Violence Detection in Video Using Spatio-Temporal Features

Fillipe D. M. de Souza*, Guillermo C. Chávez†, Eduardo A. do Valle Jr.‡ and Arnaldo de A. Araújo*
*Department of Computer Science
Federal University of Minas Gerais, Belo Horizonte - MG, Brazil
Email: {fdms,arnaldo}@dcc.ufmg.br
†Federal University of Ouro Preto, Ouro Preto - MG, Brazil
Email: gcamarac@gmail.com
‡University of Campinas, Campinas - SP, Brazil
Email: mail@eduardovalle.com

***Abstract*—In this paper we presented a violence detector built on the concept of visual codebooks using linear support vector machines. It differs from the existing works of violence detection in what concern the data representation, as none has considered local spatio-temporal features with bags of visual words. An evaluation of the importance of local spatio-temporal features for characterizing the multimedia content is conducted through the cross-validation method. The results obtained confirm that motion patterns are crucial to distinguish violence from regular activities in comparison with visual descriptors that rely solely on the space domain.**

***Keywords*-violence detection; spatio-temporal features; bags of visual words; SVM;**

I. Introduction

A topic of increasing interest in video processing is the characterization of multimedia content with regard to the presence of certain human actions. The ability of detecting different types of human actions has already been stated as a prominent task for a wide range of computer vision applications, namely, content-based video indexing, filtering of unwanted content, rating of movies, aid to human operators in real-time surveillance systems, to name just a few. Especially, detection of violent scenes receives considerable attention in surveillance systems, and filtering or rating of unwanted content. The former is justified by the need of providing people with safer public spaces, and the latter is aimed at situations where violence is considered inappropriate for the audience (*e.g.*, children).

A few difficulties arise in order to automatically characterize violence in multimedia content. To start with, its subjective nature imposes some barriers in defining what should be pointed as violence. Also some human behaviours, very similar to aggressive actions, might be misclassified. Therefore, the question of solving those ambiguities is raised to make the application feasible for efficient and robust real system. Given that the existing approaches of violence use an arbitrary notion, we also simplify the concept here by labeling as violent only those scenes containing fights (aggressive human actions), regardless of context and the number of people involved.

In this paper, we put forward a violence detector based upon local spatio-temporal features, using the "Bag of Visual Words" (BoVW) [1] representation and supervised learning with Support Vector Machine (SVM) [2]. Video elements are then classified as violent or non-violent.

The remainder of this paper is set as follows. In Section II we are concerned with the chosen type of data representation. In Section III, the proposed detector is approached in details. Lastly, the experimental results and their analysis, as well as the conclusion and future works are described in Sections IV and V, respectively.

A. Related Work

Most of the works in the literature use audio features as an additional resource to represent the video elements, combining it with visual features that provide motion information. None made use of local features such as the local spatio-temporal descriptor [3]. In addition, to the best of our knowledge, none has ever considered "visual codebooks" to bridge the gab between the underlying and meaningful coarse feature patterns and the high level semantic of interest.

In [4], an in-depth hierarchical approach was proposed for detecting distinct violent events involving two people, namely: fist fighting, hitting with objects, kicking, among others. They compute information (acceleration measure vector and its jerk) regarding the motion trajectory of image structures. However, this method presents some limitations, *e.g.* it fails when the fighters fall down, or when it involves more than two people.

Siebel and Maybank [5] developed a surveillance system for aiding human operators in monitoring undesirable events in a metro station. The system named as ADVISOR tracks and analyses people and crowd behaviours from multiple cameras. Once these data are processed the system flags a notification to the operator about a suspicious situation.

In [6], the authors argued for audio features as a self-sufficient resource for detecting violence in video. They showed the potential audio signals that should appear in violence scenes, such as screams, gunshots, explosions.

However, some sounds unrelated to violence (*e.g.*, fireworks) may lead to misclassification of scenes. Additionally, their application is focused on rating movies with respect to violent content, and the dataset is not specified.

