puis encore une fois, ça.. ça intervient à un moment ou il y avait un processus de paix quoi.

Fig. 3.5. FriDoc browser; cross-meeting navigator (left), keywords and/or captured image-based retrieval of documents, that are linked to the corresponding meeting segments and intra-meeting navigator (right) containing audio/video, documents, speech transcripts, control bar with sunBurst visualization.

3.4. Conclusion and perspectives

In this chapter, the usefulness of the documents in a multimodal environment such as meeting/lectures recordings is presented. The documents, which are in visible focus during the meeting, conferences, lectures, etc. is successfully used for the indexing of the recorded multimedia meeting data. The visible documents hold the unique temporal relationship with the meeting times, which act as semantic pointers to the meetings audio-visual streams later on. The visible documents during the meetings not only act as semantic pointers but also used as interface to access the recorded multimedia meeting data.

The proposed *DocMIR* system captures, analyzes and then automatically indexes captured multimedia meeting recordings without user intervention. The system consists mainly, of three components called *capture*, *analysis* and *indexing* and *retrieval* module. The *capture module* is simple, light-weight, scalable and distributed. It also accommodates a wide range of capture devices. It could be extended further to consider the data from white boards and personal note books. The *analysis* and *indexing module* considers the visible documents as an important aspect for the automatic indexing. It detects the entry points of a document and also identifies the corresponding original documents from the meeting repository. Furthermore, it extracts the textual content from the electronic documents and attaches it to the corresponding multimedia meeting segments. Finally, the *retrieval module* accepts both images and keywords from the user and returns back the corresponding meeting segments and original documents to the user. Basically, the user interacts with the system through the projected documents, which is the main focus during most meetings, conferences, lectures, etc.

In order to ensure the success of this system, one of the most important aspects to consider is an efficient indexing, which includes the temporal segmentation of the multimodal information through the identification of documents that appeared during the meetings. The document-based temporal segmentation of meetings is discussed and elaborated in Chapter 4. The identification of captured low-resolution documents is then explained in Chapter 5. Once the low-resolution captured document images are identified, the content of the documents can be easily and efficiently extracted without the use of *OCR*, which would not only be time consuming but also requires various systems to handle various languages.

Chapter 4

Document-based segmentation of meetings using *Slide Change Detection*

Projected documents are *virtual containers* for information related to meetings, conferences, lectures, etc. Such documents play an important role in efficient information retrieval. Often, people are interested to know about both the speaker's and participants' interactions concerning a certain point in a particular document presented by the speaker. Therefore, the captured audiovisual streams of the meetings should be fragmented into reasonably distinct smaller segments for quick non-linear access to the meetings on demand. This would help the user for instant seeking to the point of interest in the meeting using the document and/or keyword of interest without scanning from the beginning of the meeting. *SCD* consists in finding slide segment boundaries *i.e.* time instants when a slide begins and stops displaying in a given video stream containing projected slides.

In this chapter, the detailed description of the procedure for document-based segmentation of the captured meetings is presented. The organization of this chapter primarily follows the segmentation process by extracting the time boundaries of each projected document in the video stream, first. Therefore, Section 4.1 starts by giving a brief motivation to the document-based segmentation. Section 4.2 describes the existing approaches and their limitations. In Section 4.3, the proposed algorithm for the segmentation is described along with its evaluation in Section 4.4. Finally, Section 4.5 concludes with future perspectives.

4.1. Motivation

During the meetings, the projected documents such as slides hold a particular relationship with the meeting times. They appear at a specific time and remain for a definite duration in the display area as visual focus. In our case, this time information serves as indexes to the recorded meetings and is extracted from the recorded slideshow videos by analyzing its content. To extract these time boundaries, first one should detect the shots in the recorded video. The shot is the most basic unit after the single frame and is contiguously recorded audio/image sequence. Different shots are separated by a camera break or cut, which is characterized by an abrupt change from one frame to the next. It could also be separated by edited transitions such as dissolves, fade in/out and wipes where images are superimposed at a shot boundary [*Brunelli et al.*, 1999]. Most of shot detection algorithms looks for cuts and breaks in the video, thereby dividing it into distinct scenes. If there are no cuts, like in this scenario where only one camera is used and kept running continuously, these algorithms usually detect the changes such as the movement of the speakers in front of the projection screen. Furthermore, in our case the shot detection algorithm has to run on the video stream captured using low-resolution capture devices such as web-cams. Web-cams possess auto-focusing functionality, which requires a certain amount of time (~ 0.5 second) to capture a stable image after a scene change. During the transition period, each captured frame would not be the same, mainly in its color and contrast level. Thus, in our case the temporal segmentation requires further analysis.

4.2. Review of existing approaches

In this section, the state-of-the-art and existing approaches for the detection of such changes in the captured video are presented.

4.2.1. State-of-the-art methods

A video consists of three-dimensional signal (x, y, t). The horizontal direction, x and the vertical direction, y of frame reveal the flow of visual content. The last one is time, t which expresses the variations in the flow of visual content. In case of shot boundary detection, the time, t_i at which there is a significant change between the visual content of the current frame, f_i and next frame, f_{i+1} is detected and analyzed. According to *Boreczky* and *Rowe*, six major techniques have been used for shot boundary detection [*Boreczky* and *Rowe*, 1996], (1) pixel difference, (2) statistical difference, (3) histogram comparison, (4) edge difference, (5) compression difference and (6) motion vectors.

The pixel difference method is the simplest way to detect a change and is based on the pixel-by-pixel comparison between two adjacent frames in the video sequence. *Zhang et al.* considered a dissimilarity measure by counting the number of corresponding pixels in two frames with the differences in luminance exceeding a certain threshold. If this number of different pixels is large enough, the two processed frames are declared to belong to different shots [*Zhang et al.*, 1993]. *Hampapur et al.* computed a chromatic image by dividing the change in gray level of each pixel between two images by the gray level of that pixel in the second image. During dissolves and fades, this chromatic image assumes a reasonably constant value. They also computed a similar image that detects wipes [*Hampapur et al.*, 1994]. Since this technique is based on pixel-wise differences, the dissimilarity measure is highly sensitive to motion, minor changes in illumination and has heavy computational loads.

The methods based on statistical differences, require statistical functions using a robust metric for visual content discontinuities. *Zhang et al.* described a method in which an image is divided into a set of blocks and a likelihood ratio is computed for each corresponding block in the f_i and f_{i+1} frame [*Zhang et al.*, 1993]. The likelihood ratio per block is computed by considering the mean and standard deviation of intensity histogram of the block. This method is reasonably tolerant to noise, but is slow due the complexity of the statistical formulas. It also generates many false positives (*i.e.* changes not caused by a shot boundary).

Histogram-based algorithms are the most common shot boundary detection methods because of their simplicity. The histogram-based methods compute gray-level or color histograms of two adjacent images (global histograms). If the bin-wise difference between the two histograms is above a pre-fixed threshold, a shot boundary is considered [*Ueda et al.*, 1991; *Zhang et al.*, 1993].

Little et al. proposed a method that uses the differences in the size of Joint Photographic Experts Group (JPEG) compressed frames to detect the shot boundaries [*Little et al.*, 1993]. *Yeo* and *Liu* demonstrated the use of DC image difference for shot boundary detection in MPEG sequences [*Yeo* and *Liu*, 1995]. The DC image difference is measured by computing the distance between the DC images of frames. DC images are computed from frames which are divided into 8 × 8 blocks. They also applied a combination of histogram and pixel difference metrics to the DC images.

An edge-based method has been proposed by *Zabhi et al.* The method considers the number and position of edges in the detected images. The percentage of edges that enter and exit between the two frames is computed. A shot boundary is declared, when the number of edges that appear or disappear in a frame, with respect to the previous frame, is higher than a threshold [*Zabhi et al.*, 1995]. The algorithm is fairly complex, as it requires computing edges, registering the images, computing incoming and outgoing edges, and finally computing an edge fraction.

Motion vectors are also used by *Zhang et al.* and *Ueda et al.* to detect whether or not a shot was a zoom or a pan [*Zhang et al.*, 1993; *Ueda et al.*, 1991]. These motion vectors are determined from the block matching. Motion vector information can also be obtained from MPEG compressed video sequences. However, the block matching performed as part of

MPEG encoding selects vectors based on compression efficiency and thus often selects inappropriate vectors for image processing purposes [*Boreczky et al.*, 1996].

All the above-mentioned approaches in the literatures are tailored for videos which both the camera and the target objects are not fixed during capturing. As we mentioned earlier, the camera used for capturing is fixed and running continuously from the starting of the meeting recording until the end. Moreover, the projector screen, which displays the projected documents that are captured by a camera, is also fixed. Only, the content in the projected document is changing over time. Therefore, the above-mentioned approaches would not perform well in all cases and especially for *SCD*s. This is due to the fact that in real-world presentations, slides in a slideshow often have the same background and color content which generally correspond to a design pattern. In this case, only the layout, textual and the graphical content vary. Moreover, the goal is to detect a particular document that remain in visual focus for a certain period of time in the captured document video than to recognize the type of shot boundaries *i.e.* transitions such as abrupt change or gradual change (fade in/out, dissolve, wipe) rather. Therefore, the above-mentioned techniques in literature are not adapted to detect such changes, especially with a low-resolution capture devices such as web-cams, resulting in poor contrast level. Moreover, web-cams have auto-focusing function, which modifies the lighting condition and added fading during transition and thus, take nearly 0.5 second to capture stable images after a change in the projected documents (Fig. 4.1). During this transition period, the histogram techniques detect non-existent changes as we see in the Section 4.4 describing experiments and evaluation.

In order to compromise on *SCD*, *Mukhopadhyay* and *Smith* proposed a method for segmenting a slideshow video captured during a lecture. The segmentation is based on the identification of the projected slides during lectures and is described in the following subsection [*Mukhopadhyay* and *Smith,* 1999].

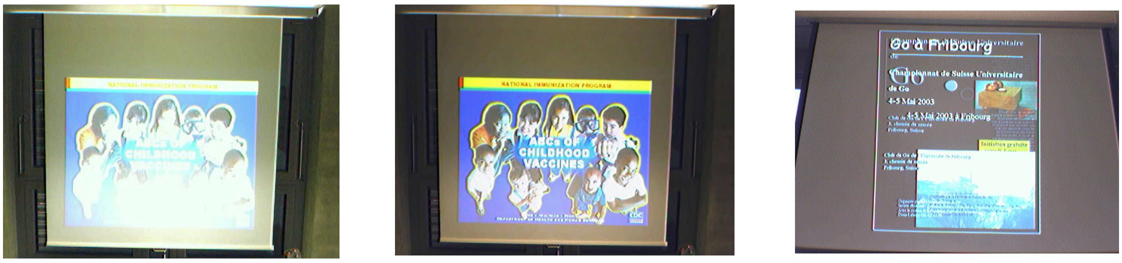


Fig. 4.1. Auto-focusing modifies the lighting conditions and fading during transition.

4.2.2. Cornell's lecture browser method

This approach by *Mukhopadhyay* and *Smith* targeted scenarios such as live lectures. In this case, the overview camera is fixed and running continuously to capture the entire lecture dais from which the presenter delivers his/her lecture. The tracking camera, which contains a built-in hardware tracker that follows the speaker, captures the head-and-shoulders shot of the presenter. The video from the overview camera contains a foreshortened version of the projected slides. Since the overview camera and the projector are fixed, the corner points of the projected slide are determined once. The *SCD* method considers the following criteria:

- The slides are presented in the same order in which they appear in the electronic file (PPT or PDF). This is not always the case since presenters often skip/move around their slides during their presentations.
- In order to validate the slide change detection, the extracted slide images must then be identified from a repository containing original slides. This is not always the case since speakers often present video clips/websites, linked to their presented slides during their presentations, which are often not in the repository.

The Cornell method uses a feature-based algorithm to detect the changes. The method clips frames from the video of the overview camera to the bounding box of the projected slides. The clipped frames are low-pass filtered for noise reduction and then adaptively thresholded [*Yanowitz* and *Bruckstein*, 1989] to produce the corresponding binary images. The distance between two successive binary images B_1 and B_2 is computed as $\Delta = (d_1 + d_2)/(b_1 + b_2)$. Where b_1 is the number of black pixels in B_1 and d_1 is the number of black pixels in B_1 whose corresponding pixels in B_2 are not black. Similarly, b_2 and d_2 are defined. A slide change is signaled whenever the distance between two successive binary images (Δ) exceeds a certain threshold. The threshold is defined so that no slide change goes undetected. Indeed, it would not matter even if some extra slide changes that do not exist were generated, since at the end, for each slide change an image is extracted and further identified from a repository. By this way, the extra slide changes detected are removed. Furthermore, the method does not mention anything about the handling of animations and gradual transitions during the presentation. For such cases, the above-mentioned *SCD* method would be computationally expensive as each animation or gradual change could result in a change of slide and thus would need to be identified from the original slide for the validation of non-existent slide change.

This method works perfectly for slideshows with high contrast and having light background with dark texts. However, it is difficult to set a unique threshold value for various slideshows having heterogeneous background color or graphical content. Finally, many extra slide changes are generated due to the auto-focusing nature of the web-cam and it is computationally expensive as each slide has to be identified in order to validate the slide change detection.

4.3. Proposed method

Our proposed method is in some aspect similar to the above-mentioned Cornell's lecture browser's method. Our proposed *SCD* algorithm has the following similarities with the above-mentioned Cornell's method:

- It is designed for videos containing projected documents, *e.g.* slideshow presentations.
- The measurement of distance between two frames is computed as used for the same in the Cornell method.

Meanwhile, the dissimilarities between both methods are listed below:

- It does not include any above-mentioned hypothesis about the slide sequences in which they are presented.
- It does not require document identification to validate the change of documents in the video sequence.
- It uses *Otsu* binarization method [*Otsu*, 1979].
- The method is two-fold, uses window-based approach and des not consider all frames in the video sequence for the processing.
- Documents, which appear less than 2 seconds, are considered as skipped.
- It detects *stable* and *unstable* periods in a video sequence and the document change detections are carried out in the *unstable* periods.
- It overcomes the auto-focusing characteristics of low-resolution capture devices such as web-cams.

To be more precise, the proposed method detects *stable* and *unstable* periods in a sequence of frames from the video of the projected documents. Each *stable* period corresponds to a projected document that stays in the visual focus whereas the *unstable* period represents either the transition of documents or animation during the presentation. The final goal is to segment the meeting document-wise and then to associate the original electronic document to each of the video segment.

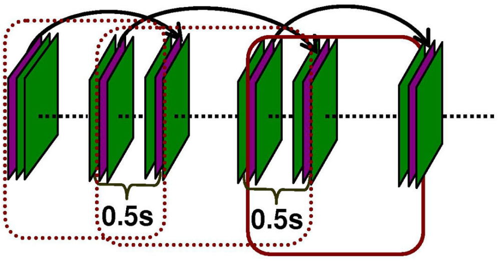


Fig. 4.2. Stability detection in the video stream using a sliding window with duration of 2 seconds.

4.3.1. Stability detection

The *stability* detection is carried out by sliding a window of 2 seconds over the sequence of frames in the captured video containing projected documents. Here, the documents that stay on the screen less than 2 seconds are defined as skipped during the presentation (Fig. 4.2) [*Behera et al.*, 2004a]. The process is mainly two-fold; (1) checking for *stability i.e.* whether the current window is *stable* or *unstable* and (2) *stability* confirmation. 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. 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 two-step procedure is explained below in detail.

- The first frame, $F_1$and the last frame, F_{N+1} in the sliding window at a given position in the sequence are converted to bi-level images using the *Otsu* segmentation method (Fig. 4.3) [*Otsu*, 1979]. For the binarization of document images, *Trier* and *Taxt* showed that the performance of *Otsu* method is best among other global methods [*Trier* and *Taxt*, 1995]. 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, Δ between them is computed using the Cornell's method as described above [*Mukhopadhyay* and *Smith*, 1999]. If $\Delta > T_1$, then the second step would be executed for the confirmation of instability; otherwise the window would be moved forward by skipping ¾th (1.5 second) of the previous position of window *i.e.* the frames which were in the last ¼th of the previous window position become the first ¼th of the current position of the sliding window. The aim is to keep the continuity of the frame sequence to avoid the unnoticed frame and could be a different document image if it appears in the first frame of the newly positioned window (Fig. 4.2). 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.

- In the second step, an individual distance Δ_i ($i = 1 \ldots N$) is computed by comparing the frame, F_1 with the rest of the N frames in the sliding window (Fig. 4.3). If the ratio $R = (\Delta_m / \Delta_v) < T_2$, then the instability is confirmed. Where $\Delta_m = \frac{1}{N}\sum_{i=1}^{N}\Delta_i$ is the mean, and $\Delta_v = \frac{1}{N}\sum_{i=1}^{N}(\Delta_i - \Delta_m)^2$ is the variance of distances. Normally, an unstable window contains two or more different kinds of frames which indicate the variance of distances in such windows would be significantly higher than those of stable windows, which contain only one kind of frames.

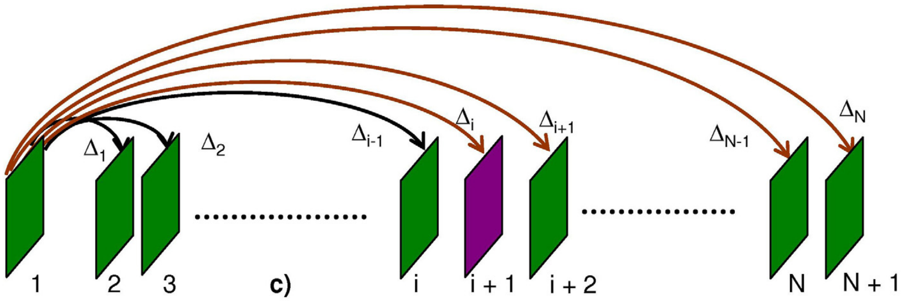


Fig. 4.3. Confirmation of instability by considering each frame in the sliding window.

4.3.2. Slide Change Detection

Once, the window is confirmed as being an *unstable* one, the exact position of the first dissimilar frame as compared 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 $max(\Delta_i)$ ($i = 1 \ldots N$) of all the distances in the window (Fig. 4.3). 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 i_{th} position is the incoming new slide document. The corresponding time for this incoming new 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 i^{th} frame and the above-mentioned two-fold *stability* inspection would be continued until the end of the video stream. 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. 4.4). The output of the algorithm is organized in an XML format with the respective time (start and stop) and type (*normal* or *skip*) attributes as shown in Fig. 4.5. If the duration $t_p - t_{p-1}$ (time between successive change detection) is less than 2 seconds then the corresponding type attribute (Fig. 4.5) is updated with *skip*; otherwise it is considered as *normal*.

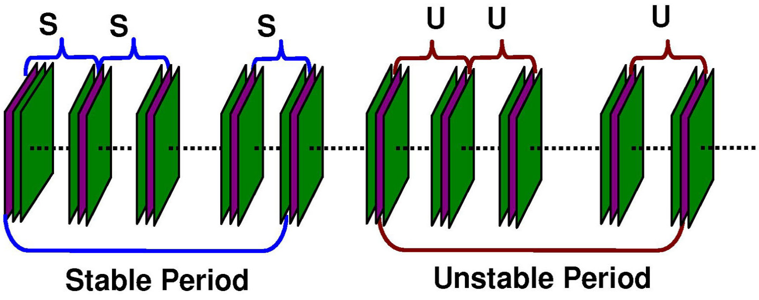


Fig. 4.4. Two or more consecutive windows of the same type are merged to form stable period or unstable period.

