Temporal Motion Vector Filter for Fast Object Detection on Compressed Video

Ronaldo Carvalho Moura, Elder Moreira Hemerly, and Adilson Marques da Cunha

***Abstract*—A novel Temporal Motion Vector Filter (TF) is presented and evaluated for real-time object detection on compressed videos in MPEG-2, MPEG-4 or H.264/AVC formats. The filter significantly reduces the noisy motion vectors that do not represent a real object movement . The filter analyses the temporal coherence of block motion vectors to determine if they are likely to represent true motion in the recorded scene.**

Experiments are performed using the CLEAR metrics for object detection and public available datasets from CAVIAR, PETS and CLEAR. These experiments demonstrate that the TF outperforms the Vector Median Filter, by providing better object detection accuracy with reduced computational complexity.

The good results obtained by the TF make it suitable as a first step towards implementing systems that aim to detect and track objects from compressed video by using motion vectors. The TF could also be used to improve other techniques based on motion vectors such as Global Motion Estimation (GME) and Motion-Compensated Frame Interpolation (MCFI).

***Index Terms*—Object detection, object tracking, video compression, motion estimation, vector median filter, spatiotemporal motion vector filter, compressed-domain segmentation, real-time segmentation, video-indexing, MPEG, H.264/AVC, global motion estimation, motion-compensated frame interpolation.**

I. Introduction

OBJECT tracking techniques aims at tracking objects in consecutive video frames. During the last two decades several techniques have been proposed for video object tracking with applications to Video Surveillance [1], Intelligent Transportation System - ITS [2], Human Machine Interface - HMI, Video Indexing [3], [4] and Shopping Behavior Analysis.

The adoption of surveillance cameras everywhere and interest in automatic video indexing served as stimulus for recent research on object tracking and behavior recognition, such as in AVSS (IEEE International Conference on Advanced Video and Signal based Surveillance from 1998, 2001, 2003, 2005-2013), PETS (IEEE International Workshop on Performance Evaluation of Tracking and Surveillance from 2000-2010,2012-2013), CLEAR (Classification of Events, Activities and Relationship Evaluation Workshop from 2006, 2007), CBMI (IEEE International Workshop on Content-Based Multimedia Indexing from 1999, 2001, 2003, 2005, 2007-2013) and ICDCS (ACM/IEEE International Conference on Distributed Smart Cameras from 2007-2013).

Despite the increasing microprocessors computational power in recent years, the processing required by object tracking techniques still consists in a bottleneck to their wider adoption, specially in low cost embedded equipment as surveillance cameras and mobile devices. To reduce this computational power demand, some techniques that extract object motion information from compressed video streams, instead of the raw video, have been developed.

By taking advantage of important information inside video compressed by standards like MPEG family, these techniques are capable of tracking an object without the need to fully decompress video data, reducing by orders of magnitude the required computational complexity. The main compressed domain information used for segmentation and tracking is the block motion vectors and the Discrete Cosine Transform (DCT) coefficients.

However, the motion vectors (*mv*) contained in the compressed video are chosen to minimize video bitstream, while maintaining its human perceptible quality, and not to represent true objects motion. Consequently, the *mv* can represent both a real object movement or two similar block textures in consecutive frames (fake movements). To make the motion vectors useful for further segmentation steps, it is necessary to remove the noisy ones, i.e., the motion vectors that do not represent a real object movement.

This paper presents a novel Temporal Motion Vector Filter (TF) to remove noisy motion vectors for object tracking purpose with low computational effort. The novelty of the TF consist in the combined use of the Equations 1 to 5 presented in Section III. The TF can be applied to motion vectors grouped in any format, with fixed block-size as in MPEG-2 and MPEG-4, or with variable block-size as in H.264.

The TF consists in a faster and more accurate replacement for Vector Median Filter, a still widely used technique for motion vector filtering. The Temporal Motion Vector Filter is intend solely to classify each motion vector as reliable or not reliable representation of the real world motion. It does not replace the techniques already proposed to estimated camera motion, or to estimate motion vectors in intra-coded blocks or frames. But the TF can be seamlessly associated with these techniques to improve their results.

In Section II, several related works for spurious motion vector removal are reviewed, especially the widely adopted Vector Median Filter. The TF approach is presented in Section III. In

Manuscript received January 28, 2013; revised March 3, 2014; accepted April 30, 2014.

Ronaldo C. Moura was with Monity, Sao Jose dos Campos, Brazil (email: ronaldo.moura@monity.com.br).

Elder M. Hermely and Adilson M. Cunha are with Instituto Tecnologico de Aeronautica, Sao Jose dos Campos, Brazil (email: hemerly@ita.br; cunha@ita.br).

This work was supported by Fundacao de Amparo a Pesquisa de Sao Paulo - Brazil (FAPESP), within the small business innovation research program PIPE n. 2004/09102-1.

Section IV, the proposed approach is tested and evaluated. Conclusions are presented in Section V.

II. Related Works

Several approaches have been proposed to object segmentation and tracking on compressed videos.

In [2] it is presented a simple car tracking technique for fixed cameras, based on motion vectors from MPEG2 video. First, a *vector median filter* is applied to all motion vectors, then nonzero motion vectors are grouped (labeled) according to their direction and magnitude proximity. Each blob is projected on previous frame, according to the mean block motion vectors value, and then matched to the nearest blob.

The well referenced Favalli work [5] presents a supervised tracking techniques based on motion vectors. The first step consists in manually selecting the frame macroblocks that must be tracked. Then, each selected macroblock is tracked in a frame by frame basis by using its motion vectors projection. In [6], a macroblock tracking technique improves [5] by creating two independent layers on the top of the macroblock grid, thereby allowing a more fine grained tracking of object boundaries with resolution superior to macroblock size.

The work in [7] presents a cascade filter for motion-vectors smoothing and noisy reduction. The cascade filter consists of a two-dimensional (spatial) Gaussian filter followed by a median filter. The cascade filter performance is compared with other filtering alternatives, as vector median filter, in one video from MPEG-7 testing dataset and presents better noisy removal results.

For a non-fixed camera, i.e., performing zoom, rotation, pan, tilt or translation operations, some techniques have been proposed to determine the Global camera Motion Estimation (GME), to deal with these operations before performing object tracking. The GME allows the segmentation of the motion-vectors associated with the camera motion (background), and also associated with a real object movement (foreground).

The Kim and Kim paper [8] presents a detailed eight parameters linear estimation model for a camera performing three-dimensional rotation and zooming, but without translation. The motion vectors with high activity in luminance, such as edges and high textures, are selected as feature point for a least-square estimator.