In [7], a violence detector consisting of 4 modules is proposed. The first module separates a video into scenes. The second is assigned the task of retrieving the skin and blood colored regions from the scene frames, which are further filtered for obtaining the regions of interest (which might correspond to violent content). Then, motion intensities of these candidate regions are computed, and those with high values are casted as pertaining to the violence class.

Finally, other works aggregate audio and visual features [8]. For instance, in [9] this combination is done using a multiview scheme, with co-training strategy for consulting both evidences, being concerned with movie rating.

II. Data Representation

Which type of image descriptor to use and how to handle the provided set of features are important decision makings in the design of a computer vision application. Until recently, there has not been a comprehensive method capable of dealing with all possible contexts yet, and probably there will not be. Therefore, a few considerations to raise are: i) *which particular information give reliable indications that a specific event (or object) has occurred in a context of interest?*, ii) *how could they be systematically obtained?*, iii) *how could they be useful to find equivalents in unknown-content data?*, iv) *which type of representation should they be encoded in?*, and v) *which techniques are available to manipulate those coded information?*. Many other questions are possible and in this section the reader is driven to the particular set of choices to deal with the detection of violence scenes.

A. On Visual Features

Violence is typically qualified by aggressive human behaviors, namely, fast movements of limbs, face punching, kicking, and other similar actions. In this regard, it is natural to think of descriptions of local motion patterns as peculiar atifacts to the application, in the sense that they correspond to movements of different parts of the object that is responsible to perform the action. Those descriptions are commonly known as local image features (or interest points), which are expected to furnish abstract and compact representation to visual data.

In the literature, many interest point detectors have been proposed including detectors of spatio-temporal features [10] [3], scale-invariant features [11], and features invariant to scale and affine transformations [12]. In particular, this work was concerned with confirming that interest point detectors of the spatio-temporal domain are more suitable and superior in providing distinctive information to describe actions characterizing violence demeanours than those only working in the spatial domain. Then, a performance comparison of the violence detector using SIFT [11] and STIP [3] was carried out.

SIFT extracts distinctive local features by computing oriented-gradient histograms. This process is accomplished in four principal computation steps, namely, detection of maximas in scale-space, selection of interest points, assignment of orientations to the interest points, and description of the interest points by measuring local gradients in their neighborhoods. So as this descriptor is very sensitive to borders, many noisy features are detected in images with cluttered backgrounds. In this case, focusing on regions belonging uniquely to the object of interest in the scene is an advantage of considering temporal information on detection of interest points.

Laptev [3] designed a differential operator for simultaneously considering extremas over the spatial and temporal scales that refer to particular patterns of events in specific locations. This method is built on the Harris [13] and Förstner [14] interest point operators, but extended for the temporal domain. Essentially, as a corner moves across an image sequence, at the change of its direction an interest point is identified. Other typical situations are established when image structures are either split or unified. For being one of the major elements in this work, a few details and mathematical considerations on the detector design are presented next.

1) Spatio-temporal interest point detector: Many interest events in videos are characterized by motion variations of image structures over time. In order to retain those important information, the concept of spatial interest points is extended to the spatio-temporal domain. This way, the local regions around the interest points are described with respect to derivatives in both directions (space and time).

At first, the selection of interest point in the spatial domain is described. The linear scale-space representation of an image can be mathematically defined as $L^{sp} : R^2 \times R_+ \mapsto R$, which is the convolution of f^{sp} with g^{sp}, where $f^{sp} : R^2 \mapsto R$ represents a simple model of an image and g^{sp} is the Gaussian kernel of variance σ_l^2. Then,

$$L^{sp}(x, y; \sigma_l^2) = g^{sp}(x, y; \sigma_l^2) * f^{sp}(x, y), \tag{1}$$

and

$$g^{sp}(x, y; \sigma_l^2) = \frac{1}{2\pi\sigma_l^2} \exp(-(x^2 + y^2)/2\sigma_l^2). \tag{2}$$