Furthermore, it has been observed that in case of animations in the presentations, the whole animation duration is detected as an unstable period. In such unstable periods, the method detects the changes for frames that have content-wise dissimilarity of at least 25%. For such changes, the intermediate frames are considered as skipped slides (Fig. 4.5) if the appearance time is less than 2 seconds; otherwise considered as a new slide. To be more precise, the finer animations (*e.g.* single line, words) in a slide are often not detected by the proposed method. However, if there is a big change (> 25%) then it is detected as a new slide. Actually, the number of stable periods corresponds to the number of slides having their type attribute set to *normal* and two stable periods are separated with an unstable period (Fig. 4.4), which depicts the transition of the previous slide to the current one. Moreover, the duration of the unstable periods is significantly less than those of stable ones and is obvious as unstable period reflects the transition of documents. On the other hand, the number of unstable periods should correspond to the number of slide transitions *i.e.* if there are *S* number of slides in the presentations, then theoretically the number of transitions would be (*S* – 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 or coming back to a previous slide. However, the aim is to detect and include even those slide documents in the annotation as this would be interesting for some of the listeners for retrieval on demand.

```
<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. 4.5. An example of SMIL file, which is the output of our slide change detection.

4.4. Experiments and evaluations

The above-mentioned method has been evaluated automatically by capturing projected slideshows (PPT, PDF). These slideshows have been collected from various presentations, which are available to public. If the real-word presentations are considered, then the ground-truth for the presentation *i.e.* the time information of each slide change is normally not available. It has to be done manually by watching the presentation video from the beginning till end, once the capturing is completed. It could be possible during the presentation by keeping notes about the time information for each slide change. The above-mentioned manual ground-truth production is not only time-consuming and tedious but also prone to errors while preparing it. In order to overcome this, an application that generates the ground-truth using SMIL (Synchronized Multimedia Integration Language) has been developed at our research group [*URL15*].

4.4.1. Experimental setup

Various presentations related to education, technical and non-technical contents have been collected. These are available on the web and mainly from conferences and seminars in various public and private sectors. In doing so, more than 3000 slides (65 slideshows) have been accumulated, which represents different varieties of presentation styles. A corpus has been built in order to balance equally various characteristics like: number of slides, background color, font color and size variability, background variability, graphics content, etc. This corpus is available in meeting data server of the *University of Fribourg* (http://diuf.unifr.ch/im2/data.html). The aim is to capture presentations as in the real world scenario. Therefore, the order of slides is kept as it is in the slideshows. The JPEG image of each slide of the slideshow is picked up with random presentation time and if it is less than 2 seconds then the type attribute is assigned as *skip*, otherwise *normal* (Fig. 4.5). In order to make the corpus more realistic, the type attribute of *skip* is added to some of the slides towards the last quarter of the slides in some slideshows. This is based on simple heuristics and is picked up randomly. This is considered by keeping in mind that the presenter often skips some of the slides towards the end of the presentation due to the shortage of presentation time. In such way one SMIL file per slideshow is generated. The SMIL file is played in the RealPlayer of the PC/Laptop connected to the projector and simultaneously the web-cam focusing on the projector screen, starts capturing. RealPlayer functions in theater mode, so that the border of the player is not visible. The PC/Laptop playing RealPlayer and the capture box controlling the camera focusing on the projector's screen get the slideshow's start and stop time from the SMIL file to start and stop, synchronously. Once the capture is over, the video is transferred to the server for later segmentation.

4.4.2. Criteria for evaluating the segmentation performance

The output of video segmentation and the ground-truth are in XML and contain attributes of image file name, start and stop times with type of slides (*skip* or *normal*). Therefore, the matching simply compares the attributes (start time, end time and type) of individual slide in the ground-truth and in the output of video segmentation (Fig. 4.3).

The evaluation is carried out using the metrics of *Recall* (*R*), *Precision* (*P*) and *F-measure* (*F*). *F-measure* conveys the combine measure of *Recall* and *Precision*. For this evaluation, these metrics are defined as follows:

$$\begin{aligned} Recall\ (R) &= \frac{\#\,correct\ slide\ changes\ detected}{\#\,total\ slides\ in\ the\ ground\text{-}truth} \\ Precision\ (P) &= \frac{\#\,correct\ slide\ changes\ detected}{\#\,total\ slide\ changes\ detected} \\ F\text{-}measure\ (F) &= 2\times\frac{R\times P}{R+P} \end{aligned} \tag{4.1}$$

The evaluation is carried out with the tolerance of one and four frames, *i.e.* for the video of 15 frames per second (FPS); the respective tolerances are of 66.67 and 266.67 millisecond. The above-mentioned metrics for the evaluation are computed presentation-wise and is considered by respecting the real-world scenario. Moreover, the effect of error on the overall performance is statistically analyzed and compared with the variation of the tolerance as well as for a fixed tolerance. The sensitivity is considered using the *Standard Error of Mean* (*SEM*) and for p number of presentations it is computed as follows:

$$E_{SEM} = \frac{\sigma}{\sqrt{p}} \text{ where } \sigma = \sqrt{\frac{\sum_{i=1}^{p}(X_i - \bar{X})^2}{p-1}} \text{ and } mean\ \bar{X} = \sum_{i=1}^{p} X_i \tag{4.2}$$

X_i is the performance of the i_{th} presentation. The *SEM* is computed for each of the above-mentioned metrics.

4.4.3. Automatic evaluation

Once the capture is over, then the *SCD* algorithm starts processing the captured video stream of the projected documents. The output of the algorithm is in XML with the following attributes of *slide id* (Fig. 4.5), *imagefile* (corresponding image file name), *st* (start time), *et* (end time) and *type* (skipped or normal appearance). The output of the *SCD* and the corresponding ground-truth (SMIL) are in XML with the same attributes. The evaluation is simply matching of the output of the *SCD* with the corresponding ground-truth. The matching follows the simple heuristics by considering one node from the *SCD* output and find the matching one from the ground-truth SMIL. The matching compares the attributes *st*, *et* and *type* of the output and ground-truth files. If the difference of the respective *st* and *et* of the source (*SCD* output) and the target (ground-truth) node is inferior to a pre-defined tolerance and both have the same *type* attribute then the target node is ascertained as the matched one. For example, assuming there are N_s and N_g number of nodes in the *SCD* output and the corresponding ground-truth, respectively. Each node of the *SCD* output and the ground-truth XML tree represents the slide (Fig. 4.5). For each node, P_i (i = 1…N$_s$) from the *SCD* output, the matching procedure would consider a counterpart node, Q_j (j = 1…N_g) from the ground-truth if and only if the following condition is satisfied.

$$\{P_i(type) == Q_j(type)\} \wedge \{\| P_i(st) - Q_j(st) \| \leq T\} \wedge \{\| P_i(et) - Q_j(et) \| \leq T\}$$

where T is the tolerance and is defined as one and four frames for a frame rate of 15 *frames per second*, as mentioned above. If the node, Q_j is found to be the matched one for P_i then both the nodes P_i and Q_j are removed from the *SCD* output and the ground-truth, respectively and both are set for the next comparison. After considering all the nodes from the ground-truth, if the corresponding match for P_i is not found then P_i is considered as not matching and removed from the *SCD* output. The same matching procedure is carried out for all other existing nodes in the *SCD* output. The matching would be terminated if the total numbers of nodes in any of the source *i.e.* either *SCD* output or ground-truth reaches zero. The corresponding metrics of *Recall* (*R*) = # *correct matched* / N_g, *Precision* (*P*) = # *correct matched* / N_s and *F-measure* (*F*) are computed (4.1). *Recall* conveys the proportion of relevant changes in the *SCD* output, out of all relevant changes available in the corresponding ground-truth. *Precision* communicates the proportion of relevant changes out of all changes in the *SCD* output. Finally, the traditional *F-measure* is used as a single measure of performance and is the weighted harmonic mean of *Recall* and *Precision* (4.1). All the above-mentioned metrics are computed for a tolerance of one and four frames per presentation.

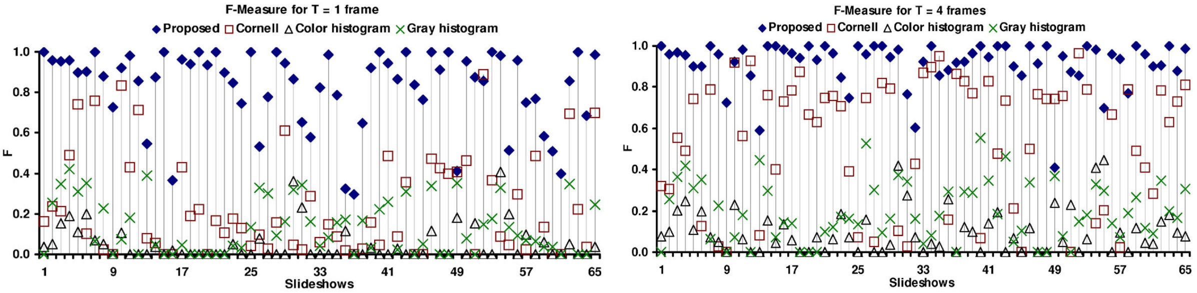


Fig. 4.6. Slideshow-wise performances of various *SCD* algorithms using *F-measure* (*F*) for tolerance of 1 (left) and 4 frames (right).

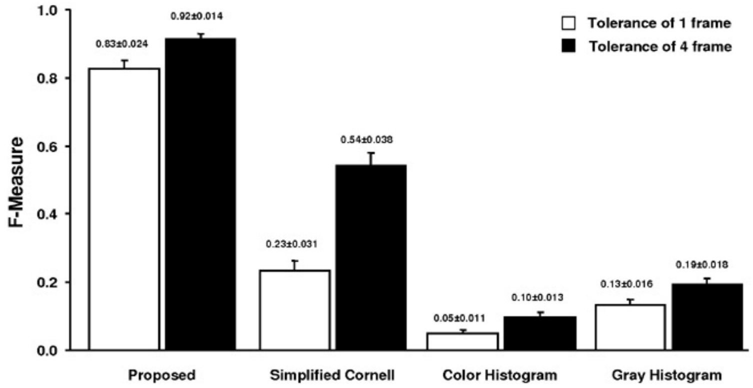


Fig. 4.7. Overall performances of various *SCD* algorithms using *F-measure* (*F*) with *SEM* for tolerance of 1 and 4 frames.

4.4.4. Performance comparison

The performance of the proposed method with the other existing methods such as Cornell, color histogram, grayscale histogram is evaluated, slideshow-wise. In this evaluation, a simplified Cornell method is used which considers neither the order in which slides are presented, nor the slide identification method to validate the slide change detection. A total of 65 slideshows have been used for the evaluation. The effect of *standard error of mean* (*SEM*) on the overall performance over 65 slideshows has also been analyzed for each of the above-mentioned method. The performance of all the above-mentioned methods is presented for a respective tolerance of one and four frames along with the *SEM*. All evaluation results are presented as mean ± *SEM*. The performance metric *F-measure* for slideshow-wise and all slideshows with the *SEM* are presented in Fig. 4.6 and Fig. 4.7, respectively. The performance metric *Recall* for slideshow-wise is presented in Fig. 4.8 and overall slideshows with *SEM* is shown in Fig. 4.9. The performance metric *Precision* for slideshow-wise and all slideshows with the *SEM* are illustrated in Fig. 4.10 and Fig. 4.11, respectively. Additionally, the overall performance of all the above-mentioned *SCD* methods are presented in Table 4.1 with the respective tolerance of one (T_1) and four (T_4) frames.

Table 4.1. Comparison of performances of various slide change detection methods

Metric	**Proposed method**		**Cornell method**		**Color histogram**		**Gray histogram**	
	T_1	T_4	T_1	T_4	T_1	T_4	T_1	T_4
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

4.4.5. Summary of segmentation performances

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 0.23±0.031 for the tolerance of 1 frame (Table 4.1, Figs. 4.7, 4.9 and 4.11). However, the method uses a slide identification mechanism based on the Hausdorff distance for confirming the slide changes [*Ruckiledge*, 1997], which should considerably increase the *Precision* as well as the processing time for non-existent extra slide changes. The high number of incorrect slide changes detected in this method, distinctly increases the computational work. This drawback is overcome by other 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 (Table 4.1, Figs. 4.9, 4.11 and 4.7). The sensitivity of each of the above-mentioned method is measured by increasing the tolerance from 1 frame to 4 frames.

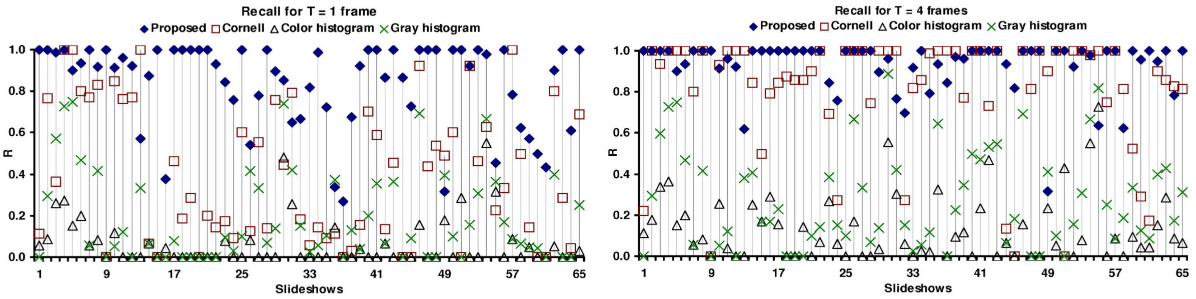


Fig. 4.8. Slideshow-wise performances of various *SCD* algorithms using *Recall* (*R*) for tolerance of 1 (left) and 4 frames (right).

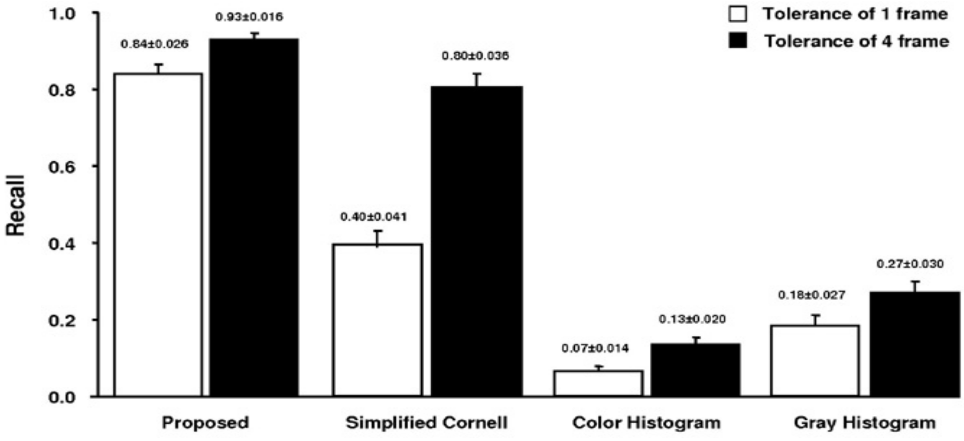


Fig. 4.9. Overall performances of various *SCD* algorithms using *Recall* (*R*) with *SEM* for tolerance of 1 and 4 frames.

Our proposed method is significantly more accurate and less sensitive since 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 1 frame (Cornell, *R*: 0.80, *P*: 0.51, *F*: 0.54) (Table 4.1). Though there is an improvement in *Recall* value (0.80) of the Cornell method for the tolerance of 4 frames (Fig. 4.8), the *Precision* does not match up to that level (0.51, Fig. 4.11). This is due to a significant number of false detections, which reduces the overall performance (*F*: 0.54, Fig. 4.7). 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 4 frames as compared to 1 frame (0.024). Moreover, with the Cornell method, the *SEM* increases from 0.031 to 0.038 for the same (Fig. 4.7). This implies that in case of the Cornell method, though the performance is improved, it shows a tendency to increase variability with the increment of tolerance. Whereas in our 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 a tolerance of 4 frames is even worse than that of the proposed method for a tolerance of 1 frame. In this evaluation, we have considered the tolerance up to 4 frames since we observed that the performance of none of the above-mentioned method vary more than 10% if the tolerance is greater than 4.

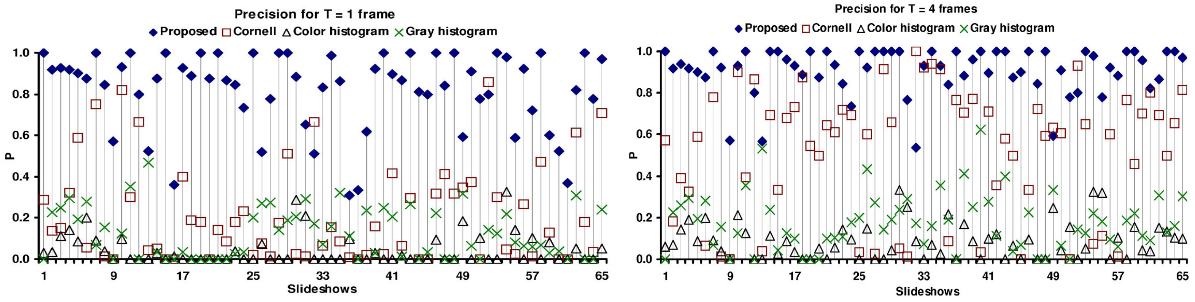


Fig. 4.10. Slideshow-wise performances of various *SCD* algorithms using *Precision* (*P*) for tolerance of 1 (left) and 4 frames (right).

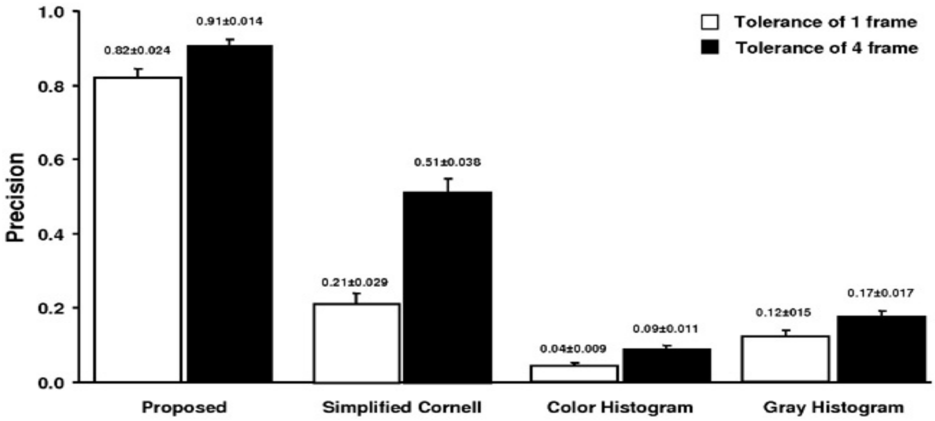


Fig. 4.11. Overall performances of various *SCD* algorithms using *Precision* (*P*) with *SEM* for tolerance of 1 and 4 frames.

The performance between the state-of-the-art-methods, *i.e.* 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 1 and 4 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 1 and 4 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 document. This results in triggering of false slide change detections in case of the color histogram method. Moreover, the *SEM* for *F-measure* is increased from 0.011 to 0.013 and from 0.016 to 0.018 for the respective color and grayscale histogram for a variation of the tolerance from 1 to 4 frames (Fig. 4.7). This signifies that except the proposed method all the other methods show the inclination to increase variability with the increment of tolerance. Since the proposed method out-performed the other existing state-of-the-art methods for the videos captured using low-resolution cameras such as web-cams, we strongly believe in further improvement of performances for those captured from high-resolution capture devices.

4.5. Conclusions and perspectives

The evaluation presented in this chapter has shown that our *SCD* algorithm goes beyond the standard state-of-the-art methods, which takes the video of the projector screen as input and provides the time-codes (start and stop time) for each projected documents. Our proposed algorithm does not include any hypothesis about the presented sequence of slides and slide identification to validate the slide change, which simplifies the slide change detection. During the segmentation, one key-frame per *stable* period is extracted. These extracted key-frames are nothing but the captured low-resolution slide images. These extracted key-frames are to be identified from the meeting repository containing the original presented electronic documents. The identification is based on the document signature and is explained in Chapter 5.

The above-mentioned *SCD* is performed the video containing the projected documents. The proposed algorithm could easily be extended for detecting animations and emphasized part of the document using laser beam, stick, etc. during the presentation. Moreover, the algorithm currently uses the video of only projected documents however it could be helpful to separate the video segment containing mixed shots of the projected documents and the rests.

Chapter 5

Low-resolution document identification

The use of mobile devices such as digital cameras, mobile phones, etc. for capturing physical documents results in low-resolution document images. These captured low-resolution document images could be queried for later retrieval of the original documents and the related information stored in a repository. The characteristics of low-resolution images affect image size, quality and compression technique (loss or lossless) used by capture devices. Furthermore, the captured images often possess perspective distortion due to the positioning of the capture devices with respect to the target physical objects.

In this chapter, we present a signature-based approach to overcome the above-mentioned drawbacks for the identification of captured documents from low-resolution handheld devices. This chapter is organized as follows: the motivation of the identification of low-resolution captured documents is presented in Section 5.1. Section 5.2 describes the existing approaches in the literature and their drawbacks for the identification of such documents. The proposed signature-based identification method is explained in Section 5.3. The rectification of the captured documents and removal of noise is explained in Section 5.4.

5.1. Motivation

Due to advancement in hardware technologies, different kinds of cameras are available on the market. The quality of the captured images is often directly proportional to the size of the capture devices. Since they are small, mobile devices often proposes low-resolution images. Therefore, our aim is to develop an algorithm for identifying documents captured from low-resolution capture devices that can be easily applicable for the high-resolution capture devices without degradation of the performance.

For most of the existing document analysis systems, the system input is either a scanned document of 300 dpi or higher, or an electronic document (*e.g.* PDF, etc.) [*Hadjar et al.*, 2004; *Haralick et al.*, 1994; *Wong et al.*, 1982]. The qualities of these documents are, in general quite high and suitable for further processing. If the perceived document image is captured from a low-resolution device and compressed with lossy format such as JPEG (50 – 100 dpi), then it becomes exceedingly difficult to analyze such documents using standard analysis systems. In most of the capture devices (digital cameras, mobile phones, etc.), JPEG compression is used in order to reduce the storage space and to speed up the processing, which unfortunately implies that more noises are brought in and therefore, some useful details are lost. Furthermore, the captured document images from such devices are often non-uniform in terms of lighting, because of the use of flash or various lighting conditions. Finally, many other distortions or incomplete information are often present (*e.g.* varying distance to the object, motion blur, occlusion, etc). The relatively low resolution, and the frequent variations in the captured environment, make noise removal and content extraction very difficult and drastically decrease the identification rate with standard methods such as using *OCR* or image comparison. For example, in our smart meeting application, projected slides of interest could be captured during a conference and used afterwards for querying the conference repository and retrieve the corresponding original document (one or more slides presented by a particular speaker), or the related audio/video sequence, or any annotations related to the stored media [*Abowd et al.*, 1996; *Lalanne et al.*, 2003; *Mukhopadhyay* and *Smith*, 1999]. For this purpose, we make the reasonable assumption that all the documents present in such environments are stored in advance in a document repository.

5.2. Existing approaches

The taxonomy of the existing document identification approaches could be broadly classified into four different classes.

- The first approach focuses on the document *layout*, which is normally used for documents such as newspaper, journal articles, book pages, etc.
- The second approach is based on the *texture*, which is one of the low-level features and often used in document analysis techniques for the partition of document into different homogeneous regions.
- The third one is based on the document content and specifically the *color* content, which is not often adopted by the document analysis techniques since the documents used in most systems are in black-and-white. However, the color content is one of the most important features for the colorful document such as newspapers, magazines, slide documents, book covers, etc. and could be used along with the layout information for an efficient recognition and retrieval of such documents.
- The last approach is specific to *slide* documents, which is the focus in this work because of their usefulness during a meetings, conferences, lectures, seminars, etc. This approach is based on document-content (mostly textual content) without considering the color and layout information.

5.2.1. Layout-based document identification

The layout analysis of a document image includes a geometric layout and logical layout analysis. The process that is used for the extraction of layout information of a document is called page segmentation. The aim of the page segmentation process is to partition document images into homogeneous regions such as texts, images, tables, drawings, rulers, etc. The geometry of document layout analysis involves specifying the geometry of the maximal homogeneous regions and their classifications (text, table, images, drawing, formulas, etc.). The logical document layout analysis determines the type of document, assigns functional labels such as title, logo, footnote, caption, etc., to each block of the page, determines the relationships of these blocks and order the text blocks according to their reading order as a one dimensional stream [*Jain* and *Yu*, 1998; *Nagy*, 2000; *Bunke* and *Wang*, 1997; *O´Gorman* and *Kasturi*, 1995]. An example of document layout analysis is illustrated in Fig. 5.1 using a newspaper image. Various homogeneous regions in the newspaper are detected using the layout analysis and they are marked with different color to visualize the separation from the neighboring homogeneous regions. The techniques for page segmentation and layout analysis are broadly partitioned into three main categories: (1) top-down, (2) bottom-up and (3) hybrid [*Okun et al.*, 1999].

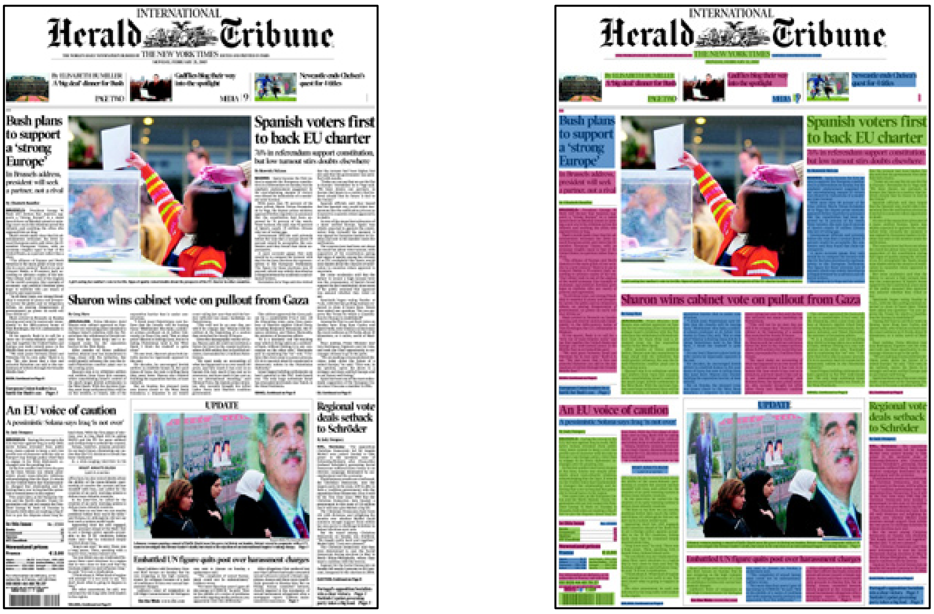

(a) (b)

Fig. 5.1. An example of page segmentation: (a) original image of a newspaper and (b) logical structure of the newspaper with text and image zones.

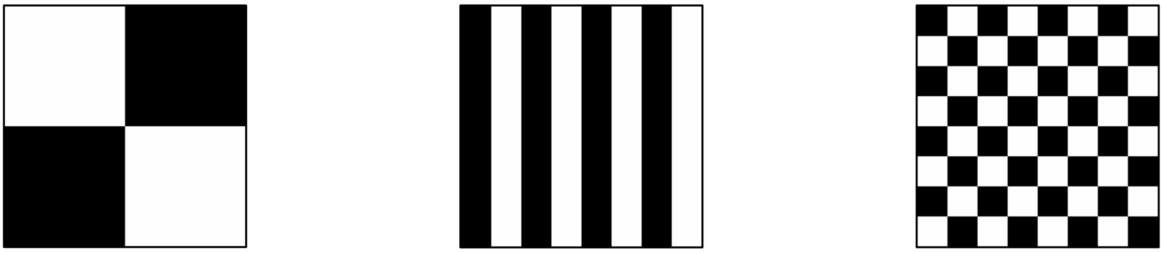
Top-down techniques look for global information on the page such as black and white strips. A common characteristic of this approach is global-to-local processing. Such approaches start by detecting the highest level of structures such as columns and graphics. Then, it partitions the page into blocks, classifies them and proceeds further by successively splitting of classified text blocks until they reach the bottom layer for small scale features like individual characters [*Krishnamoorthy et al.*, 1993; *Ha et al.*, 1995]. For this type of procedures, *a priori* knowledge about the page layout is necessary. *Wong et al.* first proposed this kind of approach in 1982 by introducing the *Run-Length Smoothing Algorithm* (*RLSA*) [*Wong et al.*, 1982]. *Wang* and *Srihari* also investigated a similar algorithm for newspaper layout analysis using projection profile [*Wang* and *Srihari*, 1989]. These approaches perform well for documents assumed to be rectangular in shape with relatively uniform fonts and sizes. However, the performance of such approaches degrades significantly when different components are closely adjacent to each other or overlapping.

Meanwhile, bottom-up methods start with the smallest elements such as pixels, merging them hierarchically in connected components such as characters, words, and then in larger structures such as paragraphs and columns [*O'Gorman*, 1993; *Drivas* and *Amin*, 1995; *Simon et al.*, 1997]. *O'Gorman* described the *docstrum* algorithm, which is a bottom-up approach based on nearest-neighborhood clustering of connected components [*O'Gorman*, 1993]. Using a character size ratio factor f_d, the connected components are separated into two groups, (a) dominant characters and (b) characters in titles and section heading. For each connected component, *K*-nearest-neighbors are found and then, text-lines are found by computing the transitive closure on within-line nearest neighbor pairings using a threshold f_t. Finally, text-lines are merged to form text blocks using a parallel distance threshold f_{pa} and a perpendicular distance threshold f_{pe}. Similarly, *Drivas* and *Amin* described a method that extracts connected components from the image; then, components of the same type are iteratively grouped together to form progressively higher-level descriptions of the documents (*e.g.* words, lines, paragraphs, etc.) [*Drivas* and *Amin*, 1995]. The disadvantage of this approach is that the time complexity is higher as compared to top-down approaches due to the identification, analysis and grouping of the connected components. In order to reduce the computational complexity, *Simon et al.* proposed a method that detects the layout hierarchy as a minimal-cost spanning tree and additional heuristics have been applied [*Simon et al.*, 1997]. Moreover, bottom-up approaches suffer from the traditional problem of incorrect segmentation due to the early errors in groupings.

The hybrid approaches do not fit into either of the above-mentioned two categories and often use the combination of both the approaches [*Wang* and *Srihari*, 1989; *Sobottka et al.*, 1999]. *Wang* and *Srihari* applied a bottom-up approach to detect text lines and non-text data, followed by top-down approach that combines the individual text lines into blocks [*Wang* and *Srihari*, 1989]. *Sobottka et al.* proposed a method that extracts text from complex color images of book and journal covers [*Sobottka et al.*, 1999]. The method uses the top-down technique to split the image into rectangular blocks and the bottom-up technique to detect homogeneous regions of arbitrary shape by using a region growing method. The results of the techniques are combined in order to verify whether the given region is text or not. A more detailed survey of above-mentioned approaches can be found in [*Haralick et al.*, 1994; *Okun et al.*, 1999].

5.2.2. Texture-based document identification

The visual texture attribute is a representation of the surface of an image object [*Haralick et al.*, 1973]. Intuitively, the term 'texture' refers to properties such as smoothness, roughness and regularity of an image object. In general, the structural homogeneity does not result from the presence of a single color or intensity, but requires the interaction of various intensities within a local region. In order to understand the basic properties of texture, let's consider the image set shown in Fig. 5.2. The first image comprises two black and two white blocks. The center image forms a striped pattern with four white and four black stripes. The rightmost image comprises of 32 black and 32 white blocks forming a check-board pattern. Each of these three images comprises of 50% white and 50% black pixels. Therefore, in order to differentiate between these three images, one could use the texture features. Often, document images contain homogeneous regions such as text, image or graphic. Text regions consist of characters with approximately the same size and line thickness that are located at a regular distance from each other. Moreover, text shows spatial cohesion *i.e.* characters of the same string are often of similar heights, gaps and orientation. Therefore, the most intuitive characteristic of text is its regularity. Such regularities have been used implicitly by considering text regions as having a certain texture with frequency components distinct from that of an image or graphic [*Wang* and *Srihari*, 1989; *Jain* and *Zhong*, 1996a; *Jain* and *Bhattacharjee*, 1992]. The problem of such approach is that the time complexity is high since different filters have to be tuned to capture a desire local spatial frequency and the orientation characteristics of a region. Therefore, many masks are used for extracting local features. Moreover, in some cases, regions of different types, having a similar texture, can be confused or merged. In the case of slide documents with non-homogeneous background, this kind of method further risk to be inefficient since the foreground objects would not be distinguished from the background texture.



Block pattern Stripped pattern Checker-board

Fig. 5.2. Three different textures with the same distribution of colors.

5.2.3. Color-based document identification

Color is considered as a low-level visual feature and is widely used in identification and retrieval of photographic images containing natural scenes, city, buildings, etc. The approach is extensively utilized in *CBIR* [*Aigrain et al.*, 1996]. *Global Color Histogram* (*GCH*) is the most conventional way of describing the color content of an image [*Bimbo*, 1999]. The *GCH* for an image is constructed by computing the normalized percentage of pixels having similar color values in an image corresponding to each color element. Considering an n-color model, a *GCH* is then an n-dimensional feature vector (h_1, h_2, …, h_n), where h_j usually represents the percentage of pixels in an image corresponding to each color element, c_j. For example, if one considers the RGB color space then the component, h_j is the unique combination of the values - Red, Green and Blue. For m-bits value of each of the Red, Green and Blue channels, the size of feature vector, n is 2^{3m}. The histogram by itself does not include any spatial information about an image. For example, all the images in Fig. 5.2 have the same number of black and white pixels (50% black and 50% white). In the first image, the black pixels are equally distributed in two different regions of the image. In the second and third image, the black pixels are equally distributed in four and thirty two different regions of the respective images. The *GCH* of the all three images are identical and the spatial information is not captured by the *GCH* (Fig. 5.3, due to black-and-white images, bars are located at 0 and 255). In this context, the identification of similar images is based on the similarity between their respective *GCH*s. A common similarity metric is based on the *Euclidean distance* between the abstracted feature vectors that represent two images, and it is defined as:

$d(Q,I)=\sqrt{\sum_{j=1}^{n}(h_j^Q-h_j^I)^2}$ where Q and I represent the query image and one of the images in the image set, respectively. h_j^Q and h_j^I represent the respective feature vectors of the color element, c_j of the above-mentioned images. The smaller distance, $d(Q, I)$ reflects a closer similarity match. The above inference stems from the fact that color histograms are mapped onto points in an n-dimensional space and similar images would, therefore appear relatively close to each other.

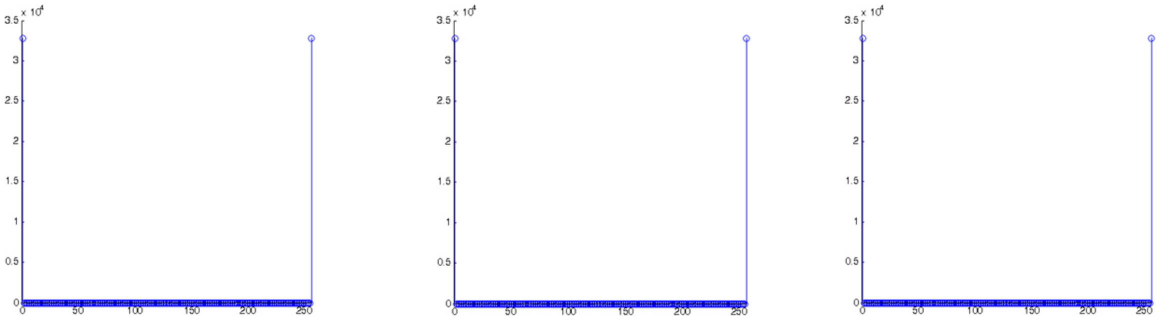


Fig. 5.3. *Global Color Histogram* (*GCH*) of the block pattern, striped pattern and checkerboard pattern (left-to-right) of the Fig. 5.2.

In case of documents, the existing document analysis tools and systems use black-and-white documents for analysis and therefore, the color feature is not much adapted [*Shin et al.*, 2001; *Wong et al.*, 1982; *Wang* and *Srihari*, 1989]. In current applications, the nature of documents is rapidly shifting from black-and-white to complex color documents such as scientific journals, newspapers, magazines, slides, advertisements, etc. For such kind of colorful document images, the color feature should not be ignored since it is one of the most useful features. However, some tools have been developed for extracting the textual content from color documents: color *OCR* [*Garcia* and *Apostolidis*, 2000; *Wu et al.*, 1999] and color string localization [*Messelodi* and *Modena*, 1999; *Sobottka et al.*, 1999; *Chen* and *Chen*, 1998; *Hase et al.*, 1999; *Jain* and *Yu*, 1998]. *Keechul et al.* presented a more detailed survey of text detection and recognition in video frames and images [*Keechul et al.*, 2004]. However, whereas the document analysis for black-and-white documents is mature, color documents analysis is still in its infancy [*Todoran et al.*, 2002].

5.2.4. Content-based slide identification

Content-based linking of presentation/meeting/lecture documents with other media has been tackled in various research projects [*Chiu et al.*, 2000b; *Franklin et al.*, 2000; *Mukhopadhyay* and *Smith*, 1999; *Ozawa et al.*, 2004; *Erol et al.*, 2003]. Such methods extract the document content rather than any layout and color information. The identification is based on either global image matching or character string comparison using *OCR*.

Chiu et al. proposed to link multimedia data automatically with a DCT-based (Discrete Cosine Transform) image matching of the slide content with truncated coefficients [*Chiu et al.*, 2000b]. The method matches the handout of slide images with the full screen images of projected slides captured by an operator and images captured from a video projector displaying slides. The method is based on the down-sampling of the slide images to 64 × 64 pixels, performing a 64-point 2D-DCT, and then using the 256 lowest frequency coefficients to retrieve slide images. Unfortunately, the method is most suitable for matching high-resolution and high quality slide images. The performance may degrade if the images are of low-resolution and not accurately segmented. Partial occlusion or presence of blur also degrades its performance.

Mukhopadhyay and *Smith* proposed a method that matches the slide images from presentations with the low-resolution clipped images from video that includes the presentation slides [*Mukhopadhyay* and *Smith*, 1999]. The method is based on first binarizing and dilating the segmented slide images and clipped video frames to highlight the text regions, and then using the Hausdorff distance to compute the similarity between the text lines. However, it works well only on slides that contain texts and the slide region should be accurately segmented. Furthermore, 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.

Ozawa et al. explained a slide identification method for lecture movies by matching characters and images [*Ozawa et al.*, 2004]. The method uses an *OCR* to recognize the text and an image matching technique (Dynamic Programming) for matching slides extracted from the video with the original slides. Their method performs well with high-resolution images and slides containing texts. The drawback of this approach is that it works only with high-resolution slide images containing texts of larger font size (> 24 points). For low-resolution images, current *OCR* techniques fail and thus, matching becomes incorrect.

Erol et al. proposed the matching of the grabbed slide images from the VGA output of the presenters' laptop/PC with the slide images from the electronic presentation documents such as PDF, PPT [*Erol et al.*, 2003]. The method uses features such as *OCR* output, edges and projection profiles for the matching. The matching is restricted presentation-wise and the use of *OCR* is computationally expensive and requires different system to deal with different languages for real-time application, as well. The method also considers the images from digital cameras. However, the captured images should contain mostly the projected area with a rotation of less than ±5 degrees and at least one text line.

5.2.5. Impediments in using the existing approaches

Most of the document image identification systems use a classifier to identify the incoming unknown document images. Algorithms used for the classification are either supervised (training requires documents with a known class label such as decision tree, neural networks, etc.) or unsupervised (training is based on the features of documents with an unknown class label such as K-means, self-organizing maps, etc.) or semi-supervised (combination of both) [*Yang* and *Ersoy*, 2003; *Jehan*, 2005]. The features for classification are mainly based on the layout structure (physical, logical) and on the content of the documents. In this scenario, the documents are captured using low-resolution capture devices and further compressed with lossy compression such as JPEG. For *e.g.* the projected part of the captured document images is of maximum 715 X 570 pixels. Unfortunately, in such cases, it is difficult to extract the complete layout structure and a clean content (using *OCR*) of the captured documents due to poor resolution. Additionally, many documents have the same layout structure. For example, very often all that the slide images in a particular presentation have the same layout structure with a different content, since they use the same design pattern. This characteristic of slideshows drastically reduces the identification rate of standard methods. Furthermore, applying *OCR* techniques is not an acceptable solution for extracting textual information from low-resolution images and therefore, an alternative method that benefit from the information stored in the original electronic document, is proposed in this thesis.

In the case of the content-based slide identification, the existing approaches are used in order to link the presented slides with the multimodal meeting recordings, by simply considering a slide as a photographic image rather than looking to the document aspect of slide images. Furthermore, such applications are mainly focused on the alignment of slides with the captured multimedia streams. However, if the captured multimodal data is efficiently indexed by considering such documents then it would help in the retrieval of the multimodal information by querying the captured projected documents using handheld devices as well as using general queries expressed in natural languages. Moreover, the applications are limited to projected documents rather than other existing paper documents such as articles, magazines, etc, which are often used in multimodal environment.

5.3. Our proposed approach: document *Visual Signature*

The proposed document identification method uses a signature, which contains shallow abstraction of the document content rather than the document itself in order to symbolize documents. The main advantages of using a shallow representation of documents is that it solves the problems of low-resolution and various distortions to the captured images implied by the handheld devices. Furthermore, the ever-increasing volume of documents that is available, the approach of relying on human-assisted annotations as a means of document abstraction is not feasible. Moreover, people are interested in fast and efficient retrieval of documents that corresponds to a queried document image. Therefore, the signature-based approach is considered which avoids the traditional classifiers. The proposed document *Visual Signature* comprises, mainly two feature-based signatures, namely: (1) *Layout Signature* and (2) *Color Signature*.

The *Layout Signature* is generated by analyzing the document's geometrical layout. The geometrical layout analysis attempts to use basic image properties and spatial relations to extract structure without reference to a particular document type. Such analysis is necessary since the input low-resolution image has no structural description. However, the low-resolution of the capture image does not allow extracting a complete layout structure. The proposed *Layout Signature* is shallow and hierarchically structured representation of the document layout structure with a zone's labeling (text, image, etc.). This *Layout Signature* is nothing but the visual perception of the document image. The matching of the *Layout Signature* follows the hierarchical structures of the signature and does not visit the entire search tree. The highest-level features stored in the signature are first compared and very bad solutions are removed from the search space. The comparison continues with the lower-level features and so on until the leaves level is reached. This method is fast, mainly because the signature hierarchy guides the search towards fruitful solution spaces. Furthermore, by alternating feature-specific comparison with global distance comparison, it guaranties that the good solutions are not avoided. The extraction of features, their hierarchical structuring to generate the *Layout Signature* and its matching is explained in detail in Chapter 6.

The *Color Signature* is computed by considering the features from the normalized GCS and the color distribution in the document geometrical plane *i.e.* in document's 2D image plane rather than in any color space such as RGB, HSV etc. The detailed extraction of the features, their structuring to generate the *Color Signature* and its matching are explained in Chapter 7.

Once the above-mentioned feature-based signatures are computed then the document signature is simply the combination of two signatures mentioned above. The captured document's signature is matched with the signatures in the repository, in order to identify the corresponding captured low-resolution document image. The signatures in the repository correspond to the electronic documents present in a repository.

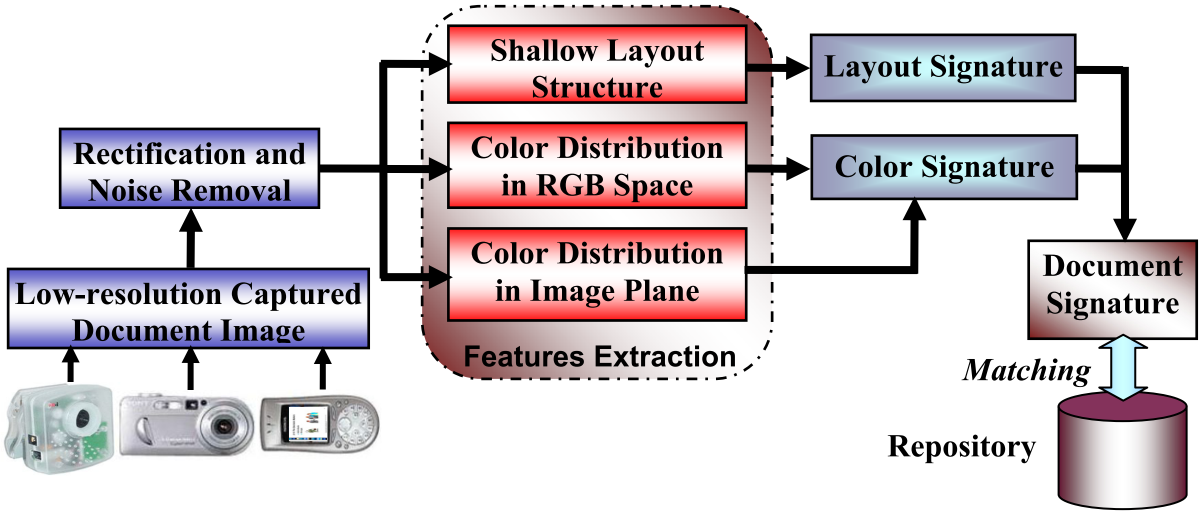

Shallow Layout Structure
Color Distribution in RGB Space
Color Distribution in Image Plane
Features Extraction
Layout Signature
Color Signature
Rectification and Noise Removal
Low-resolution Captured Document Image
Document Signature
Matching
Repository

Fig. 5.4. Low-resolution document identification: a systematic procedure for feature extraction to form the document signature and its matching from the repository.

Fig. 5.4 describes the systematic procedure for the extraction of signatures and their matching with signatures, which are already extracted from the electronic documents and stored in the repository along with the electronic documents. In the first step, the captured documents are pre-processed for the correction of perspective deformations and then low-pass filtered for the noise removal (Section 5.4). Then the pre-processed image is analyzed for the extraction of various features such as shallow layout structure (Chapter 6), color distribution in the RGB color space and in the document's 2-D image plane (Chapter 7) to form the respective layout and *Color Signature*. The document signature combines both above-mentioned signatures. The signatures of the original electronic documents are extracted by considering its Joint Photographic Experts Group (JPEG) image format and the procedure is the same as the above, except that there is no necessity for the rectification and noise removal step. In the following section, rectification and noise removal of captured documents is described.

5.4. Rectification and noise removal of captured documents

The documents captured from the projector's screen using any capture devices not only contain the projected documents but also the surrounding background. Furthermore, the captured documents often suffer geometrical distortions due to the capture environment and to the position of the captured devices. 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. Fig. 5.5 shows some of the captured document images using various low-resolution devices. From the figure, it is clear that often the captured images contain the document as well as the surrounding background and the original rectangular shape of the document appears to be a quadrangle in the captured images. This is due to the perspective distortion. Therefore, one needs to consider the four corners of the quadrangle *ABCD* (anti clock-wise) of the projected part and is to be mapped to a rectangle of common resolution of width, *W* and height, *H* (Fig. 5.6). One approach for such mapping is by using a perspective transform, which maps an arbitrary quadrilateral into another arbitrary quadrilateral while preserving the straightness of lines [*Mukhopadhyay* and *Smith*, 1999].

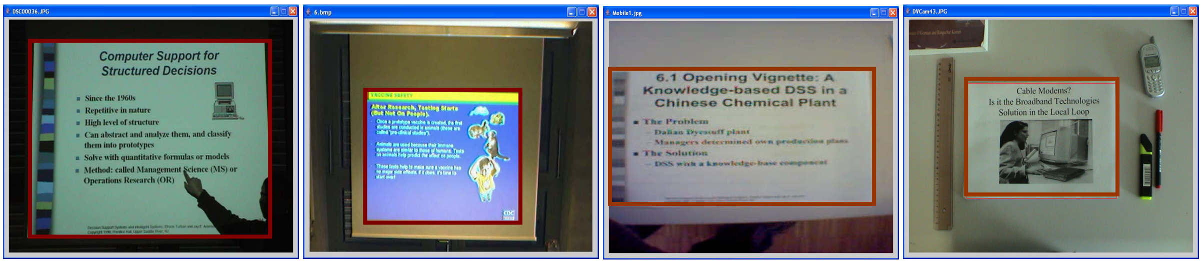

DSC00036.JPG
Computer Support for Structured Decisions
Since the 1960s
Repetitive in nature
High level of structure
Can abstract and analyze them, and classify them into prototypes
Solve with quantitative formulas or models
Method: called Management Science (MS) or Operations Research (OR)
6.bmp
Mobile1.jpg
6.1 Opening Vignette: A Knowledge-based DSS in a Chinese Chemical Plant
The Problem
Dalian Dyestuff plant
Managers determined own production plans
The Solution
DSS with a knowledge-base component
DVCam43.JPG
Cable Modems?
Is it the Broadband Technologies Solution in the Local Loop

Fig. 5.5. Documents captured from a DV camera, a web-cam, a mobile phone and a digital camera (left-to-right). The document zone is highlighted.

Fig. 5.6 illustrates the perspective transform, which transform the quadrilateral shown in the image plane to the rectangle shown on the plane $z = 0$ in the world coordinate system. The idea is, for a given coordinates of the four corners of the quadrilateral (*ABCD*) and the rectangle of height, *H* and width, *W*, compute the perspective transform that maps a new point in the quadrilateral onto the appropriate position on the rectangle. The point *A* of the quadrangle, *ABCD* is mapped to the origin, *B* to $(W, 0)$, C to (W, H), and *D* to $(0, H)$. The above-mentioned perspective transform is expressed as: $Q = M \times P$, where *Q* is a vector of world plane homogeneous destination coordinates, *P* is the vector of homogeneous image plane source coordinates and *M* is a 3×3 homogeneous transformation matrix [*Criminisi* and

Zisserman, 1999]. Since our approach is for the 2D plane so the above-mentioned perspective transform can be written as:

$$\begin{bmatrix} XW \\ YW \\ W \end{bmatrix} = \begin{bmatrix} m_{00} & m_{01} & m_{02} \\ m_{10} & m_{11} & m_{12} \\ m_{20} & m_{21} & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} m_{00}x + m_{01}y + m_{02} \\ m_{10}x + m_{11}y + m_{12} \\ m_{20}x + m_{21}y + 1 \end{bmatrix} \tag{5.1}$$

From the above equation $W = m_{20}x + m_{21}y + 1$ *i.e.* $W = \begin{bmatrix} m_{20} & m_{21} & 1 \end{bmatrix}\begin{bmatrix} x & y & 1 \end{bmatrix}^T$. The above equation could be rewritten as:

$$\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \frac{\begin{bmatrix} m_{00} & m_{01} & m_{02} \\ m_{10} & m_{11} & m_{12} \\ m_{20} & m_{21} & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}{\begin{bmatrix} m_{20} & m_{21} & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}} \Rightarrow X = \frac{m_{00}x + m_{01}y + m_{02}}{m_{20}x + m_{21}y + 1} \text{ and } Y = \frac{m_{10}x + m_{11}y + m_{12}}{m_{20}x + m_{21}y + 1} \tag{5.2}$$

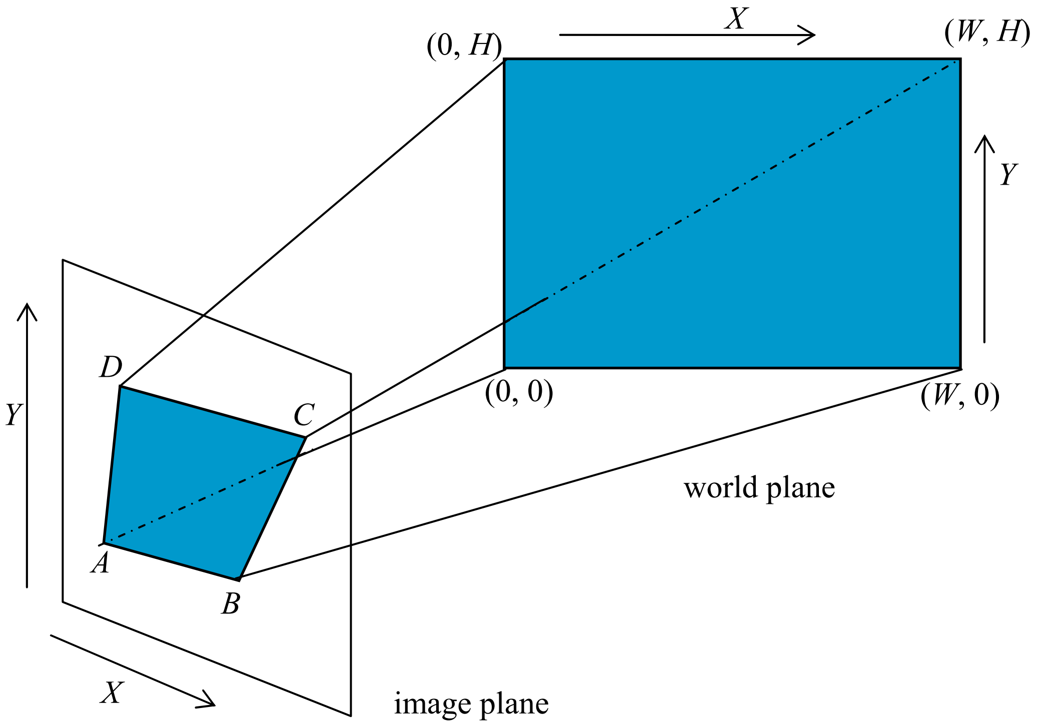


Fig. 5.6. Transformation of a quadrilateral from an image plane to world plane of world coordinate system with z = 0.

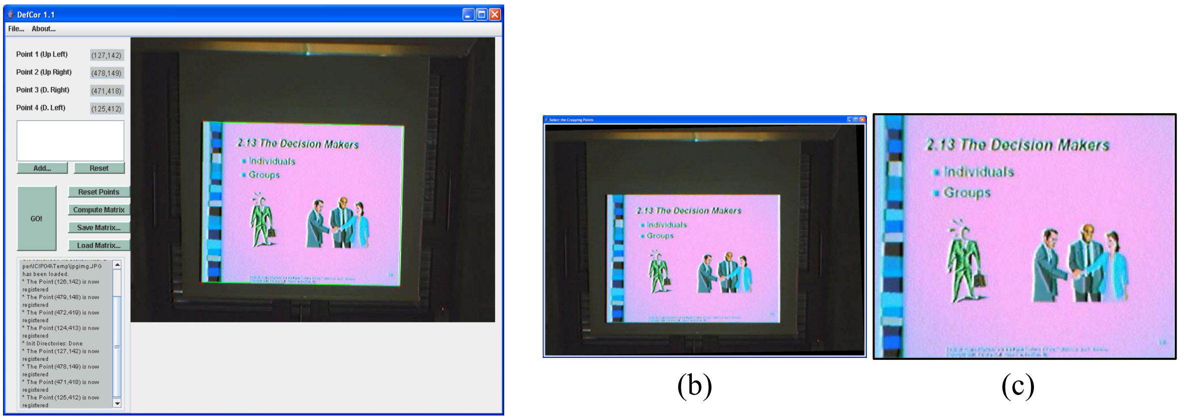
From the expression above, it is clear that the above-mentioned perspective transform is non-linear, where numerator supplies the parameter needed for affine transform, which is the linear combination of translation, rotation, scaling and/or shearing (*i.e.* non-uniform scaling in some directions) operations and the denominator supply the non-linear effect. Once the coordinates in the image plane (quadrangle) and the corresponding four coordinates in the world plane (rectangle) are defined the homogeneous matrix, *M* is computed for the perspective transformation then using these coordinates. The computation of the matrix, *M* is calculated using equation (5.2), which is straight forward but tedious. Eventually, the color (pixel values) in the rectangular image is computed through bilinear interpolation from the original image.

5.4.1. Detection of the quadrangle containing projected part

The detection of the corners of the projected part is done by using two different methods. (a) An interactive Graphical User Interface (GUI) has been developed for the manual selection of the corners of the projected documents. (b) Automatic detection of the quadrangle containing projected part in the captured document image.

In the manual selection of the corners of the projected document, the user selects the points on the captured document image by using the mouse, thus the procedure requires the interaction from the user. This method is mainly applicable for the documents which are captured using fixed capture devices. Recall from the Chapter 3, projected documents are captured as a video stream using a web-cam, which is fixed. Therefore, the selection of the corners of the quadrangle of the projected part and then the computation of the transformation matrix, *M* is done only once. The same quadrangle and transformation matrix, *M* are used for the calibration of all the document images extracted from the video stream. As it is mentioned in Chapter 4, the extracted document images from the video stream are identified from the repository containing the original documents, which are to be included with the captured audio-visual streams of the meetings. The manual selection of the four corners of the quadrangle is not only fast but also avoids any error as in case of the automatic detection. Fig. 5.7 illustrates the GUI, which is designed for novice for easy use as one has to click mouse on the four corners of the projected document.

The interactive graphical interface is comfortable for the detection of quadrangle in the captured document images from a fixed device as the interface has to be used only once. However, in case of the mobile devices such as digital cameras, mobile phones, etc. one has to use the interface for the corner detection of the quadrangle for each captured document. This is due to the fact that the device is not fixed at a particular location and in case of zoom in/out during capture. This results in the variation of the size and location of the quadrangle in the captured image. Therefore, to overcome the repetitive manual selection, an automatic detection of the quadrangle is proposed.



Point 1 (Up Left) (127,142)
Point 2 (Up Right) (478,149)
Point 3 (D. Right) (471,418)
Point 4 (D. Left) (125,412)

