Exhibit B

Shi-Kuo Chang
Zen Chen
Suh-Yin Lee (Eds.)

LNCS 2314

Recent Advances in Visual Information Systems

5th International Conference, VISUAL 2002
Hsin Chu, Taiwan, March 2002
Proceedings

Springer

Lecture Notes in Computer Science 2314

Edited by G. Goos, J. Hartmanis, and J. van Leeuwen

Springer
Berlin
Heidelberg
New York
Barcelona
Hong Kong
London
Milan
Paris
Tokyo

Shi-Kuo Chang Zen Chen Suh-Yin Lee (Eds.)

Recent Advances in Visual Information Systems

5th International Conference, VISUAL 2002
Hsin Chu, Taiwan, March 11-13, 2002
Proceedings

Springer

Volume Editors

Shi-Kuo Chang
Knowledge Systems Institute
3420 Main Street, Skokie, IL 60076, USA
E-mail: chang@cs.pitt.edu

Zen Chen
Suh-Yin Lee
National Chiao Tung University, Dept. of Comp. Science & Information Engineering
1001 Ta Hsueh Road, Hsin Chu, Taiwan
E-mail: {zchen/sylee}@csie.nctu.edu.tw

Cataloging-in-Publication Data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Recent advances in visual information systems : 5th international
conference, VISUAL 2002, Hsin Chu, Taiwan, March 11 - 13, 2002 ; proceedings
/ Shi-Kuo Chang ... (ed.). - Berlin ; Heidelberg ; New York ; Barcelona ;
Hong Kong ; London ; Milan ; Paris ; Tokyo : Springer, 2002
(Lecture notes in computer science ; Vol. 2314)
ISBN 3-540-43358-9

CR Subject Classification (1998): H.3, H.5, H.2, I.4, I.5, I.3, I.7

ISSN 0302-9743
ISBN 3-540-43358-9 Springer-Verlag Berlin Heidelberg New York

Springer-Verlag Berlin Heidelberg New York
a member of BertelsmannSpringer Science+Business Media GmbH

http://www.springer.de

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Olgun Computergrafik
Printed on acid-free paper SPIN: 10846602 06/3142 5 4 3 2 1 0

Preface

Visual information systems are information systems for visual computing. Visual computing is computing on visual objects. Some visual objects such as images are inherently visual in the sense that their primary representation is the visual representation. Some visual objects such as data structures are derivatively visual in the sense that their primary representation is not the visual representation, but can be transformed into a visual representation. Images and data structures are the two extremes. Other visual objects such as maps may fall somewhere in between the two. Visual computing often involves the transformation from one type of visual objects into another type of visual objects, or into the same type of visual objects, to accomplish certain objectives such as information reduction, object recognition, and so on.

In visual information systems design it is also important to ask the following question: who performs the visual computing? The answer to this question determines the approach to visual computing. For instance it is possible that primarily the computer performs the visual computing and the human merely observes the results. It is also possible that primarily the human performs the visual computing and the computer plays a supporting role. Often the human and the computer are both involved as equal partners in visual computing and there are visual interactions. Formal or informal visual languages are usually needed to facilitate such visual interactions.

In this conference various research issues in visual information systems design and visual computing are explored. The papers are collectively published in this volume. We would like to express our special thanks to the sponsorship of the National Science Council, ROC, the Lee and MTI Center of National Chiao Tung University, ROC, and Knowledge Systems Institute, USA.

January 2002 Shi-Kuo Chang, Zen Chen, and Suh-Yin Lee

VISUAL 2002 Conference Organization

General Chair	Shi-Kuo Chang, USA
American General Co-chair	Ramesh Jain, USA
European General Co-chair	Arnold Smeulders, The Netherlands
Asian General Co-chair	Horace Ip, ROC
Program Co-chairs	Zen Chen, ROC Suh-Yin Lee, ROC
Steering Committee	Shi-Kuo Chang, USA Horace Ip, Hong Kong Ramesh Jain, USA Tosiyasu Kunii, Japan Robert Laurini, France Clement Leung, Australia Arnold Smeulders, The Netherlands

Program Committee

Jan Biemond, The Netherlands	Carlo Meghini, Italy
Josef Bigun, Switzerland	Erich Neuhold, Germany
Shih Fu Chang, USA	Eric Pauwels, Belgium
David Forsyth, USA	Fernando Pereira, Portugal
Theo Gevers, The Netherlands	Dragutin Petkovic, USA
Luc van Gool, Belgium	Hanan Samet, USA
William Grosky, USA	Simone Santini, USA
Glenn Healey, USA	Stan Sclaroff, USA
Nies Huijsmans, The Netherlands	Raimondo Schettini, Italy
Yannis Ioanidis, Greece	Stephen Smoliar, USA
Erland Jungert, Sweden	Aya Soffer, USA
Rangachar Kasturi, USA	Michael Swain, USA
Toshi Kato, Japan	Hemant Tagare, USA
Martin Kersten, The Netherlands	George Thoma, USA
Inald Lagendijk, The Netherlands	Marcel Worring, The Netherlands
Robert Laurini, France	Jian Kang Wu, Singapore
Yi-Bin Lin, ROC	Wei-Pan Yang, ROC

Sponsors	National Science Council, ROC National Chiao Tung University, ROC Knowledge Systems Institute, USA

Table of Contents

I Invited Talk

Multi-sensor Information Fusion by Query Refinement 1
Shi-Kuo Chang, Gennaro Costagliola, and Erland Jungert