Localizing interest points means to find strong variations of image intensities along the two directions of the image. To determine those local regions, the second moment matrix is integrated over a Gaussian window having variance σ_i^2, for different scales of observation σ_l^2, which is written as the equation:

$$\begin{aligned}\mu^{sp}(.;\sigma_l^2,\sigma_i^2) &= g^{sp}(.;\sigma_i^2) * ((\nabla L(.;\sigma_l^2))(\nabla L(.;\sigma_l^2))^T) \\ &= g^{sp}(.;\sigma_i^2) * \begin{pmatrix} (L_x^{sp})^2 & L_x^{sp}L_y^{sp} \\ L_x^{sp}L_y^{sp} & (L_y^{sp})^2. \end{pmatrix} \quad (3)\end{aligned}$$

The descriptors of variations along the dimensions of f^{sp} are the eigenvalues of Eq. 3: λ_1 and λ_2, with $\lambda_1 \leq \lambda_2$. Higher values of those eigenvalues is a sign of interest point and generally leads to positive local maxima of the Harris corner function, provided that the ratio $\alpha = \lambda_2/\lambda_1$ is high and satisfies the constraint $k \leq \alpha/(1+\alpha)^2$:

$$\begin{aligned}H^{sp} &= \det(\mu^{sp}) - k.trace^2(\mu^{sp}) \\ &= \lambda_1\lambda_2 - k(\lambda_1+\lambda_2)^2 \quad (4)\end{aligned}$$

.

Analogously, the procedure to detect interest points in the scape-time domain is derived by rewriting the equations to consider the temporal dimension. Thus, having an image sequence modeled as $f : R^2 \times R \mapsto R$, its linear representation becomes $L : R^2 \times R \times R_+^2 \mapsto R$, but over two independent variances σ_l^2 (spatial) and τ_l^2 (temporal) using an anisotropic Gaussian kernel $g(.;\sigma_l^2,\tau_l^2)$. Therefore, the complete set of equations for detecting interest points described in [3] is the following.

$$L(.;\sigma_l^2) = g(.;\sigma_l^2,\tau_l^2) * f(.), \quad (5)$$

$$\begin{aligned}g(x,y,t;\sigma_l^2,\tau_l^2) &= \frac{1}{\sqrt{(2\pi)^3\sigma_l^4\tau_l^2}} \\ &\times \exp(-(x^2+y^2)/2\sigma_l^2 - t^2/\tau_l^2), \quad (6)\end{aligned}$$

$$\mu = g(.;\sigma_i^2) * \begin{pmatrix} L_x^2 & L_xL_y & L_xL_t \\ L_xL_y & L_y^2 & L_yL_t \\ L_xL_t & L_yL_t & L_t^2. \end{pmatrix} \quad (7)$$

$$\begin{aligned}H &= \det(\mu) - k.trace^2(\mu) \\ &= \lambda_1\lambda_2\lambda_3 - k(\lambda_1+\lambda_2+\lambda_3)^3, \quad (8)\end{aligned}$$

restricted to $H \geq 0$, with $\alpha = \lambda_2/\lambda_1$ and $\beta = \lambda_3/\lambda_1$, and subject to $k \leq \alpha\beta/(1+\alpha+\beta)^3$

B. *Codebooks of Visual Features*

The method proposed in this paper is built on "bag of visual features" (or bag of visual words), a concept borrowed from the field of textual information retrieval, which has been successfully applied to a large range of image processing applications [15][16]. In this approach, the feature domain is sliced into discriminative subspaces. Each subspace reflects an observed, enlightening pattern of the visual content, *e.g.* parts of animate and inanimate entities. These patterns are believed to provide instructive clues to portray a multimedia content under analysis. That is, the presence of a few of such patterns (what are referred as visual words) should indicate with certain confidence the occurrence of specific events or objects in a scene or image.