```
per\ICIP04\Temp\jpgimg.JPG
has been loaded.
* The Point (126,142) is now
registered
* The Point (479,148) is now
registered
* The Point (472,419) is now
registered
* The Point (124,413) is now
registered
* Init Directories: Done
* The Point (127,142) is now
registered
* The Point (478,149) is now
registered
* The Point (471,418) is now
registered
* The Point (125,412) is now
registered
```

(a) (b) (c)

Fig. 5.7. Deformation correction using perspective transformation (a) captured image loaded in application for correction, (b) after perspective correction, (c) background removal by cropping.

In the case of automatic detection, there is no necessity of the user interaction for the selection of the corners of the quadrangle of the projected part in the captured low-resolution document images. The system locates the corners automatically by processing the captured low-resolution documents. This automatic detection could also be used for the captured image from the fixed devices. However, the efficiency of detection of the quadrangle is far from being perfect as compared to the manual selection. It is mentioned earlier that for the fixed devices, the calibration is done only once. During the process, if there is an error in the automatic quadrangle detection then the calibration of the rest of the captured images would be erroneous. Therefore, in case of fixed devices, the detection of the corners of the quadrangle for the calibration is more appropriate using GUI-based approach without taking any risks.

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 using GUI-based approach. Nevertheless, in case of the automatic detection of the quadrangle, the error is quite rare.

The procedure for the automatic detection of the quadrangle in the captured document images is based on the Hough transform [*Jain*, 1997]. The procedure consists of the following steps: (1) Edge detection; (2) Straight lines detection and (3) Quadrangle formation.

5.4.1.1. Edge detection

An edge is the boundary between two regions in an image with relatively distinct gray-level properties [*Gonzalez* and *Woods*, 2002; *Jain*, 1997; *Lim*, 1990]. Due to the presence of noise and the ambiguity of 'distinct' gray-level pixel value, edge detection is a complex task. In 2D images, an edge is specified by its magnitude and its direction. Edge detection is often carried out by spatial derivation and thresholding. Fig. 5.8 shows an edge and its first and second order derivatives.

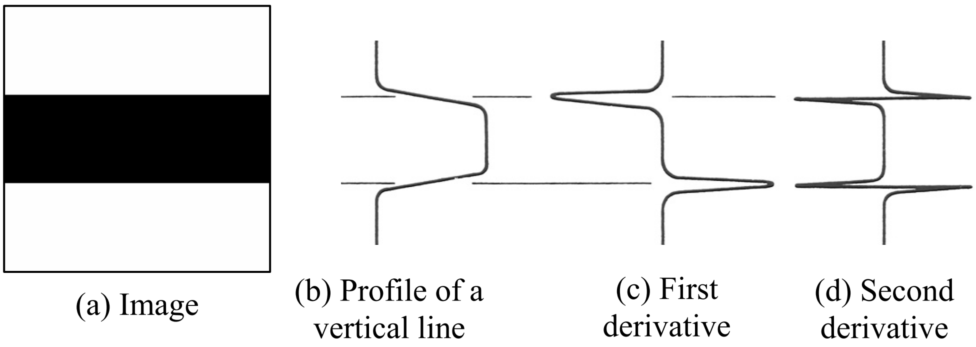


Fig. 5.8. Edge detection by derivative operator: (a) an image containing a dark strip on a light background, (b) vertical profile of a line in the image and its (c) first order derivative and (d) second order derivative [*Gonzalez* and *Woods*, 2002].

For a given image, $I(x, y)$ and its first-order derivative at location (x, y) is the gradient vector and is defined as:

$$\nabla I = \left(G_x, G_y\right) = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right) \tag{5.3}$$

It is well-known from the vector analysis that the gradient vector points in the direction of maximum rate of change of pixel value of I at (x, y). Therefore, in edge detection the edge strength is the magnitude and angle, θ represents the direction angle of the gradient vector ∇I at (x, y). From vector analysis, the magnitude and angle is defined as:

$$|\nabla I| = \left(G_x^2 + G_y^2\right)^{\frac{1}{2}} \text{ and } \theta(x, y) = tan^{-1}\left(\frac{G_y}{G_x}\right) \tag{5.4}$$

The angle is measured with respect to the x-axis. For a digital image, the gradient of the image is based on computing the partial derivatives, G_x and G_y at every pixel location. The gradient could be approximated using a difference operation and is called *masks*, which represent finite difference approximations of the orthogonal gradients, G_x and G_y [*Gonzalez* and *Woods*, 2002; *Jain*, 1997]. For example, the Sobel gradient operators in the literature above, for the edge detection are given as:

$$G_x = I \times \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \text{ and } G_y = I \times \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{5.5}$$

The Sobel operator at location (x, y) is computed by summing the intensity value of the neighboring pixels located at (x, y) using the corresponding weight of the gradient operator. For example, the gradient operator, G_x at (x, y) is reformulated as:

$$\begin{aligned} G_x = & -I(x-1, y-1) - 2I(x, y-1) - I(x+1, y-1) + 0 + 0 + 0 \\ & + I(x-1, y+1) + 2I(x, y+1) + I(x+1, y+1) \end{aligned} \tag{5.6}$$

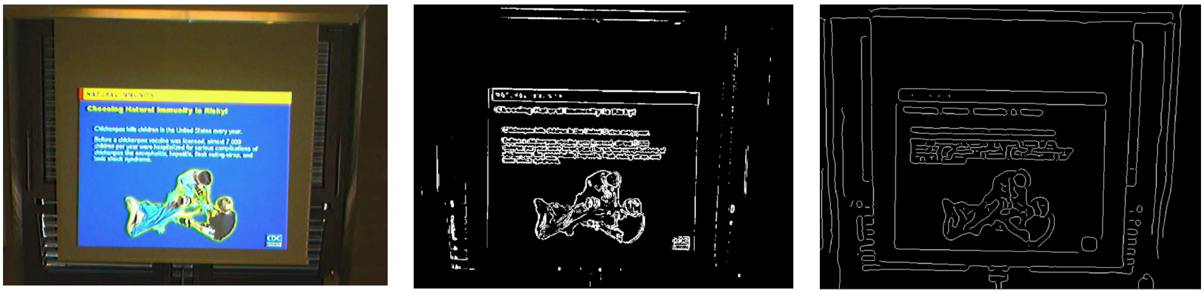
The above-mentioned computation gives a value of the gradient at the location of the center of the *masks*, which is placed over the pixel located at (x, y). The same procedure is used for the computation of vertical gradient, G_y. In order to get the next gradient value, the *masks* are moved to the next pixel location and the procedure is repeated. Edges are computed by using the local maxima of the gradient magnitude, $|\nabla I|$ (Fig. 5.8).

The above-mentioned gradient operators work best when the gray-level transition is quite abrupt *i.e.* as in a step function. As the transition region gets wider, it is more advantageous to use the second-order derivatives [*Jain*, 1997]. The second-order derivative of the given image, $I(x, y)$ is defined as:

$$\nabla^2 I = \left(G_x^{'}, G_{xy}^{'}, G_y^{'}\right) = \left(\frac{\partial^2 I}{\partial^2 x}, \frac{\partial^2 I}{\partial x \partial y}, \frac{\partial^2 I}{\partial^2 y}\right) \tag{5.7}$$

Using the second-order derivatives, the detected potential edges are zero-crossings, where the value of the derivatives changes sign as illustrated in Fig. 5.8d. Second-order derivatives are more sensitive to noise than those of first-order derivatives. However, one advantage of using zero-crossing for detecting edges is that it is easier to locate the zero-crossings than to locate the maximum gradient points with a threshold. The isotropic zero-crossing based edge operator is a direction-invariant Laplacian operator. In order to achieve stable edge detection in the presence of noise, *Marr* and *Hildreth* suggest smoothing an image with Gaussian smoothers before applying the Laplacian operator [*Marr* and *Hildreth*, 1980]. The resulting operator is referred to as the Laplacian of Gaussian (LOG) operator, which could also be implemented by difference of Gaussian (DOG).

The Canny edge detection algorithm is known to many as the optimal edge detector [*Canny*, 1986]. The algorithm uses both the first and second-order derivatives for the edge detection. Zero-crossings of the second-order derivative are detected along a line in the gradient direction, which should also have high gradient magnitude values. In this scenario *i.e.* in order to detect the edges in the low-resolution captured documents, the Canny edge detection algorithm is used. The Canny operator works in a multi-stage process. First of all the gray-scale image is smoothed by using Gaussian mask to eliminate noise. Then the smoothed image is processed to find the image gradient to highlight regions with high first-order spatial derivatives. Edges give rise to ridges in the gradient magnitude image. The algorithm, then tracks along these ridges and sets to zero all pixels that are not actually on the ridge top so as to give a thin line in the output, a process known as non-maximal suppression. The gradient array is now further reduced by hysteresis. Hysteresis is used to track along the remaining pixels that have not been suppressed. Hysteresis uses two thresholds, T_1 and T_2 with $T_1 > T_2$. Tracking could only start at a point on a ridge above T_1 and then continues in both directions from that point until the ridge falls bellow T_2. The hysteresis helps to ensure that noisy edges are not broken up into multiple fragments. The algorithm uses three parameters: σ, width of Gaussian mask used for smoothing, threshold T_1 and T_2. In this scenario, the parameters are set as $\sigma = 3$, $T_1 = 0.09$ and $T_2 = 0.01$ for the grayscale images. An example of the Canny edge detection algorithm is shown in Fig. 5.9.



a) Captured image from a web-cam

b) Edge image using Sobel gradient operators

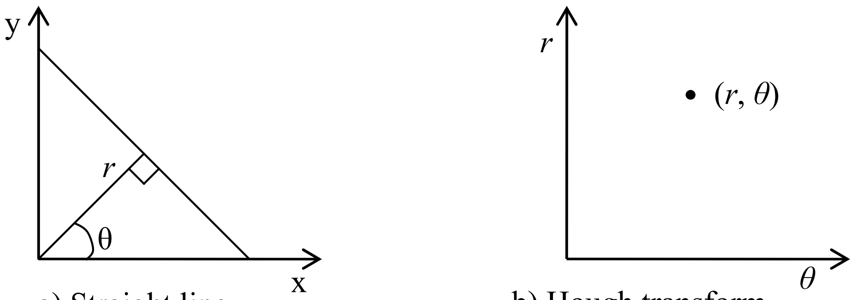
b) Edge image using Canny algorithm

Fig. 5.9. Edge image of a captured document image using Sobel operators and Canny edge detection algorithm.

5.4.1.2. Straight lines detection

Once the edge image is computed then it is processed for the detection of straight lines. Hough transform is a robust technique and more often used for the detection of straight lines [*Jain*, 1997]. The Hough transform of a line at a distance, r and orientation, θ could be represented as a point in the (r, θ) plane (Fig. 5.10).

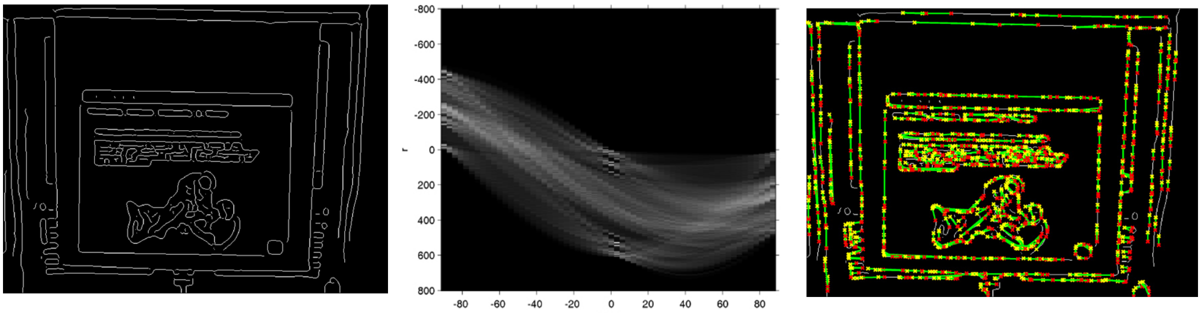
$$r = x\,cos(\theta) + y\,sin(\theta) \tag{5.8}$$



a) Straight line b) Hough transform

Fig. 5.10. The Hough transform of a line in X-Y plane to a point in R-θ plane [*Jain*, 1997].

This transform is used to detect straight lines in a given set of points. In the previous method of edge detection, the edge image is computed from a given image with the corresponding gradient vector. Therefore, the gradient direction $\theta(x, y)$ and its location (x, y) are known from the gradient vector, ∇I. By using equation (5.8), each gradient point, (x, y) in the edge image is projected in the Hough-space (r, θ) with a certain degree of tolerance. The gradient points forming a straight line should refer as close as possible to a point in the $r\theta$-space. Therefore, the idea is to subdivide the $r\theta$-space into reasonable smaller cells. At the end, cells that receive a significant number of hits represent lines that have strong edge support in the image. The orientation information is used in later stage for forming a reasonable quadrangle. In this implementation, the size of each cell in the $r\theta$-space is defined as 5 pixels by 3 degrees. Fig. 5.11b illustrates the Hough space *i.e.* $r\theta$-space of the edge image, which is computed using the Canny edge detection algorithm from the captured image in Fig. 5.9.



(a) Canny edge image (b) Hough space (r, θ) (c) Detected straight lines from the Hough space

Fig. 5.11. Detection of straight lines from the edge image of the captured image in Fig. 5.9a.

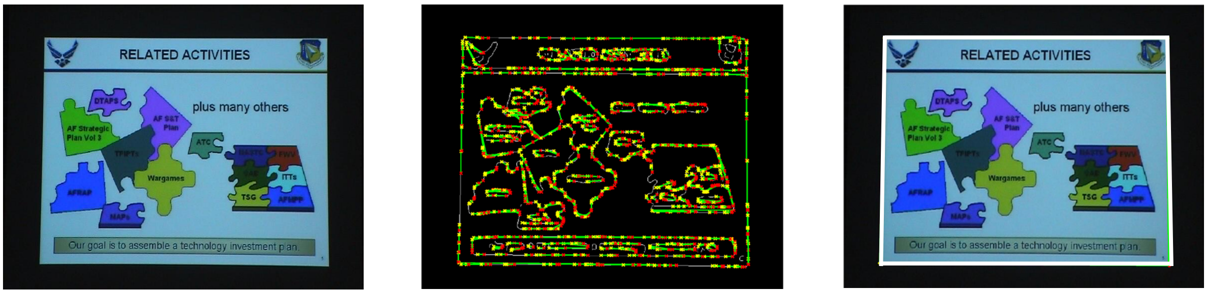
5.4.1.3. Quadrangle formation

The number of hits for each cell in the $r\theta$-space is considered for the high edge concentrations. The local maxima, whose hits are higher than 20% of the maximum number of hits in the $r\theta$-space, are considered. This is decided, in order to filter down a reasonable number of lines rather than considering all lines, which could increase the computational cost. Fig. 5.11c shows the detected straight lines from the $r\theta$-space by considering the cells having hits higher than 20% of the maximum number of hits in the $r\theta$-space. These lines are used to form the reasonable quadrangle. For each line, the starting point (x_1, y_1) and the end point (x_2, y_2) in the 2D image plane are computed from the corresponding local maxima cell in the $r\theta$-space. Often, it is observed that larger straight lines are broken into small pieces and are separated with a distance of few pixels. However, these small straight lines have close angle of orientation. First of all these straight lines are scanned. If two lines, L_1 and L_2 having the difference in orientation angle lower than 5 degrees *i.e.* $|L_1(\theta) - L_2(\theta)| < 5$ and the distance between the starting point, $L_2(x_1, y_1)$ and the end point, $L_1(x_2, y_2)$ is less than 10 pixels *i.e.* $|L_1(x_2, y_2) - L_2(x_1, y_1)| < 10$, then the lines L_1 and L_2 are merged to form a single line. The same merging procedure is continued until all the detected straight lines are visited. Now, the formation of quadrangles using these lines has to be carried on. Any four lines could form a quadrangle and therefore, the possible number of quadrangles could be high. In order to narrow down to a smaller number, the following conditions are used for the formation of quadrangles.

- The opposite lines should have approximately either the same or the opposite orientations *i.e.* 0 or 180 degrees and the difference in orientation should be less than 30 degrees. This is due to the fact that often the opposite lines are not perfectly parallel.
- The distance between opposite lines should have at least ¼th of the image width, W or height; H *i.e.* the difference in their corresponding r is higher than $W/4$ or $H/4$.
- The difference in orientation between two neighboring lines should be close to ±90 degrees and the difference in orientation should be less than thirty degrees.
- The circumference of the quadrangle should be higher than the average value of image width, W and height, H.

Once quadrangles are formed using the above-mentioned procedure, the largest quadrangle is considered if there is more than one quadrangle. The output of the above-mentioned merging procedure for the automatic detection of quadrangle is shown in Fig.

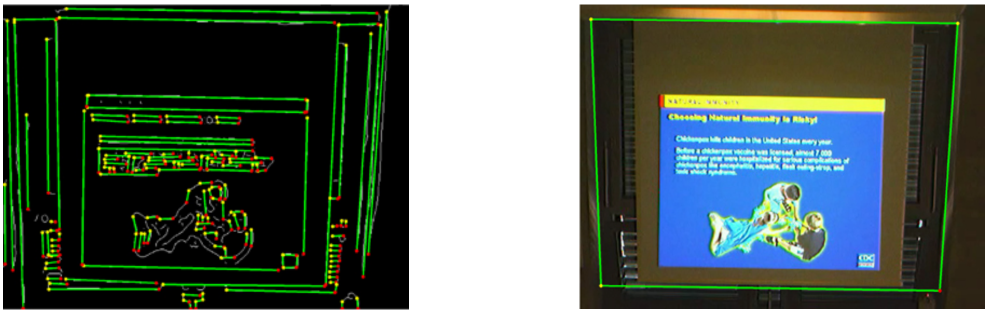
5.12. Often, the detected lines from Hough space are not very accurate. This is due to the discretization of Hough space and in most cases; it results in lines that are not properly connected at the corners of the quadrangle. This is simply handled by using line fitting in each side by considering all the edges in the neighborhood of 10 pixels and having similar orientation to the side.



(a) (b) (c)

Fig. 5.12. Automatic detection of boundary of projected part: (a) an image captured using handheld digital camera, (b) detected straight lines from Hough space, (c) derived final quadrangle from merged straight lines.

An incorrect detection of the final quadrangle after the processing of the image captured using a web-cam (Fig. 5.9a) is presented in Fig. 5.13. The detected quadrangle is the largest possible quadrangle, which surrounds the projector's screen rather than the projected part of the document. This is due to the straight lines detected at the ridge of the projector's screen that is captured as surrounding background of the projected document in the captured document image.



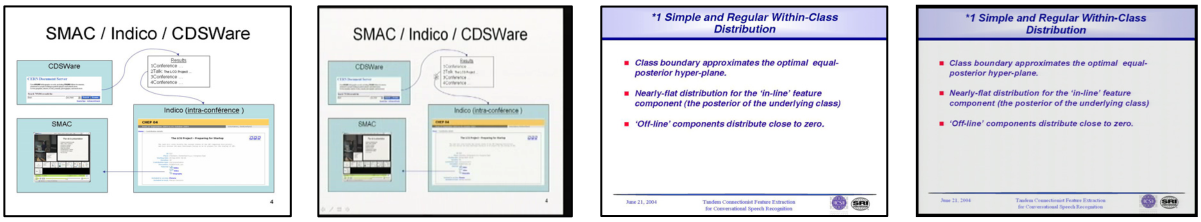
(a) (b)

Fig 5.13. (a) Merging of detected Hough lines of Fig. 5.11c to form longer straight lines and (b) the detected final quadrangle from these straight lines.

5.4.1.4. Evaluation of automatic quadrangle detection

The automatic detection of the quadrangle containing the projected part has been evaluated using projected documents, which are captured using a digital camera and a captured card that capture images from the output of the projector. The capture card has been used in the *SMAC* project [*URL1*; *URL17*]. It is mentioned earlier in Chapter 3 that the document-based automatic indexing system has been used in the *SMAC* project. In this project, the projected documents are captured as a video stream, which is taken from the output of the projector using the capture card. Then the captured document video is processed using the described *SCD* algorithm in Chapter 4. One image per projected document is extracted from the corresponding stable period of the video (Chapter 4) and processed for its identification from the repository containing original documents. Often, a boundary surrounding the projected document is superimposed by the capture card during capturing. The addition of the boundary is not uniform *i.e.* it varies from one slideshow to another, though the slides that belong to a particular slideshow have the same boundary (Fig. 5.14). Therefore, the automatic detection of the quadrangle is used to avoid the repetitive use of interactive GUI for the quadrangle selection for each slideshow. Moreover, the quadrangle in the captured images from the capture card is a rectangle and therefore, in such images the perspective correction is not required. If one considers the captured images from the capture card in Fig. 5.14, it seems that the superimposed boundary is of black pixels and it could be removed by a simple scan. However, this is not the case as those pixels in the boundary are not of the same value and are close to black in color. Therefore, the detection of rectangle is preferred rather than scanning. Furthermore, one constraint is applied to the images captured using a capture card. The constraint is based on the observation that the width of the superimposed boundary is often less than 50 pixels for the captured image resolution of 352 × 288 pixels. Therefore, the straight lines that belong to the regions towards the boundary of the captured image are looked for. The size of the region is restricted to the 1/10th of the respective width and height of the captured images. If straight lines are found in a given image, then it is considered for the detection of rectangle containing the projected document otherwise, it is treated as no superimposed boundary.

A total of 30 images are evaluated and out of which 50% are from the capture card and the rest are captured using digital camera of 1 mega pixel (Sony, DCR-TRV27E). In the case of the images from the capture card, the success rate is 100%, as expected since the projected part is already in rectangular form rather than quadrangle. In the case of the images captured using the digital camera, the success rate is 87%. The low performance is due to the fact that often the detected quadrangle is the boundary of the projector screen rather than the boundary of the projected part of the document on the screen. One of such incorrect detection of quadrangle in the captured image from a web-cam is shown in Fig. 5.13b. This is mainly due to the fact that the proposed technique considers the largest possible quadrangle in the captured document images. These detected quadrangles would include the boundary of the projector's screen if captured in the images.


SMAC / Indico / CDSWare

SMAC / Indico / CDSWare

*1 Simple and Regular Within-Class Distribution

- Class boundary approximates the optimal equal-posterior hyper-plane.
- Nearly-flat distribution for the 'in-line' feature component (the posterior of the underlying class)
- 'Off-line' components distribute close to zero.

June 21, 2004 Tandem Connectionist Feature Extraction for Conversational Speech Recognition

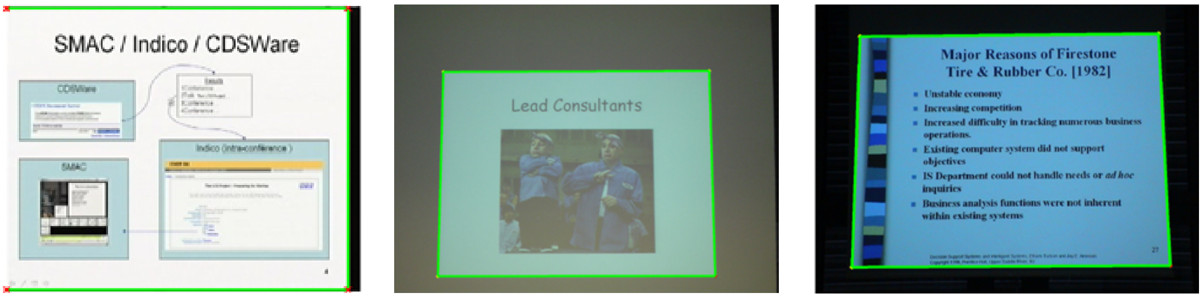
*1 Simple and Regular Within-Class Distribution

- Class boundary approximates the optimal equal-posterior hyper-plane.
- Nearly-flat distribution for the 'in-line' feature component (the posterior of the underlying class)
- 'Off-line' components distribute close to zero.

Original slide document | Corresponding output of the capture card | Original slide document | Corresponding output of the capture card

Fig. 5.14. Non-uniform boundary added to the captured document from the capture card.

The automatic detection of the quadrangle containing the projected document in the captured image from the various capture devices is illustrated in Fig. 5.15. Once the quadrangle is identified then the four corners of the quadrangle is considered for the perspective correction as described in the first part of this section. The result of the perspective correction of the respective image in the Figs. 5.15b and 5.15c is shown in the Fig. 5.16. It is mentioned earlier that the captured images from the capture card do not need the perspective correction as it is already in the rectangular shape. The computation time for the automatic detection of projected part is rather short *i.e.* within 5 seconds.

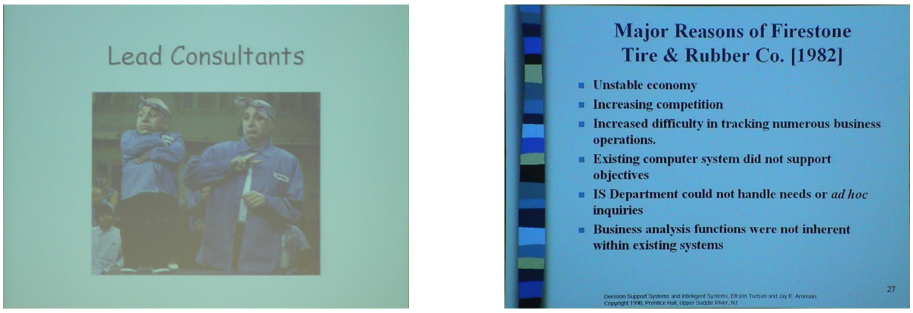

SMAC / Indico / CDSWare

Major Reasons of Firestone Tire & Rubber Co. [1982]

- Unstable economy
- Increasing competition
- Increased difficulty in tracking numerous business operations.
- Existing computer system did not support objectives
- IS Department could not handle needs or *ad hoc* inquiries
- Business analysis functions were not inherent within existing systems

(a) (b) (c)

Fig. 5.15. Detection of quadrangle containing the projected document in the captured images from (a) capture card, (b) digital camera and (c) DV-camera.



Lead Consultants

Major Reasons of Firestone Tire & Rubber Co. [1982]

- Unstable economy
- Increasing competition
- Increased difficulty in tracking numerous business operations.
- Existing computer system did not support objectives
- IS Department could not handle needs or *ad hoc* inquiries
- Business analysis functions were not inherent within existing systems

Decision Support Systems and Intelligent Systems, Efraim Turban and Jay E. Aronson
Copyright 1998, Prentice Hall, Upper Saddle River, NJ

27

Fig. 5.16. Result of the perspective correction to the captured image in the Figs. 5.15b and 5.15c.

5.4.2. Noise removal

Finally, the noise in the rectified image is removed using low-pass *Wiener* filter, which is applied to each of the RGB-channel [*Jain*, 1997]. *Wiener* filtering is a method of restoring image in the presence of blur as well as noise. The *Wiener* filter uses a pixel-wise adaptive method and is based on statistics estimated from a local neighborhood of each pixel. For a given image, I and neighborhood size of $M \times N$, *Wiener* method estimates the local image mean, $\mu(x, y)$ and standard deviation, $\sigma(x, y)$ at location (x, y) is defined as [*Lim*, 1990]:

$$\begin{aligned} \mu(x,y) &= \frac{1}{M \times N} \sum_{i=x-M/2}^{x+M/2} \sum_{j=y-N/2}^{y+N/2} I(i,j) \\ \sigma^2(x,y) &= \frac{1}{M \times N} \sum_{i=x-M/2}^{x+M/2} \sum_{j=y-N/2}^{y+N/2} (I^2(i,j) - \mu^2(x,y)) \end{aligned} \tag{5.9}$$

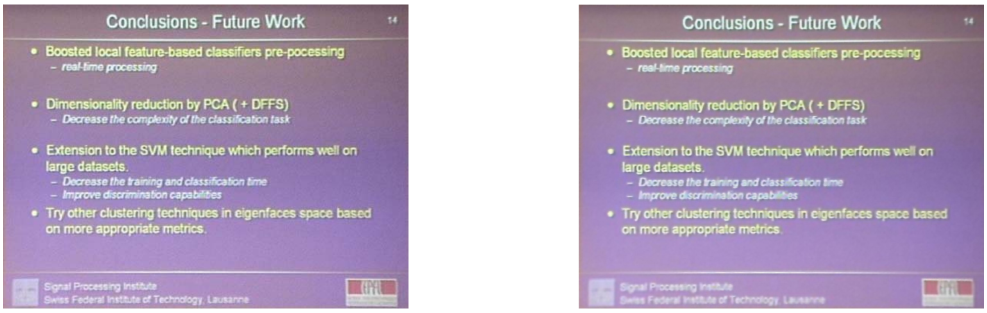
where $I(x, y)$ represents the pixel value at location (x, y) and using the above-mentioned estimated mean and standard deviation, the pixel-wise *Wiener* filtering is created as follows:

$$\begin{aligned} &W(i,j) = \mu(x,y) + \frac{\sigma^2(x,y) - v^2}{\sigma^2(x,y)} (I(i,j) - \mu(x,y)) \\ &\text{Where } x - M/2 \le i \le x + M/2 \text{ and } y - N/2 \le j \le y + N/2 \end{aligned} \tag{5.10}$$

where, v^2 is the noise variance. The type of noise in the perceived captured document images is unknown. Therefore, the noise variance, v^2 is assumed by considering the average of the local estimated variances.

$$v^2 = \frac{1}{M \times N} \sum_{i=x-M/2}^{x+M/2} \sum_{j=y-N/2}^{y+N/2} \sigma^2(i,j) \tag{5.11}$$

Considering the computational complexity and the noise level in the captured document images, the neighborhood size for the *Wiener* filtering procedure is retained as $M = N = 5$. This optimal neighborhood size is considered by looking at the blurring effect in the captured low-resolution images. More than 100 images have been processed using the neighborhood size of range from 2 to 10 and the size 5 has been retained. The proposed noise removal algorithm is illustrated in Fig. 5.17. Once the filtering is done, the resulting image is processed for the extraction of the features for both signatures.



Conclusions - Future Work 14

- Boosted local feature-based classifiers pre-pocessing
  - *real-time processing*
- Dimensionality reduction by PCA (+ DFFS)
  - *Decrease the complexity of the classification task*
- Extension to the SVM technique which performs well on large datasets.
  - *Decrease the training and classification time*
  - *Improve discrimination capabilities*
- Try other clustering techniques in eigenfaces space based on more appropriate metrics.

Signal Processing Institute
Swiss Federal Institute of Technology, Lausanne

Conclusions - Future Work 14

- Boosted local feature-based classifiers pre-pocessing
  - *real-time processing*
- Dimensionality reduction by PCA (+ DFFS)
  - *Decrease the complexity of the classification task*
- Extension to the SVM technique which performs well on large datasets.
  - *Decrease the training and classification time*
  - *Improve discrimination capabilities*
- Try other clustering techniques in eigenfaces space based on more appropriate metrics.

Signal Processing Institute
Swiss Federal Institute of Technology, Lausanne

(a) (b)

Fig. 5.17 Noise removal from low-resolution documents: (a) rectified perspective distortion of a captured slide image and (b) removal of noise using low-pass *Wiener* filter.

5.5. Conclusions and perspectives

In this chapter, the identification of captured documents from various low-resolution handheld capture devices has been discussed. Various existing approaches using document image features such as layout, texture, color, etc. have been described. The proposed identification approach uses the abstracted content of the document image rather than image itself. The abstracted content describes a document image by using features from the shallow layout structure, global color content and geometrical color distribution. The features representing abstracted content are hierarchically structured to form a document signature for fast and efficient matching of the document images. The procedures for the extraction of the above-mentioned features are described in details in the Chapter 6 and 7, respectively.

The detailed procedure for the perspective correction of the captured document images has also been presented. The perspective distortion arises due to the positioning of the target objects and the capture devices. Due to this perspective distortion, the projected rectangular documents often appear as a quadrangle, which is projected into rectangle in the 2-D world plane using perspective transform. The automatic method for the detection of the quadrangle in the captured document images could be extended for the automatic detection of documents in an image *e.g.* a printed documents laying on a table or hanging on a wall.

Furthermore, the noise in the captured document images is removed using the *Wiener* filtering method. Once the document is cleaned and rectified, the extraction of the various features to represent the corresponding signature can be performed. Next chapter describes the procedure for the extraction of these features.

Chapter 6

Document *Layout Signature*

Document images are different from natural images because they contain mainly text with a few graphics and images. Due to the very low-resolution of images of those captured using handheld devices, it is hard to extract the complete layout structure (logical or physical) of the documents and even worse to apply standard *OCR* systems. For this reason, a shallow representation of the low-resolution captured document images is proposed. This representation is called shallow layout document structure, which is close to the perception of human vision. This signature is hierarchically structured according to the document's shallow physical structure with its respective labeling (text, graphics, solid bars, etc.). This signature-based approach mainly targets for images (*e.g.* projected slides) captured using handheld mobile devices. The motivation for projected documents with such signatures is that its content is often limited and its layout varies a lot as compared to other type of documents (*e.g.* newspaper, scientific articles, magazines, etc.).

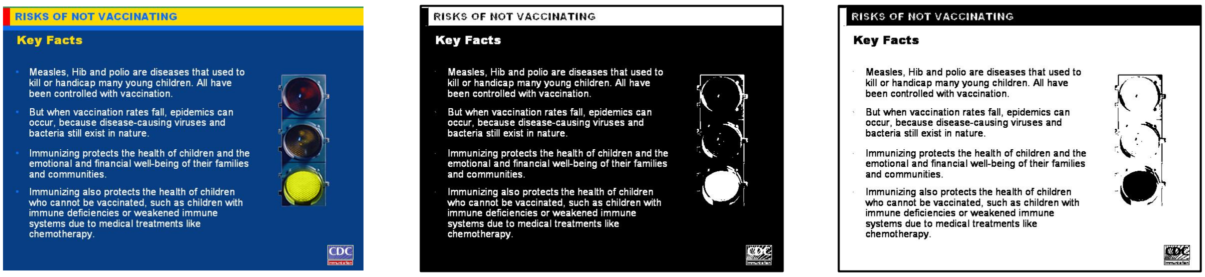
The extraction of the *Layout Signature* follows a top-down approach as described in Chapter 5. It, first considers the full document as a page and partition the page into different blocks (images, text, bars, etc.) and then subsequently, the textual blocks to text lines and then words. Due to poor resolution, it is not feasible to go up to the character level as long as the adjacent characters are overlapped in the captured documents. First, the captured document images are pre-processed for the perspective correction and noise removal as described in Chapter 5. Then, the corrected captured documents are processed for the extraction of the *Layout Signature*. In case of original electronic documents in the repository, the extraction of the same signature is straightforward; the PDF or PowerPoint form of the original electronic documents is converted into a relatively high-resolution image (TIFF, JPEG, etc.) on which the signature is computed. Finally, the captured document's signature is compared to with all the original electronic documents' signatures in order to find a match.

This chapter is organized as follows. Section 6.1 describes in details the extraction of the various layout features, which are organized in a signature called the *Layout Signature*. For fast and efficient matching of the *Layout Signatures*, the structuring of the layout features in the *Layout Signature* is necessary and is explained in the Section 6.2. Section 6.3 presents the matching of the layout features in the *Layout Signature* for the identification of the captured document images from the meeting repository. This chapter concludes in Section 6.4 along with future perspectives.

6.1. Layout features extraction

The proposed document *Layout Signature* is a hierarchically structured description of a document's shallow physical structure with its respective labeling. This signature is used to describe both a) low-resolution images resulting from the capture of projected slides and b) images converted from the original electronic slide documents. Various layout features are extracted and organized in the *Layout Signature* in a structured manner.

