Person-on-Person Violence Detection in Video Data

Ankur Datta Mubarak Shah Niels Da Vitoria Lobo
School of Electrical Engineering & Computer Science
University of Central Florida
Orlando, Florida, USA 32837
E-mail: {ankur,shah,niels}@cs.ucf.edu

Abstract

We address the problem of detecting human violence in video, such as fist fighting, kicking, hitting with objects, etc. To detect violence we rely on motion trajectory information and on orientation information of a person's limbs. We define an Acceleration Measure Vector (AMV) composed of direction and magnitude of motion and we define jerk to be the temporal derivative of AMV. We present results from several data sequences involving multiple types of violent activities.

1. Introduction and Background

The ability to detect person-on-person violence in video imagery would be very useful in both real-time camera systems and in analyzing movie data. Real-time camera systems could use violence detection for areas where peace and quiet must prevail such as quiet-zones, school playgrounds, airline cabins, airports, etc. Movie analyzing systems could use violence detection to rate movies: so children could be prevented from watching violent scenes.

To rate movies, a stochastic model was proposed in [3] to capture features such as violence, profanity, etc. This model worked with a shot transition detection model to capture the amount of motion present in a scene and hence would be unable to distinguish an action movie from a game such as basketball.

Researchers in [4] also tried to capture the degree of motion present in a scene by looking at the temporal activity and length of shots along with audio cues. Their system needs manual intervention for creation of audio samples to detect sounds associated with violence.

Both approaches above suffer from a fundamental problem: analyzing statistics of activity in complete scene rather than at object level, they only can ascertain that the movie genre is violent. They cannot tell us who is hitting who? Are there any objects being used? All of these questions can be answered by looking at violence from the object level.

Here, we present an approach to analyzing violence at the object level. We exploit the motion trajectory information of a person during violence to calculate *jerk* (reaction of the hit person). We also compute the *orientation* of arms and legs to draw certain inferences based on their motion patterns over time.

2. Overview of steps of Algorithm

The following is an overview of the algorithm. Also see Fig. 1. We skipped steps 1-2 to collate with the section numbers.

3. Background Subtraction.
4. Fit person model to silhouette and assign labels.
5. Determination of neck and shoulder.
6. Initialization of Head Tracking Box.
7. Track head using Color Sum of Squared Differences.
8. Compute *Acceleration Measure Vector* and *jerk*.
9. Compute *orientation* map for arms and legs.
10. Detect violence using objects, if object detected.
11. Detect non-violent activities, if present.
12. Refining the *orientation* data.
13. Repeat for the second person from step 4.
14. Repeat from step 3 for next frame.

3. Background Subtraction

We use an adaptive Background Subtraction method [2] that models each pixel as a mixture of Gaussian's(Fig. 2). It uses probabilistic measurements based on the mean and covariance of pixel color history and weight to determine the probability of observing the current pixel value. The background model is updated, so that it can keep up with the changes in the background. This adaptive background subtraction algorithm yields a stable background subtraction method that reliably deals with lighting changes, repetitive motions from clutter and long-term scene changes.

1051-4651/02 $17.00 © 2002 IEEE

4. Fitting a Person Model

Based on step 3 we acquire the silhouettes of moving bodies. After that we do a test based on the method described in [1] to determine if the moving body is indeed a person. We divide the bounding rectangle of the silhouette horizontally into three equal parts (H1, H2, H3). In order to extract the macroscopic features of a silhouette pattern from each part, we get the projection histogram of each part that is obtained by counting the number of black pixels in each column of the silhouette pattern(Fig. 3).

4.1. Features of Silhouette Pattern

The following features are extracted from the histogram and compared with a lookup table for 20 human models:

4.1.1 The Mean

The mean number of pixels in each box is an indicator of the human structure.

4.1.2 The Standard Deviation

We calculate the standard deviation of the number of black pixels along the x-axis. Black pixels represent the object in the scene.