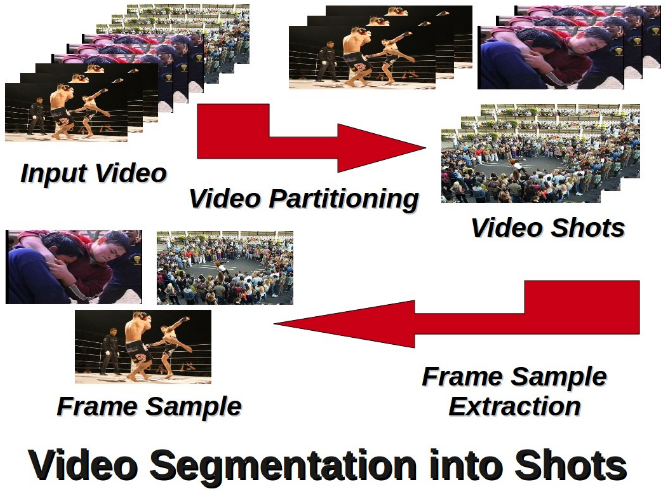

Figure 1. First step: videos are split into shots. Video shots are served as input to STIP, whereas the frame sample is input to SIFT.

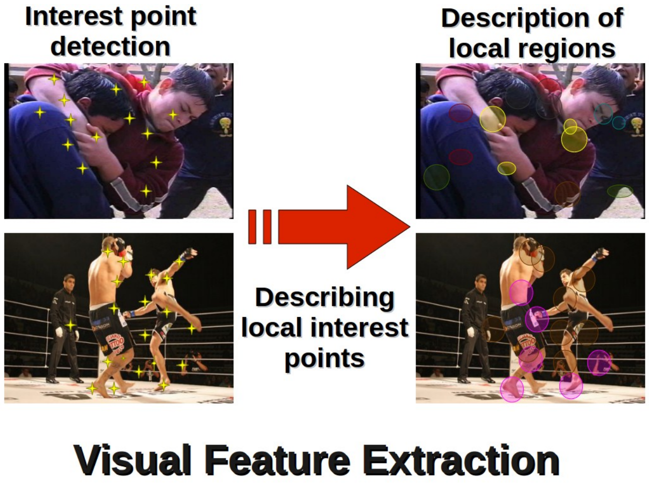
Interest point detection
Description of local regions
Describing local interest points
Visual Feature Extraction

Figure 2. Second step: the visual feature vectors are computed for each video shot.

In other words, "visual words" are distinctive feature vectors, $i.e.$ features considered informative enough to account for the underlying patterns of a set of visual data. They compose what is called the visual codebook. Depending on the intrinsic characteristics of the interest point detector and feature descriptor used to compute the feature vectors (building a feature space), the distribution of patterns in the formed feature space suffers considerable variation. Some approaches combine different organization of feature space to enhance the ability of interpretation of visual complex

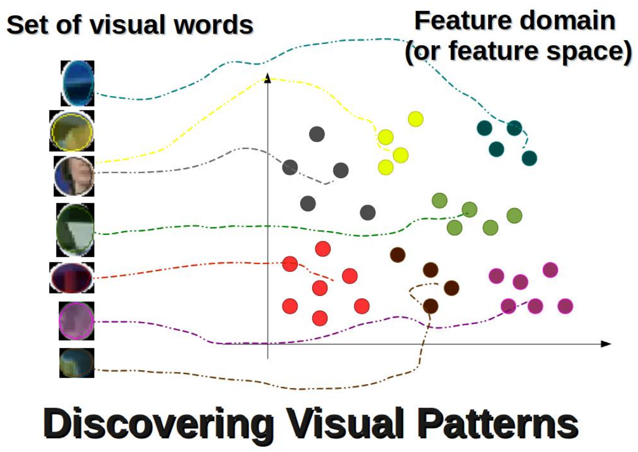

Figure 3. Third step: a clustering algorithm is applied to the feature space in order to discover the latent patterns.