In [9], Roy Wang *et al.* propose a set of confidence measures for DCT and motion-vectors based object tracking for moving camera. The motion-vectors are compared with their neighbors, resulting in separated magnitude and direction confidences. A texture confidence measure is taken by analysing regions with low AC energy in their DCT coefficients. This lower AC energy represents lower textured regions, such as roads and sky, where motion-vectors are usually less reliable. All confidence measures are then weighted and used in a recursive least square GME, to determine camera zoom, vertical and horizontal translations. The resulting motion-vectors are then processed by a 3-dimension vector median filter, and segmented with a K-means clustering followed by an Expectation Maximization (EM) clustering.

In [10], an eight parameters bilinear equation for camera global motion estimation is presented. The parameters are iteratively calculated by a least-squares estimator, by removing outliers with error greater than average error.

The well referenced Mezaris *et al.* work [3] uses the global motion estimation technique from [10] to automatically segment macroblocks in frame t, as foreground or background, and then applies the macroblock tracker [5] on foreground, thereby resulting an estimated foreground map for frame $t+1$. This estimated macroblock foreground map for frame $t+1$, and the foreground map created by application of [10] on next frame $t+1$, are intersected resulting in a filtered foreground estimation. This process is executed during n consecutive frames, providing good macroblock tracker without the burden of infinite error propagation in [5], as the tracked region of interest is constantly reset. The background with different color tones is also segmented using DC coefficients of DCT transform (Y, CB, CR) of macroblocks presented in I-frames.

The work in [11] employs the GME of six parameters from [12] to automatically segment moving objects, and then applies a median filter on foreground macroblocks along their motion trajectory in the same group of pictures, usually containing 8 frames, to filter outliers. The filtered foreground macroblocks are grouped into blobs by using timed Motion History Image technique, from [13], together with a connected component analysis. Blobs tracking are performed by 20x20 pixel window search on previous frame from estimated position of blob (center of gravity plus average motion vector).

In [4], motion-vectors and DC color coefficients are used to overcome the Mezaris *et al.*. [3] limitations for tracking object motion with small differences compared with camera motion model. The GME from [12] is also used to automatically segment moving object.

In the well referenced work [14], Babu *et al.* implement object segmentation based on motion-vectors from compressed videos. The motion vectors from P and B frames are accumulated over a few frames, median filtered, interpolated and segmented with an Expectation Maximization (EM) algorithm.

In our previous work [15], it was presented a Spatiotemporal Motion Vector Filter (STF) that removes noisy motion-vectors that do not represent a real object movement, and allows improved object detection based on the motion-vector information presented in compressed videos. The filter analyses the spatial (neighborhood) and temporal coherence of block motion vectors to determine if they are likely to represent true motion in the recorded scene. The STF was compared with the Vector Median Filter (VMF) approach, by using the CLEAR Multiple Object Detection metrics described in Kasturi *et al.* work [16]. In the two analysed scenes, from PETS and CAVIAR public video datasets, the STF outperformed the VMF, with improvements specially in highly noisy scenes.

In [17] is presented a method to estimate the reliability of motion-vectors compressed in H.264/AVC format. Each block motion-vector is compared with motion-vectors projected from previous and forward frames to determine their likelihood of representing a true object motion. The concept of motion-vector projection used in this work is analogue to the temporal analysis from the STF [15]. But while in STF a current motion-vector is recursively projected to previous frames, in [17] the motion-vectors from previous and forward frames are

projected to current frame. While in STF each mv has only one projection path, in [17] several mv can be projected to the same block, what results in true mv being averaged by noisy mv that points to the same blocks. This difference results in better STF temporal analysis performance, specially in scenes with several noisy mv surrounding true mv.

The work [18] presents a method for detecting and tracking objects from compressed H.264/AVC video using the motion-vectors and block residue. The method uses a graph based representation, with pixel blocks represented as vertices, vector of block properties (as location, motion vector direction and magnitude, and residue amount) represented as vertices attributes, and Euclidean distance between blocks properties as the edges weight. The relation between blocks of adjacent frames results in a spatio-temporal tracking graph. The work also suggests that the use of the STF [15] should improve its object boundary detection.

The Vector Median Filter Limitations

The Vector Median Filter is widely adopted for motion vectors filtering and is presented in older and newer works as in [14], [2], [11], and [9]. However, the use of two-dimensional (spatial) vector median filter presents limitations, such as:

- **Inability to filter highly noisy regions** - Low-textured regions such as floors, sky, and walls, have a high concentration of noisy motion vectors that are not removed with the vector median filter.
- **Inability to track small objects** - Small objects, with size of one or two blocks, are mostly incorrectly filtered (removed), as their neighborhood do not have the same moving pattern.

The attempt to reduce these problems, by also taking into account temporal information and creating a three dimensional (3D) vector median filter, presents another problem:

- **Inability to track fast moving objects** - Fast objects are likely to present significant block movement, while normal median filter assumes that temporally adjacent blocks represent the same data. This will cause the object to be deformed, with its front incorrectly filtered (removed) and an incorrect tail created.

III. Temporal Motion Vector Filter

The TF principle is based on the **empirical observation** of block motion vectors. Real world moving objects, as persons or cars, usually produce motion vectors with smooth variations in successive frames, while homogeneous surfaces without real object movement (as floor, wall, road or sky) produce almost random motion vectors.

Other works have already considered the temporal consistency of motion vectors to analyse their reliability as a real motion, as described in section II. But all of them are more computationally complex than the vector median filter, or demands additional information as DCT residuals in [14], hindering their adoption and implementation.

The origin of the Temporal Motion Vector Filter (TF) proposition can be traced back to the careful analysis of the two components of the Spatiotemporal Motion Vector Filter (STF) presented in our previous work [15]. By observing the individual results of the temporal analysis and the spatial analysis, it was noticed that they have similar filtering capability in less noisy scenes. In more noisy scenes, the temporal analysis had a significant smaller false detection rate. The spatial analysis was unable to deal with too much noisy, just as the Vector Median Filter (VMF). The main comparative advantage of spatial analysis was the early detection of objects, when there is not enough temporal information.

As consequence of these observations, the new TF was designed. It uses only the temporal analysis from the STF, removing the slower and less effective spatial (neighbor) analysis. Hence, the TF is a simplification of the STF and aims to produce faster results with the same filtering quality.