II Content-Based Indexing, Search and Retrieval

MiCRoM: A Metric Distance to Compare Segmented Images 12
Renato O. Stehling, Mario A. Nascimento, and Alexandre X. Falcão

Image Retrieval by Regions: Coarse Segmentation and Fine Color Description 24
Julien Fauqueur and Nozha Boujemaa

Fast Approximate Nearest-Neighbor Queries in Metric Feature Spaces by Buoy Indexing 36
Stephan Volmer

A Binary Color Vision Framework for Content-Based Image Indexing 50
Guoping Qiu and S. Sudirman

Region-Based Image Retrieval Using Multiple-Features 61
Veena Sridhar, Mario A. Nascimento, and Xiaobo Li

A Bayesian Method for Content-Based Image Retrieval by Use of Relevance Feedback 76
Ju-Lan Tao and Yi-Ping Hung

Color Image Retrieval Based on Primitives of Color Moments 88
Jau-Ling Shih and Ling-Hwei Chen

Invariant Feature Extraction and Object Shape Matching Using Gabor Filtering 95
Shu-Kuo Sun, Zen Chen, and Tsorng-Lin Chia

III Visual Information System Architectures

A Framework for Visual Information Retrieval 105
Horst Eidenberger, Christian Breiteneder, and Martin Hitz

Feature Extraction and a Database Strategy for Video Fingerprinting 117
Job Oostveen, Ton Kalker, and Jaap Haitsma

ImageGrouper: Search, Annotate and Organize Images by Groups 129
Munehiro Nakazato, Lubomir Manola, and Thomas S. Huang

Toward a Personalized CBIR System 143
Chih-Yi Chiu, Hsin-Chih Lin, and Shi-Nine Yang

IV Image/Video Databases

An Efficient Storage Organization for Multimedia Databases 152
Philip K.C. Tse and Clement H.C. Leung

Unsupervised Categorization for Image Database Overview 163
Bertrand Le Saux and Nozha Boujemaa

A Data-Flow Approach to Visual Querying in Large Spatial Databases . . . 175
Andrew J. Morris, Alia I. Abdelmoty, Baher A. El-Geresy, and Christopher B. Jones

MEDIMAGE – A Multimedia Database Management System for Alzheimer's Disease Patients 187
Peter L. Stanchev and Farshad Fotouhi

V Networked Video

Life after Video Coding Standards: Rate Shaping and Error Concealment 194
Trista Pei-chun Chen, Tsuhan Chen, and Yuh-Feng Hsu

A DCT-Domain Video Transcoder for Spatial Resolution Downconversion 207
Yuh-Reuy Lee, Chia-Wen Lin, and Cheng-Chien Kao

A Receiver-Driven Channel Adjustment Scheme for Periodic Broadcast of Streaming Video 219
Chin-Ying Kuo, Chen-Lung Chan, Vincent Hsu, and Jia-Shung Wang

Video Object Hyper-Links for Streaming Applications 229
Daniel Gatica-Perez, Zhi Zhou, Ming-Ting Sun, and Vincent Hsu

VI Application Areas of Visual Information Systems

Scalable Hierarchical Summarization of News Using Fidelity in MPEG-7 Description Scheme 239
Jung-Rim Kim, Seong Soo Chun, Seok-jin Oh, and Sanghoon Sull

MPEG-7 Descriptors in Content-Based Image Retrieval with PicSOM System 247
Markus Koskela, Jorma Laaksonen, and Erkki Oja

Fast Text Caption Localization on Video Using Visual Rhythm 259
Seong Soo Chun, Hyeokman Kim, Jung-Rim Kim, Sangwook Oh, and Sanghoon Sull

A New Digital Watermarking Technique for Video 269
Kuan-Ting Shen and Ling-Hwei Chen

Automatic Closed Caption Detection and Font Size Differentiation in MPEG Video 276
Duan-Yu Chen, Ming-Ho Hsiao, and Suh-Yin Lee

Motion Activity Based Shot Identification and Closed Caption Detection for Video Structuring 288
Duan-Yu Chen, Shu-Jiuan Lin, and Suh-Yin Lee

Visualizing the Construction of Generic Bills of Material 302
Peter Y. Wu, Kai A. Olsen, and Per Saetre

Data and Knowledge Visualization in Knowledge Discovery Process 311
TrongDung Nguyen, TuBao Ho, and DucDung Nguyen

Author Index 323

Feature Extraction and a Database Strategy for Video Fingerprinting

Job Oostveen, Ton Kalker, and Jaap Haitsma

Philips Research, Prof. Holstlaan 4, 5656 AA Eindhoven, The Netherlands
Job.Oostveen@philips.com, Ton.Kalker@ieee.org, Jaap.Haitsma@philips.com

Abstract. This paper presents the concept of *video fingerprinting* as a tool for video identification. As such, video fingerprinting is an important tool for persistent identification as proposed in MPEG-21. Applications range from video monitoring on broadcast channels to filtering on peer-to-peer networks to meta-data restoration in large digital libraries. We present considerations and a technique for (i) extracting essential perceptual features from moving image sequences and (ii) for identifying any sufficiently long unknown video segment by efficiently matching the fingerprint of the short segment with a large database of pre-computed fingerprints.