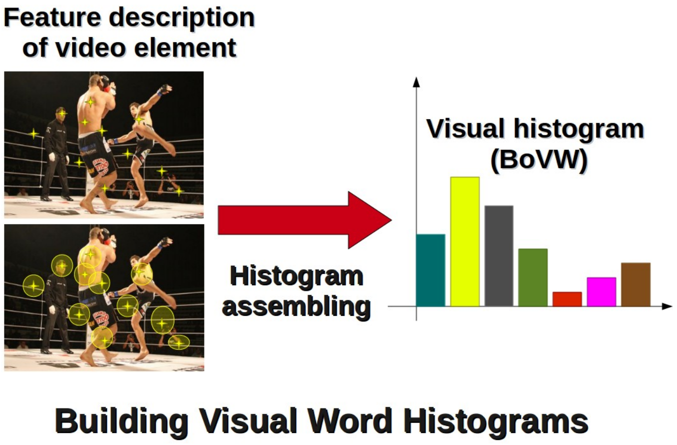

Figure 4. Fourth step: visual histograms are assembled to represent the video elements according to the occurrence of visual words in their feature space.

scenarios.

The idea is that those latent patterns encountered should be able to help narrow the gap between the seemingly meaningless coarse data and the semantic of interest. In what concerns how to represent the video elements, let the visual words be the probable attributes that any video must hold. Thus, the count of the different attribute occurrences in a specific video element, *i.e.* a visual word histogram, gives its description. A definite disposal of word frequency bestows a guess about which information it is contains with a degree of confidence.

Generally, discovering proper visual words to represent the visual content follows a standard set of steps. Once from a dataset sample the feature vectors are computed, a clustering algorithm (usually the k-means) is delegated to organize the feature vectors in groups (clusters). The assigment of feature vectors to different clusters is dictated by a similarity measure function, which is commonly the Euclidean distance function. Each formed cluster stands for a unique pattern from the multimedia dataset, and the cluster center computed by the mean of the group elements represents a visual word. Therefore, to find the correspondent visual word of new feature vector only depends on verifying to which cluster center it is most similar.

Formally, visual codebooks are generated as follows. Given a set of videos $SV = \{V_1, V_2, V_3, V_4, ..., V_n\}$, for $i \leq n$, low-level features of each video are computed by using a feature descriptor ($e.g.$ SIFT, STIP), as depicted in Figure 2. Then, each video V_i consists of a set of feature vectors, that is, $V_i = \{F_1, F_2, F_3, F_4, F_5, F_6, ..., F_k\}$, where the feature vector dimension d varies according to the interest point descriptor (feature descriptor) and the number of features extracted k depends on the amount of information available in the video that the detector and descriptor are able to snare. Those features are clustered according to their level of similarity. Therefore, g groups are formed (Figure 3), having each a d-dimensional representative c_i, $1 \leq i \leq g$. As a result, a visual codebook consists of all feature vectors c_i. Such codebook is further used as a reference to compute the visual word histograms (Figure 4) of new data. This way, a new domain is created, where each visual word corresponds to a dimension i, having histograms of visual words as elements of this space.

For example, if there is a codebook constructed by ten visual words, a visual word histogram has ten dimensions. Suppose also that some visual words (the first six words) correspond to patterns related to essential parts of a human being to recognize a person and others (the last four words) are unrelated to the description of a person. Then, if there is a visual word histogram B_i^g representing an image I_i consisting of twenty feature vectors, where sixteen are most similar to the person-related visual words and four to the unrelated-person ones, which in turn are distributed as $B_i = \{B_i^1 = 7, B_i^2 = 2, B_i^3 = 1, B_i^4 = 3, B_i^5 = 1, B_i^6 = 2, B_i^7 = 2, B_i^8 = 1, B_i^9 = 0, B_i^{10} = 1\}$, it is very likely that there is a person in this image, but the classifier is the entity tasked to give the “right” answer.

III. The Violence Detector Design

In this section, the procedures involving the violence detector are set out. The preliminary step is to segment[1] the set of videos $SV = \{V_1, V_2, V_3, V_4, ..., V_n\}$ into shots (see Figure 1). This way, each video V_i is denoted by $V_i = \{S_{i,1}, S_{i,2}, S_{i,3}, S_{i,4}, ..., S_{i,m}\}$, where m is the total number of shots that composes V_i. Next, the video shots are

[1] We have used an industry-standard software for segmentation, see at http://www.stoik.com/products/svc/

submitted to the feature extraction process (see Figure 2) using the spatio-temporal descriptor [3] (but for analysis of performance we also applied SIFT [11]).

