Fire and Smoke Detection in Video Sequences

Sergey Arseev[1], Tatiana Lukina[2], Mikhail Sindeev[1], Vadim Konushin[2], Anton Konushin[1,3]

1. Lomonosov Moscow State University, Moscow, Russia
2. Video Analysis Technologies, LLC
3. NRU Higher School of Economics, Moscow, Russia

Abstract

In this paper we present a fire and smoke detection system that uses a block-based approach. It detects smoke and fire separately in each block of a video sequence. The detection is based on spatio-temporal features of the smoke and fire, such as the fire flickering and smoke spread. Then these block detections are merged into the fire and smoke events, which are the final detections. Two versions of the algorithm are presented: in the first version the block classification is made by a heuristic classifier that doesn't use a machine learning, and in the second version the block classifiers are trained using videos from the trained set. We performed training and evaluation on a private dataset, consisting of 120 videos. Experimental results are presented.

***Keywords**: fire detection, smoke detection, block based detection.*

1. INTRODUCTION

Fire safety is as an important task nowadays as it has always been. The fire can cause tremendous property damage or even human casualties. Unfortunately, existing automatic systems still don't guarantee that the fire will be detected in time. Standard sensors require the presence of smoke particles in the sensor to set off, and the time required for the smoke to spread can be too big, especially in large rooms. More reliable sensors require installation of expensive hardware – and the more reliable it gets, the more expensive it becomes. A computer vision solution to the fire detection problem could solve these problems. The camera-based system works instantly in rooms of any size or even open air, and it doesn't require any additional hardware since it can be integrated into an existing surveillance system. The main problem here is to develop a system that would be reliable enough to replace the standard fire detectors.

.

Various approaches exist in this field of study. Usually fire detection and smoke detection are performed by separate algorithms since these are two different problems that sometimes require different methods to solve them.

Many fire detection algorithms use the block-based classification, i.e. split the video sequence into blocks and classify each block separately as fire/not fire. Panagiotis Barmpouti et al in [1] present a complex block-based fire detection system that uses a range of various features: color features (that are used as a starting point in almost any fire detection algorithm), spatial wavelet energy, spatio-temporal and temporal features such as flickering. Y. H. Habiboglu et al in [3] use a method that is based on region covariance descriptors.

Another popular approach for the fire detection is the extraction of regions of interest (RoI) instead of splitting the video into blocks. In [6], Nicholas True extracts the RoI with a background subtraction and color features and then uses the dynamic texture analysis for fire detection. Steven Verstockt et al in [7] present an approach based on the wavelet features. In [8] they build a complex multi-view system that detects fire and smoke using a number of features such as bounding box and boundary area disorder. Hongcheng Wang et al in [9] also use boundary area analysis and other motion-based spatio-temporal features in their fire detection system.

The block-based approach and RoI extraction are also popular in the smoke detection. In [4], multiple features such as area, bounding rectangle, the average and standard deviation of Y-value, and the average and standard deviation of UV-value are extracted from each RoI and then used for classification. In [5], the authors use the histogram-based approach for this purpose. Jayavardhana Gubbi et al in [2] choose the block-based approach and use wavelets and the discrete cosine transform for classification.

In this paper, we present a system that uses the block-based approach for smoke and fire detection. We use a range of spatio-temporal features based on the characteristics of smoke and fire such as the fire flickering or smoke spread. A detection example is shown in fig. 1.

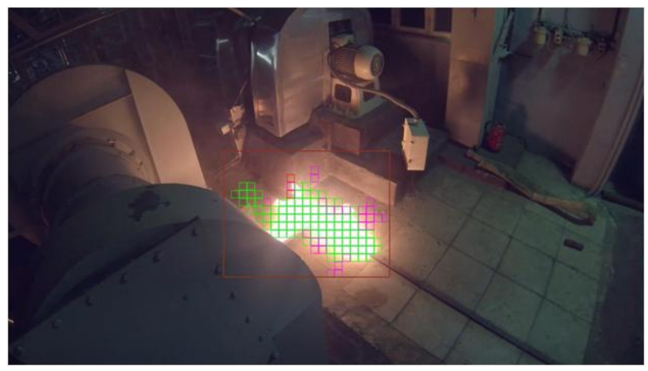

Figure 1: a detection example. Green boxes represent fire blocks, other colors are used to mark events and different types of detections

2. PROPOSED METHOD

A video sequence is processed by two separate algorithms: a smoke detection algorithm and a fire detection algorithm. These algorithms both use block-based analysis to extract spatio-temporal features from video and then filter block-level detections and merge them into smoke and fire events. They consist of these main steps:

1. The video sequence is divided into spatio-temporal blocks, i.e. each frame is divided into small squares of 16×16 pixels, and the block is a group of such squares taken across several consecutive frames. The number of frames in a block is called the block length. This division is performed separately for both algorithms, so the blocks used in smoke detection and in fire detection can have different length.
2. Each block is independently analyzed and classified as containing smoke/fire or not containing. The fire detection algorithm extracts features from the entire block while the smoke extraction algorithm first classifies each frame of the block and then makes a decision regarding the entire block.
3. The block-level detections are grouped into events. This step is required to filter occasional false detections and improve the overall reliability of the algorithm. An event is a time frame in a video containing smoke or fire, depending on the event type.