1 Introduction

This paper presents a method for the identification of video. The objective is to identify video objects not by comparing perceptual similarity of the video objects themselves (which might be computationally expensive), but by comparing short digests, also called *fingerprints*, of the video content. These digests mimic the characteristics of regular human fingerprints. Firstly, it is (in general) impossible to derive from the fingerprint other relevant personal characteristics. Secondly, comparing fingerprints is sufficient to decide whether two persons are the same or not. Thirdly, fingerprint comparison is a statistical process, not a test for mathematical equality: it is only required that fingerprints are *sufficiently* similar to decide whether or not they belong to the same person (proximity matching).

1.1 Classification

Fingerprint methods can be categorized in two main classes, viz. the class of method based on *semantical* features and the class of methods based on *non-semantical* features. The former class builds fingerprints from high-level features, such as commonly used for retrieval. Typical examples include scene boundaries and color-histograms. The latter class builds fingerprints from more general perceptual invariants, that do not necessarily have a semantical interpretation. A typical example in this class is differential block luminance (see also Section 2). For both classes holds that (small) fingerprints can be used to establish perceptual equality of (large) video objects. It should be noted that a feature extraction method for fingerprinting must be quite different from the methods used for

S.-K. Chang, Z. Chen, and S.-Y. Lee (Eds.): VISUAL 2002, LNCS 2314, pp. 117–128, 2002.

video retrieval. In retrieval, the features must facilitate searching of video clips that somehow look similar to the query, of that contain similar objects as the query. In fingerprinting the requirement is to identify clips that are perceptually the same, except for quality differences or the effects of other video processing. Therefore, the features for fingerprinting need to be far more discriminatory, but they do not necessarily need to be semantical.

Consider the example of identification of content in a multimedia database. Suppose one is viewing a scene from a movie and would like to know from which movie the clip originates. One way of finding out is by comparing the scene to all fragments of the same size of all movies in the database. Obviously, this is totally infeasible in case of a large database: even a short video scene is represented by a large amount of bytes and potentially these have to be compared to the whole database. Thus, for this to work, one needs to store a large amount of easily accessible data and all these data have to be compared with the video scene to be identified. Therefore, there is both a storage problem (the database) as well as a computational problem (matching large amounts of data).

Both problems can be alleviated by reducing the number of bits needed to represent the video scenes: fewer bits need to be stored and fewer bits need to be used in the comparison. One possible way to achieve this is by using video compression.

However, because it is not needed to reconstruct the video from the representation, at least theoretically it is possible to use less bits for identification than for encoding. Moreover, *perceptually* comparing compressed video streams is a computationally expensive operation.

A more practical option is to use a video compression scheme that is geared towards identification, more specifically to use a fingerprinting scheme. Video identification can then be achieved by storing the fingerprints of all relevant fragments in a database. Upon reception of a unknown fragment, its fingerprint is computed and compared to those in the database. This search (based on inexact pattern matching) is still a burdensome task, but it is feasible on current-day PCs.

1.2 Relation to Cryptography

We will now first discuss the concept of cryptographic hash functions and show how we approach the concept of fingerprints as an adaptation of cryptographic hash functions. Hash functions are a well-known concept in cryptography [8]. A *cryptographic hash*, also called message digest or digital signature, is in essence a short summary of a long message. Hash functions take a message of arbitrary size as input and produce a small bit string, usually of fixed size: the *hash* or *hash value*. Hash functions are widely used as a practical means to verify, with high probability, the integrity of (bitwise) large objects. The typical requirements for a hash function are twofold:

1. For each message M, the hash value $H = h(M)$ is easily computable;
2. The probability that two messages lead to the same hash is small.

As a meaningful hash function maps large messages to small hash values, such a function is necessarily many-to-one. Therefore, collisions do occur. However, the probability of hitting upon two messages with the same hash value should be minimal. This usually means that the hash values for all allowed messages have a uniform distribution. For an n-bit value h the probability of a collision is then equal to 2^{-n}.

Cryptographic hash functions are usually required to be one-way, i.e., it should be difficult for a given hash value H to find a message which has H as its hash value. As a result such functions are bit-sensitive: flipping a single bit in the message changes the hash completely. The topic of this paper, fingerprinting for video identification, is about functions which show a strong analogy to cryptographic hash functions, but that are explicitly not bit sensitive and are applicable to audio-visual data. Whereas cryptographic hashes are an efficient tool to establish mathematical equality of large objects, audio-visual fingerprint functions serve as a tool to establish *perceptual similarity* of (usually large) audio-visual objects. In other words, fingerprints should capture the *perceptually essential parts* of audio-visual content. In direct analogy with cryptographic hash functions, one would expect a fingerprint function to be defined as a function that maps perceptually similar objects to the same bit string value. However, it is well-known that perceptual similarity is not a transitive relationship. Therefore, a more convenient and practical definition reads as follows: a fingerprint function is function that (i) maps (usually bitwise large) audiovisual object to (usually bitwise small) bit strings (fingerprints) such that perceptual small changes lead to small differences in the fingerprint and (ii) such that perceptually very different objects lead with very high probability to very different fingerprints.

1.3 Fingerprinting Approaches

The scientific community seems to be favouring the terminology 'fingerprint', and for that reason this is the terminology that will be used in this paper. However, it is doubtful whether or not this is the best choice. For instance, the term fingerprinting is also used in the watermarking community, where it denotes the active embedding of tracing information.