It is important to clarify here that, as we rely on supervised learning with SVM, a mandatory step is to create a classification model. After that, the model can be updated or the classification process can be performed. Both stages depend that the features are extracted and available. Then, initially, there is neither codebook nor classification model (this latter used as input to SVM), therefore they must be created. Once they are arranged, that is, the visual codebook is built and there is one classification model at disposal, new data can be processed. A classification model is a set of information guiding the classifier to indicate to which data pattern the unknown data pertain.

In the present work, the "visual codebooks" were conceived by means of false clustering. This means that the visual words were randomly selected from the whole set of extracted visual features (using such approach allows us to save computational time and experiments have demonstrated plausible results for large amounts of data, which is the particular case). However, notice that the discovery of visual words could have been performed by using a clustering algorithm, which in theory should produce a more consistent organization of the feature space.

Thus, the system must first be provided with a sample of data to learn a classifier based on the visual word histograms of the sample. The accomplishment of this task requires a predefined set of instructions. First, the bags of visual words for each video shot $S_{i,j}$ are generated by quatifying the their respective feature space. At this point, each video V_i of the sample is represented by a set of bags $SB_i = \{B_{i,1}, B_{i,2}, B_{i,3}, ..., B_{i,m}\}$, where each bag refers to a different shot $S_{i,j}$, where m is total number of shots and $1 \leq j \leq m$. Secondly, the set of bags is annotated (each bag is labeled with a class identification number) and served as input to the linear-SVM algorithm (using libSVM [17]) in order to generate a prediction model. Experimentally, the linear kernel provided us with the best performance for all cases in comparison with the non-linear.

Once the visual codebook and prediction model are acquired, incoming data are processed, that is, segmented into shots, converted into bags of visual words, and passed to the stage in which the classifier acts. In this step, the classifier is fed with the prediction model and the not observed data in its visual-word form. In the end of the classification process, the analyzed data has a set of information associated regarding the possible classes (for example, how much of the data composition is likely connected to violent and non-violent content).

Shot classification is the final step of the detector, which is performed using support vector machines [2]. Its foundations are derived from the statistical learning theory and such methods are commonly used to either classification or regression. Technically, $n-1$ hyperplanes must be constructed and ideally have the largest possible margins among the n existing classes. The hyperplanes are responsible for establishing the subspace bounds of each class. Therefore, given a training sample of the dataset and having calculated their hyperplanes, the classifier is capable of assigning a new data to a specific class. In particular, we assume that the classes can be reasonably defined by linear hyperplanes, since especially the linear classifier sufficed to nicely categorize the testing dataset.

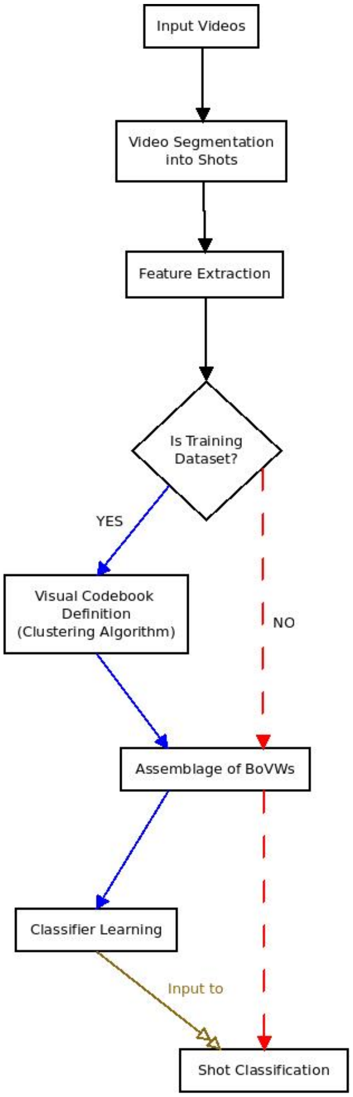

Figure 5. Working scheme of the violence detector.