Notation of TF Equations

- $(x, y)^t$: the pixel coordinate (x, y) in frame t.
- $mv(x, y)^{\overrightarrow{t,t_{ref}}}$: the motion vector of pixel $(x, y)^t$ from frame t to frame t_{ref}.
- $mv(x, y)^t$: the normalized motion vector of pixel $(x, y)^t$ from frame t to the previous frame $t - 1$, noted N and defined by Equation 1.
- $(\hat{x}, \hat{y})^{t-1}$: the estimated position of the pixel $(x, y)^t$ in last frame $t - 1$.

.

1. Motion Vector Normalization - A motion vector from a P frame references a past frame. A motion vector from a B frame makes reference to a past or a future frame. To simplify motion vector projection equations and computing data structures, motion vectors need to be normalized in order to reference only to the previous frame. This is accomplished by dividing the motion vectors by the difference between the current frame number and the reference frame number, according to Motion Vector Normalization (N) Equation 1, similarly to the process used in [14]. If the reference frame is a future frame, the divisor will be a negative number, reversing the *mv* direction. The normalized motion vector is an approximation of $mv(x, y)^{\overrightarrow{t,t-1}}$, i.e., the motion vector referencing the previous frame. In this paper the normalized motion vector $N(mv(x, y)^{\overrightarrow{t,t_{ref}}})$ is represented as $mv(x, y)^t$.

$$N(mv(x,y)^{\overrightarrow{t,t_{ref}}}) = \frac{mv(x,y)^{\overrightarrow{t,t_{ref}}}}{t - t_{ref}} \approx mv(x,y)^{\overrightarrow{t,t-1}} \quad (1)$$

Another alternative for the Motion Vector Normalization consists in discarding the B frames, and using only the motion vectors from P frames. B frames can be discarded when the frequency of P frames alone are enough to detect desired motion, usually when the camera has a broad and far view of the scene.

The TF algorithm makes no assumption about the macroblock size or format, but in the case of variable block-size, such as in H.264, it shall be useful to split the block to the smallest size available (i.e. 4x4), as presented in [18], to simplify the computing data structures.

2. Temporal Consistency Analysis - Each block center $(x, y)^t$ has its position in previous frame estimated by adding

to it the corresponding normalized motion vector, as presented in Figure 1 and Projection (P) Equation 2.

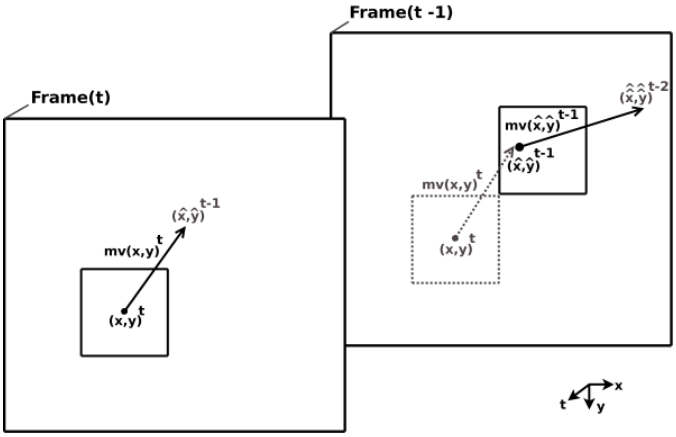


Fig. 1. Illustration of the Projection (P) Equation 2. A pixel $(x,y)^t$ added to its motion vector $mv(x,y)^t$ produces its estimated position in the previous frame, represented as $(\hat{x},\hat{y})^{t-1}$.

$$\begin{aligned} P(x,y)^{\overrightarrow{t,t-0}} &= (x,y)^t \\ P(x,y)^{\overrightarrow{t,t-1}} &= (x,y)^t + mv(x,y)^t = (\hat{x},\hat{y})^{t-1} \\ P(x,y)^{\overrightarrow{t,t-2}} &= (x,y)^t + mv(x,y)^t + mv(\hat{x},\hat{y})^{t-1} \\ &= (\hat{\hat{x}},\hat{\hat{y}})^{t-2} \\ P(x,y)^{\overrightarrow{t,t-k}} &= P\left(P(x,y)^{\overrightarrow{t,t+1-k}}\right)^{\overrightarrow{t+1-k,t-k}} \end{aligned} \tag{2}$$

Then, the motion vectors from these two related blocks, $(x,y)^t$ and its projection in previous frame $P(x,y)^{\overrightarrow{t,t-1}}$, have their direction and magnitude coherence simultaneously analyzed by using the Vector Matching Ratio (R) Equation 3.

$$R(\vec{a},\vec{b}) = \begin{cases} 1, & \|\vec{a}\| = \left\|\vec{b}\right\| = 0 \\ 1 - \frac{\|\vec{a}-\vec{b}\|}{\|\vec{a}\|+\|\vec{b}\|}, & otherwise \end{cases} \tag{3}$$

The Temporal Consistency Analysis can be recursively calculated for previous frames $t-1$, $t-2$ ... $t-n$, by means of the Temporal Consistency Index(TCI) Equation 4a.

$$TCI(mv(x,y)^t) = \sqrt[n]{\prod_{1\le i\le n} R\left(mv(P(x,y)^{\overrightarrow{t,t+1-i}}), mv(P(x,y)^{\overrightarrow{t,t-i}})\right)} \tag{4a}$$

For $n = 2$ the TCI results in Equation 4b.

$$TCI(mv(x,y)^t) = \sqrt[2]{R(mv(x,y)^t, mv(\hat{x},\hat{y})^{t-1}) \cdot R(mv(\hat{x},\hat{y})^{t-1}, mv(\hat{\hat{x}},\hat{\hat{y}})^{t-2})} \tag{4b}$$

A motion vector is considered consistent if its TCI is above a minimum threshold, as described in Temporal Motion Vector Filter (TF) Equation 5. The motion vector classified as noise is set to the background motion vector value bg_mv , for instance, $(0,0)$ in the case of static cameras or a value calculated by a global motion technique as [10] for moving cameras.

$$TF(mv(x,y)^t) = \begin{cases} mv(x,y)^t, & TCI(mv(x,y)^t) \ge \tau \\ bg\_mv(x,y)^t, & otherwise \end{cases} \tag{5}$$

Good filtering results were obtained from tested sequences by setting the number of previous frames to $n = 2$, and the temporal threshold to $\tau = 50\%$. Usually threshold selection consists in a critical part of several segmentation algorithms in the literature, making difficult to automatically apply a given algorithm to different cases without threshold tuning for each scene type. Fortunately TCI differs from these algorithms by having a good natural segmentation capability, producing most of the values near zero or near 100% in tested image sequences, even with very different conditions of illumination and scene types. As further demonstrated in Figure 8 very few TCI values are between 20% and 60%, and $\tau = 50\%$ can be used as default threshold for all scene types.