2.1 Smoke detection

The smoke detection algorithm is based on the fact that when the translucent smoke spreads over background objects, it softens edges making them less prominent but doesn't remove them completely. As the smoke spreads, the translucent smoke changes into the dense smoke that can be detected using color features but the translucent smoke is necessary for the smoke event because of a large number of dense smoke false detections, e.g. a white wall can be mistaken for a dense smoke.

2.1.1 Splitting into blocks

The smoke detection algorithm doesn't use every frame of a video, it uses every 8th frame (for 24 FPS video sequences). The number of frames in a block is 100 which equals to 800 frames or more than 30 seconds of a video sequence. To reduce the lag, blocks overlap by the time axis: each subsequent block starts 8 frames after the previous. So if the first block uses frames 1, 9, 17…, the second will use frames 9, 17, 25… and so on. Also during the start and restart (see further) of the algorithm, the blocks use less than 100 frames to prevent the initial lag.

2.1.2 Feature extraction

Each frame in a block is compared to the first frame of this block. For the i^{th} frame, its mean saturation (sat_i) and value (val_i) are calculated. Also, the frame is converted to grayscale and processed with Sobel filters to produce a gradient map ($grad_{i,x,y}$). If a sum of the gradients ($\sum_{x,y}(grad_{i,x,y})^2$) is less than 0.001, the frame is classified as containing no smoke because we cannot find smoke in a frame with no reliable gradients.

Then these two values are calculated:

$$k_i = \frac{\sum_{x,y}\left(grad_{1,x,y} * grad_{i,x,y}\right)}{\sum_{x,y}\left(grad_{i,x,y}\right)^2}$$

$$b_i = \frac{\sum_{x,y}\left(grad_{1,x,y} - \frac{grad_{i,x,y}}{k_i}\right)^2}{\sum_{x,y}\left(grad_{i,x,y}\right)^2}$$

k_i is called a gradient change amount (it represents the amount of gradient decrease in a block) and b_i is a residual value, it represents the displacement of gradients since it uses the difference of the gradient in the first frame and the gradient in the current frame normalized by the gradient change amount.

2.1.3 Frame classification

These conditions need to be met for the frame to be classified as containing smoke:

- $val_1 < 0.95$ – we don't search for smoke if the block is too bright
- $max_{x,y}\left(grad_{1,x,y}\right) > 0.6$ – the gradients on the first frame of the block must be prominent enough
- $sat_1 - sat_i > -0.02$ – the smoke is gray, so the saturation shouldn't increase
- $val_1 - val_i < 0$ – on the contrary, the value shouldn't decrease since the smoke is usually white or light gray
- $k_i < 0.7$ – edge prominence decreases
- $b_i < 0.35$ – edge position didn't change significantly, i.e. there was no occlusion by a large object.

2.1.4 Block classification

The dense smoke in a block is detected using the following conditions:

- $sat_1 < 0.2$
- $val_1 > 0.6$
- $val_1 < 0.95$
- $max_{x,y}\left(grad_{1,x,y}\right) < 0.3$

If they are met, the block is classified as containing a dense smoke.

To be classified as containing a translucent smoke a block must have at least 5 frames (not necessary consecutive) containing the smoke.

The algorithm can produce false positive results if the lighting in a scene abruptly changes, e.g. when the lights in a room are turned on/off because it can produce the same gradient and color changes that are used to detect smoke. To counter this, the algorithm stops for a while when it detects the abrupt lighting change and then reinitializes and restarts. The algorithm calculates the mean intensity for the last three frames, and if the difference between the minimum and the maximum values exceeds 0.07, it skips the next ten frames and restarts.

2.1.5 Event building

In order to exclude "noise", i.e. single false block detections, the three-dimensional (spatio-temporal) median filter with a 3-block diameter is applied to the detections. Then, two types of events are constructed:

1. The translucent smoke area growth. Over the last 5 steps (i.e. the time between block starts) of the video there should be at least 4 smoke blocks, the intersection of the smoke area for each pair of consecutive steps (of these 5 steps) should not be empty, and the union of all smoke positions should be at least 2 blocks greater than the amount of smoke on the first frame.
2. The translucent smoke changing into dense. The block is changing from translucent to dense if it contained translucent smoke 10 or less steps ago and contains dense smoke at the moment. If there are at least 3 such blocks, this is classified as an event.