A thumbnail sketch of the violence detector design is depicted in Figure 5.

IV. Experimental Results and Analysis

This section is committed to expose preliminary results regarding the detection of violence in shots. Although a fine-

HOSTED AT GORILLAFIGHTS.COM
HOSTED AT GORILLAFIGHTS.COM
HOSTED AT GORILLAFIGHTS.COM
HOSTED AT GORILLAFIGHTS.COM
tru TV
HOSTED AT COMEGETYOUSOME.COM
HOSTED AT GORILLAFIGHTS.COM
HOSTED AT GORILLAFIGHTS.COM
HOSTED AT GORILLAFIGHTS.COM

Figure 6. Content illustration of the violence video dataset.

Figure 7. Content illustration of the non-violence video dataset.

grained analysis could be interesting, a rougher categorization of the content under analysis was chosen, given the inherently anarchic nature of social networks. In addition, a clear-cut definition of what each "subclass" should contain, considering fine-grained notation, has not been established yet.

A. The Dataset

It is important to note that the literature lacks a shared violence dataset, which limits our ability to fairly compare our approach to the existing ones. Furthermore, most of the works in violence detection do not describe the dataset used in enough details to allow reproducing their results. That way, we have assembled a violence database (available under request). An illustration of the conceived database is depicted by Figures 6 and 7.

Both violence and non-violence samples are very diverse and representative, which can be found on social networks. A compilation of daily life situations in schools, ghettos, entering spaces of night clubs, matches from several sport variations (*e.g.*, soccer, hockey), traffic (and more), involving fights and professional fight scenes build the violence dataset. Scenarios are depicted by aggressive behaviors involving any number of people, in indoor and outdoor environments, with or without presence of moving objects (*e.g.*, cars) in background. The non-violence dataset is described by music clips, news broadcast videos, matches that confuse with those containing violence, pornography scenes, traffic, events with people hugging, running, among others. In total, there are 400 videos, 200 composing each category.

B. Setup

To validate the proposed violence detector, the experiments were carried out using 108 videos for each category. For extracting the set of features of the dataset, we made use of SIFT and STIP. Visual word histograms were computed to give a middle-level representation of the visual data. To construct the visual codebooks, k values ranging from 100 to 5000 were considered. The classification task performed so great to 100 as to 1000 and 5000, apart from calling for much less computational burden, leading us to naturally chose 100 as a good number of visual words to describe the dataset. To close the experimental protocol we conducted a 5-fold cross-validation, where the visual histograms were not normalized.

C. Results and Discussion

Tables I and II shows the classification performance of the method with SIFT and STIP, respectively. They reveal that although SIFT performs relatively well with 80.09% of true positive and 85.35% of true negative, there is still a considerable amount of error remaining for both sides of the coin, 14.65% false positives and 19.91% false negatives. Conversely, the STIP's performance is much superior than SIFT's, scoring 100% in detecting non-violent shots and having a minor error of 0.46% for assertion of the violent content.

These evidences make us to conclude that spatio-temporal features are decisive to better define what is in fact relevant to separate the different categories, obviously, provided that the difference among the classes strongly takes into account motion patterns. The results somehow claim how relevant is to work with the space-time domain for encountering unique characteristics of the behaviour of the interest structures in contrast to a visual descriptor that relies solely on the space domain [11].

V. Final Remarks

To sum up, we proposed and evaluated a method based on data representation using local spatio-temporal feature properties, which was abstracted with basis on the concept of

Table I
PERFORMANCE OF SHOT CLASSIFICATION USING ORIENTED-GRADIENT FEATURES WITH 100-WORD CODEBOOK.

SIFT		
(%)	**Violent**	**Non-violent**
Violent	80.09	19.91
Non-violent	14.65	85.35

Table II
PERFORMANCE OF SHOT CLASSIFICATION USING SPATIO-TEMPORAL FEATURES WITH 100-WORD CODEBOOK.