A. *Vector Matching Equations Analysis*

To numerically calculate the difference between consecutive motion vectors, a vector similarity equation must be chosen. Following are presented some equations capable of comparing the vectors $\vec{a}$ and $\vec{b}$, with angle θ between them, and $\|\vec{a}\| \ge \left\|\vec{b}\right\|$ considering its direction and magnitude. For all equations $R(\vec{a},\vec{b}) = 1$ if $\vec{a} = \vec{b} = (0,0)$.

$$Ra(\vec{a},\vec{b}) = \frac{\left\|\vec{b}\right\|}{\|\vec{a}\|}\frac{180^\circ - \theta}{180^\circ} \tag{6a}$$

$$Rb(\vec{a},\vec{b}) = \frac{\left\|\vec{b}\right\|}{\|\vec{a}\|}\frac{cos(\theta)+1}{2} \tag{6b}$$

$$Rc(\vec{a},\vec{b}) = \frac{e^{-\frac{\|\vec{a}-\vec{b}\|^2}{(\|\vec{a}\|+\|\vec{b}\|)^2}} - e^{-1}}{1-e^{-1}} \tag{6c}$$

$$Rd(\vec{a},\vec{b}) = 1 - \frac{\|\vec{a}-\vec{b}\|}{\|\vec{a}\|+\|\vec{b}\|} \tag{6d}$$

Equation 6a consists in the reference equation, representing a direct relation between vectors magnitude and angle. Equation 6b consists in a computational simplification of Equation 6a, as the cosine between vectors is easier to calculate than the angle itself, using the dot product equation $\vec{a}\cdot\vec{b} = cos(\theta)\,\|\vec{a}\|\,\left\|\vec{b}\right\|$. Equation 6c consists in a version of the equation used in [17], normalized to produce values between 0 and 1.

The vector matching ratio can be represented in terms of angle θ between vectors, and the magnitude ratio $\frac{\|\vec{b}\|}{\|\vec{a}\|}$. The Figure 2 presents how the different equations segment vectors with a matching ratio greater than 50%. For vectors pointing to the same direction ($\theta = 0°$) the matching ratio is greater than 50% if the larger vector is limited to twice the magnitude of the other vector in Equations 6a and 6b, to three times in Equation 6d, and about four times in Equation 6c.

For a matching of 50% the maximum angle between vectors ,when $\left\|\vec{b}\right\| = \|\vec{a}\|$, is 60° in Equation 6d, about 76° in Equation 6c, and 90° in Equation 6a and 6b.

With proper threshold adjustment, all these equations could be used to calculate Temporal Motion Vector Filter (TF) Equation 5. Equation 6d (also Equation 3) was selected because it needs less computer operations than Equations 6a and 6c, has a simpler mathematical representation than Equation 6b, without the need to define the larger vector or the angle between them, and presents a fair filtering capability by penalizing vectors with greater angle difference.

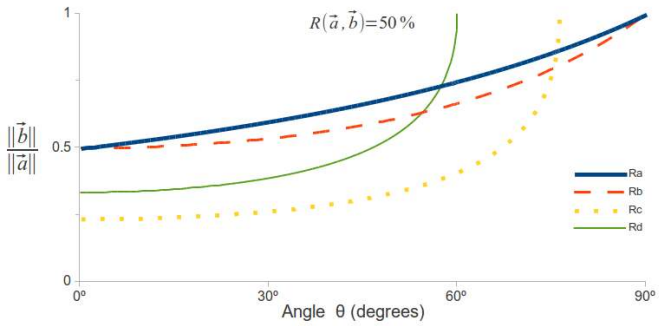


Fig. 2. Relation of magnitude ratio $\frac{\|\vec{b}\|}{\|\vec{a}\|}$ and angle θ between vectors for matching ratio of 50%. The points above each equation line produce a match greater than 50%.

B. Mathematical TF Analysis

In this section it is presented some theoretical TF algorithm analysis in order to clarify its applications and limits. In all following cases TF algorithm is applied with the default threshold of 50% and $n = 2$.

One of the most common noisy motion vector value consists in a vectors with magnitude equal to one in static cameras. This observation can be illustrated by the noisy motion vectors of test Figures 9 to 12. Let us assume the hypothesis that, for a pixel block with non-real movement, any motion vector of the set $\{(0,0),(1,0),(0,1),(-1,0),(0,-1)\}$ could be assigned with the same probability of 20%. In this synthetic case, the probability of a static block having a noisy motion vector, i.e. $mv^t \neq (0,0)$, would be 80%. But after applying the TF Equation this probability would be reduced to 16% as presented in Equation 7, with θ_1 representing the angle between mv^t and mv^{t-1}, and θ_2 representing the angle between mv^{t-1} and mv^{t-2}.

$$Prob\Big(mv^t \neq (0,0)\Big) \cdot Prob\Big(\theta_1 = \theta_2 = 0° \bigcup \theta_1 = 0°, \theta_2 = 90° \bigcup \theta_1 = 90°, \theta_2 = 0°\Big) = \frac{4}{5} \cdot \left(\frac{1}{5} \cdot \frac{1}{5} + \frac{1}{5} \cdot \frac{2}{5} + \frac{2}{5} \cdot \frac{1}{5}\right) = 16\% \tag{7}$$

The motion vectors of a object performing uniform circular motion, will be correctly processed by TF algorithm as long as object instantaneous angular velocity is limited to 60 degrees/frame. A faster angular velocity will result in consecutive motion vectors been wrongly classified as not coherent.

The linear motion of self-propelled objects, as persons and vehicles, from a stopped state to its maximum velocity can be reasonably approximated as an initial acceleration a and jerk (derivative of acceleration) determined by Equation 8a, with k representing the constant time when object reaches its maximum velocity v_{max}, as illustrated in Figure 3.