4.1.3 The Aspect Ratio of the Silhouette-Bounding Rectangle

This feature can also be used as a distinguishing feature of humans from other objects.

All the above features are normalized by the area of the rectangle. This procedure prevents our system from trying to track objects like cars and animals.

5. Determination of Neck and Shoulder

We obtain the projection histograms from previous steps, i.e., (H1, H2, H3). We then create a y-projection of the first bounding box (H1)(Fig. 4). Our observations have shown that the neck can be found by finding the maximum of the derivative of this projection. Next we calculate the position of the shoulder based on the following formula:

$$S_y = N_r + (\frac{r}{2}) \quad (1)$$

where N_r = y-coordinate of the Neck, and r is head radius, $r = (\sqrt{A(head)/\pi})$, where $A(head)$ = area of head.

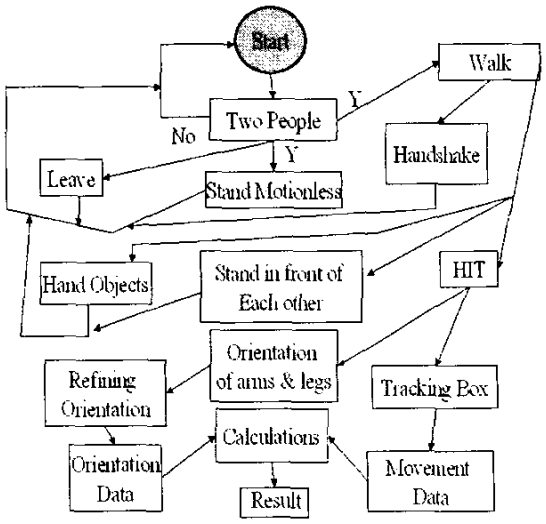

Figure 1. Finite State Machine and Flow diagram of the algorithm

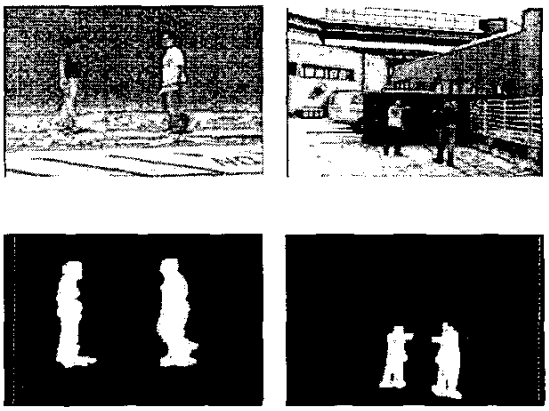

Figure 2. Dilated results from Background Subtraction for two scenes

6. Initialization of Head Tracking Box

Automatic initialization of the tracking box can be obtained after we have found the relative locations of the neck. We move upward from the neck, using region growing to get the head. After that we find the bounding rectangle of this region and that is our tracking box.

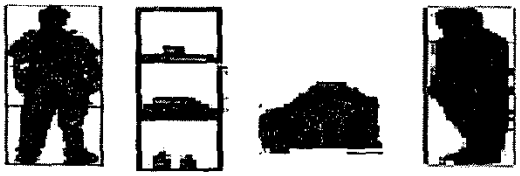

Figure 3. First two images are examples of training images and histograms used for calculation of the lookup values and the last two images are example test data

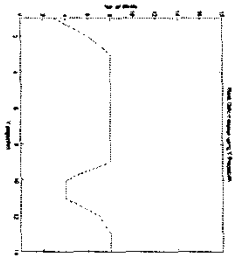

Figure 4. Example y-projection of the first bounding box (H1)

7. Computing Motion Tracks using Color Sum of Squared Differences (CSSD)

This step is used to track people so that specific inferences can be drawn from their movements. Step 6 provides input to this step. Our objective is only to get fast and reliable estimates of the motion trajectory. Therefore we use Color SSD, which is definitely an improvement over Gray Scale SSD in that color provides more information. We independently track the person in each of the color constituents, red R, green G, blue B, and then combine the results with an intra-multiplicative step, given by,