STIP		
(%)	**Violent**	**Non-violent**
Violent	99.54	0.46
Non-violent	0	100.0

visual words, applying support vector machines for learning and detection of violent content in videos. The proposed violence detector can be acceptably applied to offline processing of surveillance videos and movie rating.

As for the result analysis, it was attested that the spatio-temporal descriptor looks carefully into the dynamic aspects of the interest objects over a sequence of frames. Therefore, selecting significant patterns, which turns out to finely distinguish violence from regular scenes.

From now on, we intend to select videos having explicitly more ambiguous content in order to evaluate how the proposed detector deals with a more complex scenario. Changing the current problem to a multiclass viewpoint, with a clear-cut definition, is a point to be considered; by specializing the different types of violent behaviors and aggregating cases where violence is doubtful.

ACKNOWLEDGMENT

The authors are grateful to CNPq, CAPES, FAPESP and FAPEMIG, Brazilian research funding agencies, for the financial support to this work.

REFERENCES

[1] J. Yang, Y.-G. Jiang, A. G. Hauptmann, and C.-W. Ngo, "Evaluating bag-of-visual-words representations in scene classification," in *MIR '07: Proceedings of the international workshop on Workshop on multimedia information retrieval*. New York, NY, USA: ACM, 2007, pp. 197–206.

[2] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Min. Knowl. Discov.*, vol. 2, no. 2, pp. 121–167, 1998.

[3] I. Laptev, "On space-time interest points," *Int. J. Comput. Vision*, vol. 64, no. 2-3, pp. 107–123, 2005.

[4] A. Datta, M. Shah, and N. Da Vitoria Lobo, "Person-on-person violence detection in video data," in *ICPR '02: Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02) Volume 1*. Washington, DC, USA: IEEE Computer Society, 2002, p. 10433.

[5] N. T. Siebel and S. J. Maybank, "The advisor visual surveillance system," in *Proceedings of the ECCV 2004 workshop "Applications of Computer Vision" (ACV'04), Prague, Czech Republic*, M. Clabian, V. Smutny, and G. Stanke, Eds., May 2004, pp. 103–111.

[6] T. Giannakopoulos, D. I. Kosmopoulos, A. Aristidou, and S. Theodoridis, "Violence content classification using audio features," in *SETN*, 2006, pp. 502–507.

[7] C. Clarin, J. Dionisio, M. Echavez, and P. Naval, "Dove: Detection of movie violence using motion intensity analysis on skin and blood," in *PCSC '06: Proceedings of the 6th Philippine Computing Science Congress*. Computing Society of the Philippines, 2006, pp. 150–156.

[8] W. Zajdel, J. D. Krijnders, T. Andringa, and D. M. Gavrila, "Cassandra: Audio-video sensor fusion for aggression detection," in *IEEE Int. Conf. on Advanced Video and Signal based Surveillance (AVSS*, 2007.

[9] J. Lin and W. Wang, "Weakly-supervised violence detection in movies with audio and video based co-training," in *PCM '09: Proceedings of the 10th Pacific Rim Conference on Multimedia*. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 930–935.

[10] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *VS-PETS*, October 2005.

[11] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[12] K. Mikolajczyk and C. Schmid, "Scale and affine invariant interest point detectors," *International Journal of Computer Vision*, vol. 60, no. 1, pp. 63–86, 2004. [Online]. Available: http://lear.inrialpes.fr/pubs/2004/MS04

[13] C. Harris and M. Stephens, "A combined corner and edge detector," 1988, pp. 147–152.

[14] W. Forstner and E. Gulch, "A fast operator for detection and precise location of distinct points, corners and centres of circular features," pp. 281–305, 1987.

[15] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *In Workshop on Statistical Learning in Computer Vision, ECCV*, 2004, pp. 1–22.

[16] T. Deselaers, L. Pimenidis, and H. Ney, "Bag-of-visual-words models for adult image classification and filtering," in *ICPR*, 2008, pp. 1–4.

[17] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.