First of all, the resolution of each document *i.e.* both the image version of the original electronic documents and the pre-processed captured documents is checked for re-sampling to a common resolution format (720 X 540). If the perceived document image is resolution-wise different than the common resolution, then the re-sampling is required to bring the image up to the common resolution. This is necessary as in this scenario, the geometrical properties of the various visual features are considered during the matching for the identification of the captured document images. Then, the final image is converted to grayscale and binarized using *Otsu* segmentation method for further processing [*Otsu*, 1979]. Furthermore, looking at the mean horizontal run length of both black and white pixels the proper segmentation of foreground objects is checked. For example, for the document images having dark background and light foreground, the output of the binarization is reversed *i.e.* black background (represented as 0's) and white foreground (represented as 1's) (Fig. 6.1).



RISKS OF NOT VACCINATING

Key Facts

- Measles, Hib and polio are diseases that used to kill or handicap many young children. All have been controlled with vaccination.
- But when vaccination rates fall, epidemics can occur, because disease-causing viruses and bacteria still exist in nature.
- Immunizing protects the health of children and the emotional and financial well-being of their families and communities.
- Immunizing also protects the health of children who cannot be vaccinated, such as children with immune deficiencies or weakened immune systems due to medical treatments like chemotherapy.

(a) (b) (c)

Fig. 6.1. *Otsu* segmentation: (a) original slide document, (b) output of *Otsu* segmentation and (c) correction to the *Otsu*'s output using mean horizontal run length of white and black pixels.

The output of the binarization should be black foreground and white background for further computation of the layout features. Therefore, the mean horizontal run length of both black and white pixels in the output of the *Otsu* segmentation is computed. Normally, the mean horizontal run length of the background pixels is much higher than that of foreground pixels. If the mean horizontal run length of the black pixels is comparatively higher than that of the white pixels in the binary images of the output of *Otsu* segmentation then the black and white pixels are simply swapped for the required image. Fig. 6.1 illustrates one of such images having dark background and lighter foreground. For this particular image (Fig. 6.1b), the mean horizontal length of the black pixels is 32.3 and for white pixels it is 6.1. The image in Fig. 6.1b is corrected using this black and white run information for perfect segmentation *i.e.* black foreground and white background (Fig. 6.1c).

6.1.1. Run-Length Smoothing Algorithm

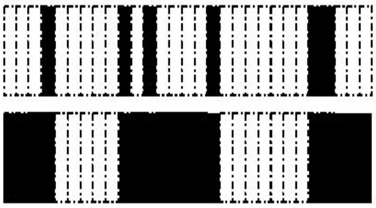
The *RLSA* is applied row-by-row and column-by-column to the above-mentioned binary document images representing white pixels by 1's and black pixels by 0's. The *RLSA* transforms a binary sequence x into an output sequence y according to the rules described by *Wong et al.* as follows [*Wong et al.*, 1982]:

- 1's in x are changed to 0's in y if the number of adjacent 1's is less than a pre-defined limit, T.
- 0's in x are unchanged in y.

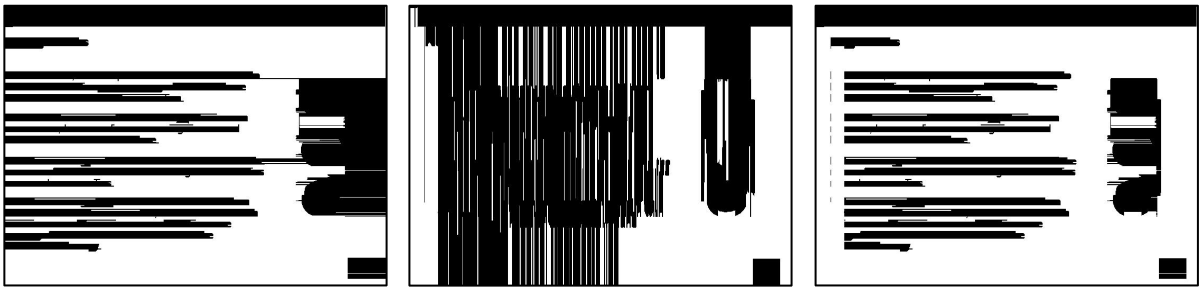
For example, with $T = 5$ the sequence x is mapped into y, which is illustrated below.

$x = (1\,1\,1\,0\,1\,1\,1\,1\,1\,0\,1\,0\,1\,1\,1\,1\,0\,1\,1\,1\,1\,1\,1\,1\,0\,0\,1\,1\,1) \Rightarrow$

$y = (0\,0\,0\,0\,1\,1\,1\,1\,1\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1\,1\,1\,1\,0\,0\,0\,0\,0) \Rightarrow$



The basic idea of the *RLSA* is to connect the neighboring black areas when they are separated by less than T pixels. The degree of connectivity depends on T, the distribution of white and black pixels in the document and the '*dpi*' (dots per inch) resolution of the document. The two distinct bit-maps are generated using the *RLSA* in both horizontal and vertical directions. Often, the spacing between the components in the document image tends to differ horizontally and vertically. Therefore, two different thresholds T_h and T_v are used for the *RLSA* in respective horizontal and vertical direction. For the slide documents, these thresholds are tuned as $T_h = 80$ and $T_v = 100$. The two bit-maps of respective *RLSA* output in horizontal and vertical directions are combined using a logical AND operator to detect various components in the document images. Additional horizontal smoothing using the *RLSA* ($T_s = 15$) produces the final segmentation result. Fig. 6.2 illustrates the *RLSA* algorithm in horizontal, vertical and combining output of the image in Fig. 6.1c. The values of these thresholds have been evaluated and tuned using about a hundred slide images.



(a) (b) (c)

Fig. 6.2. Output of the *RLSA*: (a) horizontal direction, (b) vertical direction and (c) combining both the directions, of the binary image in Fig. 6.1c.

6.1.2. Bounding box extraction

The bounding box containing each component in the final bit-map of the *RLSA* output must be located. The location of each of the bounding box is carried out by applying the connected component analysis in the final bit-map. The region containing all connected black pixels are considered as a single component and the maximum width and height of the region containing the component is the bounding box (rectangle) of that component (block). The bounding box corresponding to each component is detected and must then be labeled according to their content, so that the correct subsequent analysis is further performed (Fig. 6.2c). The labeling of each block is done after extracting the following feature vector [*Wong et al.*, 1982; *Palvidis* and *Zhou*, 1992]:

- Total number of black pixels in each block; its minimum x-y coordinates and its maximum width and height $\left(Y_{\min}, X_{\min}, W_{\max}, H_{\max}\right)$; eccentricity $E = W_{\max} / H_{\max}$ of the rectangle surrounding the block; the mean horizontal length of black runs R_m and the bounding box pixel ratio P of the original data for the block, *i.e.* before running the *RLSA*.
- The average correlation C_1 between adjacent scan lines $C_{1,y}$; the percentage of lines C_2 with $C_{1,y} > 0.8$; the average correlation C_3 between scan lines separated by 10 intervening scan lines $C_{10,y}$. The normalized correlations between scan lines at y and $y + r$ is defined as:

$$C_{r,y} = \frac{1}{L}\sum_{k=0}^{L-1}[1-2p(y,k)\oplus p(y+r,k)] \\ = 1-\frac{2}{L}\sum_{k=0}^{L-1}[p(y,k)\oplus p(y+r,k)] \tag{6.1}$$

L being the number of pixels in a scan line and $p(y, k)$ is the value of the k_{th} pixel in the scan line y using the original data of the block. The selection of 10 scan lines (10 × 75 / 72) is based on the input resolution (75 *dpi*) and on the minimum font size (10 points) used in slideshows.

6.1.3. Bounding box labeling

A unique label is further assigned to each extracted bounding box by considering the previous feature vector. The following rules are applied in order to label the segmented blocks:

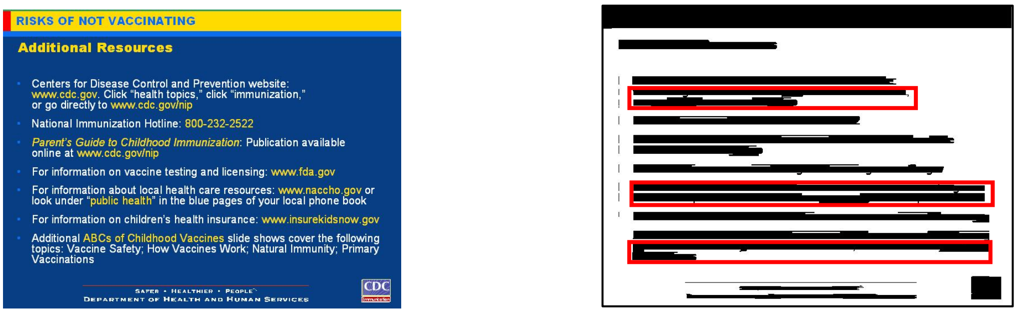
1. Text: $(C_1,C_2) > (1-\omega)$ and $C_3 < (1-\omega)$
2. Horizontal solid lines: $R_m > (1-\omega)\times W_{\max}$ and $E > 1/\omega$
3. Graphics and images: $E > \omega$ and $(C_1,C_2,C_3) < (1-\omega)$
4. Vertical solid lines: $E < \omega$ and $C_{1,2,3} < \omega$
5. Horizontal bar with text: $R_m < (1-\omega)\times W_{\max}$ and $E > 1/\omega$
6. Vertical bar with text: $E < \omega$ and $(C_1,C_2,C_3) > \omega$

The above-mentioned method has been tested on approximately one hundred documents and the parameter, ω has been set empirically to 0.2 with satisfactory performance. The minimum and maximum heights of the text are computed considering the minimum and maximum size of the fonts. For example, the minimum and maximum font sizes for a typical PowerPoint slide image are 8 and 96 points. Then the corresponding MIN_TEXT_HEIGHT and MAX_TEXT_HEIGHT are computed based on the specified minimum and maximum font sizes and the resolution (*dpi*) of the input image. For example, if the input image is 75 dpi, then MIN_TEXT_HEIGHT is 8 pixels (8 x 75 / 72) and MAX_TEXT_HEIGHT is 100 pixels (96 x 75 / 72). Then the corresponding MIN_TEXT_WIDTH and MAX_TEXT_WIDTH are computed from MIN_TEXT_HEIGHT and MAX_TEXT_HEIGHT using default width-height-ratio (7 / 12 for typical Courier font style) [*Shin et al.*, 2001]. Hence, for the configurations above the MIN_TEXT_WIDTH of 5 pixels (8 x 7 / 12) and MAX_TEXT_WIDTH of 58 pixels (100 x 7 / 12) are computed.

Each block is further processed to check whether it contains several blocks. Indeed, logos sometimes appear with text (most of the time in the title), captions are close to images and finally several images are often grouped in a same image block. Most often, they could be separated by considering the average block's height, width and also by passing them through horizontal and vertical projection profiles [*Cattoni et al.*, 1998]. Finally, when the joined blocks are separated into individual blocks, each block is re-processed in order to extract its feature vector (subsection 6.1.2) and labeled accordingly (subsection 6.1.3).

6.1.4. Text lines and words

In the previous section, the textual blocks are labeled as text but there is no further information about the text alignment (*e.g.* horizontal and vertical text lines). In this section, we discuss about the extraction of the feature vector of each text line (horizontal or vertical) for the *Layout Signature*. The feature vector for each text line is $\{Y_{\min}, X_{\min}, H_{\max}, W_{\max}, N_{\text{word}}, R_i(Y_i\ , X_i), P\}$, where N_{word} is the number of words in the text line and $R_i(Y_i\ , X_i)$ is the relative position of word i (i = 1, 2, .., N_{word}) with respect to the bounding box's $Y_{\min}$ and $X_{\min}$. The bounding box containing each text block is considered in the original binary image for the extraction of the text lines and words in each text line.



RISKS OF NOT VACCINATING

Additional Resources

- Centers for Disease Control and Prevention website: www.cdc.gov. Click "health topics," click "immunization," or go directly to www.cdc.gov/nip
- National Immunization Hotline: 800-232-2522
- *Parent's Guide to Childhood Immunization*: Publication available online at www.cdc.gov/nip
- For information on vaccine testing and licensing: www.fda.gov
- For information about local health care resources: www.naccho.gov or look under "public health" in the blue pages of your local phone book
- For information on children's health insurance: www.insurekidsnow.gov
- Additional ABCs of Childhood Vaccines slide shows cover the following topics: Vaccine Safety; How Vaccines Work; Natural Immunity; Primary Vaccinations

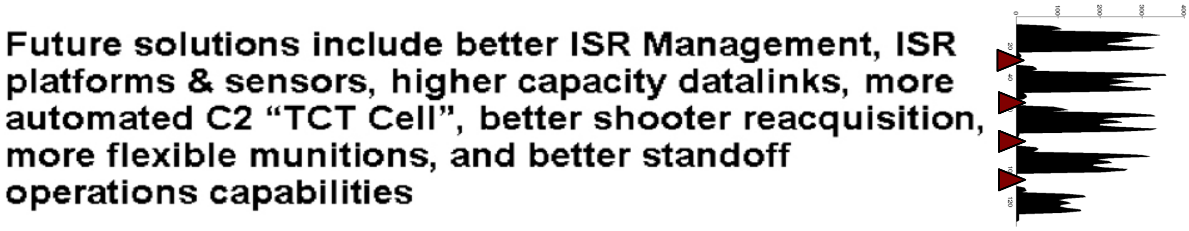
SAFER • HEALTHIER • PEOPLE™
DEPARTMENT OF HEALTH AND HUMAN SERVICES

(a) (b)

Fig. 6.3. (a) One slide document and (b) its corresponding *RLSA* output where two or more text lines are in the same text block component.

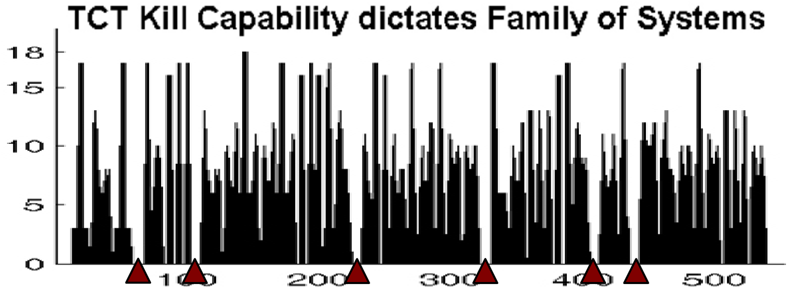
6.1.4.1. Horizontal text lines and words

Often a textual block contains more than one text line. Fig. 6.3 illustrates one of such an image in which, there are three text blocks that contain more than one text line. The maximum height, $H_{\max}$ of such blocks are normally higher than MAX_TEXT_HEIGHT. Such blocks are passed through horizontal projection profile and separated into individual blocks with one text line per block with the property of MIN_TEXT_HEIGHT < $H_{\max}$ < MAX_TEXT_HEIGHT. The horizontal projection profile counts the number of black pixels in the horizontal directions and the peaks in the projection profile correspond to text line as shown in Fig. 6.4. Two peaks are separated by a valley, which represents the space between the two consecutive text lines. The threshold for the line gap is computed adaptively by considering the width of the peaks and valleys in the projection profile. Then the number of words in each text line is computed using the vertical projection profile (Fig. 6.5). The significant valleys in the vertical projection profile show the space between two consecutive words. The threshold for word gap detection is selected by looking at a) the average column gap between connected components, b) the mean horizontal black run length R_m and c) the average height of each text line block [*Cattoni et al.*, 1998].



Future solutions include better ISR Management, ISR platforms & sensors, higher capacity datalinks, more automated C2 "TCT Cell", better shooter reacquisition, more flexible munitions, and better standoff operations capabilities

Fig. 6.4. Horizontal projection profile of a text block to separate individual text lines.



TCT Kill Capability dictates Family of Systems

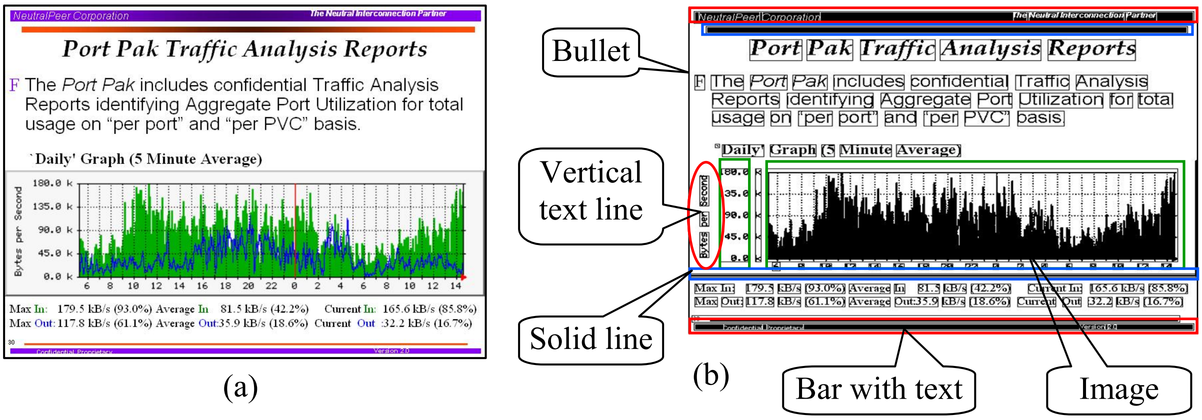
Fig 6.5. Vertical projection profile of a text line for the detection of words.

6.1.4.2. Vertical text lines and words

The vertical text lines are often wrongly segmented as multiple blocks containing one or more connected components in the final output of the *RLSA*. Such blocks are labeled as either a horizontal text or an image with the properties of MIN_TEXT_WIDTH < W_{max} < MAX_TEXT_WIDTH and MIN_TEXT_HEIGHT < H_{max} < MAX_TEXT_HEIGHT. In order to extract the vertical text lines, the vertical alignment of such blocks are looked for. Furthermore, the vertical distance between two consecutive blocks, which belongs to a vertical text line is less than MAX_ROW_GAP. The MAX_ROW_GAP is the maximum of all row-gaps between any two consecutive vertically placed horizontal text lines and is computed after the horizontal projection profile to separate the horizontal text lines from a text block component. If there are no horizontal text lines or only one, then the value is assigned to a pre-fixed threshold value, T_r of 20 pixels for a resolution of 75 *dpi*. For more than one text lines, the row-gaps between two consecutive vertically placed horizontal texts lines are considered. The procedure is illustrated by the following example. Let us consider N horizontal text lines *i.e.* $H = \{h_1, h_2, \ldots, h_N\}$ in an image, which are arranged in increasing order of their vertical locations *i.e.* $Y_{\min}(h_{i+1}) > Y_{\min}(h_{\mathrm{i}})$, where $i = 1 \ldots N - 1$. Then, the row-gap between any two consecutive text lines is computed as: $R_{\mathrm{gap}}(i) = \{Y_{\min}(h_{i+1}) - (Y_{\min}(h_i) + H_{\max}(h_i))\}$, $i = 1 \ldots N - 1$. Those row-gaps, $R_{\mathrm{gap}}(i)$ which are inferior to T_{r} are considered and the MAX_ROW_GAP is assigned as the $max(R_{\mathrm{gap}}(i))$, $i = 1 \ldots N - 1$. The threshold, T_r is considered since the row-gap between the title and the body text in a slide document is often much higher as compared to the row-gap between the horizontal text lines in the body text (Fig. 6.6a). Furthermore, the row-gap between two consecutive blocks containing more than one text line is higher than the row-gap between the horizontal text lines inside the text block (Fig. 6.3). The goal is to consider the row-gaps between the consecutive text lines, which belong to the same text block and in the *RLSA* output, it might be segmented as either individual text blocks or belong to a text block containing more than one horizontal text lines. Once the MAX_ROW_GAP is assigned, the following procedures are applied to detect the vertical text lines, whose connected components are often segmented as separate blocks. Let $B = \{b_1, b_2, .., b_N\}$ be the set of blocks with label of either text or image such that $\forall b_i \in B$:{(MIN_TEXT_WIDTH < $W_{\max}(b_i)$ < MAX_TEXT_WIDTH) AND (MIN_TEXT_ HEIGHT < $H_{\max}(b_i)$ < MAX_TEXT_HEIGHT)}. Any two blocks in B are merged, if and only if they are aligned vertically and the vertical gap between them is inferior to the MAX_ROW_GAP. Each block in set, B is compared with the rest of the blocks and if one or more blocks satisfy the condition above then they are merged *i.e.*

$$
\begin{aligned}
&MERGE(b_i, b_j) \Leftrightarrow (A_{i,j} \wedge B_{i,j} \wedge C_{i,j}) \text{ for } \forall b_i, b_j \in B : i \neq j \\
&\text{Where} \begin{cases} A_{i,j} = |X_{\min}(b_i) \; - \; X_{\min}(b_j)| < C_v \\ B_{i,j} = |X_{\min}(b_i) + W_{\max}(b_i) - X_{\min}(b_j) - W_{\max}(b_j)| < C_v \\ C_{i,j} = |\, Y_{\min}(b_j) - Y_{\min}(b_i) - H_{\max}(b_i)| < MAX\_ROW\_GAP \end{cases}
\end{aligned}
\tag{6.2}
$$

Finally, B consists of zero or more vertical text lines. Each element in B is checked and if labeled as a vertical text line then it is passed through the horizontal projection profile to compute the number of words in the text line and their relative positions as described above (Fig. 6.4). The threshold for the vertical word gap detection is selected by looking at (a) the average row gap; (b) the width and (c) the vertical mean black run length R_v of each vertical text line. Finally, the feature vector for each vertical text line is updated. In our system, the threshold, C_v is set to 5 pixels and it works for most of the slide images. The above-mentioned procedure works for the vertical text lines, which are nearly perpendicular with tolerance of 15 degrees *i.e.* 90 ± 15 degrees. If the bounding box containing text line is neither horizontal nor vertical, then with the proposed text line detection method, such lines are broken into different connected components with the labeling of either text or image. This has been considered according to the real-world slideshow presentation as the horizontal text lines are often used in presentations. The use of vertical text line is rare and is used mostly to describe the vertical axis of figures, drawings, graphics, etc. in the slide documents. Such vertical text lines are of ± 90 degree rotation of horizontal text lines. One of such vertical text line is detected in a slide document image in Fig. 6.6.


NeutralPeer Corporation
The Neutral Interconnection Partner

Port Pak Traffic Analysis Reports

F The *Port Pak* includes confidential Traffic Analysis Reports identifying Aggregate Port Utilization for total usage on “per port” and “per PVC” basis.

`Daily' Graph (5 Minute Average)

Max In: 179.5 kB/s (93.0%) Average In 81.5 kB/s (42.2%) Current In: 165.6 kB/s (85.8%)
Max Out: 117.8 kB/s (61.1%) Average Out:35.9 kB/s (18.6%) Current Out :32.2 kB/s (16.7%)

30
Confidential Proprietary
Version 2.0

(a) (b)

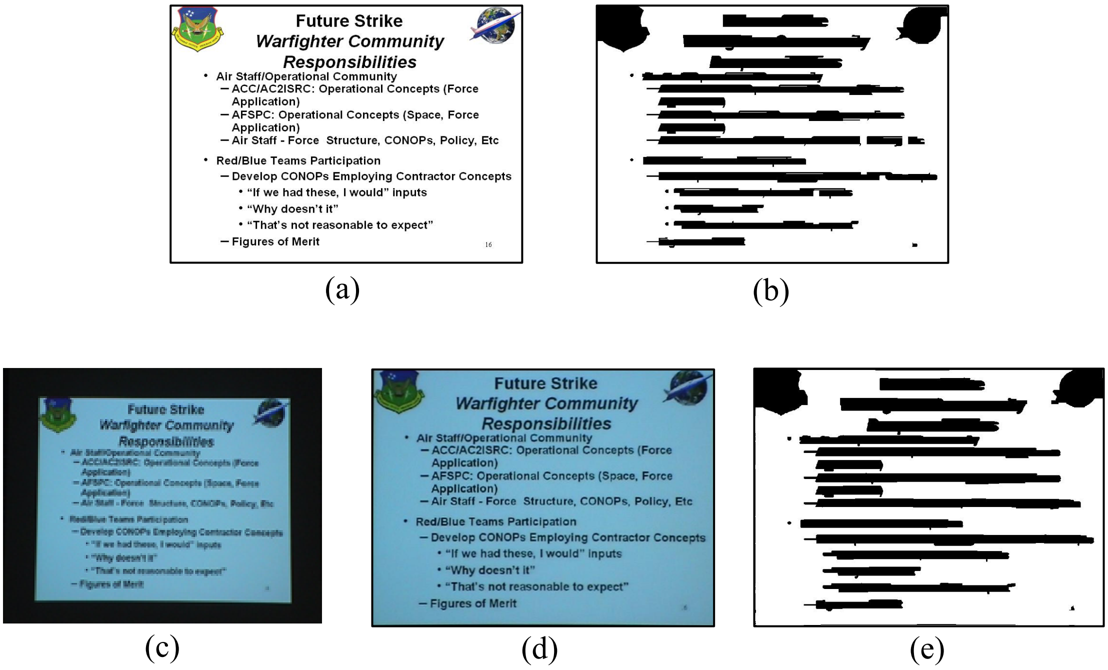
Fig. 6.6. (a) A typical slide document image containing text and graphics, (b) visual features such as horizontal text lines, vertical text lines, words in text lines, solid lines, bars with texts, images and bullets are all parts of the *Layout Signature*.

6.1.5. Image, horizontal and vertical solid lines

For these kinds of blocks, no processing is necessary as they are already tagged with the corresponding label during the bounding box labeling procedure. The feature vector ($\{Y_{\min}, X_{\min}, H_{\max}, W_{\max}, P\}$) for the *Layout Signature* of each of such block is extracted during the bounding box extraction process. Fig. 6.6 explains such layout features containing an image and two horizontal solid lines in a slide document.

6.1.6. Bullets

Bullets are often used in slideshow presentations and usually appear at the beginning of a text line. It is thus useful information to consider for the *Layout Signature*. Looking at Fig. 6.7, it is clear that in some cases bullets are attached to horizontal text lines. In other cases, bullets are segmented as separate blocks and labeled either as images or as horizontal texts having a width inferior to MAX_TEXT_WIDTH. Furthermore, bullets are often segmented as separate blocks in the original slide documents and the same bullets are attached to the respective text lines in the captured document image of the same original slide documents. One of such example is illustrated in the Fig. 6.7 in which the Fig. 6.7b shows the segmented blocks of the original image and Fig. 6.7e represents the segmented blocks of the captured image. It is clear in the figures that the bullets in the second last three text lines are segmented as separate blocks in the original slide image and the same are combined with the text line in case of the captured document image.



Future Strike
Warfighter Community
Responsibilities

- Air Staff/Operational Community
  - ACC/AC2ISRC: Operational Concepts (Force Application)
  - AFSPC: Operational Concepts (Space, Force Application)
  - Air Staff - Force Structure, CONOPs, Policy, Etc
- Red/Blue Teams Participation
  - Develop CONOPs Employing Contractor Concepts
    - "If we had these, I would" inputs
    - "Why doesn't it"
    - "That's not reasonable to expect"
  - Figures of Merit

16

(a) (b)

(c) (d) (e)

Fig. 6.7. (a) original slide document, (b) segmented blocks using connected component analysis to the final output of *RLSA*, (c) captured document image using DV-camera of the same projected slide, (d) after perspective correction and noise removal (Chapter 5), (e) segmented blocks using connected component analysis to the final output of *RLSA*.

Therefore, during the matching of the layout features of the *Layout Signature* of such document images, the matching result would be erroneous as the geometrical properties of a text line with a bullet would be different than the same without a bullet. In order to avoid this error, bullets are considered as another feature for the *Layout Signature*. Additionally, bullets are very representative of a slide document's structure. Therefore, they should not be neglected and should be recognized as they are. The extraction procedure of bullets looks for a block with constrained width and height placed horizontally before a text line. If there is no such block then it might be accompanied with the text line and the first word of the text line is analyzed for the bullet extraction. Let $B = \{b_1, b_2, .., b_{|B|}\}$ be the set of blocks labeled either as text or image, with property $\forall b_i \in B : \{W_{\max}(b_i) < MAX\_TEXT\_WIDTH$ and $H_{\max}(b_i) < MAX\_TEXT\_HEIGHT\}$. Let $L = \{l_1, l_2, .., l_{|L|}\}$ be the set of the rest of horizontal text lines. The method looks for the presence of a block b_i ahead of a text line l_j, by comparing the relative position and bounding box properties (height, width) of b_i with line l_j. Note that there can only be one bullet per text line. If the block b_i satisfies the condition given below, then the block b_i is considered as a bullet and associated with line l_j. Moreover, the line l_j is removed from L and the method continues to operate for the rest of the elements in L. Let $M = \varnothing$ be the initial empty set for bullets.

- If the condition $(p_{i,j} \wedge q_{i,j} \wedge r_{i,j})$ is satisfied then :

$$ADD(M, b_i), REMOVE(L, l_j) \text{ for } 1 \le i \le |B| \text{ and } 1 \le j \le |L|$$
$$\text{Where} \begin{cases} p_{i,j} = Y_{\min}(b_i) \ge Y_{\min}(l_j) \\ q_{i,j} = H_{\max}(b_i) \le H_{\max}(l_j) \\ s_{i,j} = X_{\min}(l_j) - X_{\min}(b_\mathrm{i}) - W_{\max}(b_i) < MAX\_ROW\_GAP \end{cases} \tag{6.3}$$

- If $L \ne \varnothing$, then possibly a bullet is within l_j and should be present in the first word. The average width and height of the connected components and the average column gap between connected components in l_j are then computed. The following steps are further performed for the bullet extraction in the first word of $\forall l_j \in L$:
  a) If there is only one connected component in the word, and if its width is inferior to the MAX_TEXT_WIDTH and if either its height is inferior to the average height or its width is inferior to the average width, then this connected component is a bullet (*e.g.* solid rectangle, circle, horizontal bar, picture, etc.).
  b) For two connected components in the word: If either the width and height of the first one is two times greater than the corresponding width and height of the next one or the height of the first one is inferior to the next one, then the entire word is a bullet (*e.g.* I., 1., 2., 3., numbering, a), 1), etc.).
  c) For more than two and less than five connected components in the word: if the height of all connected components except the last one are less or equal to the average height of the text line, and the height and width of the last connected

component is inferior to the half of the height and width of all previous ones (*e.g.* II., III., IV., etc) then the entire word is a bullet.

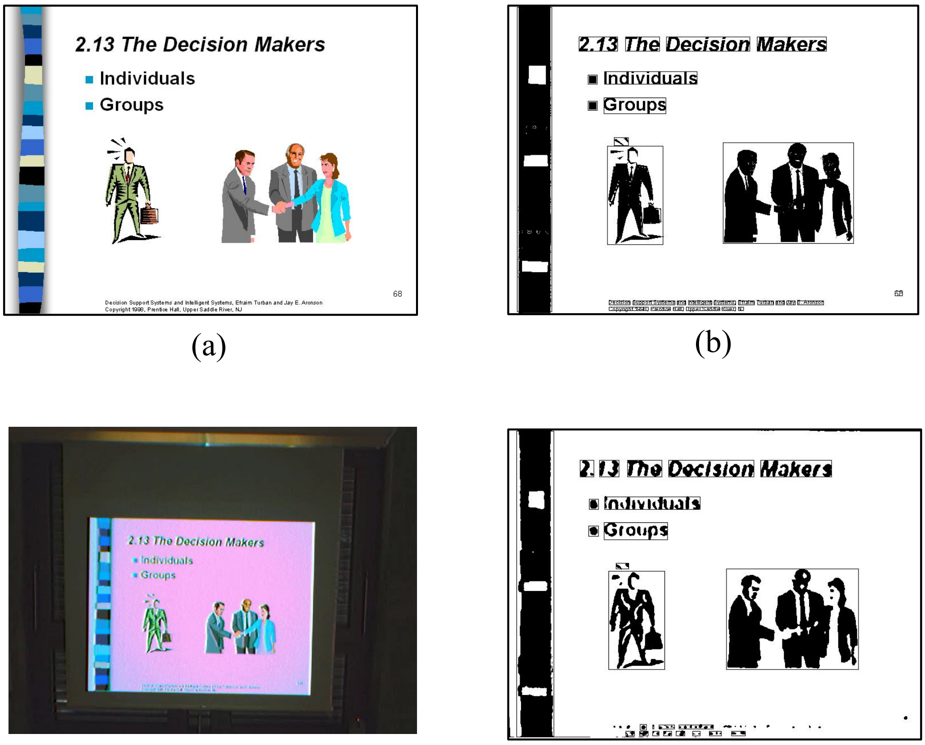
d) If a bullet is found in l_j, then l_j's feature vector is updated by removing its first word and the word is moved to the bullet set M *i.e.* $ADD\{M, FIRST\_WORD(l_j)\}$. Finally, if $M \neq \varnothing$, then the feature vector for each element in M is built for the *Layout Signature* and the feature vector for bullets is the same as that of an image $\{Y_{\min}, X_{\min}, H_{\max}, W_{\max}, P\}$.

6.1.7. Horizontal and vertical bar with text

Often in presentations, a rectangular bar behind text line is used (Fig. 6.6). This bar generally has a different background than the slides in order to highlight some textual information (below the title, above the footnotes, etc.). During binarization, often the foreground and background are reversed for such blocks. If one does not give attention to such bars with text, the corresponding block after the connected component analysis would be considered as either an image or as a horizontal (respective vertical) line. However, it is useful to analyze these kinds of blocks for adding further features to the *Layout Signature*. Thus, horizontal bars with text (respective vertical bar with text) are detected during the bounding box feature analysis and labeling. These bars with text are first converted to horizontal (respective vertical) text blocks, *i.e.* white background with black foreground. Then the new textual block is treated like other text blocks (horizontal, vertical text extractions, section 6.1.4). The feature vector of a horizontal (respective vertical) bar with text is thus the same as a horizontal (respective vertical) text line but with a different labeling of bar with text (Fig. 6.6).

6.1.8. Summary

The bounding box containing various layout features, which are extracted after completion of the processing using the procedure mentioned above to form the corresponding *Layout Signature*. Fig. 6.8 represents a typical slide images from the meeting repository and the same slide captured using a web-cam when it was projected on a screen. Upon completion of the extraction procedure, the layout features shown in Fig. 6.8b and Fig. 6.8d are the outcome from the respective original and captured image. Judging from Fig. 6.8, it is clear that the resolution of the captured image is very poor in order to apply any standard *OCR* techniques, whereas it could be applicable to the original images having homogeneous background.



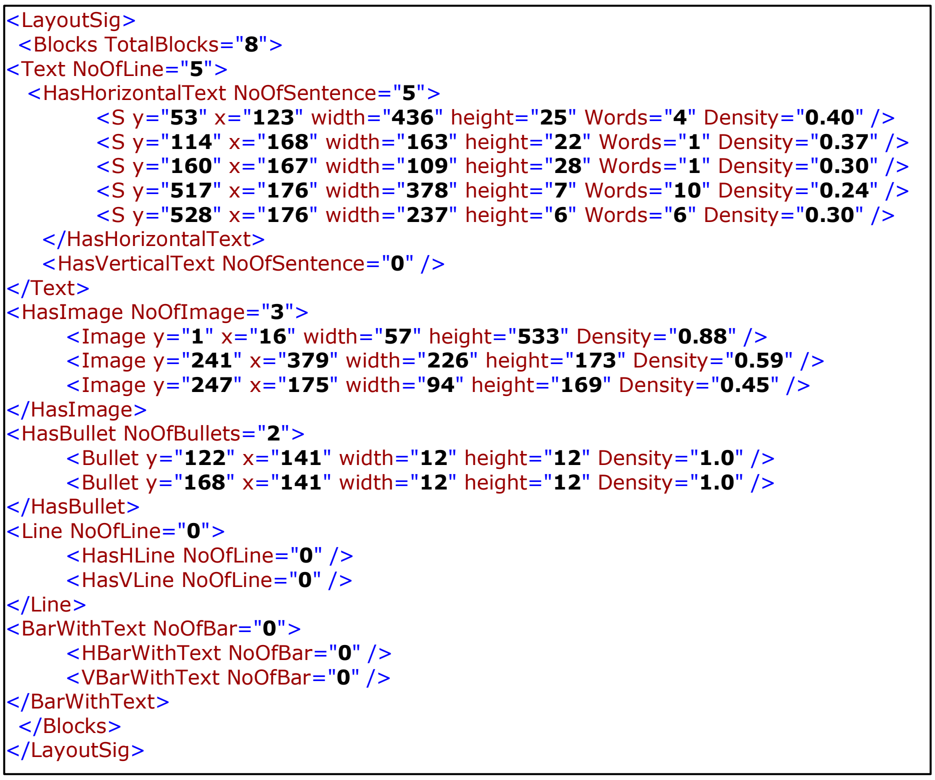
(a) (b)

(c) (d)

Fig. 6.8. (a) An original slide from the meeting repository, (b) bounding box corresponding to the layout features in the *Layout Signature*, (c) and (d) the same for the captured image from the slideshow video.

6.2. Structuring the features in the *Layout Signature*

Often, the document image identification systems use classifier to identify the incoming unknown document images. The features for classification are mainly based on the layout structure (physical, logical) and on the content (text, graphics, drawings, etc.) of the documents [*Shin et al.*, 2001; *Okun et al.*, 1999; *Yang* and *Ersoy*, 2003]. Normally, classifiers organize the electronic documents or their corresponding images in the repository by assigning a unique class identification number to a document or a set of documents. Therefore, the repository is structured according to various classes for efficient identification. In the current scenario, the main idea is to structure the *Layout Signature* rather than structuring the repository for efficient identification. The proposed *Layout Signature* is structured according to the above-mentioned layout feature's priority; the higher-level layout features appear first in the XML hierarchy and the lower level layout features stand at the leaves (Fig. 6.9). The matching of the layout features is a breadth first search approach, which considers higher-level layout features first. The breadth first search follows a level-by-level traversal. Once it finishes the comparison of all layout features in a level then it proceeds to the next level for the comparison. If one traverse from the root feature node to the leaf nodes, the priority slowly decreases as it is mentioned earlier that higher priority layout features are at the top level of the *Layout Signature* (Fig. 6.10).