Position of person in current frame =

$$CSSD(R(x,y)) * CSSD(G(x,y)) * CSSD(B(x,y)) \quad (2)$$

where

$$CSSD(\delta(x,y)) = arg\ min_{u=0..m, v=0..n}\ \hbar(x,y)$$

where

$$\hbar(x,y) = \sum_{i=0}^{k}\sum_{j=0}^{l}[F_k(x+i,y+j) - F_{k+1}(x+i+u,y+j+v)]^2$$

$F_k(x,y) = Intensity\ value\ at\ (x,y)\ in\ frame\ k.$

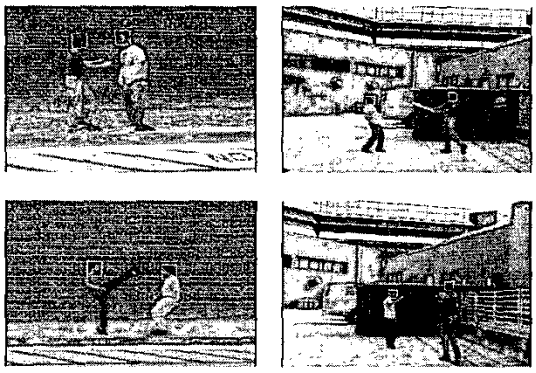

Figure 5. Tracking results using CSSD

8. Compute Acceleration Measure Vector (AMV) and Jerk

We get from step 7 the centroid of the head-tracking box. Next, we calculate the third derivative of its speed with respect to time to infer *jerk*. During violence, the motion trajectory of a person experiences a drastic change after being hit by the other person and *jerk* is an effective way to capture this behavior. The laws of physics give:

$$A(t) = dV/dt \quad \& \quad J(t) = dA/dt,$$

where $V(t)$ = velocity, $A(t)$ = acclrn., $J(t)$ = Jerk, t = time. Given $MT_i = < \vec{P}_1, \vec{P}_2, ..., \vec{P}_n >$, the motion trajectory for the i^{th} person, where $\vec{P}_i = (x,y)$ is the centroid coordinates of the tracking box in the i^{th} image. Define *Acceleration Measure Vector* (AMV) as:

$$\delta(\theta,d) = \alpha.\psi(\vec{P_{k-1}}, \vec{P_k}, \vec{P_{k+1}})i + \beta.\vartheta(\vec{P_{k-1}}, \vec{P_k}, \vec{P_{k+1}})j \quad (3)$$

where α, β are the respective weights assigned to the direction and magnitude component of acceleration, d= distance in terms of pixels.

$$\psi(\vec{P_{k-1}}, \vec{P_k}, \vec{P_{k+1}}) = (1 - \cos\theta), \quad (4)$$

where

$$\cos\theta = \frac{(\overrightarrow{P_{k-1}P_k}).(\overrightarrow{P_kP_{k+1}})}{\|(\overrightarrow{P_{k-1}P_k})\|.\|(\overrightarrow{P_kP_{k+1}})\|} \quad (5)$$

and

$$\vartheta(\vec{P_{k-1}}, \vec{P_k}, \vec{P_{k+1}}) = |(\|\overrightarrow{P_{k-1}P_k}\| - \|\overrightarrow{P_{k-1}P_k}\|)| \quad (6)$$

Then

$$jerk = \sqrt{(\frac{\partial\psi}{dt})^2 + (\frac{\partial\vartheta}{dt})^2} \quad (7)$$

AMV calculates the phenomenon that if the person has been moving in a direction for the last 'i' frames and then suddenly changes direction and magnitude of motion, then this

person is a good candidate for having been hit. If someone else is close and their limbs are extended towards the jerked person (explained in step 9), we conclude this is violence and mark the frame as a 'candidate' frame.