Although the literature on fingerprinting is still limited, in particular for video, some progress is already reported. Among others algorithms for still image fingerprinting are published by Abdel-Mottaleb et.al. [1], by J. Fridrich [5], by R. Venkatesan et al. [12,11] and by Schneider and Chang [10]. A number of algorithms for audio fingerprinting have been published. See [6] and the references therein. A number of papers present algorithms for video fingerprinting. Cheung and Zakhor ([4]) are concerned with estimating the number of copies (possibly at different quality levels) of video clips on the web. Hampapur and Bolle [7] present a indexing system based on feature extraction from key frames.

Cryptographic hashes operate on the basis of a complete message. As such, it is impossible to check the integrity or obtain the identity of a part of the message. For video fingerprint this is an undesirable property, as it means that it is impossible to identify short clips out of a longer clips. Also, for integrity

checking, one would like to be able to *localize* distortions. For this reason, it is not always appropriate to create a global fingerprint for the whole of an audio-visual object. Instead, we propose to use a *fingerprint-stream* of locally computed fingerprint bits (also referred to as *sub-fingerprints*): per time unit, a number of bits are extracted from the content. In this way, it is possible to identify also smaller sections of the original. In a typical identification scenario, the full fingerprint stream is stored in the database. Upon reception of a video, the fingerprint values are extracted from a short section, say with a duration of 1 second. The result, which we call a *fingerprint block*, is then matched to all blocks of the same size in the database. If the fingerprint block matches to a part of the fingerprint stream of some material, it is identified as that specific part of the corresponding video. If there is no sufficiently close match, the process repeats by extracting a next fingerprint block and attempting to match it.

The description above reveals two important complexity aspects of a full fledged fingerprinting system. The first complexity aspect concerns fingerprint extraction, the second concerns the matching process. In a typical application the fingerprint extraction client has only limited resources. Moreover, the bandwidth to the fingerprint matching engine is severely restricted. It follows that in many applications it is required that fingerprint extraction is low complexity and that the size of the fingerprint is either small or at least sufficiently compressible. This observation already in many cases rules out the use of semantic fingerprints, as these tend to be computationally intensive. The fingerprint matching server is in its most basic form a gigantic sliding correlator: for an optimal decision a target fingerprint block needs to be matched against *all* fingerprinting blocks of similar length in the database. Even for simple matching functions (such as bit error rate), this sliding correlation becomes infeasible if the fingerprint database is sufficiently large. For a practical fingerprint matching engine it is essential that the *proximity matching* problem is dealt with in an appropriate manner, either by including ingredients that allow hierarchical searching [6], by careful preparation of the fingerprint database [3] or both. Both types of complexities are already well recognized in the field of audio fingerprint, see for example the recent RIAA/IFPI call [9].

1.4 Overview

In this paper we introduce a algorithm for robust video fingerprinting that has very modest feature extraction complexity, a well-designed matching engine and a good performance with respect to robustness. We will present some general considerations in the design of such a video fingerprinting algorithm with a focus on building a video identification tool. In Section 2 we introduce the algorithm and discuss a number of the issues in designing such an algorithm. Section 3 contains the design of a suitable database structure. In Section 4 we will summarize our results and indicate directions for future research.

2 Feature Extraction

In this section, we present a feature extraction algorithm for robust video fingerprinting and we discuss some of the choices and considerations in the design of

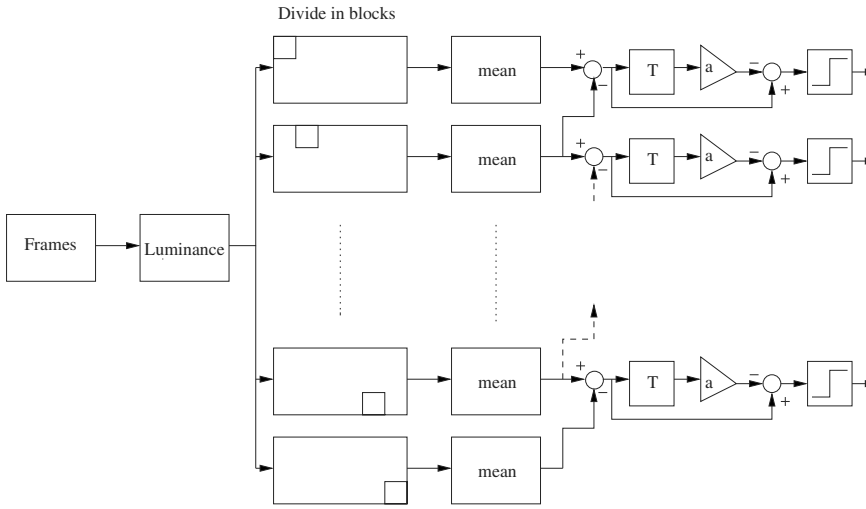

Fig. 1. block diagram of the differential block luminance algorithm

such an algorithm. The first question to be asked is in which domain to extract the features. In audio, very clearly, the frequency domain optimally represents the perceptual characteristics. In video, however, it is less clear which domain to use. For complexity reasons it is preferable to avoid complex operations, like DCT or DFT transformations. Therefore, we choose to compute features in the spatio-temporal domain. Moreover, to allow easy feature extraction from most compressed video streams as well, we choose features which can be easily computed from block-based DCT coefficients. Based on these considerations, the proposed algorithm is based on a simple statistic, the mean luminance, computed over relatively large regions. This is also approach taken by Abdel-Mottaleb [1]. We will choose our regions in a fairly simple way: the example algorithm in this paper uses a fixed number of blocks per frame. In this way, the algorithm is automatically resistant to changes in resolution.