```
<LayoutSig>
 <Blocks TotalBlocks="8">
<Text NoOfLine="5">
  <HasHorizontalText NoOfSentence="5">
       <S y="53" x="123" width="436" height="25" Words="4" Density="0.40" />
       <S y="114" x="168" width="163" height="22" Words="1" Density="0.37" />
       <S y="160" x="167" width="109" height="28" Words="1" Density="0.30" />
       <S y="517" x="176" width="378" height="7" Words="10" Density="0.24" />
       <S y="528" x="176" width="237" height="6" Words="6" Density="0.30" />
   </HasHorizontalText>
   <HasVerticalText NoOfSentence="0" />
</Text>
<HasImage NoOfImage="3">
     <Image y="1" x="16" width="57" height="533" Density="0.88" />
     <Image y="241" x="379" width="226" height="173" Density="0.59" />
     <Image y="247" x="175" width="94" height="169" Density="0.45" />
</HasImage>
<HasBullet NoOfBullets="2">
     <Bullet y="122" x="141" width="12" height="12" Density="1.0" />
     <Bullet y="168" x="141" width="12" height="12" Density="1.0" />
</HasBullet>
<Line NoOfLine="0">
     <HasHLine NoOfLine="0" />
     <HasVLine NoOfLine="0" />
</Line>
<BarWithText NoOfBar="0">
     <HBarWithText NoOfBar="0" />
     <VBarWithText NoOfBar="0" />
</BarWithText>
 </Blocks>
</LayoutSig>
```

Fig. 6.9. The *Layout Signature* of the original document image in Fig. 6.8a is represented using XML.

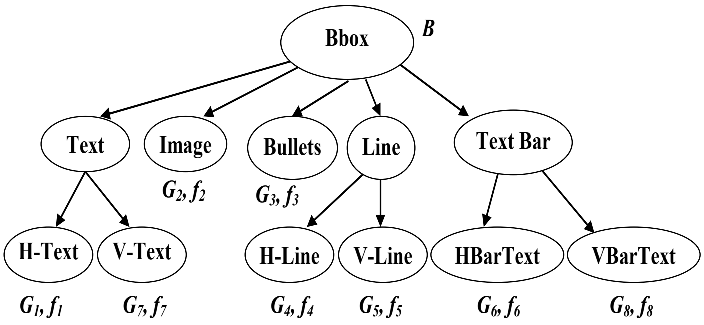


Fig. 6.10. Tree representation of layout features in a *Layout Signature*.

There are few reasons for choosing such a hierarchy and those are:

- The hierarchy of the *Layout Signature* corresponds to the extraction process of the layout features. Features requiring less processing are first extracted. They are more reliable than the lower-level features, which need more processing, and thus may introduce errors. For this reason, the layout features that take less processing time are the most reliable features and are placed at the top of the *Layout Signature* tree.

- The textual layouts vary more than other layout features in most of the slide images. For example, people often select existing design templates (*e.g.* PowerPoint application) and thus only the textual and image content varies. Thus, the textual feature is of highest priority. Observing real-world slideshow presentations, other layout features (bullets, horizontal and vertical lines, bar with text, etc) have been prioritized accordingly.
- The hierarchical representation of the *Layout Signatures* speeds-up the search during the matching by giving more importance to the high-level layout features, which narrows the search path.

An example of the current layout features' hierarchy is displayed in Fig. 6.9 and Fig. 6.10. In the following section, matching techniques based on the structured *Layout Signature*, are described.

6.3. Matching of *Layout Signatures*

The proposed signature-based matching technique has numerous advantages. Firstly, it is better than the global image matching, *e.g.* pixel-by-pixel comparison [*Mukhopadhyay* and *Smith*, 1999; *Ozawa et al.*, 2004], mainly because (a) the resolution of the extracted images is too poor for an effective matching, (b) rotation and translation affects the pixels locations, which further affects the distance computation $\{ d(A,B) \leq d(A,X) + d(X,B) \}$, where A, B and X are the respective source, target and intermediate images). Secondly, this is a novel document identification and matching technique that avoid traditional classifiers and uses a signature as a way to represent documents.

In this section, both an exhaustive and a hierarchical search technique for the identification of captured images are presented. At each feature node level (Fig. 6.9), the matching score is calculated by considering the number of total matches divided by the total number of elements located in the corresponding feature node. Both comparison directions are considered, mainly because the number of layout features and the number of elements for a particular layout feature is rarely equal for the corresponding high-resolution and low-resolution images. This fact is a direct implication of the resolution difference and of further errors in the *RLSA* method, projection profiles and bounding boxes' labeling (Section 6.1). Let us consider the following representation:

A = a queried signature;

B = a target signature;

f_A = a layout feature node of the signature A ;

f_B = a layout feature node of the signature B;

N_A = number of elements in feature node f_A;

N_B = number of elements in feature node f_B;

C_{AB} = numbers of elements found to be matched under the respective nodes f_A and f_B;

f_{AB} = matching score at feature node f_A and is computed as C_{AB} / N_A (direction $A \rightarrow B$);

f_{BA} = matching score at feature node f_B and is computed as C_{AB} / N_B (direction $B \rightarrow A$).

In a particular layout feature, the total number of matches is computed by looking at the difference in the feature vector of each element in both the feature nodes (f_A and f_B) of queried and targeted signatures. Let $V_A(l)$ and $V_B(m)$ be the feature vector of l_{th} and m_{th} element of the feature node, f_A and f_B of the respective signature A and B. The component-wise absolute distance, $d_{A,B}(l,m)$ between the feature vectors $V_A(l)$ and $V_B(m)$ is computed as:

$$d_{A,B}(l,m) = \| V_A(l) - V_B(m) \|, \text{ where } l = 1 \ldots N_A \text{ and } m = 1 \ldots N_B \tag{6.4}$$

If the above computed distance is inferior to the respective pre-defined threshold, T_v, then a match is found. The component-wise comparison is a *one-to-one* comparison and is applied to each component in the feature vector of the elements in a given layout feature node. The components in a feature vector are the geometrical properties and the pixel density of that feature such as $Y_{\min}$, $X_{\min}$, $H_{\max}$, $W_{\max}$, N_{word}, $R_i(Y_i, X_i)$, P as explained earlier in the Section 6.1. Once a match is found, then the l_{th} and m_{th} elements are removed from their corresponding feature node, otherwise only the l_{th} element is removed from the node, f_A of signature, A. At each feature node (Fig. 6.10), the matching procedure above is carried out until the number of element reaches zero at either node f_A or f_B of respective signatures (A or B). The final matching score (f) is the average of scores in both directions *i.e.* $f = (f_{AB} + f_{BA}) / 2$. The threshold value for the comparison of each component in the feature vector of the element corresponds to the same feature node in the queried and the target *Layout Signature* as shown in Table 6.1.

The thresholds are assigned empirically after evaluating more than 100 slide documents. Concerning the threshold for bounding box width $W_{\max}$ and height $H_{\max}$, the value is two times the value of the threshold for the minimum co-ordinates of the geometrical locations of the corresponding bounding box. This is due to the fact that often slideshows use an available pre-defined template in which the geometrical locations of the components such as text lines and its font size are fixed and only the length of the text lines varies. Furthermore, the captured image of the projected documents often exhibits shadow towards the boundary of the various foreground components and is due to the reflectance properties of the projector screen. Therefore, the maximum height and width ($H_{\max}$, $W_{\max}$) of the bounding box is a bit larger in the captured image than in the original slide document.

Table 6.1. Thresholds used for the layout feature vectors comparison

Feature vector elements	**Threshold**
$Y_{\min}$, $X_{\min}$	5 (pixels)
$H_{\max}$, $W_{\max}$	10 (pixels)
N_{word}	2 (word count difference)
X_i, Y_i for each word	5 (pixels)
P (Pixel density)	30% (bounding box density

6.3.1. Exhaustive search

This search technique is a brute force method for matching features in the *Layout Signature*. First, the matching score for each available layout feature (f_i, $i = 1\ldots8$, Fig. 6.10) is calculated and then the global score is computed as the ratio of the sum of all features' score upon the number of features having non-zero score. In this method, two types of mechanisms have been used for computing the global score: with weighted score *i.e.* $f_G = (\sum\omega_i f_i)$ for $i = 1\ldots8$ where ω_i is the weight of the feature, f_i and without weighted value of features *i.e.* $f_G = (\sum f_i)$ for $i = 1\ldots8$. The weight ω_i for the feature, f_i is assigned according to the priority of that feature *i.e.* its position in the hierarchically structured *Layout Signature* (Fig. 6.10). These weights are fixed and the values are shown in Table 6.2.

Table 6.2. Weights used for the various layout features

Layout features	**Weight**
Horizontal text (f_1)	0.8
Image (f_2)	0.6
Bullet (f_3)	0.5
Others (f_4, f_5, f_6, f_7, f_8)	0.3

The signature having the highest global score is returned after comparison with all the signatures in the repository.

6.3.2. Hierarchical search

This matching technique is based on simple heuristics taking into consideration the hierarchy of the features in the *Layout Signature*. Higher-level features are first compared, and then the lower-level features are matched, and so on until the matching reaches the leaves of the hierarchy. The hierarchy of the *Layout Signature* normally guides the search path. In this case the score for any feature is considered, if it is greater than a certain minimal value (in our case it is defined as 0.2). Let E be the queried signature of the captured document image and $D = \{d_1, d_2, \ldots, d_{|D|}\}$ the set having all the *Layout Signatures* in the repository and $T = 0.4$, the initial matching threshold. The proposed hierarchical search algorithm is summarized in Fig. 6.11 and follows the hierarchical steps below:

- STEP 1: $B = \{b_1, b_2, .., b_{|B|}\}$ is derived from D ($B \subseteq D$), considering all the signatures with the difference in bounding box (root of the hierarchical tree, Fig. 6.10) number is inferior to T_b *i.e.* $\forall b_i \in B \Rightarrow \exists d_i \in D : Diff\_Bbox(E, d_i) \leq T_b$.
- STEP 2: First, the horizontal text layout feature's matching-score, f_1 is computed. The subset G_1 (Fig. 6.10) is created from B ($G_1 \subseteq B$) by considering elements, whose score, f_1 is superior to T. If no element in G_1 satisfies the above condition ($f_1 > T$), then the same subset B ($G_1 = B$) is kept for the next feature comparison.
  i.e. $\forall g_i \in G_1 \Rightarrow (\exists b_i \in B : f_1(b_i, E) > T), (G_1 = B) \Leftrightarrow (G_1 = \emptyset)$.
- STEP 3: The same procedure as in step 2 is used for deriving the next subset from the previous subset for all other layout features following the hierarchy (from higher-level to lower-level features, Fig. 6.9 and Fig. 6.10) of the layout features *i.e.* $\forall g_i \in G_j \Rightarrow (\exists g_k \in G_{j-1} : f_j(g_k, E) > T), G_j \subseteq G_{j-1},\ 2 \leq j \leq 8$. At any feature level, elements fulfilling the criteria in all the previous feature levels are kept. Furthermore, at any feature level, if there is only one solution and its matching score is greater than 80%, then all other existing features for this solution are looked for. If the global matching score exceeds 90%, the search is then terminated and the current solution is returned by the system and is the winner.

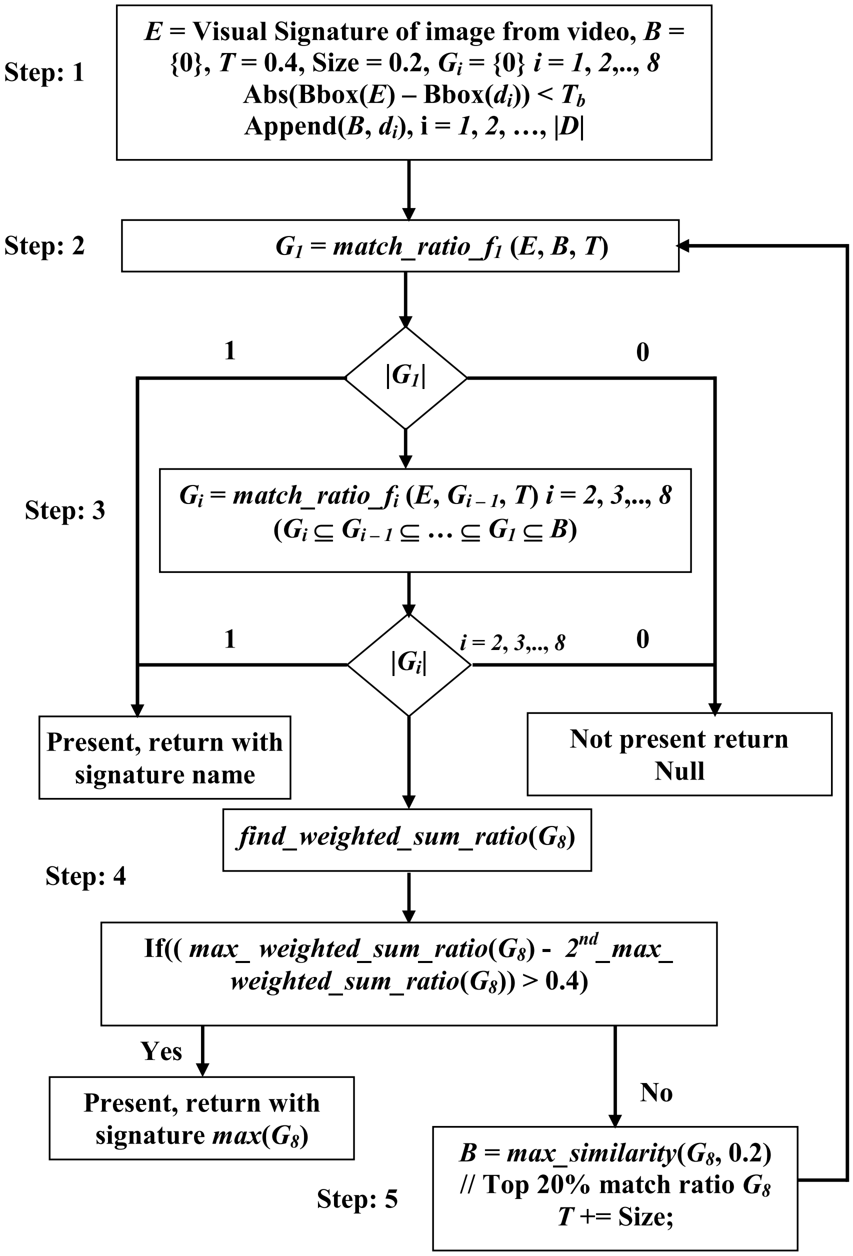

Step: 1
E = Visual Signature of image from video, B = {0}, T = 0.4, Size = 0.2, G_i = {0} i = 1, 2,.., 8
Abs(Bbox(E) – Bbox(d_i)) < T_b
Append(B, d_i), i = 1, 2, …, |D|
Step: 2
G_1 = match_ratio_f_1 (E, B, T)
1
|G_1|
0
Step: 3
G_i = match_ratio_f_i (E, G_i-1, T) i = 2, 3,.., 8
(G_i ⊆ G_i-1 ⊆ … ⊆ G_1 ⊆ B)
1
|G_i|
i = 2, 3,.., 8
0
Present, return with signature name
Not present return Null
find_weighted_sum_ratio(G_8)
Step: 4
If((max_ weighted_sum_ratio(G_8) - 2nd_max_ weighted_sum_ratio(G_8)) > 0.4)
Yes
No
Present, return with signature max(G_8)
Step: 5
B = max_similarity(G_8, 0.2)
// Top 20% match ratio G_8
T += Size;

Fig. 6.11. Flow chart of the hierarchical matching technique.

- STEP 4: When the search reaches the right-most leaf node, the number of elements in the final subset $|G_8|$ is checked. If it is more than one, a weight is assigned to each layout feature's score, according to its position in the hierarchical *Layout Signature.* The corresponding weight for each layout feature is shown in Table 2. The sum of the weighted score of all features for all elements in G_8 is then computed. If the difference between the highest weighted sum and the second highest weighted sum of G_8 is superior to 0.4, then the element having the highest weighted sum is considered as the required *Layout Signature*. Otherwise, set B is assigned to the top 20% elements in G_8 having the highest sum.

- STEP 5: T is increased with a step size of 0.2 and the same matching procedure starts again from step 2 and it continues until only one matching slide document is found or no more elements are present in the set. If there are no more elements in any of the above subset, then the slide is not present in the repository.

The threshold T_b for comparing the number of bounding box in queried and targeted *Layout Signatures* is tuned to 10 by considering many slide images. With a higher value of the threshold T_b, there would be too many signatures in the initial set B, which would drastically increase the computational cost. Whereas for a lower value, it is observed that sometimes the matching solution has been removed in the initial step.

6.4. Conclusion and perspectives

In this chapter, the identification of low-resolution documents, *e.g.* captured from mobile devices, is described using the document *Layout Signature*. The document *Layout Signature* symbolizes the abstract representation of the visual content of a document. This signature represents the shallow physical structure of the document. Due to the low resolution of the perceived captured document, it is a very difficult problem for the abstraction of a true physical layout structure. The visual features such as text lines, words, images, bullets, solid, and textual bars are extracted from the documents. The extraction of various visual features follows the procedure such as conversion of gray-scale document to bi-level images, use of *RLSA* for detection of bounding box containing the connected components and then each bounding box is labeled according to its various properties. Then each extracted component is further processed for the next level of visual aspects such as text lines, words, bullets, etc. These visual features' geometrical properties as well as pixel density are extracted and stored in the feature vector of that visual feature.

These various layout features are structured hierarchically in the *Layout Signature* according to their priority. The priority is assigned by considering the extraction process, the real-world slideshow presentation and therefore it is reliable. The structuring of the layout features helps during the matching procedure to narrow down the search path to fewer solutions at each level. Moreover, this procedure is an alternative to the use of classifiers, which instead structure the repository. The matching procedure uses the hierarchical path for the identification of an incoming low-resolution document. Two different types of matching algorithm are proposed. (a) First, a brute force matching algorithm that computes the global matching score by summing up the matching score of individual layout features. The signature having the highest global score after searching the whole repository is the winner. This method utilizes two different strategies to compute the global matching score. One uses a pre-fixed weight corresponding to each layout feature and the other does not use any weight and is only the sum of individual matching score. (b) The second matching algorithm which uses the hierarchical structure of the *Layout Signature* and at each feature node, the decision is taken either to move forward or to terminate the search. This method is computationally inexpensive as it does not consider all the signatures in the repository. Furthermore, often the matching does not require visiting all the feature nodes in the hierarchical tree. The evaluation of the above-mentioned *Layout Signature* alone and along with the *Color Signature* (Chapter 7) is presented in Chapter 8.

Though, the proposed *Layout Signature*-based identification method targets the identification of the captured slides during a presentation. The features in the proposed *Layout Signature* could be adopted for the identification of other documents like posters presented during a conference, articles or documents hanging on walls, in supermarkets for product information or real-time translation using mobile devices, etc.

Chapter 7

Document *Color Signature*

The document *Visual Signature* consists of both layout and *Color Signature*. In the previous chapter (Chapter 6), the extraction and matching of the *Layout Signature* has been detailed. In the current chapter, the extraction and matching of the *Color Signature* is presented. Color is one of the low-level visual features most often used for image retrieval applications. In the current scenario of document identification, it can be easily combined with the layout structure. Often, most of the slide images in a slideshow have similar low-level visual features (color, texture and shape). Therefore, the slide identification system should consider not only the layout structure of the slide images but also the low-level visual features. Assuming that no textual information about the content of the perceived image is given, the low-level visual features such as color [*Wan* and *Kuo*, 1996; *Sticker* and *Dimai*, 1995; *Aigrain et al.*, 1996], as well as texture [*Manjunath* and *Ma*, 1996; *Manjunath et al.*, 2001] and shape [*Jain* and *Vailaya*, 1996b; *Gary* and *Mehrotra*, 1990], are extensively used in many systems in order to retrieve images having similar content as the queried ones. Retrieval systems based on such visual features work efficiently when queried on similar images, but do not when the captured image is taken from a different angle or have a different scale [*Petkovic et al.*, 2000]. Furthermore, such features are very dependent on illumination conditions, shading and compression and for this reason we believe that distribution of features is a better visual representation, *i.e.* more robust to all the cited effects, than an individual feature vector. In our case, we considered color as one of the feature for our signature instead of the texture and shape, since often the slides in different slideshows vary in color rather than in texture and shape.

This chapter presents the extraction of various features for the *Color Signatures* and the matching procedure of the color features, which are considered for the *Visual Signature*. Section 7.1 describes the extraction of global and distributed color features by considering the normalized RGB color space and the distribution of color in the 2-D image plane. The matching of the various features such as global and distributed color features is described in Section 7.2. Finally, the chapter concludes with perspectives of the proposed document *Color Signature* (Section 7.3).

7.1. Color features extraction

The proposed *Color Signature* is the abstracted description of the color content of a document image. The feature set related to the color content are extracted by considering the color distribution in the (a) 3-D RGB color space and (b) 2-D image plane. The distribution of pixels in the RGB color space is represented with the reduced feature set rather than the global histogram. This is due to the fact that often the color in the low-resolution captured images is distorted due to the low quality of the capture devices and the varying lighting condition of the capture environment. The feature set from RGB color space is computed using normalized global color histogram and is represented as a global color feature set. This feature set is represented as an *Equivalent Ellipse* (*EE*) with six parameters (centers, axes and orientation) in the density surface that is computed using *Kernel Density Estimation* (*KDE*). The *KDE* uses the normalized and reduced global color histogram.

The feature set for the color distribution in the 2-D image plane is computed by considering the geometrical distribution of the similar pixels in the document image plane. Often, the values of pixels in the captured images and corresponding pixels in the original image are not the same due to the presence of color cast, which is the predominant superimposed color. This is due to changes in the lighting environment, surface properties of the target object and even the characteristics of the capture devices. However, the geometrical distributions of the pixels in the image plane remain preserved. Therefore, in the case of image captured from low-resolution handheld devices, the geometrical distribution of similar pixels would be more powerful than the spatial distribution in any one of the color space. This feature set is extracted first by grouping similar pixels based on their values. Then, the mean and variance of the X-Y locations (image plane) of pixels belonging to a particular group is computed. Each group of pixels is represented as an *Equivalent Rectangle* in the image plane with the center as the mean, whereas the width and height as variances in the X- and Y-directions, respectively.

Prior to the extraction of features for the *Color Signature*, the captured document images should be rectified and de-noised using a low-pass *Wiener* filter (Chapter 5). In case of the original document images, the size (height and width) of the perceived image should be checked and if necessary the resolution should be corrected using a proper sampling procedure (Section 6.1). Once, the above-mentioned pre-processing step is done then the image is further processed for the respective feature extraction.

7.1.1. Global color features

The color histogram method is commonly used for color-based image retrieval. It describes the color distribution of an image in a specific color space. Often, the RGB space is considered for the color feature extraction.

7.1.1.1. Normalized histogram computation

A standard way of generating the RGB color histogram of an image is to consider the m higher order bits of the Red, Green and Blue channels [*Swain* and *Ballard*, 1991]. The histogram consists of 2^{3m} bins, which accumulate the number of pixels having similar color values. In our approach, the generation of the color histogram has been reduced to two-dimensional chromatic space $r = R/I$ and $g = G/I$ (2^{2m} bins), where $I = R + G + B$ is the brightness, $0 \leq R, G, B \leq 2^{m-1}$ and $b = B/I$ could be represented as $1 - r - g$. The chromatic values r, g from RGB or a, b from the *Lab* are invariant to illumination. Let us consider a color image P of size $n_1 \times n_2$. Then $P = \{r_{i,j}, g_{i,j}\}$ could be represented with the chromatic values, where $i = 1 \ldots n_1$ and $j = 1 \ldots n_2$. The reduced color histogram $h(r, g)$ in rg- space is obtained as:

$$\begin{aligned} &r = \mathrm{int}(Mr_{i,j}),\ g = \mathrm{int}(Mg_{i,j}),\ M = 2^m - 1 \\ &h(r,g) = \frac{\#\, pixels\ fall\ in\ bin\ r,g}{n_1 \times n_2},\ 0 \leq r,g \leq M \end{aligned} \tag{7.1}$$

Finally, the similarity between any two images, I_p and I_q is very often measured by computing the similarity distance between their respective histograms, h_p and h_q. *Minkowski* distance is one of the most popular methods used to measure the similarity distance and is defined as:

$$D(I_p, I_q) = \sum_{x=0}^{M}\sum_{y=0}^{M}\{[h_p(x,y) - h_q(x,y)]^n\}^{n^{-1}} \tag{7.2}$$

The different values of parameter, n gives us different distance measures, *e.g.* when $n = 1$ we get the *Manhattan* distance, and for $n = 2$, the *Euclidian* distance.

Another measure of the similarity distance of the two histograms is expressed as the intersection of the histograms [*Swain* and *Ballard*, 1991] and is defined as:

$$D(I_p, I_q) = 1 - \frac{\sum_{x=0}^{M}\sum_{y=0}^{M}\min\{h_p(x,y), h_q(x,y)\}}{|h_p|} \tag{7.3}$$

In the histogram representation the drawback is that the shape of the histogram strongly depends on the number of pixels and the method used for image representation. If the image size is small, then there are very few points available for the histogram, which gives rise to erroneous results for the histogram-based comparison. In order to overcome the above-mentioned problems, we propose in the following section a smooth non-parametric estimation of the color distribution, instead of a discrete histogram representation, based on the concept of non-parametric density estimation [*Scott*, 1992].

7.1.1.2. Kernel Density Estimation

Density estimation describes the process of obtaining the *probability density function* (*pdf*) $f(x)$ from an observed random quantity. In general, the density functions of the random samples are unknown. The simplest and oldest form of the density estimation is histogram. In this case, the sample space is first divided into a grid of width, h. Then, the density at the center of the grid is estimated by $f(x)$ = *# samples in one bin* / *h*. In such estimation, the drawbacks are (1) the offset dependence, (2) the lack of differentiability, (3) sensitive to the rotation of coordinate axis and (4) in higher dimensions it causes sparse occupancy.

The drawbacks above are overcome by the *KDE* procedures. However, most non-parametric methods require either all samples or extensive knowledge of the problem. In this technique, the underlying probability density function is estimated by placing a kernel function on every sample in the sample space and then summing up all the functions for each sample. Given a one dimensional sample space $X = \{x_i\}$, where $i = 1 \ldots N$, the kernel density at any point x is estimated as:

$$f'(x) = \sum_{i=1}^{N} w_i K\left(\frac{x - x_i}{h}\right) \tag{7.4}$$

where K is the kernel function, which determines the shape of the ‘‘bumps’’ placed around the data points in the sample space, h is the bandwidth of the kernel and w_i is the weighting coefficients. Normally, the value of w_i is constant and is $1/(Nh)$. The multivariate kernel density in case of d-dimensional sample space is defined as:

$$f'(x) = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{h_1 \ldots h_d} \kappa\left(\frac{x_{i_1} - x_1}{h_1}, \ldots, \frac{x_{i_d} - x_d}{h_d}\right) \tag{7.5}$$

where $h_1 \ldots h_d$ the bandwidths for each dimension and κ is the d-dimensional kernel function. The d-dimensional kernel functions are commonly represented as the product of the one-dimensional kernel functions *i.e.*

$$\kappa(u_1, u_2, \ldots, u_d) = K(u_1)K(u_2) \ldots K(u_d) \tag{7.6}$$

In our approach, the two-dimensional chromaticity *rg*-space is used with the same bandwidth in both dimensions ($h_1 = h_2 = h$, *i.e.* radial-symmetric kernel function). The resulting kernel density estimation in two-dimensional space is:

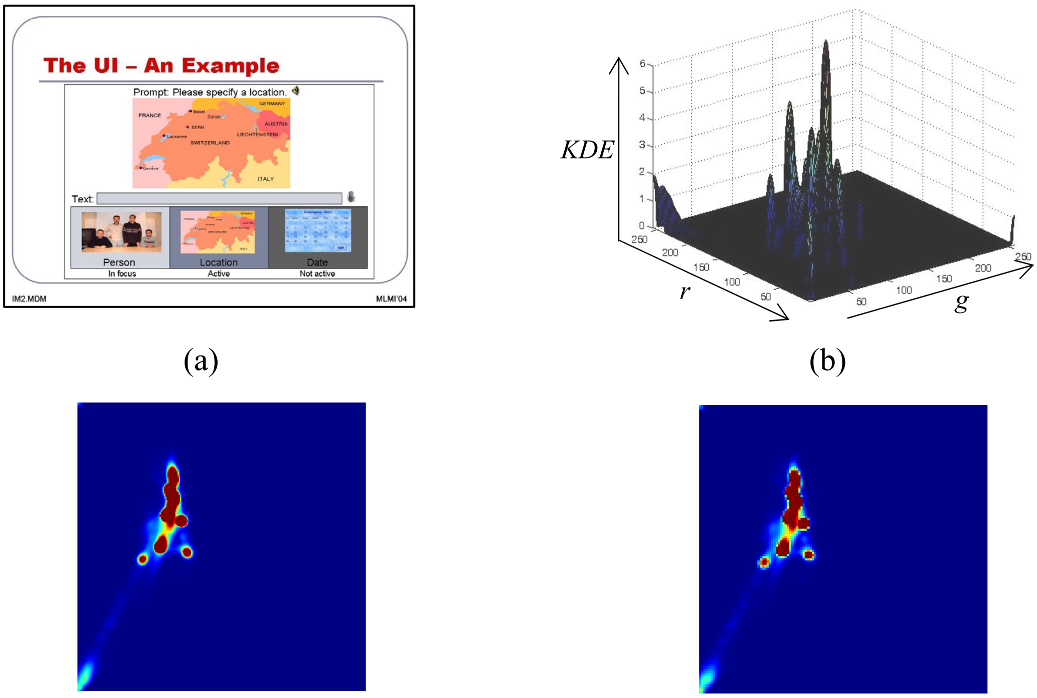
$$f^{'}(x) = \frac{1}{Nh^2} \sum_{i=1}^{N} \left\{ \prod_{j=1}^{2} K\left(\frac{x_{i_j} - x_j}{h} \right) \right\} \tag{7.7}$$

The estimation of the kernel density depends on the kernel function and the bandwidth, *h*. The kernel function decides the shape of the "bumps" placed around the sample for a given bandwidth. We consider the *Epanechnikov* kernel, which has been shown to be robust to outliers and optimum in the sense of having the minimum Mean Integrated Square Error (MISE) in comparison with other kernels [*Silverman*, 1986].

$$K(u) = \begin{cases} \frac{1}{2} c_d^{-1} (d+2)(1 - u^T u) & \text{if } u^T u \text{ p } 1 \\ 0 & \text{otherwise} \end{cases} \tag{7.8}$$

where c_d is the volume of the unit *d*-dimensional sphere and *u* is the *d*-dimensional data point. For the density estimation, the shape of the density function is heavily dependent on the chosen bandwidth. The small values of *h* result in spiky density estimation, which shows the spurious features. On the other hand, too large values of *h* lead to over-smoothed density estimation that masks the structural features. In this case, the value of *h* is set to 2.5 and 2.0 for the respective original and captured images after evaluating 100 different images for the different values of *h* ranging from 1.5 to 3.5. Furthermore, the above-mentioned density estimation is evaluated using the optimal Gaussian kernel and obtained the similar results [*Parzen*, 1962]. Fig. 7.1 illustrates the *KDE* of a sample slide document for the bandwidth of 2.5.

Jones and *Rehag* reported that 77% of the possible 24-bit RGB colors were never encountered on images collected from the web [*Jones* and *Rehag*, 2002]. Furthermore, it is observed that no perceptive degradation of the *KDE* for 7-bits compared to 8-bits per RGB channels (Fig. 7.1), indicating that reducing the color space do not affect much the color density estimation. Since the color feature is not used in our method to identify the exact matching of the slide but the slideshows or groups of slides having similar background pattern and color. Therefore, it is judged reasonable to consider for the *KDE*, the 7 most significant bits (*msb*) of each of the RGB channels. This reduces the sample space to ¼th of the actual one, and thus heavily speeds-up the computation time of the *KDE*. Moreover, further reduction of bits is not considered since the goal is to represent the *KDE* surface as an *Equivalent Ellipse* (*EE*) for the matching rather than the surface itself. If one goes for further reduction then the surface area would be too small and could be difficult to differentiate the *KDE* surface of two different document images.

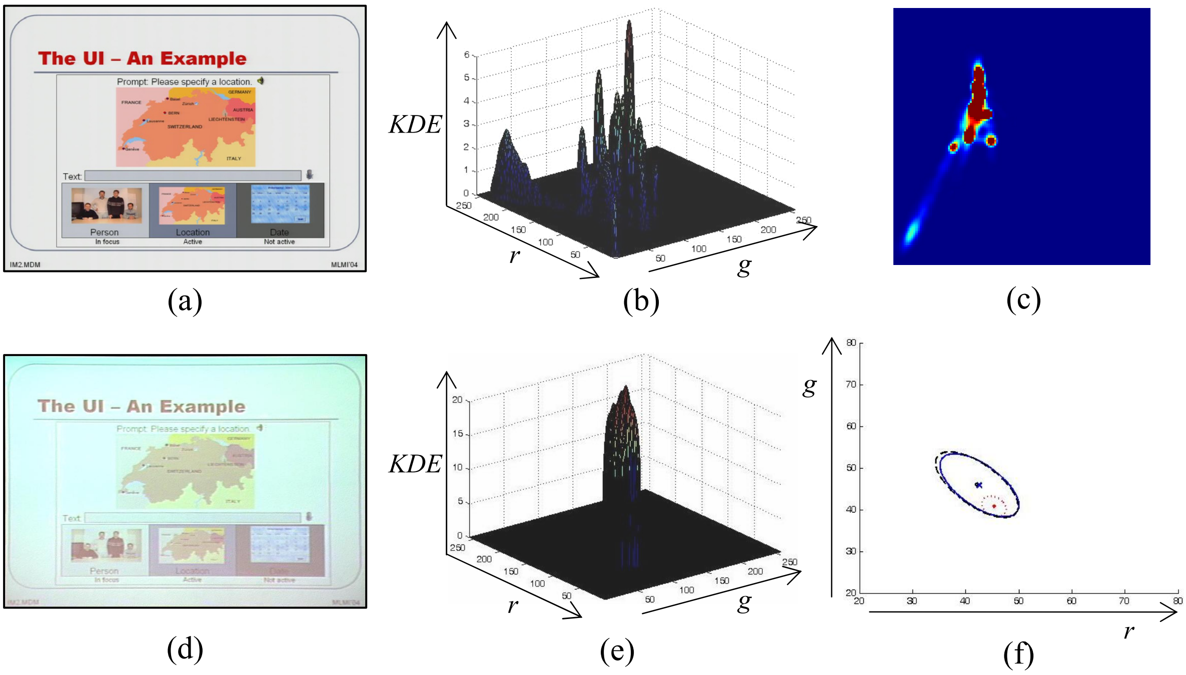


(a) (b)

(c) (d)

Fig. 7.1. (a) Original image, (b) *KDE* of the color distribution in the *rg*-color space, (c) its pseudo-color representation for true color (24-bits) and d) reduced color (21-bits).

The similarity between two images could be measured by computing the distance between their respective *KDE* of the histograms using the equation either (7.2) or (7.3). This distance-based similarity measurement is known to perform well for images of the same size with negligible color distortions. In this scenario of the captured documents, one faces a problem of non-uniform color shifting in the captured image as compared to the original image. This shift of color is due to the presence of color cast, which is the predominant superimposed color. The color cast is because of variations in the lighting conditions or to the capture device properties. Fig. 7.2 illustrates the *KDE* of the captured images using two different capture devices. One is a projector and other is a handheld digital camera. The color of the image from projector's output (Fig. 7.2a) is quite close to the original image (Fig. 7.1a) whereas it is very much distorted in the captured image from the handheld device (Fig. 7.2d). This causes the position(s) of the peak(s) and valley(s) in the density surface is quite similar in the original and the image from the projector. However, in case of the image from the handheld device (Fig. 7.2e), the position(s) of the peak(s) and valley(s) in the density surface far differ in comparison to that of the original image and the image from the projector. Thus, the standard histogram-based similarity distance would not perform efficiently for the images captured using handheld devices. Furthermore, the aim is to represent the documents with their corresponding signature and identification is based on the matching of the signatures. It is not wise to keep all values of the density surface, which not only takes more matching time but also storage space. So, the reduction of feature space is a better option for both storage and fast matching.



(a) (b) (c)

(d) (e) (f)

Fig. 7.2. (a) Captured image of Fig. 7.1a from projector, (b) its *KDE* of the color distribution, and (c) pseudo-color representation for 21-bits in the *rg*-space, (d) rectified image captured using a digital camera, (e) its *KDE* of the color distribution and (f) *Equivalent Ellipse* of the density surfaces of the original and the two captured image.

7.1.1.3. Equivalent Ellipse representation

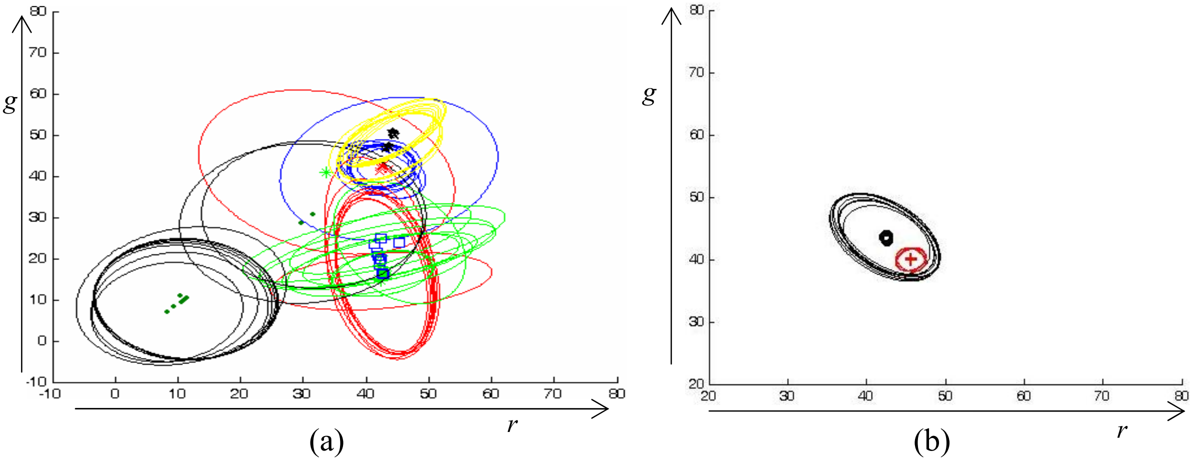
The color features are computed from the estimated density surface of the normalized color using the *KDE*. Once this is done, the distribution of the density surface in the *rg*-plane of image colors is then analyzed by looking at its kernel density distribution $K_d(r, g)$. The mean (μ_r, μ_g) and variance (σ_r, σ_g) of the density surface in the *rg*-plane is computed as:

$$\begin{aligned} \mu_r &= \int_r rK_d(r,g)dr \ , \ \mu_g = \int_g gK_d(r,g)dg \\ \sigma_r^2 &= \int_r (r-\mu_r)^2 K_d(r,g)dr \ , \ \sigma_g^2 = \int_g (g-\mu_g)^2 K_d(r,g)dg \end{aligned} \tag{7.9}$$

Then, the density distribution of each surface is associated to an *EE* with its center $C = (\mu_r, \mu_g)$, semi major axis $a = max\ (\sigma_r, \sigma_g)$, semi minor axis $b = min\ (\sigma_r, \sigma_g)$ and an orientation angle of θ. The orientation, θ of the *EE* representing density surface, is computed using the least-squares fit of ellipse to 2-D points in the density surface [*Fitzgibbon et al.*, 1999]. Although the density surface of the original and captured images are not the same but, it is observed that most of the properties (eccentricity, orientation, etc.) of the *EE* of both the captured and original images are often preserved and that only the *EE* location is shifted (Fig. 7.2f). The feature vector for the color is finally $c_f = \{\mu_r, \mu_g, \sigma_r, \sigma_g, \theta, d\}$, where d is the density of the estimated kernel density distribution over the elliptical surface area. When the axes (a and b) are equal, then the *EE* becomes *Equivalent Circle* (*EC*). In this case, the orientation angle θ is not considered as this is not applicable for a circle. However, practically it is not feasible comparing Fig. 7.2f, which shows the *EE* of the original image of Fig. 7.1a same image captured using a projector and a handheld digital camera, one could observe that the *EE* of the original and the same from the projector is very close (two large ellipses). The *EE* of the image from the handheld device is small and placed inside the other two ellipses. However, if one gives a closer look at the orientation and the eccentricity, they are quite close to that of the ellipses from the original and captured image from the projector. Only the position of the ellipse is shifted. Fig. 7.3a shows the *EE* of 50 slides randomly picked up from 5 different slideshows (10 each) and it is possible to observe most of the slides within a slideshow have similar color since the properties of *EE* are close. In some cases only the centers of *EE* are adjacent to each other but the orientation and axes are dissimilar, which help to differentiate slides with different colors. Similarly, for a given set of slide documents from a presentation, the properties (eccentricity, orientation, etc.) of the *EE* of both the captured and the original images are preserved and that only the location is shifted (Fig. 7.3b).

7.1.2. Distributed color features

Although, the colors in the captured images from the low-resolution devices are distorted due to changes in the lighting environment or even of capture devices, nevertheless, the geometrical distributions of the color in the image plane remain preserved. Therefore, for the color-based identification of low-resolution captured images, the geometrical color distribution is more stable than the color distributions in any one of the color space (RGB, Lab, YUV, etc.), if one consider the 'sustainability' of features extracted from color content of the image.



(a) (b)

Fig. 7.3. *Equivalent Ellipse* representation of the estimated color densities in the reduced *rg*-space of (a) slide documents randomly picked from 5 different slideshows and (b) original and captured slide documents from a single slideshow.

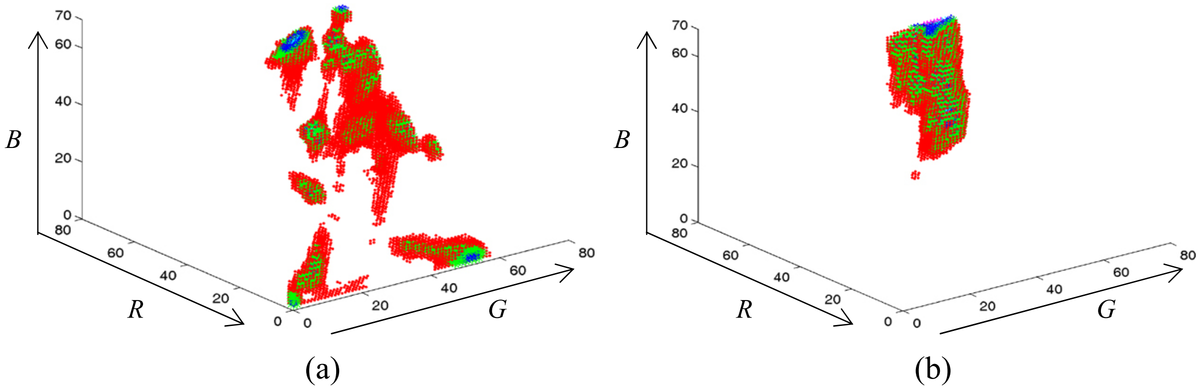
In the proposed geometrical color distribution method, the features are extracted by projecting the global color histogram in document's 2-D image plane. The features are computed with the assimilation of the two different features such as the color feature and the geometrical layout feature. The feature 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. The grouping of pixels uses the clustering methods and could be divided into two basic types: hierarchical and non-hierarchical clustering [*Jehan*, 2005]. Hierarchical clustering proceeds successively by either splitting larger clusters or by merging smaller clusters into larger ones. Non-hierarchical clustering, on the other hand, attempts to decompose the data set into a set of disjoint clusters directly. In this case, the non-hierarchical clustering is used as the pixels of the perceived document image are to be separated into different groups based on the color content of the document image. A commonly used non-hierarchical clustering method is K-means clustering algorithm. The K-means algorithm requires prior knowledge of a pre-defined fixed number of clusters, which is K from its name, while "means" stands for an average location of all pixels belonging to that particular cluster, which is in the 3-D RGB color histogram in this case [*Yang* and *Ersoy*, 2003]. In this case, the value of K is derived from the number of predominant peaks in the RGB color histograms rather than using a pre-defined fixed value as the color content of the incoming document is unknown. The "means" refers to the *centroids* of the K clusters. The value of K and the *centroid* of each cluster are derived from the RGB color histogram of the document. An adaptive assignment of K could be used, starting with a minimum value of $K = 2$ and adaptively increase the number of cluster till the improvement in error falls below a threshold or a maximum number of clusters is reached [*Sural et al.*, 2002]. The drawback of this adaptive clustering is the processing time and assignment of the maximum number of clusters, which is unknown for an incoming document image. Furthermore, the clustering time for a given data set is dependent on the number of clusters and the initialization of the clusters centroids.

7.1.2.1. Reduced 3-D histogram

The value of K is derived from the reduced RGB color histogram. The standard method for creating an RGB color histogram is to consider the m higher order bits of each channel and then accumulate the pixels having similar color values in the 2^{3m} bins. If human vision is considered, then there is a large amount of redundancy in 24-bit RGB representation of color images. *Wang et al.* have reported that representing each of the RGB channel with only 4-bits introduced little or even no perceptible visual degradation [*Wang et al.*, 2004]. Furthermore, the aim here is to estimate the number of dominant colors *i.e.* number of peaks in the 3-D color histogram. For this reason, a $64 \times 64 \times 64$ ($m = 6$) color image is considered for generating the histogram rather than a true color image $256 \times 256 \times 256$ ($m = 8$). This is achieved by simply performing a 2-bit right-shifting on each RGB channel. The resulting histogram is smoothened by convolving the 3-D *Gaussian* window of size 7.

7.1.2.2. Valid peaks

Once the reduced histogram is generated and smoothened, subsequently the numbers of predominant peaks, which consist of more than 10% of the total number of pixels, are located. The final numbers of peaks are those that posses a distance superior to certain threshold, between neighboring peaks (Fig. 7.4). The *Euclidian* distance between the co-ordinates of the located peaks in the 3-D histogram is used for comparing with the threshold for final selection. The threshold for the distance is defined as 20 for the original documents and 15 for the captured documents. These thresholds are defined after evaluation over a hundred document images. The threshold for the original image is higher as compared to the captured one. This is due to the reason that the 3-D color histogram of the captured document images is often converged as that of the original one (Fig. 7.2). This is due to the presence of color cast as explained earlier.



(a) (b)

Fig. 7.4. Histogram for peaks detection of (a) original image (Fig. 7.1a) and (b) its captured image using a digital camera (Fig. 7.2d) in reduced 6-bits RGB space.

7.1.2.3. K-mean cluster initialization

In the K-means clustering algorithm, one of the potential problems is that the choice of the number of pre-fixed clusters *i.e.* K and could be critical as different types of clusters might be became apparent when K is changed. Furthermore, a good initialization of the clusters centroids is also crucial as some clusters may even be left empty if their centroids lie initially far from the actual distribution of pixels in RGB color histogram. Therefore, the initialization of the K-means clustering techniques plays an important role. In order to overcome the pre-defined fixed value of K and random initialization to the *centroids* of the K-means algorithm, these values are computed from the color histogram, in which the detected peaks represent the predominant color content in the perceived document image. Once the numbers of effective peaks are decided then the value is assigned to K for the K-mean clustering. The centroid of each cluster is initialized with the average RGB values of the pixels surrounding the respective peak in the 3-D histogram. The *centroids* are the points in the three dimensional space of the RGB color histogram. The number of clusters and the cluster *centroids* are fed to the K-means clustering algorithm. Therefore, the processing time is extremely fast in comparison with random seeding or adaptive clustering.

7.1.2.4. K-mean clustering technique

K-means algorithm uses an iterative procedure, in which the criterion function is that the distance of pixels from their nearest cluster *centroids*. For the task of partitioning (clustering) N pixels into K disjoint subsets is referred as to minimize the sum-of-squares distance (E) criterion. The distance, E is computed as:

$$E = \sum_{k=1}^{K} \sum_{n \in S_k} | x_n - \mu_k |^2 \tag{7.10}$$

where x_n is the n_{th} pixel and μ_j is the *centroid* of the pixels in S_k. The steps of K-means algorithm are described below:

- Initialize the value of K (number of clusters) and the cluster *centroids*, which are computed from the color histogram as described above.
- Assign each pixel to the *centroid* nearest to that pixel until all pixels have been assigned to K clusters, exclusively.
- Updates the centroid of each cluster by averaging the values of all pixels that belong to the same cluster.
- Verify if the centroid of any cluster still changes markedly. If so, this procedure is repeated from step 2. Otherwise, all pixels are assigned to their corresponding cluster and all the clusters are finalized.

Once the clustering is completed, the pixels in the same clusters would have the similar RGB values that represent a unique color in the document image. The pixels that belong to the different clusters are assumed different. The measure of “nearest” clusters to a given pixel is often performed by computing *Euclidian* distance. The *Euclidian* distance between a given pixel $p(r_p, g_p, b_p)$ and the centroid of a cluster $c(r_c, g_c, b_c)$ in three dimensional RGB color space is defined as:

$$\begin{aligned} & d_E(p,c) = \sqrt{(r_p - r_c)^2 + (g_p - g_c)^2 + (b_p - b_c)^2} = \sqrt{(p-c)^t (p-c)} \\ & \text{and } d_E(p,0) = \| p \|_2 = \sqrt{{r_p}^2 + {g_p}^2 + {b_p}^2} = \sqrt{p^t p} \end{aligned} \tag{7.11}$$

$d_E(p,0)$ is the *Euclidian* norm of p, which follows that all pixels with the same distance of the origin $\| p \|_2 = a$ satisfying ${r_p}^2 + {g_p}^2 + {b_p}^2 = a^2$, which is the equation of a sphere. This means that all pixels of the observation, p contribute equally to the *Euclidian* distance of p from the center. That means *Euclidian* distance is appropriate for pixels that are uncorrelated and have equal variances. However, this is not the case for a low-resolution image. Therefore, in order to overcome the above-mentioned drawbacks a statistical distance *i.e. Mahalanobis* distance is considered to adjust the correlations and different variances. Correlation indicates associations between the pixels. The *Mahalanobis* distance between the pixel, $p(r_p, g_p, b_p)$ and *centroid*, $c(r_c, g_c, b_c)$ is defined as [*Zhang* and *Lu*, 2003]:

$$d_M(p,c)=\sqrt{\left(\frac{r_p-r_c}{s_r}\right)^2+\left(\frac{g_p-g_c}{s_g}\right)^2+\left(\frac{b_p-b_c}{s_b}\right)^2}=\sqrt{(p-c)^t S^{-1}(p-c)}$$

$$d_M(p,0)=\sqrt{\left(\frac{r_p}{s_r}\right)^2+\left(\frac{g_p}{s_g}\right)^2+\left(\frac{b_p}{s_b}\right)^2}=\sqrt{p^t S^{-1} p}\ and\ S=\begin{pmatrix} s_r^2 & 0 & 0 \\ 0 & s_g^2 & 0 \\ 0 & 0 & s_b^2 \end{pmatrix} \tag{7.12}$$

$S=diag(s_r^2,s_g^2,s_b^2)$ is the weight for the corresponding *r*, *g* and *b* values of the pixel and all pixels with the same distance of the origin $\| p \| = a$ satisfy

$$\left(\frac{r_p}{s_r}\right)^2+\left(\frac{g_p}{s_g}\right)^2+\left(\frac{b_p}{s_b}\right)^2=a^2 \tag{7.13}$$

This is the equation of an *ellipsoid* centered at origin with principal axes parallel to the co-ordinate axes. This implies that the pixels with high variability receive less weight than pixels with low variability. Therefore, the computation of the *Mahalanobis* distance for each of the pixels takes the variability of that pixel into account when determining its distance. In the implementation, the distance is computed adaptively by considering the co-variance matrix, *S*. For example, in case of completely correlated or constant pixels, which often occurs in original document image and the matrix, *S* is singular *i.e.* *det*(*S*) = 0 that means the distance computation using *Mahalanobis* distance is not feasible (7.12). Therefore, in this scenario, the *Euclidian* distance is used (7.11). Furthermore, to speed up the clustering procedure, one constraint of convergence rate *i.e.* percentage of pixels are assigned to their respective cluster, is applied. The value of the constraint is of 95% and 98% for the respective original image and captured image. The constraint for the captured document image is higher than the original one and is due to the distortions in the captured image as explained earlier. For example, for an image of size 720 × 540 pixel lines and *K* = 5, the clustering takes less than 3 iterations for a convergence rate of 99%, while the random initialization takes more than 15 iterations to converge. This is due to the fact that the initialization plays a crucial role in K-means clustering algorithm. In some cases of random initialization, it is observed that the perceived *centroid* of the cluster is significantly shifted from the actual *centroid* of the cluster.

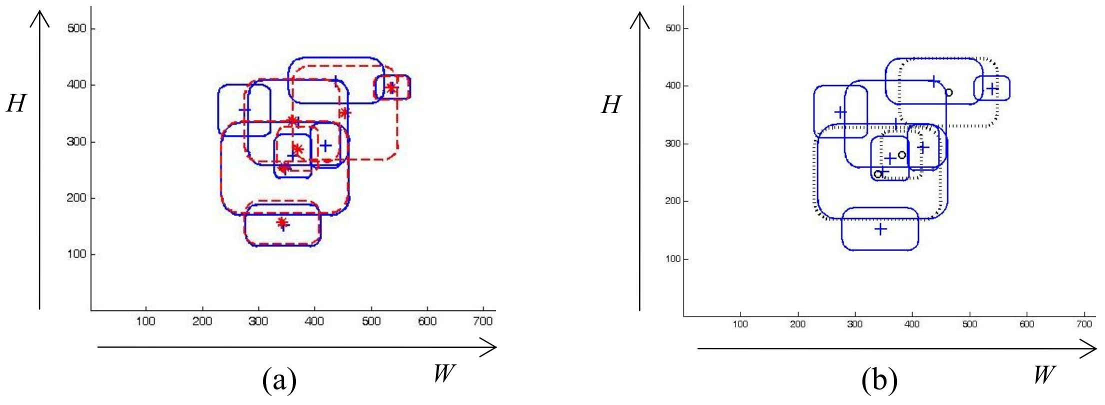
7.1.2.5. Equivalent Rectangle representation

Once the pixels are grouped into *K* number of clusters, then the geometrical distribution of pixels in each cluster is looked for. The geometrical distribution signifies as how pixels that belong to a cluster are physically located in the image 2-D plane rather than in the space of

3-D color histogram. The color features that correspond to the geometrical distribution of pixels in a cluster are represented with *Equivalent Rectangle*. The center and the sides (width and height) of the rectangle are symbolized as the geometrical mean and variances of the location of pixels in the 2-D image plane. The color features corresponding to the geometrical distribution of each cluster, $i = 1 \ldots K$ in the 2-D image plane of resolution $H \times W$, are computed as follows (center of the equivalent rectangle $C_{x,i}$ and $C_{y,i}$, width $R_{x,i}$, and height $R_{y,i}$):

$$\left.\begin{aligned} C_{x,i} &= \frac{1}{N_i}\sum_{\forall p\in i} X_p, C_{y,i} = \frac{1}{N_i}\sum_{\forall p\in i} Y_p \\ R_{x,i} &= \frac{1}{N_i}\sum_{\forall p\in i} (X_p - C_{x,i})^2 \\ R_{y,i} &= \frac{1}{N_i}\sum_{\forall p\in i} (Y_p - C_{y,i})^2 \end{aligned}\right\} i = 1 \ldots K \tag{7.14}$$

The X_p and Y_p are the horizontal and vertical location of pixel p, respectively and N_i is the number of pixels per cluster, i. The other features included in the *Color Signature* are the cluster density $d_i = N_i / (H \times W)$. Moreover, the features related to the color distribution in the 3-D RGB space per cluster, are also extracted in similar fashion and are the mean ($M_{r,i}$, $M_{g,i}$, $M_{b,i}$) and variance ($V_{r,i}$, $V_{g,i}$, $V_{b,i}$) for each channel (7.14). Though these features are not helpful for the identification of captured documents since one faces the problem of non-uniform color shifting in the captured image as compared to the original image and is due to the presence of color cast as mentioned earlier. However, these features are useful for the original documents, which are subjected to identification from a repository containing a set of original documents. The highest peak in the histogram (Fig. 7.4) is considered the background color since the number of pixels in a uniform background is generally significantly greater than that of the foreground, which is obvious in documents such as slides. Fig. 7.5 represents such features with the *Equivalent Rectangle* having a center representing the mean and the width and height of the rectangle representing the variance of the geometrical locations of the pixels in the clusters. The rectangles with solid boundaries are derived from the original image and dotted boundaries, from the captured images. It is observed that the rectangles from the captured image from the projector are closer to the rectangles of the original image (Fig. 7.5a) than those captured from a digital camera (Fig. 7.5b). This is due to color deformations and low-resolution of the captured image as explained earlier.

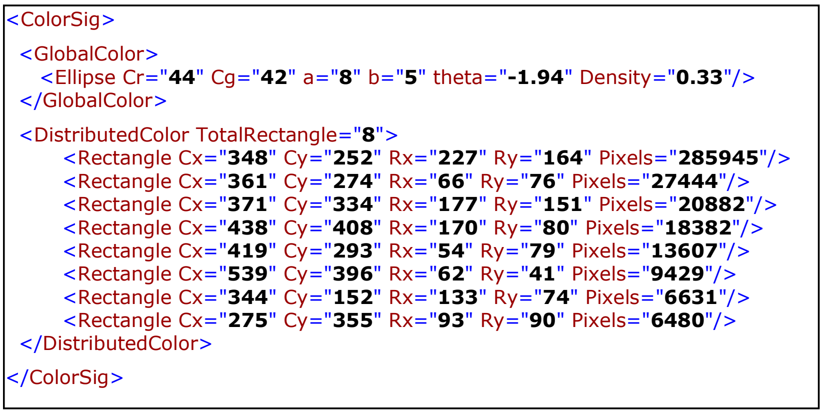


(a) (b)

Fig. 7.5. Clusters are represented with *Equivalent Rectangle* in image plane (a) clusters from original slide (Fig. 7.1a, solid boundary) and output from projector (Fig. 7.2a) (b) clusters from original slide and captured from a digital camera (Fig. 7.2d).

7.1.3. *Structuring of features in Color Signature*

Once the features corresponding to the *Color Signature* are extracted, they are hierarchically structured in the signature for efficient matching. The number of global color features is fixed and independent of the color content in the documents as it is represented with the *EE* (Section 7.1.1). Only the values of the parameters of the *EE* change rather than the number of parameters, which are always six. Therefore, there is no necessity of structuring it. The first node (GlobalColor) of Fig. 7.6 represents the global color features with the six parameters of the *EE*.