9. Compute the Orientation Map for Arms and Legs

This step is performed simultaneously with step 8. It monitors specific body parts, getting its input from step 6. We move outwards from the shoulder point and traverse the silhouette boundary to get the *orientation* of the upper arm, which is then further refined in step 12.

Experiments show that traversing up to the 'torso' is usually enough to get the upper arm's *orientation*. Using box H2 from step 4, we traverse the outer boundary of the silhouette to get the *orientation* for the legs.

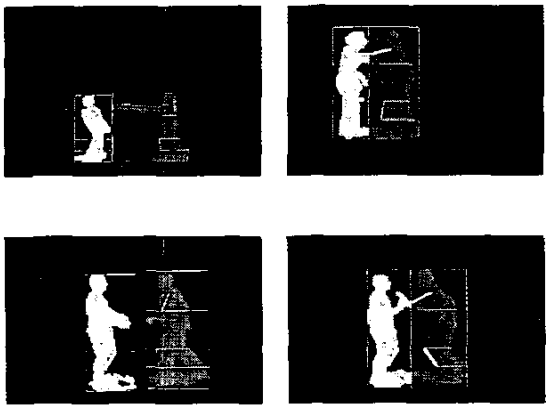

Figure 6. White lines on body give orientation. Top left picture depicts violence using an object

$$Orientation\ \theta = \tan^{-1}\left(\frac{\frac{\partial \vec{P}_2}{dy} - \frac{\partial \vec{P}_1}{dy}}{\frac{\partial \vec{P}_2}{dx} - \frac{\partial \vec{P}_1}{dx}}\right), \qquad (8)$$

where $\vec{P}_1, \vec{P}_2$are the position vectors of shoulder points and where traversal stops.

We keep track of these *orientations* for every frame. During violence, people raise arms and/or legs, and hence the *orientation* of hand and/or leg starts to change towards being parallel/negative to the ground plane. Our system measures this feature and marks the frame in which the *orientation* of hand/leg is close to being parallel or has negative slope to the ground plane as a 'candidate' frame. From this approach we also get enough information to decisively say who the hitter is and who is being hit. Whichever person's rate of change of *orientation* is faster gets the label of being the 'hitter'. After crosschecking with step 8 we remove all the redundant 'candidate' frames and instead mark one particular frame as being the 'violent frame'. The way we determine the 'violent frame' from the 'candidate' frames is that we look for a frame which has the most approval from step 8 and the frame which is chosen by this step and we take the average of the two to get the 'violent frame'. In figure 10, the horizontal lines across the x-projection of motion trajectory and *orientation* map prove that there exists a temporal consistency between the two maps, which means that both step 8 & 9 mark their 'candidate' frames at approximately the same time instant, so the average of the these two is very close to the correct 'violent' frame.