To ease the discussion, we introduce some terminology. The bits extracted from a frame will be refereed to as *sub-fingerprints*. A *fingerprint block* then denotes a fixed number of sub-fingerprints from consecutive frames.

Our goal is to be able to identify short video clips and moreover to localize the clip inside the movie where it originates from. In order to do this, we need to extract features which contain sufficient high frequency content in the temporal direction. If the features are more or less constant over a relatively large number of frames, then it is impossible to localize exactly the clip inside the movie. For this reason, we take differences of corresponding features extracted from subsequent frames. Automatically, this makes the system robust to (slow) global changes in luminance. To arrive at our desired simple binary features, we only retain the sign of the computed differences. This immediately implies robustness to luminance offsets and to contrast modifications. To decrease the complexity

of measuring the distance between two fingerprints (the matching process), a binary fingerprint also offers considerable advantages. That is, we can compare fingerprints on a bit-by-bit basis, using the Hamming distance as a distance measure. Summarizing, we discard all magnitude information from the extracted filter output values, and only retain the sign.

The introduction of differentiation in the temporal direction leads to a problem in case of still scenes. If a video scene is effectively a prolonged still image, the temporal differentiation is completely determined by noise, and therefore the extracted bits are very unreliable. Conceptually, what one would like is that fingerprints do not change while the video is unchanged. One way to achieve this is by using a conditional fingerprint extraction procedure. This means that a frame is only considered for fingerprint computation if it differs sufficiently from the last frame from which a fingerprint was extracted [2]. This approach leads, however, to a far more difficult matching procedure: the matching needs to be resistant to the fact that the fingerprint extracted from a processed version of a clip may have a different number of *sub-fingerprints* than the original. Another possibility is to use a different temporal filter which does not completely suppress mean luminance (DC). This can be achieved in a very simple manner by replacing the earlier proposed FIR filter kernel $[\ -1\ 1\ ]$ by $[\ -\alpha\ 1\ ]$, where α is a value slightly smaller than 1. Using this filter the extracted fingerprint will be constant in still scenes (and even still regions of a scene), whereas in regions with motion the fingerprint is determined by the difference between luminance values in consecutive frames.

In addition to the differentiation in the time domain, we can also do a spatial differentiation (or more generally, a high-pass filter) on the features extracted from one frame. In this way, also the correlation between bits extracted from the same frame is decreased significantly. Secondly, application of the spatial filter avoids a bias in the overall extracted bits, which would occur if the new temporal filter were applied directly to the extracted mean luminance values[1]. For our experiments, the results of which will be presented below, we have used the following algorithm.

1. Each frame is divided in a grid of R rows and C columns, resulting in $R \times C$ blocks. For each of these blocks, the mean of the luminance values of the pixels is computed. The mean luminance of block (r, c) in frame p is denoted $F(r, c, p)$ for $r = 1, \ldots, R$ and $c = 1, \ldots, C$.
2. We visualise the computed mean luminance values from the previous step as frames, consisting of $R \times C$ "pixels". On this sequence of low resolution gray-scale images, we apply a spatial filter with kernel $[\ -1\ 1\ ]$ (i.e. taking differences between neighbouring blocks in the same row), and a temporal filter with kernel $[\ -\alpha\ 1\ ]$, as explained, above.
3. The sign of the resulting value constitutes the fingerprint bit $B(r, c, p)$ for block (r, c) in frame p. Note that due to the spatial filtering operation in the previous step, the value of c ranges from 1 to $c - 1$ (but still, $r = 1, \ldots, R$). Thus, per frame we derive $C \times (R - 1)$ fingerprint bits.

[1] Without spatial differentiation the fingerprint values before quantization would have a larger probability of being positive than negative

Summarizing, and more precisely, we have for $r = 1, \ldots, R$, $c = 1, \ldots, C-1$:

$$B(r,c,p) = \begin{cases} 1 \text{ if } Q(r,c,p) \geq 0, \\ 0 \text{ if } Q(r,c,p) < 0, \end{cases}$$

where

$$Q(r,c,p) = (F(r,c+1,p) - F(r,c,p)) - \alpha\,(F(r,c+,p-1) - F(r,c,p-1))\,.$$

We call this algorithm "differential block luminance". A block diagram, describing this is depicted in Figure 1. These features have a number of important advantages:

- Only a limited number of bits is needed to uniquely identify short video clips with a low false positive probability
- the feature extraction algorithm has a very low complexity and it may be adapted to operate directly on the compressed domain, without a need for complete decoding
- The robustness of these features with respect to geometry-preserving operations is very good

A disadvantage may be that for certain applications the robustness with respect to geometric operations (like zoom & crop) may not be sufficient. Experimental robustness results are presented in section 2.1, below.

For our experiments we used $\alpha = 0.95$ and $R = 4$, $C = 9$. This leads to a fingerprint size of 32 bits per frame, and a block size 120×80 pixels for NTSC video material. Matching is done on the basis of fingerprint bits extracted from 30 consecutive frames, i.e., $30 \times 32 = 960$ bits.

2.1 Experimental Results

Extensive experiments with the algorithm described above are planned for the near future. In this article we report on the results of some initial tests. We have used six 10-second clips, taken from a number of movies and television broadcasts (with a resolution of 480 lines and 720 pixels per line). From these clips, we extracted the fingerprints. These are used as "the database". Subsequently, we processed the clips, and investigated how this influences the extracted fingerprint. The test included the following processing:

1. MPEG-2 encoding at 4 Mbit/second;
2. median filtering using 3×3 neighbourhoods;
3. luminance-histogram equalisation;
4. shifting the images vertically over k lines ($k = 1, 2, 3, 4, 8, 12, 16, 20, 24, 32$);
5. scaling the images horizontally, with a scaling factor between 80% and 120%, with steps of 2%.

