J. Vis. Commun. Image R. 22 (2011) 432–439

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Automatic video activity detection using compressed domain motion trajectories for H.264 videos

Haowei Liu [a,*], Ming-Ting Sun [a], Ruei-Cheng Wu [b], Shiaw-Shian Yu [b]

[a] University of Washington Seattle, WA 98195, USA
[b] Industrial Technology Research Institute, Hsinchu, Taiwan

ARTICLE INFO

Article history:
Received 25 June 2010
Accepted 24 March 2011
Available online 29 March 2011

Keywords:
Activity detection
Vehicle tracking
Compressed domain processing
H.264 videos
Object classification
Trajectory classification
Motion vector tracking
Residual error tracking

ABSTRACT

Most automatic event detection methods for video surveillance detect target events based on features extracted in the pixel domain. However, in practice, surveillance videos are often compressed. It is desirable to perform automatic event detection in the compressed domain directly so that the video does not need to be decoded for analysis purpose. In this paper, we investigate the use of motion trajectories for video activity detection in the compressed domain. We show it is possible to extract reliable motion trajectories directly from compressed H.264 video streams. To overcome the problems caused by unreliable motion vectors, we propose to include the information from the compressed domain prediction residuals to make the tracking more robust. We use a real world application of detecting vacant or occupied parking spaces to demonstrate the effectiveness of our proposed approach. We also demonstrate the robustness of our approach to different encoder settings, and lighting conditions.

1. Introduction

Over the past few years, research on video surveillance systems with automatic event detection has permeated the field of video analysis and computer vision. Most of the video surveillance systems detect target events based on features extracted using object segmentation and tracking in the pixel domain. However, in practice, surveillance video is often compressed for transmission or storage. It is desirable to be able to perform automatic event detection in the compressed domain directly, so that the video does not need to be decoded for the analysis purpose. This is especially important for real-time low-cost software implementations where the computation complexity is critical or for wireless surveillance systems where the power consumption is a major concern.

Current pixel-domain tracking algorithms usually perform background subtraction or segmentation to extract moving objects, and subsequently build the correspondences among the objects across different video frames using models such as Kalman filters or particle filters. An extensive review of the state of the art tracking algorithms can be found in [1].

From MPEG to H.264, the culmination of video compression technology has led to a vast amount of video databases, consisting of a huge amount of compressed video clips, making people consider compressed-domain event detection. In [2], Lie presents an approach to object tracking for MPEG-2 video sequences by macroblock linking. After linkage, paths are pruned and merged into motion trajectories. Fonseca and Nesvadba [3] propose to use color and texture information decoded from the I frames along with the motion vectors (MV's) in MPEG-4 videos for general object tracking applications. In [4], the authors perform background subtraction on DC images combined with motion vectors to segment pedestrians. In [5], it formulates a background model in the compressed domain and concludes that as long as the background model is linear, it can be formulated well in the compressed domain.

Most of the aforementioned compressed-domain object tracking methods rely on motion vectors. One major problem in compressed-domain motion trajectory extraction using motion vectors is that video compression standards actually do not specify how to perform motion estimation. A standard-conforming video encoder can use different methods to perform motion estimation resulting in different motion vectors. Small search windows can also result in unreliable motion vectors when the motions exceed the search range. Motion estimation is a block-matching algorithm, aiming at finding the best matching block (i.e., the block with the most similar pattern) in the reference frames. Thus, the resulting motion vectors may be unreliable

* Corresponding author.
E-mail addresses: hwliu@u.washington.edu (H. Liu), mts@u.washington.edu (M.-T. Sun), AllenRCWu@itri.org.tw (R.-C. Wu), ssyu@itri.org.tw (S.-S. Yu).

1047-3203/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.jvcir.2011.03.010

and not necessarily represent the underlying motions. In an extreme case, a video encoder could completely skip the motion estimation process and use (0,0) for all motion vectors. This will cause severe problems to the motion trajectory extraction in the compressed domain. It is desirable to have a motion trajectory extraction algorithm which can work robustly under situations with unreliable motion vectors.

H.264 is the current state-of-the-art video compression standard. It can achieve two to three times better performance than MPEG-2. Compared to MPEG-2, H.264 supports variable-block-size motion estimation. The block-size can be 16×16, 16×8, 8×16, 8×8, 8×4, 4×8, or 4×4. It also supports multiple reference frames and intra-prediction. Although many compressed domain techniques have been proposed, they are mainly for MPEG-2. With so many new functionalities introduced in H.264, it is not clear how these approaches would work under these new features or different parameter settings. Hence, it would be interesting to investigate the effects of variable-block-sizes, different motion estimation methods, multiple reference frames, and intra-prediction on the motion trajectory extraction method.