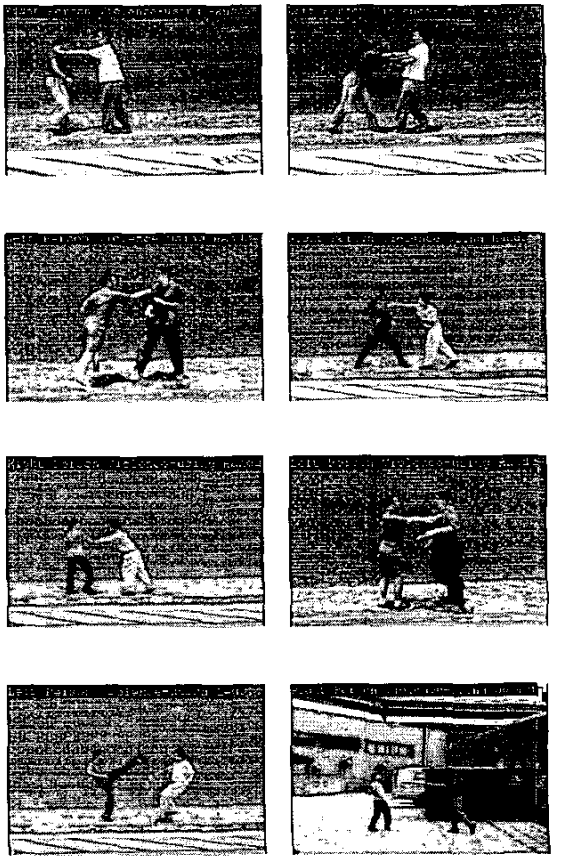

Figure 7. Violence detected using steps 8 and 9. In lower two frames some distance was maintained between the two persons.

If the sequence meets other conditions which are: more than 2 'violent frames' and has also previously passed tests

like presence of two people (calculated from number of connected components) and people approach before hitting (calculated from the motion history of silhouettes) then we label the sequence as being a 'violent' sequence.

10. Detecting Tool-Mediated Violence

Detection of violent acts involving objects is a hard problem. Nevertheless we have made some progress towards this goal. Suppose a person enters the scene. Then at that time instant we do not have the information whether or not the person has some object because it is hard to classify any thing in the hand as a potential weapon or not, moreover the object might not be even visible, due to occlusion. So we wait for the first violent act to occur. When using step 8 and 9 we detect some on going violent activity, we check for the following conditions in linear order:

10.1. Distance Between People:

Since the object will be used with the hand, then it serves to increases the distance between people at the time of impact.We add the length of shoulder to legs for each person and check whether this is smaller than the current distance between shoulders. If so, then we mark this frame as a 'candidate-object frame' but it still needs approval from the next two steps.

10.2. Absence of Skin at Point of Impact

Using color predicate [5] we check for skin at point of impact. If an object is used for violence then skin should not be present at the point of impact.

10.3. Presence of Skin at Point of Holding

We then look for skin by traversing the boundary of the object, which helps us in getting the point where the hand is holding the object.

11. Detecting Non-Violent Activities

Not every meeting between two people ends up being violent. Therefore it is very important to detect non-violent activities also to prevent raising false alarms. Non-violent activities for our purpose include walking, handshakes, object handovers and finger pointing.

11.1. Walking

Walking is the easiest case to detect, the silhouettes are always in motion and neither the *jerk* nor the *orientation* map detects anything, so we label such a case as walking.

11.2. Handshakes

During handshakes there's no *jerk* involved but there's a change in *orientation*, which has a periodicity to it; we label such a pattern as handshakes.

11.3. Object Handovers

Object handovers are challenging because if the object is small then its very hard to detect it but if the object is big then by comparing the area change near the hand after they separate is enough to tell who possesses the object.

11.4. Finger Pointing

Finger pointing is a very interesting phenomenon because most of the times finger pointing is a precursor to the actual fight happening. During finger pointing there's no *jerk* involved but *orientation* map of arm is parallel/negative to the ground which can be detected and labelled.

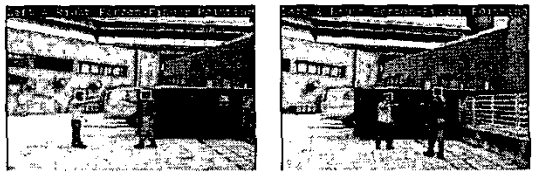

Figure 8. Finger-Pointing Detected

12. Refining the Orientation Data

We have also investigated refining the *orientation* data we get from traversing the silhouette. The basic premise of this work is that even though the results from the silhouette data are robust enough when the arms or the legs are outside the body parameter, the *orientation* lacks precision when the arm or leg is within the body perimeter. To correct this, we use the Radon Transform to get the *orientation* of body parts when they are inside the body parameter. We select a small patch around the shoulder location as determined before in component 3 and feed that data to the Radon Transform which outputs the projection data of line parameters and then we use a simple threshold to select the strongest line out of the projection data. This strongest line is usually the strongest edge around the shoulders, which is the line between shoulder and the chest. For the legs this line corresponds to their outer most edge. Therefore we are able to obtain a much more accurate *orientation* of the hand and legs when they are inside the body perimeter.