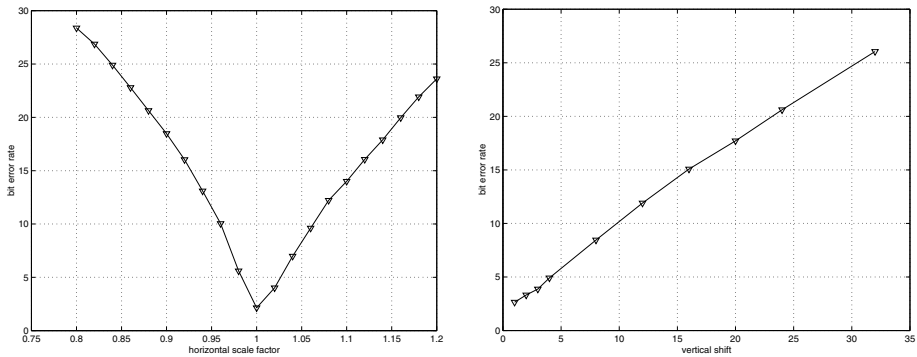

Fig. 2. Robustness w.r.t. horizontal scaling (left graph) and vertical shifts (right graph)

The results for scaling and shifting are in Figure 2. The other results are reported below:

MPEG-2 encoding:	11.8%
median filtering:	2.7%
histogram equalisation:	2.9%

The results indicate that the method is very robust against all processing which is done on a local basis, like for instance MPEG compression or median filtering. In general the alterations created by these processes average out within the blocks. Processing which changes the video more in a global fashion is more difficult to withstand. For instance, global geometric operations like scaling and shifting lead to far higher bit error rates. This behaviour stems from the resulting misalignment of the blocks. A higher robustness could be obtained by using larger blocks, but this would reduce the discriminative power of the fingerprint.

3 Database Strategy

Matching the extracted fingerprints to the fingerprints in a large database is a non-trivial task since it is well known that proximity matching does not scale nicely to very large databases (recall that the extracted fingerprint values may have many bit errors). We will illustrate this with some numbers, based on using the proposed fingerprinting scheme (as described in Section 2), in a broadcast monitoring scenario. Consider a database containing news clips with a total duration of 4 weeks (i.e., $4 \times 7 \times 24 = 672$ hours of video material). This corresponds to almost 300 megabytes of fingerprints. If we now extract a fingerprint block (e.g. corresponding to 1 second of video, which results in 30 sub-fingerprints) from an unknown news broadcast, we would like to determine which position in the 672 hours of stored news clips it matches best. In other words we want to find the position in these 672 hours where the bit error rate is minimal. This

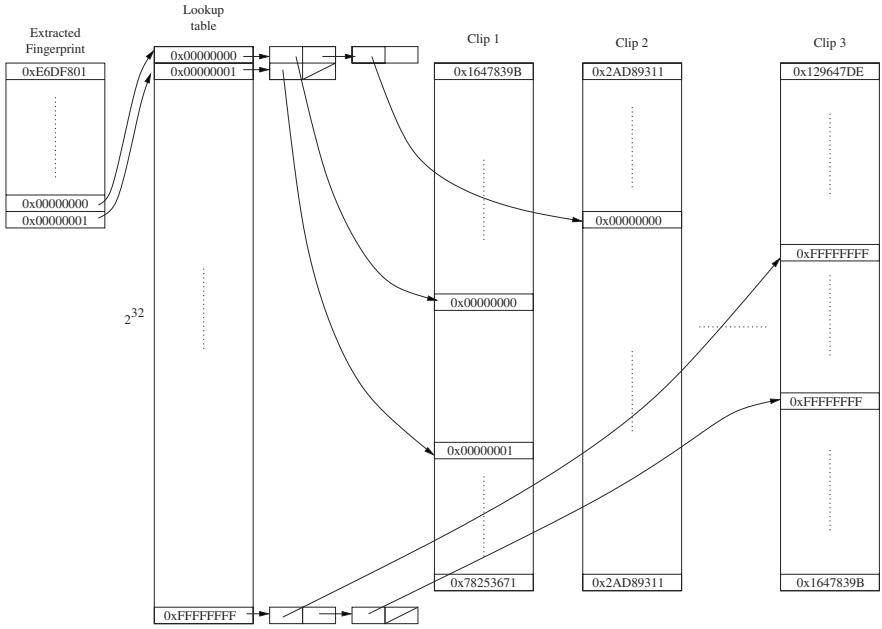

Fig. 3. database layout

can be done by brute force matching, but this will take around 72 million comparisons. Moreover the number of comparisons increases linearly with the size of the database.

We propose to use a more efficient strategy, which is depicted in Figure 3. Instead of matching the complete fingerprint block, we first look at only a single sub-fingerprint at a time and assume that occasionally this 32-bit bit-string contains no errors. We start by creating a lookup table (LUT) for all possible 32-bit words, and we let the entries in the table point to the video clip and the position(s) within that clip where this 32-bit word occurs as sub-fingerprint. Since this word can occur at multiple positions in multiple clips the pointers are stored in a linked list. In this way one 32-bit word is associated with multiple pointers to clips and positions. The approach that we take bears a lot of similarity to inverted file techniques, as used commonly in text retrieval applications. Our lookup table is basically an index describing for each sub-fingerprint (word) at which location in which clip it occurs. The main difference with text retrieval is that due to processing of the video we need to adapt our search strategy to the fact that sub-fingerprints will frequently contain (possibly many) erroneous bits.