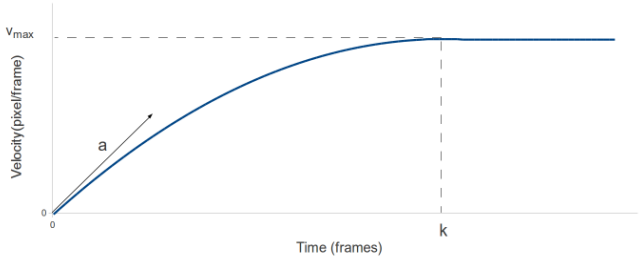


Fig. 3. Modeling of objects linear motion, as persons and vehicles, with initial acceleration a and constant jerk $j(t) = -a/k$ until maximum velocity.

$$j(t) = \begin{cases} -a/k, & t < k \\ 0, & otherwise \end{cases} \tag{8a}$$

$$v(t) = \begin{cases} at - \frac{at^2}{2k}, & t < k \\ 0.5ak, & otherwise \end{cases} \tag{8b}$$

$$mv(t) = \int_{t-1}^{t} v(t)dt \tag{8c}$$

$$= \begin{cases} a\frac{-3t^2+(3+6k)t-3k-1}{6k}, & t \leq k \\ \int_{t-1}^{k} at - \frac{at^2}{2k}dt + \int_{k}^{t} 0.5akdt, & k < t < k+1 \\ 0.5ak, & t \geq k+1 \end{cases}$$

$$R(mv(t), mv(t-1)) = 1 - \frac{mv(t) - mv(t-1)}{mv(t) + mv(t+1)} \geq 0.5$$

For $t \leq k$

$$3t^2 - 12t - 6kt + 12k + 10 \leq 0$$

$$2 + k - \sqrt{k^2 + \frac{2}{3}} \leq t \leq 2 + k + \sqrt{k^2 + \frac{2}{3}} \tag{8d}$$

Equation 8c determines the motion vector of an object correctly detected by the motion estimation algorithm. In the

first frames after an object starts its linear acceleration the TF algorithm will wrongly classify the motion vectors as noisy, until motion vectors of consecutive frames are similar enough. Equations 8d and 8c imply that, for any acceleration, if $k \geq 2$ and $t \geq 2$ then Vector Matching Ratio (R) will be greater or equal to 50%. Therefore, the TF will correctly classify the motion vectors of an object performing the linear motion of Figure 3 after 3 frames, no matter what its acceleration is.

IV. Experimental Results

The CLEAR Multiple Object Detection metrics described in Kasturi *et al.* work [16] were used to numerically compare the capability of the Vector Median Filter (VMF), Spatiotemporal Motion Vector Filter (STF), and the proposed Temporal Motion Vector Filter (TF) to correctly detect true objects motion.

CLEAR MOD metrics notation

- N_{frames}: the number of frames in video sequence.
- $G_i^{(t)}$: the *i*th ground truth object in frame *t*.
- $D_i^{(t)}$: the *i*th detected (by the evaluated technique) object in frame *t*.
- $N_G^{(t)}$: number of ground truth objects in frame *t*.
- $N_D^{(t)}$: number of detected objects in frame *t*.
- $N_{mapped}^{(t)}$: number of match pairs between ground truth and detected objects in frame *t*.

The **Multiple Object Detection Accuracy - MODA** metric uses the number of missed detections m_t , the falsely identified objects fp_t, to assess the accuracy aspect of the object detection algorithm.

$$MODA = 1 - \frac{\sum_{t=1}^{N_{frames}} (m_t + fp_t)}{\sum_{t=1}^{N_{frames}} N_G^{(t)}} \tag{9}$$

The **Multiple Object Detection Precision - MODP** gives the average overlapping ratio (match ratio) between the bounding-boxes of ground-truth and detected objects, as defined in Equation 10. It does not take into consideration the missed or falsely identified objects.

$$MODP = \frac{\sum_{t=1}^{N_{frames}} \sum_{i=1}^{N_{mapped}} \frac{\left|G_i^{(t)} \cap D_i^{(t)}\right|}{\left|G_i^{(t)} \cup D_i^{(t)}\right|}}{\sum_{t=1}^{N_{frames}} N_{mapped}^{(t)}} \tag{10}$$

The following steps were adopted to convert motion vectors to objects, so they can be analyzed by CLEAR MOD metrics: a given block was considered as foreground if its motion vector has a value different from the background, i.e., (0,0) in the case of static cameras. Two foreground motion vectors were grouped in the same blob (object) if they were 8-connected neighbors and have a Vector Matching Ratio (R), Equation 3, greater than 40%.

After the segmentation, objects with size below a given threshold were ignored to reduce the huge number of false detections. The minimum size threshold was tuned to each filter configuration to obtain the best possible MOD metrics.

Four configurations of motion vector filters were comparatively tested:

- **none** - No motion vector filter is applied before segmentation. Objects with size of 6 blocks or smaller are ignored.
- **VMF** - The Vector Median Filter is applied before segmentation. Objects with size of 6 blocks or smaller are ignored.
- **STF** - The Spatiotemporal Motion Vector Filter is applied before segmentation. Objects with size of 1 block are ignored.
- **TF** - The Temporal Motion Vector Filter is applied before segmentation. Objects with size of 1 block are ignored.

The *usf_date* software from [16] was used to calculate the metrics. The public available ground-truths of CLEAR and PETS datasets were converted to the VIPER XML format accepted by the *usf_date* software. Table I displays information about the video sequences used by the experimental tests.

TABLE I
List of public available video datasets used in performance evaluation.

Sequence	Resol.	Frames	Objects	Compression
CAVIAR Fight OneManDown	384x288	803	2036	MPEG-4, with EPZS motion estimator, GOP = 12, I and P-frames
PETS2001 Dataset1 Testing/Camera1	768x576	2500	7849	MPEG-4, with EPZS motion estimator, GOP = 128, I and P-frames
PETS2001 Dataset1 Testing/Camera2	768x576	2500	7849	MPEG-4, with EPZS motion estimator, GOP = 128, I and P-frames
CLEAR2006 PVTRA101a01	720x480	6567	1875	MPEG-4, with EPZS motion estimator, GOP = 128, I and P-frames

The performance analysis for the video sequence are summarized by Figures 4, 5, 6, 7. Qualitative analysis are provided by Figures 9, 10, 11, 12, were the motion vectors values are in white color, and boundaries of blobs in yellow. The motion vectors with value (0,0), of blocks belonging to background, are not displayed. The motion vector value is drawn over each block with the layout $\boxed{\begin{smallmatrix} dx \\ dy \end{smallmatrix}}$.

The TF allows a superior object detection accuracy, as summarized in Figure 4. In the sequences with more noisy motion vectors, Fight_OneManDown (Figure 9) and PVTRA101a01 (Figure 12), the superior filtering capability of the TF becomes more evident. The TF significantly outperforms the VMF object detection accuracy for sequence Fight_OneManDown (MODA 45% against -22%) and sequence PVTRA101a01 (MODA 31% against -14%). The TF and STF have very similar accuracy, except in sequence Fight_OneManDown, when the temporal analysis performed by the STF produces a high number of false detections.