```
<ColorSig>
 <GlobalColor>
   <Ellipse Cr="44" Cg="42" a="8" b="5" theta="-1.94" Density="0.33"/>
 </GlobalColor>
 <DistributedColor TotalRectangle="8">
      <Rectangle Cx="348" Cy="252" Rx="227" Ry="164" Pixels="285945"/>
      <Rectangle Cx="361" Cy="274" Rx="66" Ry="76" Pixels="27444"/>
      <Rectangle Cx="371" Cy="334" Rx="177" Ry="151" Pixels="20882"/>
      <Rectangle Cx="438" Cy="408" Rx="170" Ry="80" Pixels="18382"/>
      <Rectangle Cx="419" Cy="293" Rx="54" Ry="79" Pixels="13607"/>
      <Rectangle Cx="539" Cy="396" Rx="62" Ry="41" Pixels="9429"/>
      <Rectangle Cx="344" Cy="152" Rx="133" Ry="74" Pixels="6631"/>
      <Rectangle Cx="275" Cy="355" Rx="93" Ry="90" Pixels="6480"/>
 </DistributedColor>
</ColorSig>
```

Fig. 7.6. *Color Signature* consisting global color features (*Equivalent Ellipse*) and distributed color features (*Equivalent Rectangle*) of the slide document of Fig. 7.1a in XML format.

On the other hand, the distributed color features varies according to the predominant color content of the documents. The distributed color features are represented with the *Equivalent Rectangle* and the number of rectangles symbolizes the number of predominant color content (Section 7.1.2). Since the number of predominant color content varies from document to document, therefore in the signature these color features should be organized hierarchically for efficient matching. The structuring of the distributed color features in the signature are carried out by keeping the clusters' properties in descending order of cluster density, d_i ($i = 1 \ldots K$), since the cluster having the highest number of pixels covers more area in the image than the others. The second node (DistributedColor) of the *Color Signature*, which is represented with XML hierarchy, represents such color features (Fig. 7.6).

7.2. Matching of document *Color Signatures*

The extracted signature of the queried document image is matched with the signatures of all the original electronic documents stored in a repository for identification. The signature of a document consists of two parts, namely (a) *Layout Signature* and (b) *Color Signature*. The document signature, called *Visual Signature*, conveys one's visual perception of a document. In the previous chapter (Section 6.3), the matching of the *Layout Signature* is presented whereas in this section, the matching of the *Color Signature* is presented.

7.2.1. *Global color feature matching*

The matching of the global color features is simple and is based on the comparison of absolute distances between the six parameters of the *EE* of the queried and target signature to a certain threshold, T_G. During the matching of *Visual Signatures*, the purpose of the global color features is to filter down the initial solution set to a reasonable number of solutions for fast and efficient matching on *Layout Signatures*. This is due to the reason that slide documents that belong to a particular presentation, often have the similar background pattern and color, indicating that they share a similar distribution of the kernel density; or in other words, the properties of the *EE* in the *rg*-plane are similar. Therefore, this color features could not be used for the final identification. Once the queried image is identified from a particular presentation, further identification of the slide documents would be performed using either the layout feature-based or distributed color feature-based or combination of both. The evaluation for the matching of this feature alone and the effect of combined features such as distributed color features and layout features is presented in the next chapter (Chapter 8).

7.2.2. Distributed color feature matching

The matching algorithm takes into account the number of *Equivalent Rectangles* and their properties in both the original and queried *Color Signature* of the document's *Visual Signature*. It has been mentioned earlier that the rectangle (cluster) of the highest density represents the background color and the others, the foreground. The matching follows the top-down approach. The rectangle having the highest cluster density is first compared, followed by those in the decreasing order of densities until the last rectangle in the *Color Signature* (Fig. 7.6). Due to the presence of color cast, the number of rectangles in the captured image is often 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. 7.4). The idea is to imitate the geometrical distribution of the clusters of the captured image as in the original image, by merging the rectangles in the original signature and *vice versa* (in case of divergence). This helps to bring the centroids of the resulting rectangles in both images closer. On the other hand, splitting of the rectangles rather than merging is not feasible, since the locations of each pixel are not in the feature set. The properties of each rectangle of the queried signature are first compared using *one-to-one* mappings with the rectangles in the target signature.

In *one-to-one* mappings, each rectangle of the source signature is compared with all the rectangles in the target signature by computing the absolute distance between the properties (center and size) of the respective rectangle. For example, let's say there are K_s and K_t number of rectangles in the respective source and target *Color Signature*. For each rectangle, $i = 1 \ldots K_s$ in the source signature one more matched rectangles, $j = 1 \ldots K_t$ from the target signature is considered for the matching set if and only if the following conditions are satisfied:

a) the difference in cluster density is inferior to a pre-fixed threshold *i.e* $||d_i - d_j|| < T_{D;}$
b) the distance between the center of the respective rectangle is less than a pre-defined threshold *i.e.* $C_i = ||C_{x,i} - C_{x,j}|| + ||C_{y,i} - C_{y,j}|| < T_C$;
c) the difference between the sides of respective rectangle is inferior to a threshold *i.e.* $R_i = ||R_{x,i} - R_{x,j}|| + ||R_{y,i} - R_{y,j}|| < T_S$.

If the matching set contains more than one rectangle, then the rectangle having minimum $(C_i + R_i)$ is considered for the final matched rectangle. The procedure is carried out for each rectangle in the source signature and continued until all the rectangles in the source signature are visited. For any rectangle, if no match is found using the above-mentioned *one-to-one* mappings then the *one-to-many* mappings are followed.

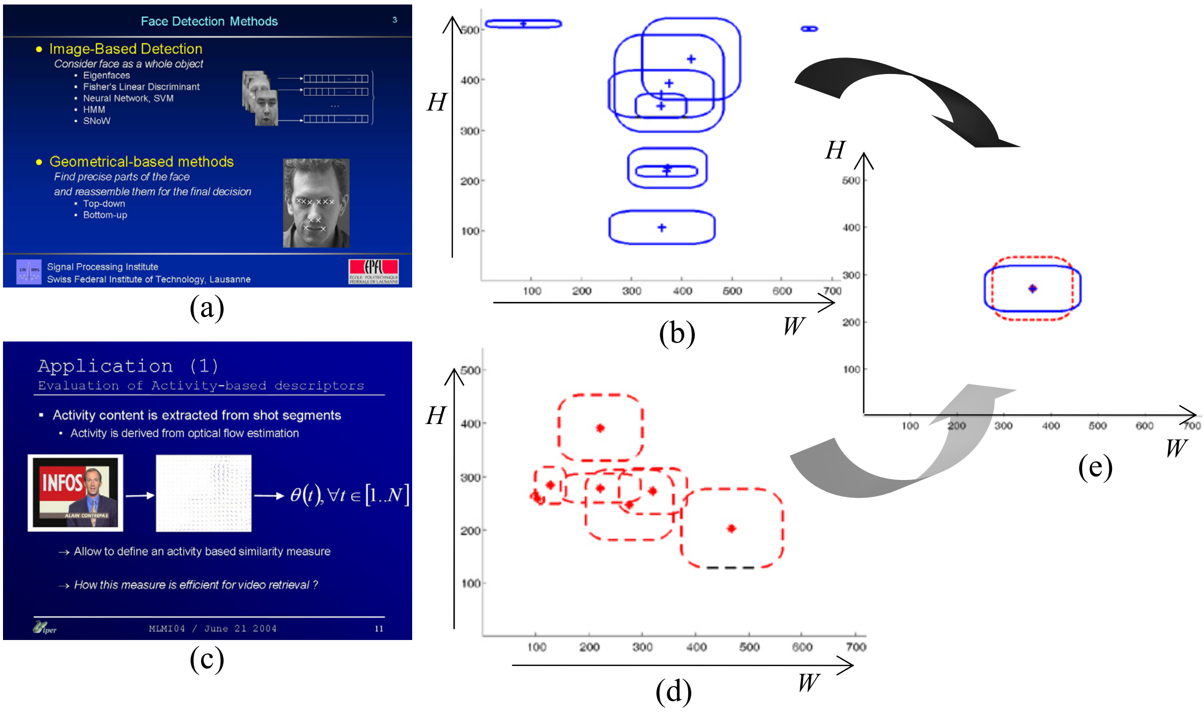
In *one-to-many* mappings, two or more rectangles in the target signature are merged and the resulting rectangle's properties are compared with the source rectangle. Two or more rectangles in the target *Color Signature* are merged if and only if the following conditions are satisfied:

- the resulting cluster density;
- the subsequent geometrical centroid from the merged rectangles are close to the cluster density and geometrical centroid of the source rectangle to which the resulting rectangle is to be compared.

In the same example as mentioned above, for *one-to-many* mappings any two rectangles (p_{th} and q_{th}, $1 \leq p$, $q \leq K_t$) in the target *Color Signature* are merged and compared with the rectangle, i ($1 \leq i \leq K_s$) of the source *Color Signature*. The properties such as the density, the center and the sizes of the resulting rectangle (j) from the merged rectangles (p_{th} and q_{th}) is computed as:

$$\begin{gathered} d_j = (N_p + N_q)/N, \;\; 1 \leq p,q \leq K_t, \; N = W \times H = \#\,total\ pixels \\ C_{x,j} = (N_p C_{x,p} + N_q C_{x,q})/(N_p + N_q) \; and \; C_{y,j} = (N_p C_{y,p} + N_q C_{y,q})/(N_p + N_q) \\ R_{x,j} = (N_p R_{x,p} + N_q R_{x,q})/(N_p + N_q) \; and \; R_{y,j} = (N_p R_{y,p} + N_q R_{y,q})/(N_p + N_q) \end{gathered} \tag{7.15}$$

The resulting rectangle j is considered for the matching 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 matching set contains more than one rectangle, then the rectangle with the center and cluster density closer to the rectangle, i (minimum distance) is finally considered. The above-mentioned procedure is carried out with all possible combinations of the rectangles in the target *Color Signature* and after merging, the matching procedure is the same as the *one-to-one* mappings except the size of rectangle. If no match is found, then the same procedure is carried out with *many-to-one* followed by *many-to-many* mapping with the same matching procedure, as explained above. In the process of merging, the maximum number of rectangles for merging is restricted to half of the total number of rectangles. This restriction is imposed to reduce the number of solutions in the final set. For example, for most slide documents if one merges all the rectangles of a signature to a single rectangle, the final rectangle often has the similar properties as that of the target one. This is due to the reason that often people use pre-defined slideshow templates. This is illustrated in Fig. 7.7 as the *Equivalent Rectangles* from two different slides are merged to form the respective single rectangle and the properties of both the final merged rectangle is very similar in both center and sides (Fig. 7.7e).


Face Detection Methods

3

- Image-Based Detection
  Consider face as a whole object
  - Eigenfaces
  - Fisher's Linear Discriminant
  - Neural Network, SVM
  - HMM
  - SNoW
- Geometrical-based methods
  Find precise parts of the face
  and reassemble them for the final decision
  - Top-down
  - Bottom-up

Signal Processing Institute
Swiss Federal Institute of Technology, Lausanne

Application (1)
Evaluation of Activity-based descriptors

- Activity content is extracted from shot segments
  - Activity is derived from optical flow estimation

$\theta(t), \forall t \in [1..N]$

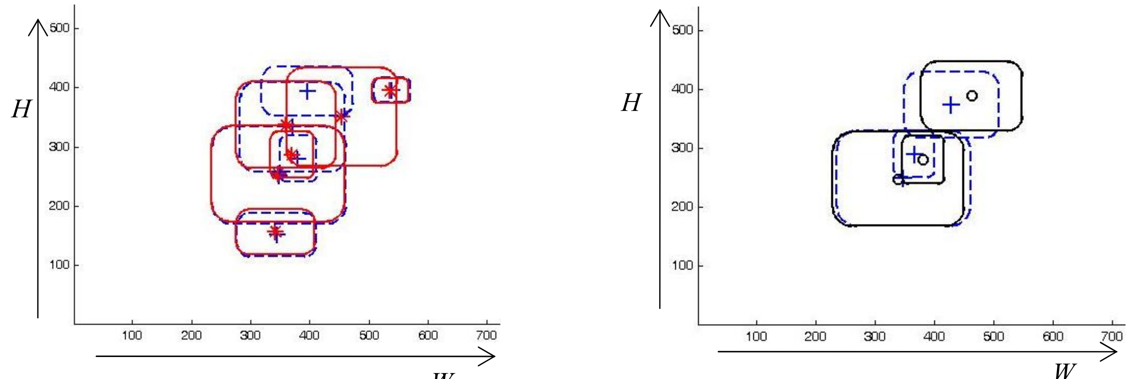
→ Allow to define an activity based similarity measure

→ *How this measure is efficient for video retrieval ?*

MLMI04 / June 21 2004 11

Fig. 7.7. Merging of *Equivalent Rectangles*: (a) and (c) original slide from two different presentation, (b) and (d) corresponding *Equivalent Rectangles*, (e) the *Equivalent Rectangles* of each image is merged to represent single rectangle.

In the current scenario of slide documents, after the evaluation of over hundred slides the threshold, T_D for the cluster density is set to 20% of the maximum density of the source and the target rectangles. Similarly, the threshold for the center, T_C and sizes, T_S for the comparison of source and target rectangle is set as 20 and 10, respectively. Fig. 7.8 shows an example of such a merging of *Equivalent Rectangles*. Before merging, there are 8 rectangles in the original image and only 6 in the captured image from the output of a projector. The rectangles in the projector image are compared using the *one-to-one* mapping and then followed by *one-to-many* mappings to form the final 6 rectangles in the original image for the comparison. Similarly, the rectangles in both the original image and the captured image from DV camera are merged to three. This is the optimal number of rectangles among all possible combinations to bring them closer. This merging procedure considers all the possible mappings *i.e.* starting from *one-to-one* followed by *one-to-many*, *many-to-one* and finally *many-to-many*. From the figure, it is easily understood that the clusters are brought closer after using the above-mentioned 'merge-and-compare' algorithm.



(a) (b)

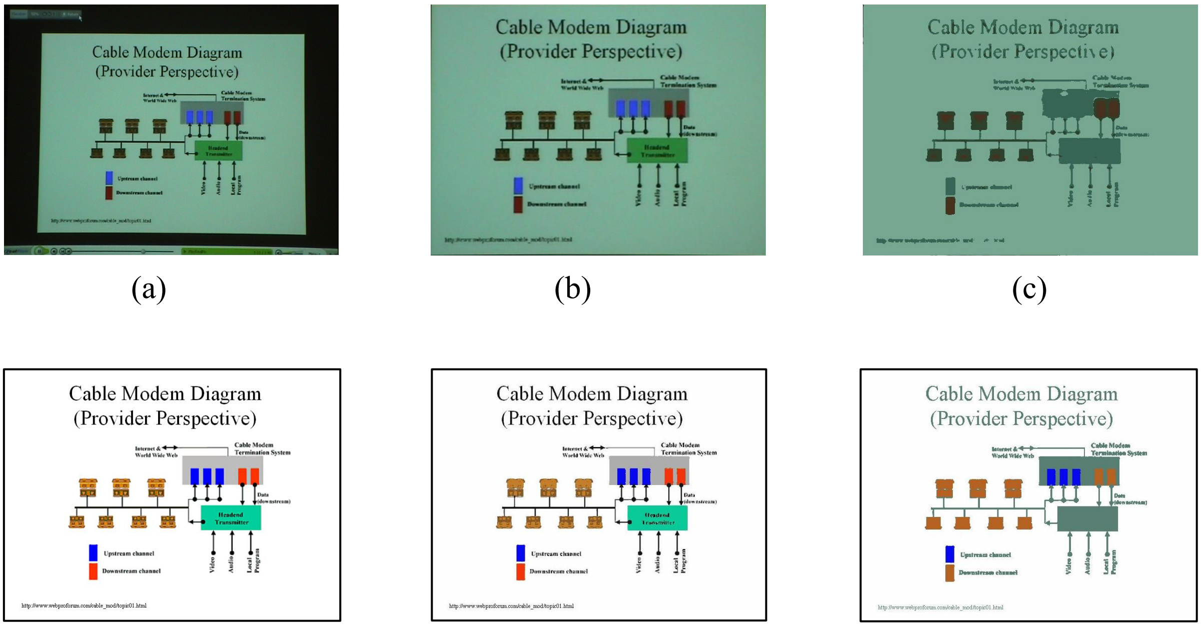
Fig. 7.8. (a) Rectangles from the original slide image are merged (dotted) to imitate the rectangles in the captured image from the projector (Fig. 7.5a), (b) rectangles from both the original and the captured image from a digital camera are merged to bring them closer for the comparison.

The above-mentioned 'merge-and-compare' algorithm would be easily understandable from the effect of merging of clusters (rectangles) reflected on the document image (Fig 7.9). There are a total of 8 clusters in the original image (Fig. 7.9e) and 4 clusters in the captured image of the original one before merging (Fig. 7.9c). In this example, the clusters in the original image are merged to 4 in order to imitate the geometrical color distribution of the captured image. From the figure, it is clear that after merging, the geometrical distribution of colors (*i.e.* the physical location of the pixels that belongs to a particular cluster) of the final image of the original (Fig. 7.9f) and the captured (Fig. 7.9c) is quite close as compared to before merging.

7.3. Conclusion and Perspectives

In this chapter, the procedure for the extraction of color features to form a document *Color Signature* is explained. The *Color Signature* is a part of the document *Visual Signature*, which is combined with the *Layout Signature* (Chapter 6). The *Color Signature* mainly consists of two feature sets, one is the global color feature set which is represented with the *EE* having six parameters and the other one is the distributed feature set, which is represented with the *Equivalent Rectangles* with one rectangle per color of the document image.

The *EE* is computed from the density surface, which is computed using the non-parametric kernel density estimation from the normalized RGB color histogram. In order to avoid the color cast, which is the superimposed predominant color in the captured document image, the *EE* representation of the global color content is proposed. Furthermore, the total number of features in this feature set is reduced to six for fast matching and to avoid unnecessary storage space. Moreover, the global color features is used to filter down the solution set to a reasonable smaller number of solutions rather than using for the final identification. This method has been suggested by considering real-world presentation, since design templates are often used and thus, the slides of that presentation would have the same design templates, layout and color content. The matching of the *EE* is implied by simply considering the *Euclidian* distance between the parameters of the ellipses.



(a) (b) (c)

(d) (e) (f)

Fig. 7.9. (a) Captured slide image using a handheld digital camera, (b) pre-processed captured image, (c) formation of image after clustering (4 clusters), d) original slide image, (e) formation of image after clustering (8 clusters) and (f) formation of image after merging of clusters in original image to 4 clusters.

The proposed distributed color features for the *Color Signature* are considered because the color in the captured document images often vary with the lighting conditions, properties of the capture devices and the compression technique used. Nevertheless, the physical location of pixels in the document image is distorted, which justified using color information to make the document identification more robust. The features are extracted by grouping the pixels of similar color using K-means clustering. The initialization to the K-means clustering is done by considering the predominant peaks from the RGB color histogram and is fast and efficient as compared to the adaptive clustering technique. Then each cluster is represented with *Equivalent Rectangles* by computing the geometrical mean and variance of the location of pixels that belongs to that cluster. The number of rectangles represents the number of major colors in the document images. The extracted rectangles are structured in the *Color Signature* by considering the cluster density of that rectangle. It is organized in the decreasing order of the cluster density. The matching of the rectangle is carried out using a *one-to-one* method followed by *one-to-many* succeeded with *many-to-one* and *many-to-many* mappings between the source and target *Color Signatures*. The matching compares the properties of the rectangle such as center, sides and cluster density.

The evaluation of the matching of *Color Signature* alone and combined with the *Layout Signature* is presented in next chapter (Chapter 8). Though the proposed *Color Signature* is used in this work for the identification of the captured slide documents, it could also be used for a wide range of applications, *e.g.* the pattern matching and identification of photographic images consisting of natural scenes, cities, mountains, etc.

Chapter 8

Visual Signature evaluation: combining *Layout* and *Color Signature*

The document *Visual Signature* is a representation of the document with three different feature sets, which are computed by processing of document image. These feature sets consist of one set for the *Layout Signature* (Chapter 6) and two sets for the *Color Signature* (Chapter 7). The document *Visual Signature* combines the three above-mentioned feature sets. The encapsulated signature-based representation of documents is considered in order to identify them fast and efficiently for real-time applications. Our proposed *Visual Signature* describes the visual aspects of documents.

In this chapter, the procedure for the construction of the *Visual Signature* of the document using the above-mentioned three feature sets is explained in Section 8.1. The identification of low-resolution documents is carried out with the matching of the *Visual Signature* and is described in Section 8.2. The major part of this chapter describes the experiments and evaluations of the proposed signature-based identification and is presented in Section 8.3. Finally, this chapter concludes with future perspectives in Section 8.4.

8.1. Constructing document *Visual Signature*

The document *Visual Signature* consists of two signatures called *Layout Signature* (Chapter 6) and *Color Signature* (Chapter 7). The extraction and structuring of layout features to form the *Layout Signature* and its matching is detailed in Chapter 6. Similarly, the extraction of the feature sets for the *Color Signature* and their matching is explained in Chapter 7. The document *Visual Signature* hierarchically structures these feature sets for an efficient matching. The priority of the feature sets are shown in Fig. 8.1 with the left node, *i.e.* the *Layout Signature* being of higher priority than the *Color Signature*. At the intermediate and leaf level, the priority decreases from left-to-right (Fig. 8.1). The layout feature set of the *Layout Signature* is given higher priority as it is considered as a local feature. The priority of individual feature inside the layout feature set is explained in Chapter 6. Next to the layout feature set, the distributed color feature set is given higher priority followed by the global color feature set of the *Color Signature*. This is due to the fact that often the global color content of the slide documents belongs to a particular presentation and thus shares a common global color rather than the distributed color. The idea of structuring the *Visual Signature* is that it helps to narrow down the scope of search during the matching to find a fruitful solution.

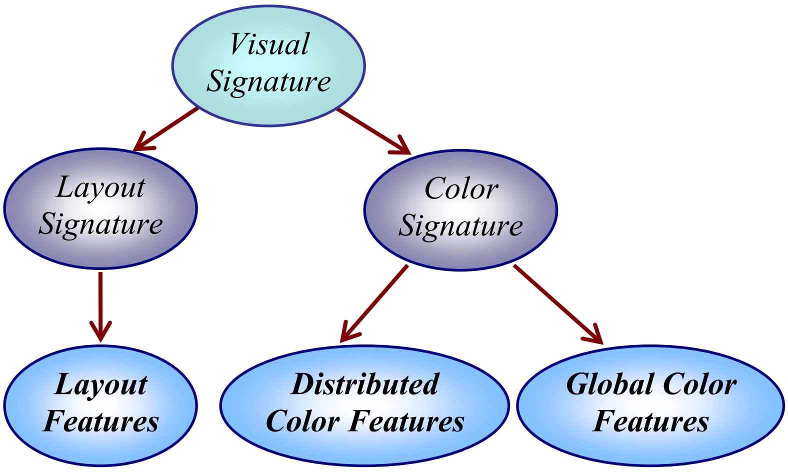


Fig. 8.1. The top-down structure of the *Visual Signature* of a document consists of layout and *Color Signature* at the intermediate level and the corresponding feature sets at the leaf level.

8.2. Matching of *Visual Signatures*

In order to identify a captured document from a document repository, the *Visual Signature* of the queried document must be matched with the corresponding *Visual Signatures* of the documents kept in the repository. It is mentioned earlier that before commencing the matching procedure, all documents in the repository are processed for the extraction of their corresponding *Visual Signature*. The matching procedure considers the global-to-local criteria *i.e.* global feature sets, which is of lower priority, are first compared to filter down the initial larger solution set to a reasonable smaller solution set (Fig 8.1). This criterion is considered in order not to lose the target solution at the initial comparison. Then, the final identification is done using local feature set (highest priority feature set) in combination with the global feature set. This global-to-local matching procedure helps to speed up the matching rather than the comparison of each feature set in the source and target signature. The effect of the global-to-local matching could easily be understood in the next section of experiments and evaluations. However, to consider the effect of both global and local feature set, the possible combination of the various feature set has been considered. The matching of the individual feature set is the same as it is described in the respective chapters (*i.e.* layout feature set in Chapter 6, global color feature set and distributed color feature set in Chapter 7). In this chapter, the matching strategies of the orders for various feature set are explained and analyzed.

8.3. Experiments and evaluations

The evaluation of the above-mentioned *Visual Signature*-based identification is carried out by capturing the projected documents using various capture devices. First, the corpus for the evaluation has been constructed. As it is mentioned earlier in Chapter 4, various presentations available on the web and related to private, public and academic institutions have been collected. Therefore, more than three hundred presentations have been accumulated, which represent different varieties of presentation styles. During this accumulation, the presentations are not organized according to their various characteristics like: number of slides, background color, font color and size variability, background variability, graphics content, etc. They are just deposited in the repository with the image version (JPEG) of each slide in the presentations, after which all the images in the repository are processed for the extraction of their corresponding *Visual Signature* and kept in the repository. Once, an image is received for identification, the system does the same with it for the extraction of its *Visual Signature*. The *Visual Signature* of the queried image is then matched with all the *Visual Signatures* in the repository. If a match is found, the corresponding image of the matched signature is supplied by the system.

8.3.1. Experimental setup

The experimental setup is prepared by considering all the available types of capture devices in the market. The possible devices for the capturing projected documents could be categorized into two groups and is based on mainly the effect of lighting condition of the capture environment and device characteristics:

- The capture devices such as web-cams, digital still cameras, digital video cameras and mobile phones with camera, whose image quality are dependent on both the lighting conditions and device characteristics. It is also dependent on the position (distance and orientation) of the capture device with respect to the target object. Furthermore, the surface properties (*e.g.* reflectance) of the target object also plays a role in the quality of the captured document images. This category can be further distinguished into two groups: (a) *controlled* (the position of the captured devices is fixed by an operator), (b) *uncontrolled* (the position is not fixed).

- The second category of capture devices consists of those, for which the quality of the captured images does not depend on either the environmental condition or the surface properties of the target objects. It only depends on the characteristics of the capture devices, *e.g.* the output from a projector, capture card, etc. In case of projector connected to the output of the laptop/pc/workstation, the output from the projector is directly captured as a video stream or sequence of images. Similarly, the input of the capture card is connected to a laptop/PC/workstation and the capture card digitizes the input signal and saved it as a video stream or image sequences.

Following data sets are considered for the evaluation of our proposed *Visual Signature*-based identification of low-resolution captured documents.

8.3.1.1. Smart Meeting Room data set

In the *Smart Meeting Room* of the *University of Fribourg*, the projected documents are captured using the following hardware:

- *FireWire webcams*: Fire-i™ Digital Camera.
- *Digital video camera*: Sony, DCR-TRV27E, PAL, 1 mega pixels.
- *Digital still camera*: Sony, DSCP7 Cyber-shot 3.2 mega pixels.

Video streams containing projected documents captured using a webcam/DV-camera are processed using our proposed *SCD* (Chapter 4) for the extraction of the image of each projected document. These images of captured documents are then, to be identified from the repository containing original electronic documents. The resolution of the captured documents using various capture devices are given below:

- *Webcam*: 640 × 480 pixel lines with 96 *dpi*, the projected part in the captured image is 345 × 270 pixel lines (*controlled*).
- *DV-camera*: 720 × 576 pixel lines with 96 *dpi*, the projected part in the captured image is 560 × 450 pixel lines (*controlled*).
- *Digital still camera*: 2048 × 1536 pixel lines with 72 *dpi*, the average size of projected part in the captured images is around 1400 × 1000 pixel lines (*uncontrolled*).

8.3.1.2. MLMI'04 data set

The data captured from the international workshop on *Multimodal Interaction and Related Machine Learning Algorithms 2004* (*MLMI'04*) is used for our evaluation. In this workshop, there are 32 scientific presentations, which are captured and archived. Following hardware is used:

- *Digital video camera*: ParkerVision pan-tilt-zoom camera (www.parkervision.com).
- *Projector*: 3M multimedia projector MP8749.

The captured data includes the audio-video recordings, projected documents from projector and the video stream of the projected documents. The presented documents mainly consist of text, graphics as well mathematical formulas. The specialty of these documents is that each electronic presentation file has unique design pattern. 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 the projector having a dimension of 1036 × 776 pixel lines 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 images of the projected documents are also extracted from the compressed video stream (DivX compression, www.divx.com), which is captured using the ParkerVision pan-tilt-zoom camera. A total of 674 documents have been first extracted from the stable period using the *SCD* algorithm (Chapter 4). The dimension of the captured document images is 720 × 576 pixel lines with the resolution of 96 *dpi*, whereas the dimension of the projected part is 590 × 518 pixel lines.

8.3.2. Evaluation procedure and evaluation metrics

An application has been developed for the automatic evaluation of the proposed *Visual Signature*. The application first considers the repository containing original electronic documents and then takes an image or a set of images as input to deliver the respective identified documents from the repository. For our evaluation procedure, the matching is performed by considering the whole repository without organizing the electronic documents in the repository according to their order of appearance during the presentation. *Mukhopadhyay* and *Smith* applied the above-mentioned restrictions along with the matching procedure restricted to slideshow-wise rather than whole repository [*Mukhopadhyay* and *Smith*, 1999]. It is mentioned earlier that the *Visual Signatures* corresponding to the original electronic documents are stored in the repository along with the original documents images. The queried images are first processed to build their corresponding *Visual Signatures* and then, the application finds the best matching document *Visual Signature*. The application returns original document, which correspond to the matched *Visual Signature*.

Recall and *Precision* metrics are used for the evaluation. The computation of these performance metrics are different from those used for the evaluation of the proposed *SCD* algorithm in Chapter 4. For document identification, performance metrics *Recall* (*R*), *Precision* (*P*) and *F-measure* (*F*) are defined as:

$$R = \frac{D_c}{D_c + D_n}, P = \frac{D_c}{D_c + D_f}, F = \frac{2 \times R \times P}{R + P}$$

$$\text{Where } \begin{cases} D_c = \ \#\text{correct documents retrieved} \\ D_n = \ \#\text{documents not retrieved} \\ D_f = \ \#\text{incorrect documents retrieved} \\ D_c + D_n = \#\text{total documents queried are in the repository} \\ D_c + D_f = \#\text{total documents returned by the system} \end{cases} \tag{8.1}$$

In this evaluation, the queried documents are already in the repository. Thus, the system returns either a document (correct or wrong) or null, when the system could not make a trustful decision. Hence the following rules could be stated: $(D_c + D_f \leq D_c + D_n) \Rightarrow (D_f \leq D_n) \Rightarrow (R \leq P)$ and $(D_n - D_f)$ is the number of times the system returns null. *R* conveys the identification rate from the user part (*i.e.* ratio of the correct documents identified upon total documents queried) whereas *P* expresses the trustfulness (accuracy) of the answer returned by the system (*i.e.* ratio of the correct documents identified upon total documents returned by the system). Finally, *F* corresponds to the combined performance of both *R* and *P*. As mentioned earlier, the queried documents are already in the repository and therefore, another way to evaluate the result is to consider the identification rate and *Rejection Rate*. The identification rate is the same as the above-mentioned *Recall*. The *Rejection Rate* is defined as the ratio of number of times the system returns null (rejected) upon the total number of documents queried *i.e. Rejection Rate* = $(D_n - D_f) / (D_c + D_n)$. The specification of the system used for the evaluation is 1.7 GHz, 512 MB and Pentium 4 PC.

8.3.3. Performance of the document Layout Signature

In this section the performance of the document *Layout Signature* contained in the document's *Visual Signature* is presented. In Chapter 6, the extraction procedure of the layout features, hierarchically structured to form the *Layout Signature* and their matching procedures (exhaustive and hierarchical) have been explained. For this experiment, matching algorithms are applied to the *Smart Meeting Room* data set in which images are captured using a FireWire webcam (Section 8.3.1.1). A total of 626 slide documents (16 presentations) are captured and queried to a repository containing more than 1000 original slide documents.

The performances of the three matching algorithms have been compared: two exhaustive searches (with and without weighting mechanisms) and one hierarchical search (Chapter 6). The results are displayed in Table 8.1. For the exhaustive search, the average *Recall* (66%) and *Precision* (71%) with weighted features is better than the respective *Recall* (62%) and *Precision* (67%) without weight, which tells about the benefits of having weighted layout features for matching.

Table 8.1. Results of evaluation for images captured using a webcam

Matching method	Recall (R)	Precision (P)	F-measure (F)	Rejection rate
Exhaustive without weight	0.62	0.67	0.65	0.07
Exhaustive with weight	0.66	0.71	0.68	0.07
Hierarchical	0.54	0.91	0.65	0.41

The hierarchical search drastically increases the average *Precision*, *i.e.* the trustfulness of the system (Table 8.1), indicating that when the system retrieves a document, the result can be trusted in 91% of the cases. The low *Recall* value (54%) is due to either the removal of solutions in the initial matching step, which is the bounding box comparison, or due to the fact that not even a single feature of the *Layout Signature* qualifies the minimum matching threshold (Chapter 6, Section 6.3). However, this could be corrected by setting up the various thresholds properly by the hierarchical algorithm and by enhancing the bounding box extraction procedure as well as the matching technique, so that no good solution is removed in the initial step. Finally, from a preliminary study, it seems that the hierarchical search is only 2-3 times faster than the exhaustive search. However, with the increasing number of images in the repository, this ratio would proportionally grow and the proposed hierarchical search would become greatly necessary for real-time applications, in order to avoid uninteresting search spaces. However, in exhaustive search with weight only 71% (67% without weight) of the answers are correct as compared to hierarchical search, which is of 91%. The rejection rate in hierarchical search is higher *i.e.* 41% as compared to exhaustive search which is only 7%. This conveys that the system returns null in case it is unable to take a decision rather than providing an incorrect solution.

In the exhaustive searches, two sets of slide images gave back *Recall* and *Precision* values inferior to 40%, which drastically decreased the overall average performance. But the performance is better in the hierarchical search for the same set of images [*Behera et al.*, 2004b]. Indeed, in the exhaustive search, the solution having the highest non-zero global score is returned and it may not be the correct one, whereas in the hierarchical search, the score is feature-specific and is compared to a threshold for acceptance in each feature level.

Table 8.2. Results of evaluation for images captured using a digital still camera

Matching methods	Recall (R)	Precision (P)	F-measure (F)	Rejection rate
Exhaustive without weight	0.46	0.73	0.56	0.37
Exhaustive with weight	0.64	0.78	0.70	0.18
Hierarchical	0.43	0.71	0.54	0.39

In the second set of experiments, with the *Smart Meeting Room* data set in which retrieval accuracy of the projected slides captured using a digital still camera (3.2 Mpixel) is measured (Table 8.2). A total of 20 different images were taken during the presentation and queried to the same number (1000) of slides in the repository. In this case the performance is comparatively lower than that of the web-cam. The main reason is that the distance varies and the camera could be rotated in comparison to web-cam, which is fixed. This experiment is performed without perspective corrections of the captured image, for a preliminary evaluation. A more subsequent evaluation of such captured images is presented in the following section.

In this evaluation process, both the *Recall* and *Precision* should be increased in order to improve the identification accuracy from both part of the user and system. The lower values of *Recall* and *Precision* are due to a) the existence of tables, small font size (< 10 points), and complex figures, which obstructs the extraction of effective layout features for the *Layout Signature* and b) in some cases, the extracted slide images were so bad (distorted or too small) that the matching gave back no result. The number of documents (D_c) retrieved correctly could be increased by enhancing the bounding box extraction procedure along with the matching technique as explained in Chapter 6. Both *Recall* and *Precision* are directly proportional to D_c (R and $P \propto D_c$), therefore the increment in D_c not only increases R and P but also decreases D_n (correct documents not retrieved) and D_f (incorrect documents retrieved). Moreover, P is inversely proportional to D_f ($P \propto 1 / D_f$) and since D_f could be decreased by tuning the threshold so that null is returned instead of incorrect documents, then P could be increased without affecting *Recall* and D_c. In the following experiments, there is an improvement in performance by considering the other feature sets.

8.3.4. Rules for combining two or more features set