By inspecting the lookup table for each of the 30 extracted sub-fingerprints a list of candidate clips and positions is generated. With the assumption that occasionally a single sub-fingerprint is free of bit errors, it is easy to determine whether or not all the 30 sub-fingerprints in the fingerprint block match one of

the candidate clips and positions. This is done by calculating the bit error rate of the extracted fingerprint block with the corresponding fingerprint blocks of the candidate clips and positions. The candidate clip and position with the lowest error rate is selected as the best match, provided that this error rate is below an appropriate threshold. Otherwise the database reports that the search could not find a valid best match. Note that in practice, once a clip is identified, it is only necessary to check whether or not the fingerprints of the remainder of the clip belong to the best match already found. As soon as the fingerprints no longer match, a full structured search is again initiated.

Let us give an example of the described search method by taking a look at Figure 3. The last extracted fingerprint value is 0x00000001. The LUT in the database points only to a certain position in clip 1. Let's say that this position is position p. We now calculate the bit error rate between the extracted fingerprint block and the block of song 1 from position p-29 until position p. If the two blocks match sufficiently closely, then it is very likely that the extracted fingerprint originates from clip 1. However if the two blocks are very different, then either the clip is not in the database or the extracted sub-fingerprint contains an error. Let's assume that the latter occurred. We then try the one but last extracted sub-fingerprint (0x00000000). This one has two possible candidate positions, one in clip 2 and one in clip 1. Assuming that the bit error rate between the extracted fingerprint block and the corresponding database fingerprint block of clip 2 yields a bit error rate below the threshold, we identify the video clip as originating from clip 2. If not, we repeat the same procedure for the remaining 28 sub-fingerprints.

We need to verify that our assumption that every fingerprint block contains an error-free sub-fingerprint is actually a reasonable assumption. Experiments indicate that this is actually the case for all reasonable types of processing. By the above method, we only compare the fingerprint blocks to those blocks in the database which correspond exactly in at least one of their sub-fingerprints. This makes the search much faster compared to exhaustive search or any pivot-based strategy [3], and this makes it possible to efficiently search in very large databases. This increased search speed comes at the cost of possibly not finding a match, even if there is a matching fingerprint block in the database. More precisely, this is the case if all of the sub-fingerprints have at least one erroneous bit, but at the same time the overall bit error rate is below the threshold. We can decrease the probability of missed identifications by using *bit reliability* information. The fingerprint bits are computed by taking the sign of a real-valued number. The absolute value of this number can be taken as a reliability measure of the correctness of the bit: the sign of a value close to zero is assumed to be less robust than the sign of a very large number. In this way, we can declare q of the bits in the fingerprint unreliable. To decrease the probability of a missed recognition, we toggle those q bits, thus creating 2^q candidate sub-fingerprints. We then do an efficient matching, as described above, with all of these sub-fingerprints. If one of these leads to a match, then the database fingerprint block is compared with the originally extracted fingerprint. If the resulting bit error rate of this final comparison is again below the threshold then we have a successful identifi-

cation. Note that in this way the reliability information is used to generate more candidates in the comparison procedure, but that it has no influence on the final bit error rate.

In [6] we have described a method for audio fingerprinting. The database strategy described there is the same as the one in this paper, except for some of the parameter values (in case of audio, matching is done based on fingerprint blocks which consist of 256 sub-fingerprints, corresponding to 3 seconds of audio). With this audio database we have carried out extensive experiments, that show the technical and economical feasibility to scale this approach to very large databases, containing for instance a few million songs.

An important figure of merit for a fingerprinting method is the false positive probability: The probability that two randomly selected video clips are declared similar by the method. Under the assumption that the extracted fingerprint bits are independent random variables, with equal probability of being 0 or 1, it is possible to compute a general formula for the false positive probability: Let a fingerprint consist of R sub-fingerprints and let each sub-fingerprint consist of C bits. Then for two randomly selected fingerprint blocks, the number of bits in which the two blocks correspond is binomially (n, p) distributed with parameters $n = RC$ and $p = \frac{1}{2}$. As RC is large, we can approximate this distribution by a normal distribution with mean $\mu = np = RC/2$ and variance $\sigma^2 = np(1-p) = RC/4$. Given a fingerprint block B_1 the probability that less than a fraction α of the bits of a randomly selected second fingerprint block B_2 is different from the corresponding bits of B_1 equals

$$P_f(\alpha) = \frac{1}{\sqrt{2\pi}} \int_{(1-2\alpha)\sqrt{n}}^{\infty} e^{-\frac{x^2}{2}} dx \;=\; \frac{1}{2} \operatorname{erfc}\left(\frac{1-2\alpha}{\sqrt{2}} \sqrt{n} \right).$$

Based on this formula, we can set our threshold for detection. In our experiments we used $n = 960$. Setting the threshold $\alpha = 0.3$ (i.e., declaring two clips similar if their fingerprint blocks are different in at most 30% of the bit positions), the false positive probability is computed to be in the order of 10^{-35}. In practice the actual false positive probability will be significantly higher due to correlation between the bits in a fingerprint block. Currently, we are in the process of studying experimentally the correlation structure, and adapting our theoretical false positive analysis accordingly.