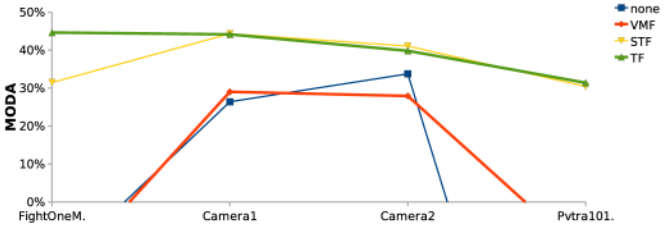


Fig. 4. Object detection accuracy - MODA.

In the less noisy sequences Camera1 (Figure 10) and Camera2 (Figure 11), the VMF has an object detection accuracy near to that exhibited by TF. In these sequences, even without any *mv* filter, the minimum size filter applied to segmented blobs presents a detection accuracy as good as the VMF.

The missed detections account for the great majority of detection errors, and are 10 times more frequent than false detections, as presented in the detailed metrics of TF in Figures 9, 10, 11, and 12. Without these missed detections, the accuracy metric would reach a value as high as 95% in evaluated sequences. These missed detections occur mainly because objects near each other with similar velocity are grouped as one big object, instead of several smaller ones, as can be visualized in the person group with Figure 10 frame 950, and within Figure 12 frame 1900.

This object grouping also leads to significant reduction in bounding box overlap precision. The develop of techniques to further decompose these near objects would bring great improvements in metrics results.

Objects without movement, that do not have motion vectors, also accounts for a significant part of missed detections, as the woman standing near the window within Figure 9 frame 150, and the stopped white van within Figure 11 frame 2350.

The TF and STF presented an almost constant bounding box overlap precision (MODP) in the video sequences, with values between 41% and 47%. In sequences Camera1, Camera2, and PVTRA101a01 the TF, STF, and VMF presented similar bounding box precision as presented within Figure 5. The MODP metric is calculated only in the detect objects, so the smaller objects missed by VMF and detect by TF tend to favor the VMF average bounding box precision. This explains the small MODP advantage of the VMF in Camera1 and Camera 2. Nevertheless, in the highly noisy sequence Fight_OneManDown, the TF outperforms the VMF (MODP 47% against 19%).

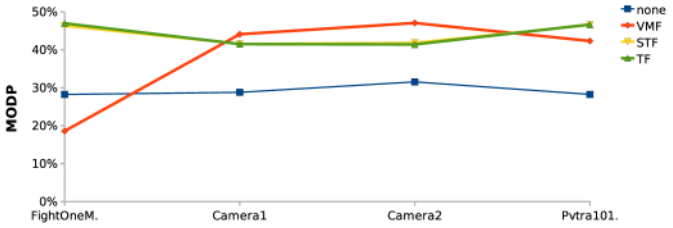


Fig. 5. Object detection bounding box precision - MODP.

The computational effort for executing the four tested configurations, including the motion vector and object segmentation, is presented within Figures 6 and 7. As video sequences have different resolutions, the frames per second (fps) where scaled proportionally to the number of pixels presented in a 720 x 480 image within Figure 6. The tests were executed in a Compaq Presario M2000 Notebook produced in 2005, with AMD Sempron-2800 1.60 GHz processor, and 650 MB of DDR DIMM memory.

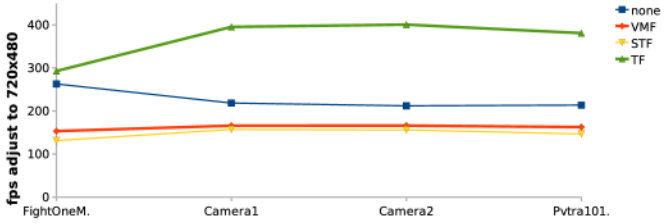


Fig. 6. Measured frames per second adjusted to 720 x 480.

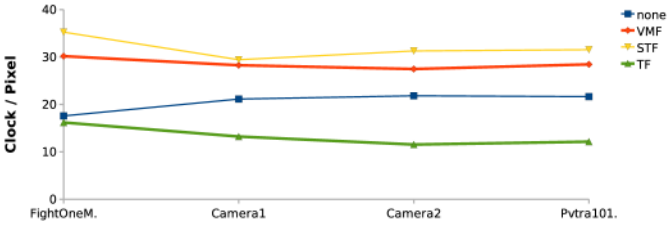


Fig. 7. Measured clocks consumed per video pixel.

The $clock/pixel$ is equal to the number of processor clock consumed by an algorithm divided by the number of pixel of a video sequence($width \cdot height \cdot frames$). The $clock/pixel$ is a better measure than fps, because it is less dependent on image size, and computer processor clock.

The object detection with TF is twice faster than VMF or STF in the video sequences presented in Table I. Table II displays the theoretical number of computer operations necessary to compute the filters, based on their equations. The VMF requires five times more operations than the TF. The VMF with 3 x 3 window requires the computation of 36 vector euclidean distance calculations, and consequently 36 square roots. The TF requires the computation of only 7 square roots.

TABLE II
THEORETICAL INSTRUCTIONS TO FILTER ONE MOTION VECTOR BLOCK.

	Sum/Sub	Mult	Div	Cmp	Sqrt
VMF	108	72	0	36	36
TF	18	12	2	1	7

The use of TF produces even faster results than not using any motion vector filter. The higher number of blobs created in segmentation step explain the reduced fps for the configuration without any motion vector filter (configuration *none*) compared to the TF, presented in Figure 6.

The experiments demonstrate that the Temporal Motion Vector Filter (TF) outperforms the Vector Median Filter (VMF) with better object detection accuracy (MODA), lower computational complexity, and better bounding box overlap precision (MODP) in noisy scenes. The TF presents half the computational cost of the Spatiotemporal Motion Vector Filter (STF), while preserving the same accuracy and precision. The results obtained with the TF make it suitable as a first building block of any system that aims to detect and track objects from compressed video, by using its motion vectors.

