October 1995
Volume 13 Number 4

acm Series on
Computing
Methodologies

acm Transactions on Information Systems

Special Issue on Video Information Retrieval

acm PRESS

Published by ACM, the First Society in Computing

acm Transactions on Information Systems

ACM
1515 Broadway
New York, NY 10036
Tel: (212) 869-7440

ACM European Service Center
Avenue Marcel Thiry 204
1200 Brussels, Belgium
Tel: 32 2 774 9602
Fax: 32 2 774 9690
Email: acm_europe@acm.org

ACM Transactions on Information Systems (ISSN 1046-8188) is published 4 times a year in January, April, July, and October by the Association for Computing Machinery, Inc., 1515 Broadway, New York, NY 10036. Second-class postage paid at New York, NY 10001, and at additional mailing offices. Postmaster: Send address changes to Transactions on Information Systems, ACM, 1515 Broadway, New York, NY 10036.

For subscription and submissions information, see inside back cover.

Motion Recovery for Video Content Classification

NEVENKA DIMITROVA and FOROUZAN GOLSHANI
Arizona State University, Tempe

Like other types of digital information, video sequences must be classified based on the semantics of their contents. A more-precise and completer extraction of semantic information will result in a more-effective classification. The most-discernible difference between still images and moving pictures stems from movements and variations. Thus, to go from the realm of still-image repositories to video databases, we must be able to deal with motion. Particularly, we need the ability to classify objects appearing in a video sequence based on their characteristics and features such as shape or color, as well as their movements. By describing the movements that we derive from the process of motion analysis, we introduce a dual hierarchy consisting of spatial and temporal parts for video sequence representation. This gives us the flexibility to examine arbitrary sequences of frames at various levels of abstraction and to retrieve the associated temporal information (say, object trajectories) in addition to the spatial representation. Our algorithm for motion detection uses the motion compensation component of the MPEG video-encoding scheme and then computes trajectories for objects of interest. The specification of a language for retrieval of video based on the spatial as well as motion characteristics is presented.

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems; I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding—*motion*

General Terms: Algorithms, Design

Additional Key Words and Phrases: Content-based retrieval of video, motion recovery, MPEG compressed video analysis, video databases, video retrieval

1. INTRODUCTION

Applications such as video on demand, automated surveillance systems, video databases, industrial monitoring, video editing, road traffic monitoring, etc. involve storage and processing of video data. Many of these applications can benefit from retrieval of the video data based on their content. The problem is that, generally, any content retrieval model must have the capability of

This article is a revised version with major extensions of an earlier paper which was presented at the ACM Multimedia '94 Conference.
Authors' addresses: N. Dimitrova, Philips Laboratories, 345 Scarborough Road, Briarcliff Manor, NY 10562; email: nvd@philabs.philips.com; F. Golshani, Department of Computer Science and Engineering, Arizona State University, Tempe, AZ 85287-5406; email: golshani@asu.edu.

dealing with massive amounts of data. As such, classification is an essential step for ensuring the effectiveness of these systems.

Motion is an essential feature of video sequences. By analyzing motion of objects we can extract information that is unique to the video sequences. In human and computer vision research there are theories about extracting motion information independently of recognizing objects. This gives us support for the idea of classifying sequences based on the motion information extracted from video sequences regardless of the level of recognition of the objects. For example, using the motion information we can not only submit queries like "retrieve all the video sequences in which there is a moving pedestrian and a car" but also queries that involve the exact position and trajectories of the car and the pedestrian.

Previous work in dynamic computer vision can be classified into two major categories based on the type of information recovered from an image sequence: recognition through recovering structure from motion and recognition through motion directly. The first approach may be characterized as attempting to recover either low-level structures or high-level structures. The low-level structure category is primarily concerned with recovering the structure of rigid objects, whereas the high-level structure category is concerned primarily with recovering nonrigid objects from motion. Recovering objects from motion is divided into two subcategories: low-level motion recognition and high-level motion recognition. Low-level motion recognition is concerned with making the changes between consecutive video frames explicit (this is called optical flow [Horn and Schunck 1981]). High-level motion recognition is concerned with recovering coordinated sequences of events from the lower-level motion descriptions.