2.2 Fire detection

2.1.1 Splitting into blocks

In the fire detection algorithm, a block has a size of 16×16 px × 32 consecutive frames. Contrary to blocks in the smoke detection algorithm, they don't overlap with each other.

2.1.2 Feature extraction

The fire detection is performed in two separate ways: the detection of a flickering fire with reflections on nearby objects and the detection of moving flares.

The first step of the flickering fire detection is the construction of a two-dimensional mask consisting of pixels that meet the following conditions:

- More than 20% of the time the pixel has a "flame-like" color. i.e. $r > g$ && $g \geq b$ && $r > 110/255$, where r, g and b are respectively the red, green and blue channels.
- The minimum and the maximum color values (taken separately by each color channel) during the block are calculated. The length of the line between them (called the min-max line) should be more than 0.2. It is required to filter the noise that has lesser change magnitude. See fig. 2 for illustration.
- The maximum pixel color deviation from the min-max line (the distance between the pixel color and its projection to the line) should be less than 0.1. Pixels in the fire block have similar color and flicker consistently, i.e. their color should stay close to the min-max line during the block.
- The maximum pixel deviation should be less than 0.25 × the length of the min-max line. This is required to filter the noise.

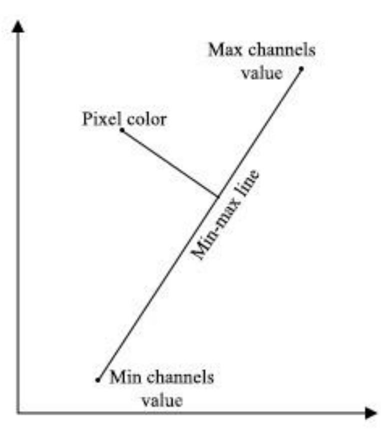

Figure 2: the illustration of a min-max line (projected from 3D color space to 2D) and a pixel color projection onto it.

Then this mask is eroded with a 7×7 pixel square as a structural element. If the final mask contains less than 3% of block pixels, the block is classified as not containing fire.

After that, an oscillation value is calculated for each pixel in each frame. This value equals to the distance between the projection of the pixel on a min-max line and the minimum color. The oscillation is normalized to the length of the min-max line.

A mean oscillation value by all pixels inside a mask is calculated for each frame, so we get a vector of 32 (block length) mean oscillation values. Two values are calculated for this vector: the number of local maxima (*LocMax*) and the number of intersections with the mean value (*MeanIntersection*). These values represent the mean flame flickering over the duration of the block.

An additional value is calculated for the entire block (including those pixels that didn't get into the mask): the *consensus* value that equals to the number of pixels whose oscillation at least 80% of the time didn't deviate from the mean by more than 0.2.

The flare detection algorithm calculates the intensity of each pixel on each frame and compares it to the two thresholds: "background" – less than 0.6 and "fire" – greater than 0.95. Then it calculates how many times during the block the pixel changed its state between "background" and "fire".

2.1.3 Block classification

The flickering detection algorithm uses this formula to determine if a block contains fire or not:

$$LM = 1 - 25 * \left(\frac{ocMax}{BlockLength} - 0.3\right)^2$$

$$MI = 1 - 15 * \left(\frac{MeanIntersection}{BlockLength} - 0.3\right)^2$$

$$FireLevel = \frac{(LM + MI) * consensus}{2}$$

The *FireLevel* value increases as the oscillation features get closer to the certain values representing the "ideal" fire. If the *FireLevel* is > 0.3, the block is classified as the fire block.

The algorithm also detects "bright" blocks. Only the blocks on the edges of the flame show the flickering behavior while the middle part of the flame can get plain white due to the lack of dynamic range in the video camera. The block is considered "bright" if at least 60% of its pixels meet the following conditions:

- $I > 0.7$
- $r \geq g$
- $r \geq b$

  or $I > 0.95$

where I is the pixel color intensity and r, g and b are the values of the red, green and blue channels.

Each spatial block has a counter that helps the algorithm to adapt to the lighting change. At the beginning of the video, all bright blocks are given the counter value of 1 and all dim (not bright) blocks have the value of 0. Then, in each subsequent frame the bright blocks counter increments by 1 until it reaches 25 and the dim block counter decrements by 1 until it reaches 0. Overall, to be considered bright the block should have the counter value from 1 to 15.

If the frame has more than 3 bright blocks and at least 1 flickering fire block, all bright blocks are treated as flickering fire blocks for the purpose of the event construction.

To be classified as a flare the block should have at least five pixels that changed their state between "fire" and "background" at least four times.

2.1.4 Event building

A two-step median filtering is performed for the flickering fire blocks:

1. If at least 3 blocks in the spatial vicinity of the block contain flickering fire, this block is also classified as containing fire.
2. Then, the blocks are filtered by a median filter on the time axis.