The document's *Visual Signature* contains more than one feature sets (Fig. 8.1). The *Color Signature* of the *Visual Signature* contains the global feature sets and the *Layout Signature* consists of local feature sets. The local features are more powerful than the global ones for the identification of low-resolution captured documents. Often, projected documents have the same global features (global and distributed color features). 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 for the final identification. The *Color Signature* contains two types of features set, global color feature set and distributed color feature set (Fig. 8.1). For the identification of captured documents, the distributed color feature set would be more trustworthy than the global color feature set as it dictates the distribution of the similar pixels in the image X-Y plane, whereas the global color feature set contains the global color content (Chapter 7). Therefore, the matching of documents using only global color feature set is not feasible as most of the documents have a similar color content, which is often distorted due to the presence of color cast. To identify the captured documents, the matching of the signatures is carried out individually using each feature set as mentioned above and combination of them. For the combined feature sets matching, the global-to-local rule has been used. Using global feature set, it is easy to identify a set of similar documents as most of them share the same global feature. However, using global feature set, it is hard to take the decision for the exact match as all the documents in the set share common global features. For example, consider all the slide documents from a real-world presentation, in which majority of the slide documents contain same global color content. This is because presenters often use the existing template for the preparation of presentations. Moreover, the template for the presentation file *i.e.* either PPT (Microsoft PowerPoint) or PDF (Adobe Acrobat) varies from person to person. Furthermore, the same template is often kept throughout of his/her presentation. Using global color feature set, it is hard to identify the exact slide document as most of the slide documents that belong to a presentation or a particular template have the same color content. Nevertheless, it is easy to identify a set of documents that belongs to a particular presentation or a template using the global color feature set. Whenever, two or more feature sets are considered for the identification, two types of matching rules are proposed for the identification of captured documents. One is sequential matching and the other is fused matching. In case of sequential matching, the matching procedure considers a single feature set at a time; followed by another till the desired solution is reached. In this scenario, some solutions are sometimes removed too early due to the fact that the first feature set tested is weaker as compared to all other existing features. Therefore, a better strategy to overcome this drawback consists in fusing the features sets. In case of fused matching, the global feature set is used to filter down the solution set to a reasonable smaller set and then the fusion of the global and local feature sets are used for the final identification. While comparing the global feature sets, the threshold for matching is set in such a way that no solution should be removed in the initial step of comparison. The rule uses the global to local feature set matching and the order is from right-to-left of the leaf nodes of Fig. 8.1. The fusion of the feature sets is explained in the Section 8.3.7.

8.3.5. Performances of sequential matching

In this experiment, the effect of sequential matching of feature set is considered. It is mentioned earlier that in case of sequential matching the matching procedure performs the global-to-local matching. The two global feature set such as global color and distributed color feature sets of the *Color Signature* is considered (Chapter 7). It would be interesting to see the effect of global feature set in the performance as compared to only local features set such as layout features of the *Layout Signature* (Chapter 6). The main goal is to reduce the matching time without degrading the performance using the global feature sets to filter down the solution set prior to identification using only local layout feature set.

8.3.5.1. First experiment: effect of global color feature

The first set of experiment is carried out by considering the effect of global color feature set *i.e.* the parameters of *EE* as described in Chapter 7, on the layout feature set of the *Layout Signature*. First, all the original slide images in the repository are filtered out according to their global color similarity (*EE*) with a queried document image that reduces the size of the search space. The slide documents having the *EE* close (distance inferior to a threshold T_c) to the color feature of the queried image are considered. Let $S = \{s_1, s_2,\ldots, s_n\}$ be the set of signatures in the repository. After the matching of *EE*, a new set $S_c = \{s_1, s_2,\ldots, s_m)$ is derived from S where $m \leq n$. Then, the layout-based visual feature matching is performed on the set S_c for the final detection of the queried slide images. For the matching of layout features of the *Layout Signature*, the exhaustive search with weight (Chapter 6) is used as it gives a better performance than others.

In this evaluation, the *Smart Meeting Room* data set is used, in which 310 projected slides from 14 different presentations have been captured using the digital video camera (Section 8.3.1.1). The captured documents are queried on a repository containing 1500 slide documents from 45 different presentations. In this evaluation, prior to the extraction of color and *Layout Signatures*, the captured images are pre-processed for the corrections of geometrical deformations as described in Chapter 5. The identification rate, comprising *Recall* and the rejection rate metrics are used for the evaluation.

Table 8.3. Results of sequential matching of global color and layout feature set for images captured using a digital video camera

Matching procedures	**Search space**	**Identification rate**	**Rejection rate**	**Matching time in second**
Layout feature set of *Layout Signature*	1.00	0.88	0.02	2.71
Global color feature set of *Color Signature* plus layout features	0.42	0.90	0.02	1.29

The first row of Table 8.3 represents the results for the matching of layout features alone; whereas the last row shows the results for the combined feature sets (sequential), *i.e.* global color plus layout features. The identification rate of the combined feature sets is slightly better than the layout features alone (90% and 88%, respectively). Even if in the tested repository, most of the slides have little color variations, the average search space has been reduced to 42% when using the global color feature, which is an encouraging result for more colorful repository [*Behera et al.*, 2005a]. The matching time is dependent on the size of the repository and for each *Layout Signature* the matching time is directly proportional to the number of elements in each feature node, which is dependent on the physical content of the corresponding document (Chapter 6). For the global color feature, the matching time is dependent only on the global color content and thus the number of parameters corresponding to *EE* is constant for each comparison. Therefore, in the combined features, not only the identification rate is improved but also the identification time is reduced due to the reduction in number of matching parameters. In the worst scenario, the number of elements in the search space could be the same as in the whole repository when all the documents have similar color content. The matching time using the sequential feature set is even less than 50% of the matching time of layout feature set alone. This is due to the reduction of search space using global color features. Identification using global color feature set alone is not a good idea as most of the slide documents that belong to a presentation or a template often shares the same color content. However, in the next section (Section 8.3.8) the evaluation is also performed using individual feature sets.

8.3.5.2. Second experiment: effect of distributed color feature

In the second set of experiment, the effect of other global feature set *i.e.* the distributed color feature set of the *Color Signature* on the layout feature set of the *Layout Signature* is examined. The procedure for the matching strategy is the same as the global-to-local feature set matching. In this evaluation, the distributed color feature set used for the reduction of the solution set and then the layout feature set of *Layout Signature* is used for the final identification. The procedure is the same as in the first experiment except that the global color feature set is replaced by the distributed color feature set. The threshold for the comparison of distributed color features is set in such a way that no solution should be left out in the reduced solution set. The detailed procedure for matching of the distributed color feature set, which is represented with *Equivalent Rectangle*, is explained in Chapter 7. In this evaluation set, the size of the repository as well as the size of query set is increased in comparison to previous set. A total of 355 slide documents from 16 different presentations are captured using the same digital video camera and queried on a repository containing 2000 slide documents from 60 different presentations. The captured document images are pre-processed for the geometrical deformations as described in Chapter 5. The same metric of identification rate and rejection rate is used for the evaluation.

Table 8.4. Results of sequential matching of distributed color and layout feature set for images captured using a digital video camera

Matching procedures	**Search space**	**Identification rate**	**Rejection rate**	**Matching time in second**
Layout feature set of *Layout Signature*	1.00	0.79	0.02	4.02
Distributed color feature set of *Color Signature* plus layout features	0.16	0.89	0.03	1.20

In Table 8.4, the average identification result is shown. It shows that not only the identification rate of the color and layout (89%) is better than the layout alone (79%) but also the drastic reduction of the search space (16%). This is due to prior filtering with the distributed color feature set of the *Color Signature*, which results in the reduction of the processing time (1.2 sec). Recall from the last experiment, the reduction of search space was only 42% whereas in this case, it is much lower even though the size of repository is increased from 1500 to 2000. This is due to the fact that the distributed color feature set is more powerful than the global color feature set and is explained in detail in Chapter 7. Due to the increment in the size of repository, the matching time for the layout features of the *Layout Signature* is increased from 2.71 to 4.02 second. This is obvious as the matching time increases with the size of repository. The performance using only layout features is decreased (79%) as compared to the last experiment (88%). This is due to the addition of 45 more slide documents from two more presentations to the query set in comparison to the last experimental set up. The identification of these two presentations is the lowest (35% and 37%), which is due to the non-uniform (vertical color gradient), textured background [*Behera et al.*, 2005b]. In the *Layout Signature*, the dark part of the background is segmented as an image, which is the same for all slides in the presentations. In the case of the *Color Signature*, the distributed color features approximated the color gradient with 3 different clusters, which are represented with the corresponding *Equivalent Rectangle* (Chapter 7). The centroids of these clusters are different due to the variation of the foreground objects, which helps in identification. This is a very encouraging result for handling the non-uniform background. The rejection rate is nearly 2% and is the same as the previous experimental set up. Upon analysis of the results for rejected documents, it is observed that the documents are too noisy for such comparison.

8.3.5.3. Third experiment: combining all

In the third set of experiments, the evolution is carried out using the *MLMI'04* data set (Section 8.3.1.2). Since the document videos from the *MLMI'04* conference are available presentation-wise, one could identify the extracted document from the video, like-wise. However, for the document image-based retrieval of the captured conference, the queried document image should be compared with all the documents presented during the conference. For this purpose, in this experimental setup, the queried document image is compared with the whole repository (Table 8.5) and presentation-wise (Table 8.6). For this evaluation, the matching of layout features of the *Layout Signature*, the exhaustive search with weight

(Chapter 6) is used and the metrics of recognition rate and rejection rate are used as in previous experiments. Moreover, in this experiment, the evaluation is carried out by comparing the individual feature set as well as the sequential matching of feature set by considering the global-to-local rule and using the possible combination of three feature sets such as layout, global color and distributed color feature set.

Table 8.5. Evaluation of the performance using single and sequential matching of various feature set by considering whole repository

Matching of feature set	**Captured images from projector's output**			**Captured images using ParkerVision camera and compressed using DivX**		
	Identification rate	*Rejection rate*	*Time in second*	*Identification rate*	*Rejection rate*	*Time in second*
F_1	0.93	0.002	1.57	0.73	0.074	1.54
F_2	0.16	0.000	0.33	0.01	0.000	0.34
F_3	0.86	0.005	0.42	0.54	0.008	0.37
F_2 then F_1	0.96	0.000	1.43	0.73	0.000	1.33
F_3 then F_1	0.96	0.000	0.43	0.78	0.050	0.48
F_2 then F_3	0.87	0.000	0.39	0.62	0.030	0.36

The feature sets F_1: layout features of the *Layout Signature*, F_2: global color feature set (*EE*) of the *Color Signature* and F_3: distributed color feature set (*Equivalent Rectangle*) of the *Color Signature* (Fig. 8.1).

Table 8.6. Evaluation of the performance using single and sequential matching of various feature set by considering presentation-wise

Matching of feature set	**Captured images from projector's output**			**Captured images using ParkerVision camera and compressed using DivX**		
	Identification rate	*Rejection rate*	*Time in second*	*Identification rate*	*Rejection rate*	*Time in second*
F_1	0.94	0.003	0.15	0.78	0.133	0.15
F_2	0.25	0.000	0.06	0.08	0.000	0.06
F_3	0.89	0.002	0.10	0.68	0.004	0.09
F_2 then F_1	0.96	0.000	0.12	0.79	0.000	0.14
F_3 then F_1	0.96	0.000	0.08	0.82	0.020	0.11
F_2 then F_3	0.92	0.000	0.07	0.75	0.001	0.08

As mentioned before, the feature set F_2 and F_3 consist of the global color features. The performance using these feature set alone is not efficient (2nd and 3rd rows of Table 8.5 and 8.6). When feature set, F_2 and F_3 are combined with the feature set F_1, which is the local visual feature set, the respective increment in performance of 80% and 10% for the images from projector's output (Table 8.5; 4th and 5th rows). This is due to the fact that the slide documents in a presentation often have the same color but different layout structure. The standalone performance of the global color feature set, F_2 is quite inferior to that of distributed color feature set, F_3. It shows that the global color content of the slide documents in a slideshow does not vary significantly, whereas the distribution of the similar pixel (color) in the 2-D image plane does. However, the performance using the global color feature set, F_2 is quite good and is due to the fact that the color of some of the slide documents in a presentation is significantly different than the rest of the slide documents. The performance using feature set, F_1 is lower as compared to the performance of F_1 combined with F_2 and F_3 (4th and 5th rows of Table 8.5 and 8.6) since the slide documents in some of the presentations have either a non-uniform (gradient variation) and/or complex background (textured), which creates intricacy in the extraction of local visual feature set, F_1. In most of the cases, the whole slide of such background is considered as a single image feature of the feature set, F_1. Therefore, in this case by combining the local and global feature set, not only increases the identification rate but also reduces the signature matching time, in seconds as compared to a single feature set. The performance of the images captured from the projector's output 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. This introduces errors during the extraction of the features for the respective signatures. Furthermore, this property mainly affects the layout features of the *Layout Signature*, which is considered as the local features and is more reliable than the other features. In the next section, the fusion strategy for the above-mentioned feature sets is explained. In this evaluation, ideally the rejection rate should be zero as the queried documents are already present in the meeting repository (Table 8.5 and 8.6). Therefore, the system should return a solution, which could be either correct or false. Moreover, in some cases it is observed that the system 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 digital video camera. In this evaluation, the identification is done presentation-wise as well as considering the whole repository. The standalone and combined performance of all the feature sets for the identification in the presentation-wise evaluation (Table 8.5) is much better than that of considering the whole repository. This is obvious since in the case of presentation-wise identification, the target repository is restricted to the slide documents of a single presentation rather than the whole repository containing the slide documents from all presentations. The presentation-wise identification also consumes less matching time as compared to the whole repository due to the reduction of number of documents in the target repository.

8.3.6. Fusion strategies

In sequential matching of the feature sets as described above, the required solution is sometimes removed in the initial steps during the filtering of the solution set. Furthermore, in some cases the feature set used for filtering is more powerful than that of feature set used for the final identification. For example, in this case the distributed color feature set, F_3 is used for filtering down the solution and then layout feature set, F_1 is used for final identification. In case the image is too noisy then the whole document is often considered as a single image feature of the layout feature set, F_1. In such cases, the distributed color feature set is more powerful than the layout feature set, which is considered for the final identification. In order to avoid the drawback above, it is better to fuse the various features set for the final identification rather than considering only the local visual feature set. However, for the real-time applications, the matching time should be as less as possible without degradation of the performance. Therefore, it is necessary to reduce the solution set to a reasonable numbers and then the final identification could be carried on by fusing the various feature sets. Furthermore, it is often observed that several slide documents have the same local visual features but different global features. Therefore, the final identification by matching only local visual features results in incorrect solution. For example, the last slide of various presentations often contains the text ‘Thank You’ or ‘Question’ only. Most of them are having same local visual features; however they often differ in the global features such as the color content. Therefore, it is necessary to combine both the feature sets for an efficient identification.

The fusion of two or more feature sets is possible if they are from the same feature space otherwise it could produce an erroneous fusion. In this scenario of three feature sets, the layout feature set, F_1 and the distributed color feature set, F_3 are extracted by considering the document image plane (Chapter 6 and Chapter 7). Whereas, the global color feature set, F_2 is extracted from the normalized 2-D color histogram of documents. Therefore, the feature set F_2 could not be fused with the other two, whereas feature sets F_1 and F_3 could be considered for the fusion. For the fusion of feature sets F_1 and F_3, two types of fusion strategies are considered and are of (1) linear fusion and (2) non-linear fusion. In the linear fusion, the final matching score is computed by simply summing up the scores obtained from the comparison of two feature sets F_1 and F_3 ($S_f = F_1 + F_3$). The final identification is based on the final score *i.e.* the solution having the highest score is the winner.

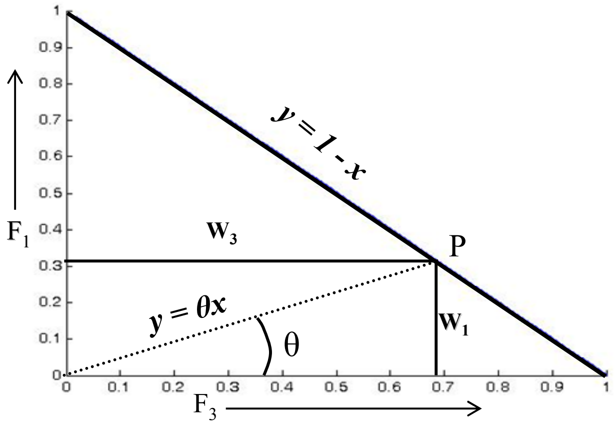


Fig. 8.2. Derivation of weighting function W_1 and W_3 for the non-linear fusion using the matching score of feature sets, F_1 and F_3.

In case of non-linear fusion, the final score is computed from the standalone score by multiplying a weighting function to individual score *i.e.* $S_f = W_1 F_1 + W_3 F_3$, where $W_1 + W_3 = 1$ and $W_1 = f(F_1, F_3)$. The above weighting function is derived using the Fig. 8.2. As it is mentioned earlier that both the features sets F_1 and F_3 belong to same feature space, therefore the matching score of both feature sets are normalized to one. The angle θ between the score is computed as $\theta = \tan^{-1}(F_1 / F_3)$, then the point P, which is the intersection of the straight lines $x + y = 1$ and $y = \theta x$, is located (Fig. 8.2). The vertical and horizontal distances of the point, P are the respective weight of W_1 and W_3.

$$\begin{gathered} W_1 + W_3 = 1 \text{ and } \frac{F_1}{F_3} = \frac{W_1}{W_3} \\ W_1 + W_1 \frac{F_2}{F_1} = 1 \Rightarrow W_1 = \frac{F_1}{F_1 + F_3} \text{ and } W_3 = \frac{F_3}{F_1 + F_3} \end{gathered} \tag{8.2}$$

Using the value of W_1 and W_3 from the above equation (8.2), the non-linear fused final matching score is computed as:

$$S_f = \frac{F_1^2}{F_1 + F_3} + \frac{F_3^2}{F_1 + F_3} \tag{8.3}$$

The above non-linear fusion (8.3) as well as the linear fusion strategy is used for the evaluation.

8.3.7. Performance using fused layout and color features

The above-mentioned fusion strategies have been evaluated using the same data set presented in the second experiment (Section 8.3.5.2). The evaluation is carried out by extending the top *N* solutions rather than the top one solution as in the second set of experiments. The value of *N* is assigned to top-one, best-five and best-ten. This implies whether the solution is present in a set, which consists of top-*N* solution. In this experiment, only the identification rate is considered as it was seen that the rejection rate was 2-3% in the second experiment and could be acceptable. The result of this experiment is shown in Table 8.7.

Table 8.7. Results of sequential versus fused matching of distributed color and layout feature set for images captured using a digital video camera

Matching using fusion strategies of feature sets F_1 and F_3	**Identification rate**		
	Best-one	**Top-five**	**Top-ten**
Sequential matching ($F_3 \rightarrow F_1$)	0.89	0.92	0.93
Linear fusion $F_1 + F_3$	0.88	0.90	0.91
Non-linear fusion using equation (8.3)	0.91	0.94	0.95

In case of sequential matching of the features set, the performance is 89%, whereas it is 92% and 93% of the time within the top five and top ten, respectively. Theoretically, the performance of linear fusion (88%, 90% and 91% for respective best one, top five and top ten) should be better than the sequential matching performance. However it is not the case. After analyzing some of the solutions, it is observed that the distributed color feature (F_3) score of some solutions, which are close behind the layout feature's (F_1) matching score of the final solution, is higher. In the sequential matching the solution having the highest layout feature (F_1) score is picked up. However, in linear fusion the maximum of summation of both feature score is considered and the solution having a higher distributed color feature's (F_3) and close behind the layout feature's (F_1) score is picked up. Nevertheless, the performance degradation in comparison to the sequential matching is very less, which is less than 2%. However, this drawback is overcome by applying the non-linear fusion. In the non-linear fusion, the weighting function itself is the function of both the feature sets F_1 and F_3. The weighting function gives more weight to the higher score than the lower ones by computing from feature sets F_1 and F_3. Let's assume two solutions, s_1 and s_2 in the final set and having matching score of $s_1(F_1) = 0.8$, $s_1(F_3) = 0.1$ and $s_2(F_1) = 0.7$, $s_2(F_3) = 0.3$ for the respective feature sets, F_1 and F_2. If the sequential matching is considered then the final solution is s_1 as $s_1(F_1) > s_2(F_1)$. For the linear fusion the final solution is s_2 as $s_2(F_1) + s_2(F_2) > s_1(F_1) + s_1(F_2)$. In case of the non-linear fusion using equation (8.3), the matching score for s_1 is 1.38 and s_2 is 0.58. Therefore, in this case both sequential and non-linear fusion gives the correct result whereas the linear fusion picks the wrong one. In case of the non-linear fusion of the feature sets, the results are obtained as 91%, 94% and 95% for the respective best one, top five and top ten solutions. The non-linear performance is nearly 3% higher than the sequential matching and is better than the linear fusion [*Behera et al.*, 2005c].

From the experiment mentioned above, it is observed that the performance of non-linear fusion of the feature sets F_1 and F_3 gives a better result than the sequential matching and linear fusion. Therefore, the above non-linear fusion strategy is applied to the matching of document *Visual Signature* for the evaluation of the *MLMI'04* data set (Section 8.3.1.2), which has been presented in the third experimental set up is used. In the third experiment, the sequential matching of the possible combinations of various feature sets such as layout feature set (F_1) of *Layout Signature*, global color feature set (F_2) and distributed color feature set (F_3) of the *Color Signature*. In this evaluation the fusion of feature sets F_1 and F_3 is used for the final identification in the reduced solution set as they belong to the same feature space as explained earlier. The filtering of solution set is applied using the rule of global-to-local feature set (Fig. 8.1) *i.e.* feature set, F_2 is first used to filter candidates followed by feature set, F_3 for further filtering and finally the non-linear fusion of features score of F_1 and F_3 is used for identification the final solution. In this evaluation, the best solution is considered rather than the top-N solution. The evaluation is carried out presentation-wise and the whole repository, using the performance metrics of identification rate and rejection rate as presented before. The performance of the identification of documents using their *Visual Signature* is presented in Table 8.8.

Table 8.8. Evaluation of the matching of document *Visual Signature*

Matching of document *Visual Signature*	**Captured images from projector's output**			**Captured images using ParkerVision camera and compressed using DivX**		
	Identification rate	*Rejection rate*	*Time in second*	*Identification rate*	*Rejection rate*	*Time in second*
Whole repository	0.97	0.02	1.03	0.83	0.014	0.95
Presentation-wise	0.98	0.02	0.14	0.87	0.05	0.13

In this evaluation, the performance of document *Visual Signature* by considering both the presentation-wise and whole repository using non-linear fusion of color and layout features is better than the sequential matching of the respective feature sets. In case of the non-linear fusion, though the performance improvement is less (1-2%) as compared to sequential matching for the comparatively high resolution images from the projector's output, however it is significantly increased (5%) for the low-resolution images captured using the handheld devices.

8.3.8. Performance comparison

The performance of the method has been evaluated using the *MLMI'04* data set. The same data has been used by *Daddaoua et al.* by applying different version of *OCR* systems such as *Trans2*, *TransAll* and *TransBest* for keyword-based retrieval [*Daddaoua et al.*, 2005]. However, they used *OCR* only on considerably high resolution images such as those captured from the projector's output without considering images from handheld devices (digital video camera). Furthermore, the computational cost for the *OCR* performance is not mentioned. The *OCR* performance is presented slide-wise, presentation-wise and using whole repository. For their evaluation, they used metric of *term recall TR* and *term precision TP* (Table 8.9).

Table 8.9. Performances of various *OCR* systems on slides captured from the projector's output in the *MLMI'04* data set [*Daddaoua et al.*, 2005]

OCR method	**slide**		**presentation**		**database**	
	TR	*TP*	*TR*	*TP*	*TR*	*TP*
Trans2	72.4	77.3	67.5	77.0	71.4	77.4
TransBest	77.0	78.4	72.6	77.3	76.7	79.0
TransAll	80.9	65.5	76.2	62.3	80.8	62.0

It would be possible to identify the captured documents by comparing its text strings with those in the original documents. However, if one considers *OCR* and the proposed *Visual Signature*-based identification, the latter performed much better (Table 8.8 and 8.9). Moreover, the identification rate of low-resolution documents from handheld devices is significantly better than that of the *OCR* performances using comparatively high resolution images (projector's output). The best performance for *OCR* using slideshow-wise is 76.2% whereas the identification rate of the proposed method is 98%.

The performance of our proposed *Visual Signature*-based identification method is also compared with the DCT-based image matching with truncated coefficients, proposed by *Chiu* *et al.* [*Chiu et al.*, 2000b]. The evaluation is carried out using *MLMI'04* data set (Section 8.3.1.2).

The performance of DCT-based method (Table 8.10) for high quality document images (projector output) is comparable with our *Visual Signature*-based method (Table 8.8). However, the performance is significantly poor (25%) for the low-resolution documents captured using handheld devices as compared to our method (83%), with comparable matching time (0.95 sec). Moreover, the performance using single feature set such as layout features (73%) and distributed color features (54%) of our *Visual Signature* (Table 8.5), is much better than the DCT-based method for identification of low-resolution documents. In the case of DCT-based approach, the presentation-wise matching performance is better than considering the whole repository. However, in comparison to our method (Table 8.8), the performance is far low (Table 8.10), even if one considers one feature sets (layout, distributed color) of our *Visual Signature* (Table 8.6).

Table 8.10. Performance evaluation of the matching method proposed by *Chiu et al.*

Matching	**Captured images from projector's output**			**Captured images using ParkerVision camera and compressed using DivX**		
	Identification rate	*Rejection rate*	*Time in Second*	*Identification rate*	*Rejection rate*	*Time in Second*
Whole repository	0.96	0	0.92	0.25	0	0.91
Presentation-wise	0.96	0	0.08	0.66	0	0.09

8.4. Conclusions and perspectives

In this chapter, the formation of the document *Visual Signature*, which combines layout and *Color Signatures*, is presented. The proposed *Visual Signature* contains three feature sets: layout feature (local), global color and distributed color (global) features, which are presented in previous chapter. The major part of the chapter describes the evaluation of the proposed signature-based identification. The matching of signatures is presented using various strategies such as the sequential matching of various feature sets as well as the linear and non-linear combinations of feature sets. The evaluation is performed by considering the captured images from various available handheld devices as well as the output from a video projector. The evaluation uses real data captured during the International Workshop on *Multimodal Interaction and Related Machine Learning Algorithms* (*MLMI* 2004). The proposed signature-based identification excelled the existing approaches such as the use of various *OCR* systems and DCT-based image comparison. Furthermore, the retrieval of the original electronic documents, as well as the related audio-visual streams can be retrieved by querying the images captured from the handheld devices. The matching time for the signature-based identification is fast and could easily be applicable to real-time applications.

Chapter 9

Conclusions and perspectives

This thesis has investigated methods for developing an efficient system for capturing, analyzing and indexing multimedia data captured from multimodal environments such as meetings, conferences, lectures, etc. Nowadays, all the activities in a multimodal environment are captured and archived for later access and browsing. Therefore, the main purpose for building such a system was to fulfill the needs of the ever-growing content-based multimedia archiving, indexing, retrieval and management. In such environments, the *human-to-human* interactions often use different modalities such as voice, gesture, etc. In this thesis, we have shown the role of documents as an additional modality to bridge the gap between temporal data (*e.g.* audio/video) and static information (documents). Furthermore, the presented documents exhibit an important source of information, pointers to other captured multimedia streams and provide natural interfaces for browsing multimedia recordings. We constructed the whole system using the following four tasks: a) recording, b) detection of documents in the audio-visual stream captured using low-resolution handheld devices, c) identification of the detected documents from the meeting repository containing original electronic documents and d) retrieval of multimedia data by querying captured/original presented documents and/or keywords.

9.1. Major achievements and contributions

The major contributions of the thesis are described in the following sections along with a brief summary related to above-mentioned tasks.

9.1.1. Recording task

The presented recording system (Chapter 3) synchronously captures the raw data of the meeting room activities using various capture devices connected to it. The architecture of the system is of a *master-slave* model, which is simple, distributed, light-weight, scalable and can accommodate a wide range of capture devices. The synchronization among various capture devices is one of the most important aspects of the system and is handled by defining a global clock in the *master* PC to which all the capture boxes communicate co-operatively.

9.1.2. Document-based segmentation task

The proposed technique for segmentation (Chapter 4) of meetings/lectures is based on projected documents, which are captured as a video stream by our recording module. The state-of-the-art scene change detection algorithms could not be used since the video is captured from a fixed camera and contain only projected documents. The proposed method uses feature-based algorithms that consider the stability rather than changes in video sequences and out-performed the existing state-of-the-art techniques. Among the novel findings of our proposed method are that it does not need to consider the order in which the documents are presented, which is not the case in every presentation. Moreover, it does not require document identification methods to confirm the change of documents in a video sequence. Furthermore, it overcomes the auto-focusing characteristics of light-weight capture devices such as webcams, which takes nearly half a second to capture stable images, once there is a change of documents. During the transition period, each frame in the sequence is significantly different and the existing state-of-the-art methods detect numerous document changes in the transition period, whereas our stability-based approach overcomes this problem.

9.1.3. Document identification task

Our proposed document identification method is based on the matching of *Visual Signature* (Chapter 5), which contains hierarchically structured features extracted from the visual information of the document content. There are two sub-signatures called document *layout signature* (Chapter 6) and document *Color Signature* (Chapter 7) which constitute the *Visual Signature*. The novelty of our proposed technique is that it combines low-level visual features such as color, its spatial distribution and the layout features of the document, for an efficient identification. The main objective was to handle very poor quality captured documents *i.e.* low-resolution and varying lighting conditions, which is quite difficult to analyze using the current document image analysis and retrieval systems. Our proposed matching strategies based on sequential and multi-level fusion of features out-performed the existing state-of-the-art approaches (Chapter 8). Furthermore, the performance of multi-level fusion strategy excelled over the sequential matching strategy. The proposed signature-based matching is fast, efficient and can be easily applicable to real-time applications. Another interesting aspect of our method is that it allows a robust content extraction of poor quality document images for later *Information Retrieval* (IR). Indeed, the content can be extracted efficiently from the original electronic documents rather than using *OCR* on the low-resolution captured images. Use of *OCR* on low-resolution document images not only would give poor performance but is also computationally expensive and requires different packages to handle different languages.