4 Conclusions

In this paper we have presented fingerprinting technology for video identification. The methodology is based on the functional similarity between fingerprints and cryptographic hashes. We have introduced a feature extraction algorithm, the design of which was driven by minimal extraction complexity. The resulting algorithm is referred to as differential block luminance. Secondly we have outlined a structure for very efficiently searching in a large fingerprint database.

The combination of these feature extraction and database algorithms results in a robust and very efficient fingerprinting system. Future research will

be mainly focusing on extracting even more robust features, still under the constraint of limited complexity of the extractor and manageable fingerprint database complexity.

References

1. M. Abdel-Mottaleb, G. Vaithilingam, and S. Krishnamachari. Signature-based image identification. In *SPIE conference on Multimedia Systems and Applications II*, Boston, USA, 1999.
2. J. Bancroft. Fingerprinting: Monitoring the use of media assets, 2000. Omnibus Systems Limited, white paper. see http://www.advanced-broadcast.com/.
3. E. Chavez, J. Marroquin, and G. Navarro. Fixed queries array: A fast and economical data structure for proximity searching. *Multimedia Tools and Applications*, 14:113–135, 2001.
4. S.S. Cheung and A. Zakhor. Video similarity detection with video signature clustering. In *Proc. 8th International Conference on Image Processing*, volume 2, pages 649–652, Thessaloniki, Greece, 2001.
5. J. Fridrich. Robust bit extraction from images. In *Proc. IEEE ICMCS'99*, volume 2, pages 536–540, Florence, Italy, 1999.
6. J. Haitsma, T. Kalker, and J. Oostveen. Robust audio hashing for content identification. In *International Workshop on Content-Based Multimedia Indexing*, Brescia, Italy, 2001. accepted.
7. A. Hampapur and R.M. Bolle. Feature based indexing for media tracking. In *Proc. International Conference on Multimedia and Expo 2000 (ICME-2000)*, volume 3, pages 1709–1712, 2000.
8. A.J. Menezes, S.A. Vanstone, and P.C. van Oorschot. *Handbook of Applied Cryptography*. CRC Press, 1996.
9. RIAA-IFPI. Request for information on audio fingerprinting technologies, 2001. http://www.ifpi.org/site-content/press/20010615.html, http://www.riaa.com/-pdf/RIAA_IFPI_Fingerprinting_RFI.pdf.
10. M. Schneider and S.F. Chang. A robust content based digital signature for image authentication. In *Proceedings of the International Conference on Image Processing (ICIP) 1996*, volume 3, pages 227–230, 1996.
11. R. Venkatesan and M.H. Jakubowski. Image hashing. In *DIMACS conference on intellectual property protection*, Piscataway, NJ, USA, 2000.
12. R. Venkatesan, S.M. Koon, M.H. Jakubowski, and P. Moulin. Robust image hashing. In *Proceedings of the International Conference on Image Processing (ICIP)*, 2000.

Author Index

Abdelmoty, Alia I. 175

Boujemaa, Nozha 24, 163
Breiteneder, Christian 105

Chan, Chen-Lung 219
Chang, Shi-Kuo 1
Chen, Duan-Yu 276, 288
Chen, Ling-Hwei 88, 269
Chen, Trista Pei-chun 194
Chen, Tsuhan 194
Chen, Zen 95
Chia, Tsorng-Lin 95
Chiu, Chih-Yi 143
Chun, Seong Soo 239, 259
Costagliola, Gennaro 1

Eidenberger, Horst 105
El-Geresy, Baher A. 175

Falcão, Alexandre X. 12
Fauqueur, Julien 24
Fotouhi, Farshad 187

Gatica-Perez, Daniel 229

Haitsma, Jaap 117
Hitz, Martin 105
Ho, TuBao 311
Hsiao, Ming-Ho 276
Hsu, Vincent 219, 229
Hsu, Yuh-Feng 194
Huang, Thomas S. 129
Hung, Yi-Ping 76

Jones, Christopher B. 175
Jungert, Erland 1

Kalker, Ton 117
Kao, Cheng-Chien 207
Kim, Hyeokman 259
Kim, Jung-Rim 239, 259
Koskela, Markus 247
Kuo, Chin-Ying 219

Laaksonen, Jorma 247
Lee, Suh-Yin 276, 288
Lee, Yuh-Reuy 207
Leung, Clement H.C. 152
Li, Xiaobo 61
Lin, Chia-Wen 207
Lin, Hsin-Chih 143
Lin, Shu-Jiuan 288

Manola, Lubomir 129
Morris, Andrew J. 175

Nakazato, Munehiro 129
Nascimento, Mario A. 12, 61
Nguyen, DucDung 311
Nguyen, TrongDung 311

Oh, Sangwook 259
Oh, Seok-jin 239
Oja, Erkki 247
Olsen, Kai A. 302
Oostveen, Job 117
Qiu, Guoping 50

Saetre, Per 302
Saux, Bertrand Le 163
Shen, Kuan-Ting 269
Shih, Jau-Ling 88
Sridhar, Veena 61
Stanchev, Peter L. 187
Stehling, Renato O. 12
Sudirman, S. 50
Sull, Sanghoon 239, 259
Sun, Ming-Ting 229
Sun, Shu-Kuo 95

Tao, Ju-Lan 76
Tse, Philip K.C. 152

Volmer, Stephan 36

Wang, Jia-Shung 219
Wu, Peter Y. 302

Yang, Shi-Nine 143

Zhou, Zhi 229