Figure 9. Some results of better orientation after using Radon Transform

13. Results

The system runs on a Windows PC. Eight different people were tested performing various kinds of violent and non-violent activities. Data sets were taken with a 3CCD Sony stationary camera.

Name	Total Frames	# of violent Acts	# Detected
Omar_Yun1	510	6	6
Omar_Yun2	279	1	1
Yun_Omar1	400	2	2
Yun_Omar2	239	1	1
Zeeshan_Cen1	474	4	4
Zeeshan_Cen2	333	5	4
Jaime_Jigna1	329	2	2
Jaime_Xuan2	408	3	3
Joanna_Xu1	281	4	4
Joey_Dave1	310	8	7
Joey_Dave2	249	1	1
Dave_Will_1	130	1	1
Joey_Joanna	210	2	2
Kris_Rusty1	465	6	6
Kris_Rusty2	304	2	2

14. Discussions/Future Work

The table above shows very positive results as all our data sets conformed to the basic assumptions. However, our system breaks down if people in the scene instead of hitting each other start to wrestle. Also if during violence, if one of the persons falls down then also our system will break down because currently we have assumed upright silhouettes. Additionally gang or group violence will also cause malfunction. Sometimes the skin detection algorithm breaks down and steps 10.2-10.3 malfunction. All these cases are future work.

15. Conclusion

We presented a system to track and monitor an area for violent actions between people. We do this by doing a combined analysis of two independent approaches, both of which give results that are reliable and when combined together, the inferences become very robust. The system has been tested on a variety of people with different physical builds and under various backdrop conditions. We have also tested our system for possible situations like object handovers, handshakes and normal walking.

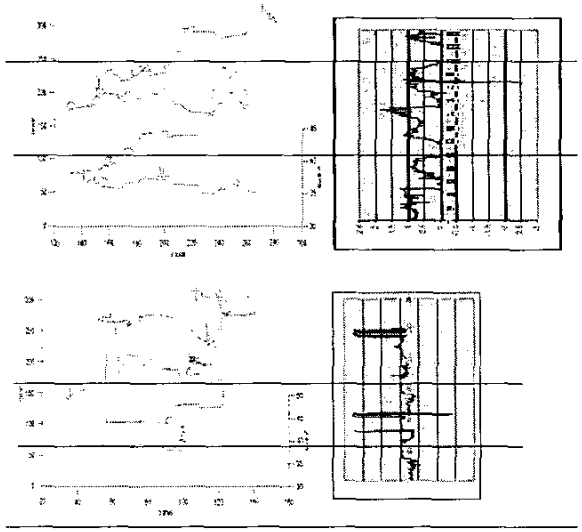

Figure 10. Motion Trajectory's x-projection and Orientation map. The horizontal lines in graph prove the temporal consistency between the two maps.

References

[1] Kuno, Y.; Watanabe, T, Shimosakoda Y., Nakagawa S., *"Automated Detection of Humans for Visual Surveillance Systems"*, Proc. 13th ICPR, vol 3, 1996, pp: 865 -869.

[2] C. Stauffer, E Grimson, *"Learning Patterns of Activity using Real Time Tracking"*, PAMI, Vol. 22, Aug. 2000

[3] N. Vasconcelos, Lippman. *"Towards Semantically Meaningful Feature Spaces for the Characterization of Video Content"*. ICIP 1997, Vol. 1, pp 25-28

[4] Nam, J., Alghoniemy, M., *"Audio-visual content-based violent scene characterization"*, ICIP 98, pp 353-357.

[5] Kjedlsen R., Kender J. *"Finding skin in color image"*. Face and Gesture Recognition, 1996, pp 312-317.