Compression is an inevitable process when dealing with large multimedia objects. Digital video is compressed by exploiting the inherent redundancies that are common in motion pictures. Compared to encoding of still images, video compression can result in huge reductions in size. In the compression of still images, we take advantage of spatial redundancies caused by the similarity of adjacent pixels. To reduce this type of redundancy, some form of transform-based coding (e.g., Discrete Cosine Transform, known as DCT) is used. The objective is to transform the signal from one domain (in this case, spatial) to the frequency domain. DCT operates on 8×8 blocks of pixels and produces another block of 8×8 in the frequency domain whose coefficients are subsequently quantized and coded. The important point is that most of the coefficients are near zero and after quantization will be rounded off to zero. Run-length coding, which is an algorithm for recording the number of consecutive symbols with the same value, can efficiently compress such an object. The next step is coding. By using variable-length codes (an example is Huffman tables), smaller code words are assigned to objects occurring more frequently, thus further minimizing the size.

Our aim in the coding of video signals is to reduce the temporal redundancies. This is based on the fact that, within a sequence of related frames, except for the moving objects, the background remains unchanged. Thus to reduce temporal redundancy a process known as motion compensation is used. Motion compensation is based on both predictive and interpolative coding.

MPEG (Moving Pictures Expert Group) is the most general of the numerous techniques for video compression [Furht 1994; LeGall 1991; Mattison 1994]. In fact, the phrase "video in a rainbow" is used for MPEG, implying that by adjusting the parameters, one can get a close approximation of any other proposal for video encoding. Motion compensation in MPEG consists of predicting the position of each 16×16 block of pixels (called a macroblock) through a sequence of predicted and interpolated frames. Thus we work with three types of frames—namely, those that are fully coded independently of others (called reference frames or I-frames), those that are constructed by prediction (called predicted frames or P-frames), and those that are constructed by bidirectional interpolation (known as B-frames). It begins by selecting a frame pattern which dictates the frequency of I-frames and the intermixing of other frames. For example, the frame pattern IBBPBBI indicates (1) that every seventh frame is an I-frame, (2) that there is one predicted frame in the sequence, and (3) that there are two B-frames between each pair of reference and/or predicted frames. Figure 1 illustrates this pattern.

Our approach to extracting object motion is based on the idea that during video encoding by the MPEG method, a great deal of information is extracted from the motion vectors. Part of the low-level motion analysis is already performed by the video encoder. The encoder extracts the motion vectors for the encoding of the blocks in the predicted and bidirectional frames. A macroblock can be viewed as a coarse-grained representation of the optical flow. The difference is that the optical flow represents the displacement of individual pixels while the macroblock flow represents the displacement of macroblocks between two frames. At the next, intermediate level, we extract macroblock trajectories which are spatiotemporal representations of macroblock motion. These macroblock trajectories are further used for object motion recovery. At the highest level, we associate the event descriptions to object/motion representations.

Macroblock displacement in each individual frame is described by the motion vectors which form a coarse optical-flow field. We assume that our tracing algorithm is fixed on a moving set of macroblocks and that the correspondence problem is elevated to the level of macroblocks instead of individual points. The advantage of this elevation is that even if we lose individual points (due to turning, occlusion, etc.) we are still able to trace the object through the displacement of a macroblock. In other words, the correspondence problem is much easier to solve and less ambiguous. Occlusion and tracing of objects which are continuously changing are the subject of our current investigations.

In Section 2 of this article we survey some of the research projects related to our work. In Section 3 we present the object motion analysis starting from the low-level analysis through the high-level analysis. We discuss the importance of motion analysis and its relevance to our model which is presented in Section 3.4. Section 4 introduces the basic OMV structures (object, motion,

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.