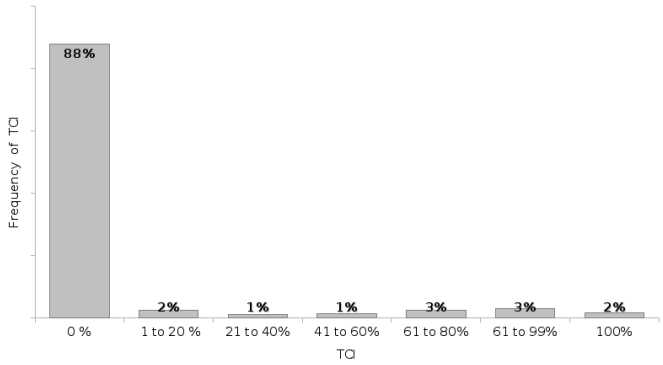
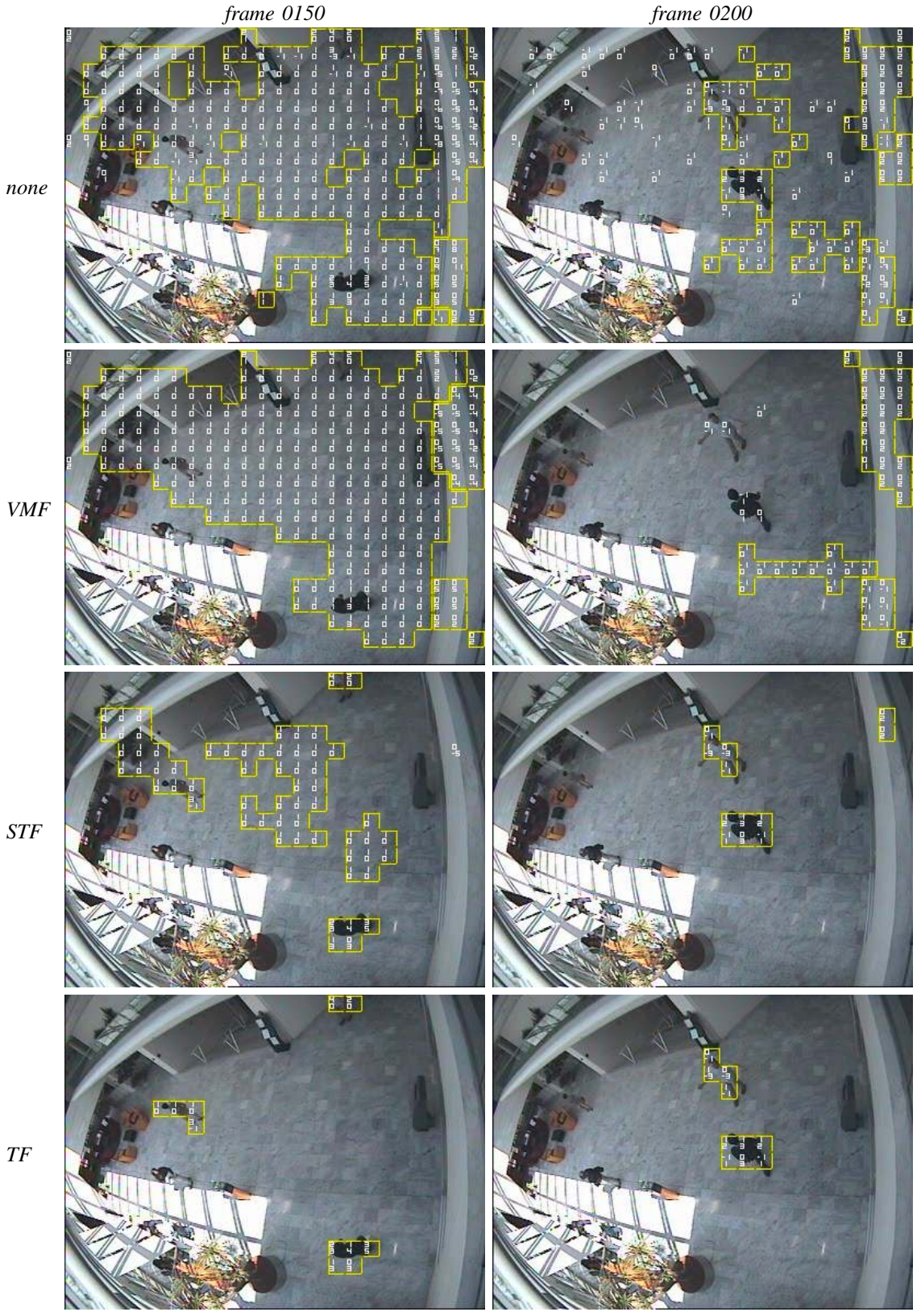


Fig. 8. Histogram of TCI results for non static motion vectors in sequence PETS2001 Camera1. The TCI of static motion vectors (0,0) are not represent in this histogram, as they only produce two TCI values: 0% or 100%. Others videos sequences have a similar histogram.

The histogram in Figure 8 indicates that 88% of TCI results for non static motion vectors are equal to zero in sequence PETS2001 Camera1. Others videos sequences have a similar histogram. The TCI naturally segments most of motion vectors to zero, and the threshold value τ of TF Equation 5 allows only a fine-tuning of the filter. Setting this threshold does not represent a critical part of the TF Filter, since values between $\tau = 20\%$ and $\tau = 60\%$ produce similar filtering results.



Filter	detec.	missed	false detec.	MODP	**MODA**	fps	**clock/pixel**
none	716	1320	1085	28%	-18%	822	18
VMF	285	1751	740	19%	-22%	479	30
STF	997	1039	356	46%	31%	410	35
TF	974	1062	65	47%	45%	915	16

Fig. 9. Object detection in sequence CAVIAR FightOneManDown, by using different motion vector filters.

Filter	detec.	missed	false detec.	MODP	**MODA**	fps	**clock/pixel**
none	3342	4507	1269	29%	26%	171	21
VMF	2302	5547	23	44%	29%	130	28
STF	3777	4072	297	42%	44%	123	29
TF	3779	4070	313	42%	44%	309	13

Fig. 10. Object detection in sequence PETS2001 Camera1, by using different motion vector filters.

Filter	detec.	missed	false detec.	MODP	**MODA**	fps	**clock/pixel**
none	3367	4428	735	32%	34%	166	22
VMF	2279	5516	101	47%	28%	130	27
STF	3651	4144	450	42%	41%	122	31
TF	3652	4143	547	41%	40%	313	12

Fig. 11. Object detection in sequence PETS2001 Camera2, by using different motion vector filters.

Filter	detec.	missed	false detec.	MODP	**MODA**	fps	**clock/pixel**
none	586	1289	2798	28%	-118%	214	22
VMF	367	1508	636	42%	-14%	163	28
STF	682	1193	111	47%	30%	147	32
TF	677	1198	88	47%	31%	381	12

Fig. 12. Object detection in sequence Clear2006 PVTRA101a01, by using different motion vector filters.