Motivated by providing a low cost surveillance solution, in this paper, we propose an approach to activity detection using compressed domain motion trajectories and show as an example a parking-lot application. Our main contributions are summarized as follows:

1. We propose an approach to robustly extract reliable motion trajectories directly from compressed H.264 video streams with variable block-size multiple reference frames motion estimation and intra-prediction video encoding settings.
2. We propose an approach that tackles the problem of unreliable motion vectors based on compressed domain prediction residuals.
3. We investigate the robustness of motion trajectories under different H.264 coding modes such as variable block size, motion estimation methods, intra-prediction, and poor lighting conditions.
4. We propose a new scheme for detecting parking spaces based on motion trajectories, which could complement other pixel-based parking space detecting methods. Use this application as an example, we demonstrate the effectiveness of our proposed compressed-domain event detection approach.

The organization of the rest of paper is as follows. In Section 2, we describe the proposed motion trajectory extraction method for H.264 bit-streams. In Section 3, we discuss the use of the motion trajectories for parking space detection as an example application. In Section 4, we discuss the robustness and complexity of our approach under different conditions. Finally, we conclude the paper in Section 5. We assume an IPPP GOP structure throughout the paper.

2. Forming robust motion trajectories in the H.264 compressed domain

2.1. Trajectory forming using block motion vectors for variable block-sizes and multiple reference frames

In this section, we discuss the extraction of object trajectories in H.264 compressed videos.

Compared to the motion estimation in previous video coding standards, H.264 introduces variable block-size and multiple reference frame motion estimation. The implication of variable block-size motion estimation is that motion vectors can be of different granularities covering variable-size block areas. To handle the variability, we compute the motion vector in the smallest granularity, i.e., the 4×4 block, based on the motion information in the compressed bit stream. For example, if the motion vector for a 16×16 macroblock (MB) is V, then we can obtain the motion vector for each 4×4 block belonging to the MB by duplicating V. The motion trajectory extraction is then performed at the 4×4-block level. The effect of using the 4×4 blocks instead of the 16×16 MB in MPEG-2 in the motion trajectory extraction is that the motion vectors from a smaller block-size may be less reliable, since the smaller blocks have a higher probability to match to similar blocks which do not represent true motions. On the other hand, using a larger block-size such as a 16×16 MB, it has a higher probability that the MB may contain different objects with different motions, which could also cause inaccurate motion vectors.

We handle the case of multiple reference frames as follows. As shown in Fig. 1, let $A^{F_t} = (A_x^{F_t}, A_y^{F_t})$ be the coordinates of block A in frame F_t, $MV^A = (MV_x^A, MV_y^A)$ the motion vector pointing to F_{t-n}, then the corresponding location A'' of block A in frame F_{t-n} can be estimated using

$$\begin{cases} A_x^{F_{t-n}} = A_x^{F_t} + MV_x^A \\ A_y^{F_{t-n}} = A_y^{F_t} + MV_y^A \end{cases} \tag{1}$$

The corresponding location of block A in frame F_{t-n+1}, F_{t-n+2}, $\ldots, F_{t-1}$ can then be interpolated using A^{F_t} and $A^{F_{t-n}}$. If the location in frame F_{t-i} does not align with the block boundary (the shaded area in Fig. 1 shows all possible blocks overlapping with the matching block), the one with the largest overlapping area is selected as the corresponding block. It should be noted that the corresponding block A' in Frame $t-i$ could have a motion vector pointing to a different block A''' in other frames, resulting in a relatively noisy trajectory structure. However, as shown later, these noisy trajectories could be pruned, merged, and smoothed to result in clean trajectories.

Given a compressed video sequence $\{F_t\}_{t=1}^N$ where N is the number of video frames in the video sequence, with the motion vector field associated with each frame F_t, we could construct trajectories by linking the corresponding blocks across the set of frames.

2.2. Dealing with unreliable or missing motion vectors: forming complementary trajectories based on prediction residuals

One major problem of solely depending on the approach in the previous section is that it makes the assumption that the motion vectors highly reflect the true motions. However, it is not uncommon in real world scenarios that the motion vectors are poor indications of true motions, depending on different encoder implementations such as the search range or motion estimation algorithms. As mentioned, video coding standards do not specify how to perform motion estimation. Motion estimation is just a block matching process. It could match to a block with a similar pattern which does not represent the true motion. For example, in the extreme, the encoder could use (0,0) for all the motion vectors, in which case, the aforementioned approach would fail miserably.

To address this issue and make the tracking more robust, we propose to use the sum of squared DCT coefficients of the residual errors as another modality of signal to make the motion trajectory extraction more robust. When the motion vector is not reliable due to the limited search range or being set to (0,0), the prediction residuals often will be relatively large. By considering both MV's and the prediction residuals, we cover both cases that a motion block is tracked well and not tracked well using MV's.

Specifically, other than the motion vectors, we also keep track of the sum of squared values of the DCT coefficients of the prediction residuals. The intuition is that, in most surveillance applications,