9.1.4. Retrieval and browsing task

This thesis further proposed a document- and keyword-based retrieval and browsing method. The document-based storyboard was generated for an effective browsing of the captured multimedia data. The main idea behind the document-based storyboard is that often presenters' talk evolves around the content of the projected documents. Concerning the keyword-based retrieval, the system extracts keywords from the presented documents using natural language processing techniques and compares them with the queried keywords. An initial cluster of documents corresponding to the queries is presented to the user. Then, the user can select according to his/her document of interest. The major achievements of this document-based browsing/retrieving method is that one can enquire for the recorded meetings/lectures to the system using the projected/discussed documents (electronic or image). This adds document-centric meetings/lectures on demand in addition to the existing browsing task.

9.2. Limitation of the current work and possible future research directions

In the current system, the captured multimedia data is segmented by considering the projected documents, which are captured as a video stream using a camera focused on the projector screen. The proposed segmentation technique (*SCD*) is not applicable to captured video streams containing mixed shots of the presenter and of the projected documents. For example, the recording of classroom lectures, in which one stream is often captured with shots of both the instructor and projected slides and switching between them is done by an operator. One way to extend the system for segmenting such videos is to separate the video containing projected documents from the rest. Then the proposed *slide change detection* technique (Chapter 4) can be used on the segments containing documents for further segmenting to the granularity of slide-level.

One of the future aspects to work on is the identification of occluded documents. This often happens, due to the presence of the speaker or of a stick pointer in front of the screen during the presentation. This results in the identification of partial documents.

The other future aspect to continue is the detection of scrolling and/or zooming of documents during a presentation which could be annotated during indexing. Furthermore, this scrolling and/or zooming also results in identification of partial documents. The other information related to presentation such as detection of laser pointers, on the fly modification of documents or use of markers to highlight the important content could be handy for annotation. Moreover, detection of video clips, animation, web pages embedded in slides and browsing of web pages during the presentation also provides useful information, which could provide other indices for indexing.

Future investigations on the improvement in performance of identification of poor quality captured document images need to be perused for developing solid document detection and recognition applications and related multimedia annotation and retrieval applications. The following future research could be performed for improving performances of such systems:

- *Multi-level document image segmentation*: This would be useful for the extraction of features for the *Layout Signature* (Chapter 6). The layout features are extracted from the black-and-white strip of the document image. However, if the document has a textured and/or gradient background, the extracted features might not be correct, and therefore lead to erroneous identification result. In our proposed signature-based technique, this drawback is overcome by distributed color features of the *Color Signature* (Chapter 7). However, the layout features can be extracted using multi-level document segmentation in which each color component of the document image could be considered prior to layout feature extraction using a clustering technique. Afterwards, our proposed layout feature extraction technique would be applied to each color cluster and the connected component analysis (CCA) is then used for the final layout features from the cluster-wise layout features. Therefore, the current excellent identification result obtained could further be improved. This is due to the fact that the layout features are more effective than the color features.
- *Color cast correction*: This could be applied to the captured low-resolution color document images, which often suffer from the superimposed non-uniform color called color cast. The color cast varies from one capture device to another as well as the lighting condition and distance from the target objects. A solid color cast correction technique, which is independent of capture devices as well as capture environments, could improve the extraction and matching of color features that would result in the improvement of identification performances.

- *Extraction of embedded textual content*: The embedded text in the figures/images, which are incorporated in electronic documents, can not be extracted using our current *Xed* API. This textual content often illustrates the corresponding figures/images and would be useful information for keyword-based search and retrieval. Therefore, these can be extracted using available *OCR*, once figures/images are segmented from the corresponding documents.
- *Mathematical expressions*: They can be automatically translated to the set of keywords using *optical formula recognition* (OFR). This also helps in keyword-based indexing and retrieval applications.

9.3. Impact of this thesis

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. However, the challenge is to be able to deliver a system that could handle low-resolution documents, which are often captured using handheld devices since the existing approaches to document management are yet to support this category of documents. Therefore, the outcome of our research has high impacts on prominent areas such as:

- mobile translation, which extracts and translates textual contents and visual signs for tourism and visually impaired people;
- detection of multiple instance of same documents (duplicates);
- digital media asset managements;
- document cataloging.

Therefore, our proposed low-resolution document identification method imposes a novel technique towards an efficient management of documents captured from handheld devices. Furthermore, our proposed document integration method with other digital media narrows down the gap between the temporal data (audio, video, etc.) and static information (documents), which was one of the main goals of this thesis. Further details on this work can be found in the publications (Curriculum Vitae) that have been published in various research domains: document analysis (International Conference on Document Analysis and Recognition), document engineering (DocEng) and multimedia systems (International Conference on Multimedia and Expo, International Journal of Signal Processing, International Conference on Pattern Recognition and Computer Vision).

Bibliography

[1] G. D. Abowd, C. G. Atkeson, C. H. A. Feinstein, C. Hmelo, R. Kooper, S. Long, N. N. Sawhney and M. Tani. Teaching and learning as multimedia authoring: The classroom 2000 project. In *Proc. of ACM Multimedia*, pp. 187–198, November 1996.

[2] P. Aigrain, H. Zhang and D. Petkovic. Content-based representation and retrieval of visual media: a state-of-the-art review. *Multimedia Tools and Applications*, 3(3), pp. 179–202, 1996.

[3] J. Ajmera, G. Lathoud and I. McCowan. Clustering and segmenting speakers and their locations in meetings. In *Proc. of IEEE ICASSP*, pp. 605–608, May 2004.

[4] M. Amlani and R. Kasturi. A query processor for information extraction from images of paper-based maps. In *Proc. of RIAO*, pp. 991–1000, March 1988.

[5] A. Behera, D. Lalanne and R. Ingold. Combining color and layout features for the identification of low-resolution documents. *Int'l Journal of Signal Processing*, ISSN: 1304–4478, Vol. 2(1), pp. 7–14, 2005a.

[6] A. Behera, D. Lalanne and R. Ingold. Enhancement of layout-based identification of low-resolution documents using geometrical color distribution. In *Proc. of ICDAR*, pp. 468–472, August-September 2005b.

[7] A. Behera, D. Lalanne and R. Ingold. Influence of fusion strategies on feature-based identification of low-resolution documents. In *Proc. of ACM Symposium on Document Engineering*, pp. 20–22, November 2005c.

[8] A. Behera, D. Lalanne and R. Ingold. Looking at projected documents: Event detection and document identification. In *Proc. of IEEE Intl. Conf. on Multimedia and Expo*, pp. 2127–2130, June 2004a.

[9] A. Behera, D. Lalanne and R. Ingold. Visual signature based identification of low-resolution document images. In *Proc. of ACM Symposium on Document Engineering*, pp. 178–187, October 2004b.

[10] M. Bett, R. Gross, H. Yu, X. Zhu and Y. Pan *et al*. Multimodal meeting Tracker. In *Proc. of RIAO*, April 2000.

[11] M. H. Bianchi. AutoAuditorium: a fully automatic, multi-camera system to televise auditorium presentations. In *Proc. of Joint DARPA/NIST Smart Spaces Workshop*, http://www.autoauditorium.com/nist/autoaud.html, July 1998.

[12] A. D. Bimbo. Visual information retrieval. *Morgan Kaufmann Publishers, Inc.*, 1999.

[13] J. S. Boreczky and L. A. Rowe. Comparison of video shot boundary detection techniques. In *Proc. of SPIE Storage and Retrieval for Still Image and Video Databases IV*, Vol. 2670, pp. 170–179, 1996.

[14] J. A. Brotherton. Enriching everyday experiences through the automated capture and access of live experiences: eClass: building, observing and understanding the impact of capture and access in an educational domain. *PhD Thesis*, December 2001.

[15] J. A. Brotherton, J. R. Bhalodia and G. D. Abowd. Automated capture, integration, and visualization of multiple media streams. In *Proc. of IEEE Int. Conf. on Multimedia Computing and Systems*, pp. 54–63, 1998.

[16] R. Brunelli, O. Mich and C. M. Modena. A survey of the automatic indexing of video data. *Journal of Visual Communication and Image Representation*, Vol. 10(2), pp. 78–112, 1999.

[17] H. Bunke and P. S. P. Wang. Handbook of character recognition and document image analysis. *World Scientific Publishing Co. Pte. Ltd.*, 1997.

[18] J. F. Canny. A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 8(1), pp. 679–698, 1986.

[19] R. Cattoni, T. Coianiz, S. Messelodi and C. M. Modena. Geometric layout analysis techniques for document image understanding a review. *Technical Report ITC-IRST*, 1998.

[20] W. Y. Chen and S. Y. Chen. Adaptive page segmentation for color technical journals' covers images. *Image and Vision Computing*, Vol. 16(3), pp. 855–877, 1998.

[21] P. Chiu, A. Kapuskar, S. Reitmeier and L. Wilcox. NoteLook: Taking notes in meetings with digital video and ink. In *Proc. of ACM Multimedia*, pp. 149–158, October-November 1999.

[22] P. Chiu, A. Kapuskar, S. Reitmeier and L. Wilcox. Room with a rear view: Meeting capture in a multimedia conference room. *IEEE Multimedia*, Vol. 7(4), pp. 48 – 54, 2000a.

[23] P. Chiu, J. Boreczky, A. Girgensohn, and D. Kimber. LiteMinutes: An internet-based system for multimedia meeting minutes. In *Proc. of 10th Int'l World Wide Web Conference*, pp. 140–149, May 2001.

[24] P. Chiu, J. Foote, A. Girgensohn and J. Boreczky. Automatically linking multimedia meeting documents by image matching. In *Proc. of ACM Hypertext*, pp. 244–245, 2000b.

[25] A. Criminisi, I. Reid and A. Zisserman. A plane measuring device. *Image and Vision Computing*, Vol. 17 (8), pp. 625–634, 1999.

[26] R. Cutler, Y. Rui, A. Gupta, J.J. Cadiz, I. Tashev and L. He *et al.* Distributed meetings: A meeting capture and broadcasting system. In *Proc. of ACM Multimedia*, pp. 503 – 512, December 2002.

[27] N. Daddaoua, A. Vinciarelli and J.-M. Odobez. OCR based slide retrieval. In *Proc. of ICDAR*, pp. 945–949, August-September 2005.

[28] D. Doermann. The indexing and retrieval of document images: A survey. *Technical Report LAMP-TR-0013*, University of Maryland, February 1998.

[29] D. Doermann, E. Rivlin, and I. Weiss. Applying algebraic and differential invariants for logo recognition. *Machine Vision and Applications*, Vol. 9(2), pp. 73–86, 1996.

[30] D. Doermann, H. Li and O. Kia. The detection of duplicates in document image databases. In *Proc. of ICDAR*, pp. 314–318, August 1997.

[31] D. Drivas and A. Amin. Page segmentation and classification utilizing bottom-up approach. In *Proc. of ICDAR*, pp.610–614, August 1995.

[32] J. P. Eakins and M. E. Graham. Content-based image retrieval. *A report to the JISC technology applications programme*, Institute for Image Data Research, University of Northumbria, October 1999.

[33] S. Elrod, R. Bruce, R. Gold, D. Goldberg and F. Halasz *et al.* Liveboard: A large interactive display supporting group meetings, presentations and remote collaboration. In *Proc. of CHI*, pp. 599–607, May 1992.

[34] B. Erol, J. J. Hull and D. S. Lee. Linking multimedia presentations with their symbolic source documents: algorithm and applications. In *Proc. of ACM Multimedia*, pp. 498 – 507, November 2003.

[35] A. W. Fitzgibbon, M. Pilu and R. B. Fisher. Direct least squares fitting of ellipses. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 21(5), pp. 476–480, 1999.

[36] D. Franklin, S. Bradshaw and K. J. Hammond. Jabberwocky: you don't have to be a rocker scientist to change slides for hydrogen combustion lecture. In *Proc. of Intelligent User Interface*, pp.98–105, January 2000.

[37] C. Garcia and X. Apostolidis. Text detection and segmentation in complex color images. In *Proc. of ICASSP*, pp. 2326–2330, June 2000.

[38] J. E. Garyand and R. Mehrotra. Similar shape retrieval using a structural feature index. *Information Systems*, Vol. 18(7), pp. 525–537, October 1990.

[39] W. Geyer, H. Richter and G. D. Abowd. Towards a smarter meeting record—capture and access of meetings revisited. *Multimedia Tools and Applications*, Vol. 27(3), pp. 393–410, 2005.

[40] W. Geyer, H. Richter, L. Fuchs, T. Frauenhofer, S. Daijavad and S. Poltrock. A team collaboration space supporting capture and access of virtual meetings. In *Proc. of ACM SIGGROUP Conf. on Supporting Group Work*, pp. 188–196, 2001.

[41] A. Ginsberg and S. Ahuja. Automating envisionment of virtual meeting room histories. In *Proc. of ACM Multimedia*, pp. 65–75, November 1995.

[42] A. Girgensohn, J. Boreczky, L. Wilcox and J. Foote. Facilitating video access by visualizing automatic analysis. In *Proc. of Human-Computer Interaction INTERACT*, pp. 205–212, 1999.

[43] R. C. Gonzalez and R. E. Woods. Digital image Processing. *Prentice Hall*, 2002.

[44] V. N. Gudivada and V.V. Raghavan. Spatial similarity based retrieval in image databases. *Symposium on Document Analysis and Information Retrieval*, pp 255–270, 1993.

[45] J. Ha, R. Haralick and I. Phillips. Recursive X-Y cut using bounding boxes of connected components. In *Proc. of ICDAR*, pp. 952–955, August 1995.

[46] K. Hadjar, M. Rigamonti, D. Lalanne and R. Ingold. Xed: A new tool for eXtracting hidden structures from Electronic Documents. In *Proc. of Int'l Workshop on Document Image Analysis for Libraries (DIAL)*, pp. 212–224, 2004.

[47] A. Hampapur, E.T.Weymouth and R. Jain. Digital video segmentation. In *Proc. of ACM Multimedia*, pp. 357–364, October 1994.

[48] R. Haralick. Document image understanding: geometric and logical layout. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, Vol. 8, pp. 385–390, 1994.

[49] R. M. Haralick, K. Shanmugan and I. Dinstein. Textural features for image classification. *IEEE Trans. on Systems, Man and Cybernetics*, Vol. 3(6), pp. 610–621, 1973.

[50] H. Hase, T. Shinokawa, M. Yoneda, M. Sakai and H. Maruyama. Character string extraction from a color document. In *Proc. of ICDAR*, pp. 75–78, September 1999.

[51] L. He, Z. Liu and Z. Zhang. Why take notes? Use the whiteboard system. In *Proc. of ICASSP*, Vol. V, pp. 776–779, April 2003.

[52] D. Hindus and C. Schmandt. Ubiquitous audio: Capturing spontaneous collaboration. In *Proc. of Computer Supported Cooperative Work (CSCW)*, pp. 210–216, November 1992.

[53] J. Hull and J. Cullen. Document image similarity and equivalence detection. In *Proc. of ICDAR*, pp. 308–312, August 1997.

[54] J. J. Hull. Document image matching and retrieval with multiple distortion-invariant descriptors. In *Proc. of Document Analysis System (DAS)*, pp. 383–400, 1994.

[55] J. Hunter and S. Little. Building and indexing a distributed multimedia presentation archive using SMIL. In *Proc. of ECDL*, September 2001.

[56] A. K. Jain and B. Yu. Automatic text location in images and video frames. *Pattern Recognition*, Vol. 31(12), pp. 2055–2076, 1998.

[57] A. K. Jain and S. Bhattacharjee. Text segmentation using gabor filters for automatic document processing. *Machine Vision and Application*, Vol. 5, pp. 169–184, 1992.

[58] A. K. Jain and Y. Zhong. Page segmentation using texture analysis. *Pattern Recognition*, Vol. 29, pp. 743–770, 1996a.

[59] A. K. Jain. Fundamentals of Digital Image Processing. *Prentice-Hall*, 1997.

[60] A. K. Jain and A. Vailaya. Image retrieval using color and shape. *Pattern Recognition*, Vol. 29 (8), pp. 1233–1244, 1996b.

[61] A. Janin, J. Ang, S. Bhagat, R. Dhillon and J. Edwards, *et al.* The ICSI meeting project: Resources and research. In *Proc. NIST ICASSP 2004 Meeting Recognition Workshop*, May 2004.

[62] T. Jehan. Creating music by Listening. *PhD. Thesis*, MIT, September 2005.

[63] M. J. Jones and J. M. Rehag. Statistical color models with application to skin detection. *Intl. Journal of Computer Vision*, Vol. 46 (1), pp. 81–96, 2002.

[64] T. Kato. Database architecture for content-based image retrieval. In *Proc. of SPIE: Image Storage and Retrieval Systems*, Vol. 1662, pp. 112–123, 1992.

[65] J. Keechul, K. I. Kim and A. K. Jain. Text information extraction in images and video: a survey. *Pattern Recognition*, Vol. 37(5), pp. 977–997, 2004.

[66] M. Krishnamoorthy, G. Nagy, S. Seth and M. Viswanathan. Syntactic segmentation and labeling of digitized pages from technical journal. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 15(7), pp. 737–747, 1993.

[67] D. Lalanne, A. Lisowska, E. Bruno and M. Flynn *et al.* The IM2 multimodal meeting browser family. *Interactive Multimodal Information Management Tech. Report*, IDIAP, Margtiny, Switzerland, March 2005.

[68] D. Lalanne, R. Ingold, D. V. Rotz, A. Behera and D. Mekhaldi. Using static documents as structured and thematic interfaces to multimedia meeting archives. In *Proc. Int'l Workshop on Multimodal Interaction and Related Machine Learning Algorithms (MLMI)*, LNCS, Vol. 3361, pp. 87–100, 2004.

[69] D. Lalanne, S. Sire, R. Ingold, A. Behera, D. Mekhaldi and D. V. Rotz. A research agenda for assessing the utility of document annotations in multimedia databases of meeting recordings. In *Proc. of 3rd Int'l Workshop on MDDE, in conjunction with VLDB*, pp 47–55, 2003.

[70] D. S. Lee, B. Erol, J. Graham, J. J. Hull and N. Murata. Portable meeting recorder. In *Proc. of ACM Multimedia*, pp. 493–502, December 2002.

[71] J. S. Lim. Two-dimensional signal and image processing, *Prentice Hall*, 1990.

[72] T. D. C. Little, G. Ahanger, R. J. Folz, J. F. Gibbon and F. W. Reeve *et al.* A digital on-demand video service supporting content-based queries. In *Proc. of ACM Multimedia*, pp. 427–436, August 1993.

[73] M. Liwicki and H. Bunke. IAM-OnDB - an on-line english sentence database acquired from handwritten text on a whiteboard. In *Proc. of ICDAR*, Vol. 2, pp. 956 – 961, August-September 2005.

[74] B. S. Manjunath and W.Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18(8), pp. 837–842, 1996.

[75] B. S. Manjunath, J. R. Ohm, V. V. Vasudevan and A. Yamada. Color and texture descriptors. *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 11(6), pp. 703–715, 2001.

[76] D. Marr and E. Hildreth. Theory of edge detection. In *Proc. of the royal society of London*, B–207, pp. 186–217, 1980.

[77] S. Messelodi and C. Modena. Automatic identification and skew estimation of text lines in real scene images. *Pattern Recognition*, Vol. 32(5), pp. 791–810, 1999.

[78] S. Minneman, S. Harrison, B. Janssen, T. Moran and G. Kurtenbach *et al.* A confederation of tools for capturing and accessing collaborative activity. In *Proc. of ACM Multimedia*, pp. 523–534, November 1995.

[79] S. Mukhopadhyay and B. Smith. Passive capture and structuring of lectures. In *Proc. of ACM Multimedia*, pp. 477–487, October-November 1999.

[80] G. Nagy. Twenty years of document image analysis in PAMI. *IEEE Trans on Pattern Analysis and Machine Intelligence (PAMI)*, Vol. 22(1), pp. 38–62, 2000.

[81] L. O´Gorman and R. Kasturi. Document image analysis. *IEEE Computer Society Press*, 1995.

[82] L. O'Gorman. The document spectrum for page layout analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 15(11), pp. 1162–1173, 1993.

[83] O. Okun, D. Doermann and M. Pietikainen. Page segmentation and zone classification: The state of the art. *Technical Report LAMP-TR-036*, University of Maryland, 1999.

[84] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. on Systems, Man and Cybernetics*, Vol. 9(1), pp. 62–66, 1979.

[85] N. Ozawa, H. Takebe, Y. Katsuyama, S. Naoi and H. Yakota. Slide identification for lecture movies by matching characters and images. In *Proc. of SPIE-Document Recognition and Retrieval XI*, Vol. 5296, pp. 74–81, 2004.

[86] T. Palvidis and J. Zhou. Page segmentation and classification. *Computer Vision, Graphics, and Image Processing*, Vol. 54, pp. 484–496, 1992.

[87] E. Parzen. On estimation of a probability density function and mode. *Ann. Math. Stat.*, Vol. 33, pp. 1065–1076, 1962.

[88] E. Pedersen, K. McCall, T. P. Moran and F. Halasz. Tivoli: An electronic whiteboard for informal workgroup meetings. In *Proc. of INTERCHI*, pp. 391–389, April 1993.

[89] M. Petkovic′. Content-based video retrieval. In *Proc. of Int'l Conf. on Extending Database Technology*, pp 74–77, 2000.

[90] C. V. Remco and M. Tanase. Content-based image retrieval systems: A survey. *Technical Report UU-CS-2000-34*, Utrecht University, 2000.

[91] W. J. Rucklidge. Efficiently locating objects using the Hausdorff distance. *Int'l Journal of Computer Vision*, Vol. 24(3), pp. 251–270, 1997.

[92] Y. Rui, A. Gupta, J. Grudin and L. He. Automating lecture capture and broadcast: technology and videography. *ACM Multimedia Systems*, Vol. 10, pp. 3–15, 2004.

[93] S. Santini and R. C. Jain. The graphical specification of similarity queries. *Journal of Visual Languages and Computing*, Vol. 7, pp. 403–421, 1997.

[94] D. W. Scott. Multivariate density estimation. *New York: John Wiley*, 1992.

[95] C. Shin, D. Doermann and A. Rosenfeld. Classification of document pages using structure-based features. *Int'l Journal of Document Analysis and Recognition (IJDAR)*, Vol. 3, pp. 232–247, 2001.

[96] B. W. Silverman. Density estimation for statistic and data analysis. *New York: Chapman and Hall*, 1986.

[97] A. Simon, J. Pret and A. Johnson. A fast algorithm for bottom-up document layout analysis. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 19, pp. 273–276, 1997.

[98] K. Sobottka, H. Bunke and H. Kronenberg. Identification of text on colored book and journal covers. In *Proc. of ICDAR*, pp. 57–62, September 1999.

[99] A. Steinmetz and M. Kienzle. The e-Seminar lecture recording and distribution system. In *Proc. of SPIE MultiMedia Computing and Networking (MMCN)*, Vol. 4312, pp. 25 – 36, 2001.

[100] L. Stifelman, B. Arons and C. Schmandt. The audio notebook: Paper and pen interaction with structured speech. In *Proc. of SIGCHI*, pp. 182–189, 2001.

[101] M. Stricker and A. Dimai. Color indexing with weak spatial constraints. In *Proc. of SPIE: Storage and Retrieval for Image and Video Database IV*, Vol. 2670, pp. 29–41, 1996.

[102] P. Suda, C. Bridoux, B. Kammerer and G. Maderlechner. Logo and word matching using a general approach to signal registration. In *Proc. of ICDAR*, pp. 61–65, August 1997.

[103] H. Sundaram and S. Chang. Video analysis and summarization at structural and semantic levels. *Multimedia Information Retrieval and Management: Technological Fundamentals and Applications, book chapter in D. Feng eds.*, Springer, March 2003.

[104] S. Sural, G. Qian and S. Paramanik. Segmentation and histogram generation using the HSV color space for image retrieval. In *Proc. of IEEE ICIP*, pp. 589–592, 2002.

[105] M. Swain and D. Ballard. Color Indexing. *Int'l Journal of Computer Vision*, Vol. 7(1), pp. 11–32, 1991.

[106] T. F. Syeda-Mahmood. Indexing of technical manual document databases. In *Proc. of SPIE - Storage and Retrieval for Image and Video Databases III*, pp. 430–441, 1995.

[107] L. Todoran, M. Worring and A. W. M. Smeulders. Data groundtruth, complexity, and evaluation measures for color document analysis. In *Proc. of 5th Int'l Workshop on Document Analysis Systems (DAS)*, pp. 519–531, August 2002.

[108] Ø. D. Trier and T. Taxt. Evaluation of binarization methods for document images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 17(3), pp.312–315, 1995.

[109] H. Ueda, T. Miyatake and S.Yoshizawa. IMACT: An interactive natural-motion-picture dedicated multimedia authoring system. In *Proc. of ACM CHI*, pp. 343–350, 1991.

[110] X. Wan and C. C. J. Kuo. Color distribution analysis and quantization for image retrieval. In *Proc. of SPIE*, Vol. 2670, February 1996.

[111] B. Wang, X. –F. Li, F. Liu, and F. –Q. Hu. Color text image binarization based on binary texture analysis. In *Proc. of ICASSP*, pp. 585–588, May 2004.

[112] D. Wang and S. N. Srihari. Classification of newspaper image blocks using texture analysis. *Computer Vision, Graphics, and Image Processing*, Vol. 47, pp. 327–352, 1989.

[113] K. Weber and A. Poon. Marquee: a tool for real-time video logging. In *Proc. of CHI*, pp. 58-64, April 1994.

[114] S. Whittaker, P. Hyland and M. Wiley. Filochat: Handwritten notes provide access to recorded conversations. In *Proc. of CHI*, pp. 271–277, April 1994.

[115] L. D. Wilcox, B. N. Schilit and N. Sawhney. Dynomite: A dynamically organized ink and audio notebook. In *Proc. of CHI*, pp. 186–193, 1997.

[116] C. G. Wolf and J. R. Rhyne. Facilitating review of meeting information using temporal histories. *Technical Report RC 19811*, IBM T.J. Watson Research Center, 1992a.

[117] C. G. Wolf, J. R. Rhyne and L. Briggs. Communication and information retrieval with a pen-based meeting support tool. In *Proc. of ACM CSCW*, pp. 322-329, 1992b.

[118] P. Wolf, W. Putz, A. Stewart, A. Steinmetz and M. Hemmje *et al.* LectureLounge - experience education beyond the borders of the classroom. *Int'l Journal on Digital Libraries*, Vol. 4(1), pp. 39–41, 2004.

[119] K. Y. Wong, R. G. Casey and F. M. Wahl. Document analysis system. *IBM J. Res. Dev.*, Vol. 26, pp. 647–656, 1982.

[120] V. Wu, R. Manmatha and E. M. Riseman. Textfinder: An automatic system to detect and recognize text in images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 21(11), pp. 1224–1229, 1999.

[121] J. Y. Yang and O. K. Ersoy. Combined supervised and unsupervised learning in genomic data mining. *Technical Report TR-ECE 03-10*, Purdue University, 2003.

[122] S. D. Yanowitz and A. M. Bruckstein. A new method for image segmentation. *Computer Graphics, Vision and Image Processing*, Vol. 46(1), pp. 82–95, 1989.

[123] B. L. Yeo and B. Liu. Rapid scene analysis on compressed video. *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 5(6), pp. 533–544, 1995.

[124] R. Zabhi, J. Miller and K. Mai. A feature-based algorithm for detecting and classifying scene breaks. In *Proc. of ACM Multimedia*, pp. 189–200, November 1995.

[125] D. Zhang and G. Lu. Evaluation of similarity measurement for image retrieval. In *Proc. of IEEE Int'l Conf. on Neural Network and Signal Processing*, pp. 928–931, 2003.

[126] H. Zhang, A. Kankanhalli and S.W. Smoliar. Automatic partitioning of full-motion video. *ACM Multimedia Systems*, Vol. 1(1), pp.10–28, 1993.

Webliography

[*URL1*] http://www.eif.ch/projets/smac/, *SMAC*, *Smart Multimedia Archive for Conferences*, http://ditwww.epfl.ch/SIC/SA/SPIP/Publications/article.php3?id_article=1002, FI 1/06, http://ditwww.epfl.ch/SIC/SA/SPIP/Publications/article.php3?id_article=830, FI 2/05.

[*URL2*] http://bmrc.berkeley.edu/frame/projects/lb/index.html, LectureBrowser, University of Berkeley CA.

[*URL3*] http://lecturelounge.ipsi.fraunhofer.de, LectureLounge, Fraunhofer-IPSI, Darmstadt, Germany.

[*URL4*] http://www.cc.gatech.edu/fce/eclass/, eClass, Georgia Institute of Technology, Atlanta.

[*URL5*] http://www.informatik.uni-mannheim.de/informatik/pi4/projects/emulib/, EmuLib, Educational Multimedia Library Project, University of Mannheim, Germany.

[*URL6*] http://www1.cs.columbia.edu/CLIC/, CLIC: The Columbia Computer Science Lab and Interactive Classroom.

[*URL7*] http://www.tk.informatik.tu-darmstadt.de/Forschung/DLH, Digital Lecture Hall (DLH), Technical University of Darmstadt, Germany.

[*URL8*] http://www.cs.washington.edu/education/dl/presenter/, Class Room Presenter, University of Washington, Seattle.

[*URL9*] http://penance.is.cs.cmu.edu/meeting_room/, Meeting Room, Carnegie Mellon University.

[*URL10*] http://www.autoauditorium.com/, Foveal Systems AutoAuditorium.

[*URL11*] http://www.icsi.berkeley.edu/Speech/mr/mtgrcdr.html, The Meeting Recorder Project, International Computer Science Institute (ICSI), Berkeley.

[*URL12*] http://www.idiap.ch/~moore/meeting/, IDIAP Smart Meeting Room, IDIAP Research Institute, Martigny, Switzerland.

[*URL13*] http://www.smarttech.com/, SMART Technologies Inc.

[*URL14*] http://www.polyvision.com/, PolyVision Corporation.

[*URL15*] http://www.w3.org/TR/SMIL2/, Synchronized Multimedia Integration Language (SMIL 2.1) Specification, W3C Recommendation, February 2005.

[*URL16*] http://www.labtec.com/, Labtec Inc.

[*URL17*] http://www.qnap.com.tw/, QNAP Systems, Inc.

Curriculum Vitae

Name : **Ardhendu Behera**
Date of Birth : July 1, 1977
Permanent Address : Kukudapada, Balia, Balasore, Orissa, India-756001

EDUCATIONS AND EMPLOYMENTS

Year	University/Company	Positions/Degree
July 2002- May 2006	Department of Computer Science, University of Fribourg, Switzerland	**PhD. in Computer Science**
May 2006	**PhD in Computer Science; Thesis**: A Visual Signature-based Identification Method of Low-Resolution Document Images and its Exploitation to Automate Indexing of Multimodal Recordings	
2001	Sun Microsystems India Ltd., Divyashree Chambers, Off Langford Road, Shantinagar, Bangalore-560 027, India	**Member of Technical Staff**
1999-2001	Department of Electrical Engineering and Department of Computer Science, Indian Institute of Science, Bangalore, India.	**Master of Engineering; System Science and Automation**
2001	**Master of Engineering (M. Eng.) in System Science and Automation; Thesis**: Enhancements to MPEG-4 Advanced Audio Coding	
1995-1999	Motilal Neheru National Institute of Technology (MNNIT), formerly Motilal Neheru Regional Engineering College (MNREC), Allahabad, U.P., India.	**Bachelor of Engineering (Hons.) (Electrical Engineer)**
1999	**Bachelor in Electrical Engineering; Thesis**: Digital Techniques for Protection of Alternators	

COMPUTER EXPERIENCE

- **Operating Systems**: UNIX/Linux and Windows variants
- **Languages**: C++, VC++, C, Java, Matlab

WORSHOPS UNDERGONE

- International Workshop on Camera-based Document Analysis and Recognition (Seoul, S. Korea) : 2005
- Troisième Cycle Romand d'Informatique, “Mixed Reality and Computing in the Physical World,” Fribourg, Switzerland : 2005
- 1st International Workshop on Machine Learning for Multimodal Interaction (MLMI) 2004, Martigny, Switzerland : 2004
- Troisième Cycle Romand d'Informatique, “Multimodal and Mobile Interfaces,” Anzère, Switzerland : 2004
- Troisième Cycle Romand d'Informatique, “Question/Answering & Web Searching,” Champéry, Switzerland : 2003
- Troisième Cycle Romand d'Informatique, “ Biometrics for Secure Person Authentication,” Fribourg, Switzerland : 2003

POSTERS AND ABSTRACTS PRESENTED

1. **Ardhendu Behera**, Denis Lalanne and Rolf Ingold, “A Feature-based Method for Identification of Low-resolution Documents Captured using Handheld Devices during Meetings,” Swiss National Center of Competence in Research (NCCR) on Interactive Multimodal Information Management (IM2) Summer Institute 2005. 14-17 November, 2005. Lausanne, Switzerland.

2. **Ardhendu Behera** and Denis Lalanne, Looking at meeting documents: Events detection and documents identification, Joint AMI/PASCAL/IM2/M4 Workshop on Multimodal Interaction and Related Machine Learning Algorithms (MLMI), 21-23 June, 2004, Martigny, Switzerland.

3. **Ardhendu Behera** and Denis Lalanne, “Detection and Recognition of Visible Documents,” 2nd Joint Meeting of IM2, 12-13 February 2004, Martigny, Switzerland.

4. **Ardhendu Behera** and Denis Lalanne, Detection and Identification of Visible Documents and Related Phenomena, IM2 Summer Institute, 7-8 October 2003, Crans-Montana, Switzerland.

OTHER CONTRIBUTIONS

- Launching of Smart Multimedia Archive for Conferences, SMAC in collaboration with the European Organization for Nuclear Research, CERN and the Ecole d'Enginéur et Architec, Fribourg scheduled on January 2006.
- Member of the scientific committee of World Enformatica Congress (WEC) and additional reviewer of International Conference on Pattern Recognition 2006 (ICPR), IAPR Workshop on Document Analysis Systems (DAS) 2005 and 2006, International Conference on Document Analysis and Recognition (ICDAR) 2005, ACM Symposium on Document Engineering (DocEng) 2004 and 2005.

RESEARCH PUBLICATIONS

1. **Ardhendu Behera**, Denis Lalanne and Rolf Ingold, "DocMIR: An Automatic Document-based Indexing System for Meeting Retrieval," International Journal of Multimedia Tools and Applications, 2005 (submitted).

2. **Ardhendu Behera**, Denis Lalanne and Rolf Ingold, "Influence of Fusion Strategies on Feature-based Identification of Low-resolution Documents," The ACM Symposium on Document Engineering (DocEng) 2005, 2 - 4 November, Bristol, UK, pp. 20-22.

3. **Ardhendu Behera**, Denis Lalanne and Rolf Ingold, "Enhancement of Layout-based Identification of Low-resolution Documents using Geometrical Color Distribution," International Conference on Document Analysis and Recognition (ICDAR) 2005, IEEE Computer Society, August 29-1st September, Seoul, Korea, pp. 468-472.

4. **Ardhendu Behera**, Denis Lalanne and Rolf Ingold, "Combining Color and Layout Features for the Identification of Low-resolution Documents," International Journal of Signal Processing (IJSP), ISSN:1304-4478, Vol 2, No 1, pp. 7-14, 2005.

5. **Ardhendu Behera**, Denis Lalanne and Rolf Ingold. "Color and Layout-based Identification of Documents Captured from Low-resolution Handheld Devices," International Conference on Pattern Recognition and Computer Vision 2005, 25-27 February, Istanbul, Turkey, 2nd World Enformatika Congress, WEC'05, ISBN 975-98458-3-0, Vol 1, pp. 51-54.

6. **Ardhendu Behera**, Denis Lalanne and Rolf Ingold. "Visual Signature based Identification of Low-resolution Document Images," The ACM Symposium on Document Engineering (DocEng) 2004, Milwaukee, Wisconsin, USA, October 28-30, 2004, pp. 178-187.

7. **Ardhendu Behera**, Denis Lalanne and Rolf Ingold. “Looking at projected documents: Event detection & Document Identification,” IEEE International Conference on Multimedia and Expo (ICME) 2004, Taipei, Taiwan, June 2004, pp. 2127-2130.

8. Denis Lalanne, Rolf Ingold, Didier von Rotz, **Ardhendu Behera**, Dalila Mekhaldi and Andrei Popescu-Belis - “Using static documents as structured and thematic interfaces to multimedia meeting archives,” 1st International Workshop on Machine Learning for Multimodal Interaction (MLMI) 2004, Martigny, Switzerland, LNCS, Vol. 3361, pp. 87-100.

9. Denis Lalanne, Stephane Sire, Rolf Ingold, **Ardhendu Behera**, Dalila Mekhaldi and Didier von Rotz, “A research agenda for assessing the utility of document annotations in multimedia databases of meeting recordings,” 3rd International Workshop on Multimedia Data and Document Engineering (MDDE), Berlin, Germany, pp. 47-55.