V. Conclusion

A novel Temporal Motion Vector Filter (TF) was proposed and evaluated in this paper. The experiments demonstrate that the TF outperforms the Vector Median Filter (VMF) with better object detection accuracy (MODA), lower computational complexity and better bounding box overlap precision (MODP) in noisy scenes. The TF exhibits half the computational cost of the Spatiotemporal Motion Vector Filter (STF), while preserving the same accuracy and precision. The results obtained from the TF make it suitable as a first step towards implementing systems for detecting and tracking objects from compressed video using its motion vectors.

Future works on the Temporal Motion Vector Filter could be:

- Evaluation of the TF as part of a complete object tracking system with video sequences in MPEG-4 and H.264 formats.
- Integration of the TF to global motion estimation (GME) techniques. The TF could be used to select the reliable motion vectors in the least square estimator of [10]. By discarding the noisy motion vector in earlier iterations, the technique [10] may converge faster, and with better results.

References

[1] I. Haritaoglu, D. Harwood, and L. S. Davis, "W4: real-time surveillance of people and their activities," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 809–830, Aug. 2000.

[2] F. Bartolini, V. Cappellini, and C. Giani, "Motion estimation and tracking for urban traffic monitoring," in *Image Processing, 1996. Proceedings., International Conference on*, vol. 3. Lausanne: IEEE, Sep. 1996, pp. 787–790.

[3] V. Mezaris, I. Kompatsiaris, N. V. Boulgouris, and M. G. Strintzis, "Real-time compressed-domain spatiotemporal segmentation and ontologies for video indexing and retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 5, pp. 606–621, May 2004.

[4] F. Manerba, J. Benois-Pineau, R. Leonardi, and B. Mansencal, "Multiple moving object detection for fast video content description in compressed domain," *EURASIP J. Adv. Signal Process*, vol. 2008, p. 5, 2008.

[5] L. Favalli, A. Mecocci, and F. Moschetti, "Object tracking for retrieval applications in mpeg-2," *IEEE : Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 3, pp. 427–432, 2000.

[6] R. De Sutter, K. DeWolf, S. Lerouge, and R. Van de Walle, "Lightweight object tracking in compressed video streams demonstrated in region-of-interest coding," *EURASIP J. Appl. Signal Process.*, vol. 2007, no. 1, pp. 59–59, 2007.

[7] A. Ahmad, D.-Y. Chen, and S.-Y. Lee, "Robust object detection using cascade filter in mpeg videos," in *Proceedings of the IEEE Fifth International Symposium on Multimedia Software Engineering*. USA: IEEE, 2003, pp. 196–203.

[8] E. T. Kim and H.-M. Kim, "Efficient linear three-dimensional camera motion estimation method with application to video coding," *Journal of Optical Engineering*, vol. 37, pp. 1065–1077, Mar. 1998.

[9] R. Wang, H.-J. Zhang, and Y.-Q. Zhang, "A confidence measure based moving object extraction system built for compressed domain," in *Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium on*. Geneva , Switzerland: IEEE, 2000, pp. 21–24.

[10] T. Yu and Y. Zhang, "Retrieval of video clips using global motion information," *Electronics Letters*, vol. 37, no. 14, pp. 893–895, Jul. 2001.

[11] C. Käs and H. Nicolas, "An approach to trajectory estimation of moving objects in the h.264 compressed domain," in *PSIVT '09: Proceedings of the 3rd Pacific Rim Symposium on Advances in Image and Video Technology*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 318–329.

[12] M. Durik and J. Benois-Pineau, "Robust motion characterisation for video indexing based on mpeg2 optical flow'," in *Proceedings of Second International Workshop on Content-Based Multimedia Indexing*, Brescia, Italy, 2001, pp. 57–64.

[13] G. R. Bradski and J. W. Davis, "Motion segmentation and pose recognition with motion history gradients," *Mach. Vision Appl.*, vol. 13, no. 3, pp. 174–184, 2002.

[14] R. V. Babu, K. R. Ramakrishnan, and S. H. Srinivasan, "Video object segmentation: a compressed domain approach," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, pp. 462–474, 2004.

[15] Ronaldo C. Moura and Elder M. Hemerly, "A spatiotemporal motion-vector filter for object tracking on compressed video," in *Advanced Video and Signal Based Surveillance, IEEE Conference on*. Boston, USA: IEEE Computer Society, 2010, pp. 427–434.

[16] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, R. Bowers, M. Boonstra, V. Korzhova, and J. Zhang, "Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 319–336, Feb. 2009.

[17] S. D. Bruyne, C. Poppe, S. Verstockt, P. Lambert, and R. V. de Walle, "Estimating motion reliability to improve moving object detection in the h.264/avc domain," in *IEEE International Conference on Multimedia and Expo 2009. ICME 2009.* New York, USA: IEEE, 2009, pp. 330–333.

[18] H. Sabirin and M. Kim, "Moving object detection and tracking using a spatio-temporal graph in h.264/avc bitstreams for video surveillance," *IEEE Transactions on Multimedia*, vol. 14, no. 3, pp. 657–668, 2012.

Ronaldo Carvalho Moura received the degree of Computer Engineer, in 2004, and the degree of Master of Science in Electronic Engineering and Computer Science, in 2010, both from the Instituto Tecnológico de Aeronáutica, Brazil.

Founder of startup company Monity, since 2005 he works as Researcher and Team Leader for the development of new technologies and products in video processing, embedded systems, and information technology.

Elder Moreira Hemerly (S'86–M'89) received the Ph.D. degree in Electrical Engineering from Imperial College, London, U.K., in 1989.

Currently, he is Professor of Control Systems in the Electronics Division, Technological Institute of Aeronautics, Brazil. His current research interests include system identification, adaptive control and signal processing.

Adilson Marques da Cunha received the Bachelor degree in Air Force Course for Pilots from Brazilian Air Force Academy, in 1970, the Bachelor in Business Administration from Centro de Ensino Unificado de Brasília, in 1979, the degree of Master of Science in Information Systems from Air Force Institute Of Technology- USA, in 1984, and Ph. D. degree in Information Systems from George Washington University-USA, in 1987.

He is Professor in Computer Science Division, Technological Institute of Aeronautics, Brazil. He current research and teaching interests include software engineering; real-time embedded systems; software quality, dependability and safety; CASE tools; CNS-ATM; database system; and artificial intelligence.