

Cairo University

Faculty of Engineering

Department of Computer Engineering

**Crashing Detection Module**

**Vif Descriptor**

Report to explain the approach we took to make a crashing detection Module

**Introduction**

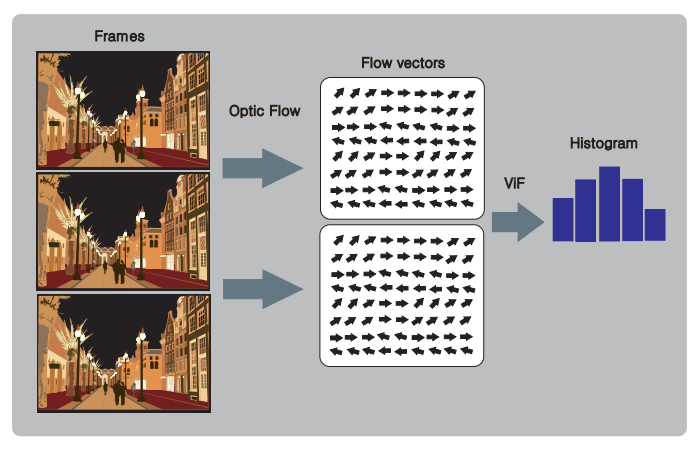
We need a way to describe the flow of the scene to detect anomaly in the flows that could be caused by crashing.

We here consider how flow vector magnitudes change over time for short frame sequences, are represented using the VIolent Flows (ViF) descriptor. ViF descriptors are then classified as either violent or non-violent using linear SVM.

**Why Violent Flows (ViF) descriptor ?**

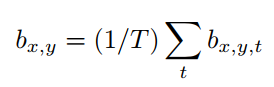
we are going to use the ViF descriptor because of the very low cost and

acceptable accuracy. The ViF descriptor regards the statistics of magnitude changes of flow vectors over time as we see here:



In order to get these vectors, [6] used the optical flow algorithm proposed by [11] named Iterative Reweighted Least Squares (IRLS), but in this context, we used the ViF descriptor with Horn-Schunck [8] as optical flow algorithm proposed by [1].

The ViF descriptor is presented in algorithm [1], here we get a binary, magnitude-change, significance map bt for each frame ft. Then we get a mean magnitude-change map, for each pixel, over all the frames with the equation:



Then the ViF descriptor is a vector of frequencies of quantized values bx;y.

Then the car crash detector is trained using a SVM classifier

Given a video sequence S of frames {f1, f2, . . .} we

consider two related but different tasks. The first is anomaly classification: The video S is assumed to be segmented temporally, containing T frames portraying either

anomaly or non-anomaly event behavior. The goal is to classify S accordingly. The second is anomaly detection: Here, we assume an input stream of frames and the goal is to detect the change from anomaly to non-anomaly behaviour, with

the shortest delay from the time (frame) that the change occurred. Moreover, as mentioned above, this goal must be achieved with processing performed faster than frame-rate.

Existing work [30] has shown that under certain circumstances, less than ten video frames are required for reliable action classification. We consider such sub-second delays acceptable for a detection system and so reduce the second

problem to the first by processing short frame sequences separately, classifying each one as either anomaly or non-anomaly; a detection is reported once an anomaly sub-sequence of frames is thus encountered.

**ViF representation Algorithm**

Given a sequence of frames, S, we produce the VIolence Flows (ViF) descriptor by first estimating the optical flow between pairs of consecutive frames. This provides

for each pixel px,y,t, where t is the frame index, a flow vector (ux,y,t, vx,y,t), matching it to a pixel in the next frame t + 1. Here, we consider only the magnitudes of these vectors:



Doing so is in some sense a throwback to some early action recognition techniques which also relied on flow-vector magnitudes for processing actions [21]. There are some important differences, however, between those earlier approaches and our own.

Unlike previous methods, we do not consider the magnitudes themselves, but rather how they change over time.

Our rationale is that although flow vectors encode meaningful temporal information, their magnitudes are arbitrary quantities: they depend on frame resolution, different motions in different spatio-temporal locations, etc. By comparing magnitudes we obtain meaningful measures of the significance of observed motion magnitudes in each frame compared to its predecessor.

for each pixel in each frame we obtain a binary indicator bx,y,t, reflecting the significance of the change of magnitude between frames:



Where θ is a threshold adaptively set in each frame to the average value of |mx,y,t − mx,y,t−1|. Doing so provides us with a binary, magnitude-change, significance map bt for each frame ft. We next compute a mean magnitude-change

map by simply averaging these binary values, for each pixel,

over all the frames ft ∈ S:



In its simplest form, the ViF descriptor is a vector of frequencies of quantized values bx,y.

The ViF descriptor is therefore produced by partitioning b into

M × N non-overlapping cells and collecting magnitude change frequencies in each cell separately. The distribution of magnitude changes in each such cell is represented by a fixed-size histogram. These histograms are then concatenated into a single descriptor vector.

**Implementation**

Here are the steps that we implemented in the program for Crashing Detection Module “Vif Descriptor”:-

Input: List of Frames : Sequence of frames

Output : Histogram(bx;y; n\_bins = 336)

1. Loop over frames
2. Resize the frames to (134,100)
3. get prev, current and next frames using subsample of 3
4. get optical flow for prev and current frame
5. get optical flow for current and next frame
6. calculate delta between the difference of their magnitudes
7. get the mean of the delta and this is theta the limit
8. if the pixal above the theta then increment 1 to the flow
9. go to step 1 untill all frames processed
10. get the mean of the flow
11. Partition the frame to blocks and loop on them
12. Calculate the histogram for every block Append to the feature vector
13. Return the feature vector

**Processing Time Takes**

As we don’t take the whole frame to process then the processing time differes but on averge it takes a [7 to 18] ms and we hope this speed don’t affect on us when we attached later to the optical flow.

**Comparison to Opencv Method**

We compared our implementation to opencv method and the accuracy was very close you can’t even determine which is better, but on the other side the speed of opencv method is much faster as results showed it’s faster than our implementation - even after optimizing it - with [60 – 80]% faster than us and that is because their function is implemented by c/c++ which is already compiled.

**References**

[6] T. Hassner, Y. Itcher, and O. Kliper-Gross, “Violent flows: Real-time

detection of violent crowd behavior,” in Computer Vision and Pattern

Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on, June 2012, pp. 1–6.

[11] C. Liu, “Beyond pixels: exploring new representations and applications

for motion analysis,” Ph.D. dissertation, Citeseer, 2009.

[8] B. K. Horn and B. G. Schunck, “Determining optical flow,” in 1981

Technical symposium east. International Society for Optics and

Photonics, 1981, pp. 319–331.

[1] V. M. Arceda, K. F. Fabián, and J. Gutiérrez, “Real time violence

detection in video,” IET Conference Proceedings, pp. 6 (7 .)–6 (7 .)(1),

Apr 2016. [Online]. Available: http://digital-library.theiet.org/content/

conferences/10.1049/ic.2016.0030

[30] K. Schindler and L. V. Gool. Action snippets: How many

frames does human action recognition require? In CVPR,

pages 1–8, 2008.

[21] J. Little and J. Boyd. Recognizing people by their gait: the

shape of motion. J. of Comp. Vision Research, 1(2):1–32,

1998.