If the last 4 frames contain at least 4 flickering fire blocks that don't change their position across these 4 frames, then the fire event is detected.

If a flare block keeps its position for at least 5 consecutive frames, the event is also detected.

2.3 Alternative block classification

We also implemented an alternative method of block classification (and frame classification in the smoke detection algorithm). Instead of the described heuristic formulas, we used the automatic classifiers trained with the AdaBoost technique. We chose the ensemble of 5 decision trees for the smoke detection and 250 decision trees for the fire detection.

Due to the lack of the block-wise markup in our data set (see further), we used the initial version of the algorithm to generate features according to the following rules:

- If the algorithm classifies the block as containing fire (or smoke) and the block falls in the time constraints of the respective event, then all features extracted from it are written in the positive set.
- It the algorithm classifies the block as containing fire or smoke but the block doesn't fit into any of the respective events (i.e. a false positive results), then all features extracted from it are written in the negative set.
- If the algorithm classifies the block as not containing fire or smoke, and the block doesn't fit into any of the respective events (to prevent the false negative results from being considered as negative examples), then all features are written in the negative set with the probability of 1/50.

The $sat_1 - sat_i$, $val_1 - val_i$, k_i and b_i values were used as features for the smoke classifier. The fire classifier used the $LocMax$, $MeanIntersection$ and $consensus$ values.

3. DATA SET

We have marked a private dataset provided by Video Analysis Technologies, LLC, consisting of 120 video sequences. The sequences have varied length (from one minute to ten hours) and are captured with multiple cameras. Most of them contain scenes with smoke and/or fire while the rest are used to detect the false positive results. The video sequences containing events were marked. The block-wise or pixel-wise marking is extremely tedious so we used the simplified time-wise marking: the beginning and end times of each event were stored. 90 video sequences were used as a training set and 30 as a testing set.

The total length of video sequences containing smoke and fire is approximately 5.5 hours. There are 45 fire events in these video sequences with the total length of 20 minutes and 84 smoke events with the total length of 1 hour and 45 minutes.

Scenes in the video sequences vary greatly and include different scenarios such as wiring ignition, trash can catching fire from a cigarette etc. The algorithm performs better on some scenarios over another. For example, figure 3 shows a difficult scene: the smoke comes out pretty dense (while the algorithm is focused on translucent smoke) and the fire is occluded by smoke which makes it harder for the fire detection algorithm to detect flares and flickering fire. Only a few blocks (green boxes for fire and blue boxes for smoke) were detected in this frame.

Figure 3. A difficult scene.

4. EXPERIMENTAL RESULTS

The results are presented in the table 1.

	Base algorithm	Modified algorithm
Precision (fire)	0.54	**0.90**
Recall (fire)	**0.93**	0.82
F-score (fire)	0.65	**0.87**
Precision (smoke)	0.93	**0.98**
Recall (smoke)	**0.79**	0.76
F-score (smoke)	0.87	**0.88**

Table 1. Experimental results

As seen from the table, the modified algorithm generally works better than the base algorithm but the recall of the base algorithm is higher. For the fire detection, the value with high F-score was chosen to show the significant precision and overall score improvement while keeping a decent recall value. The training is complicated because the size of the training set is relatively low, and that is why most formulas were created manually. The lack of open data sets also complicates the comparison of the results to the results of other researchers.

The algorithm speed allows it to work in real time on a 3 GHz Intel Core i3 processor which means that this approach can be used not only for the video sequence analysis but also for the video stream analysis.

5. CONCLUSION

A new fire and smoke detection system has been presented in this paper. It uses the block-based approach and spatio-temporal features for both fire and smoke detection. The system consists of two separate algorithms: the fire detection algorithm which, in turn, consists of a flickering fire detection and flare detection sub-algorithms and the smoke detection algorithm that detects translucent smoke utilizing the fact that the smoke gradually spreads over the scene, softening edges but not removing them completely until it becomes dense enough. The system shows high fire and smoke detection results on various video sequences on a private data set.

6. ACKNOWLEDGEMENTS

This work was supported by RFBR grant no. 14-01-00849.

7. REFERENCES

[1] Panagiotis Barmpouti, Kosmas Dimitropoulos, Nikos Grammalidis. *Real time Video Fire Detection using Spatio-Temporal Consistency Energy* // Advanced Video and Signal Based Surveillance (AVSS), 2013 10th IEEE International Conference on . Krakow, 2013, P. 365 – 370.

[2] Jayavardhana Gubbi, Slaven Marusic, Marimuthu Palaniswami. *Smoke Detection In Video Using Wavelets And Support Vector Machines* // Fire Safety Journal. 2009. 44. N 8. P. 1110-1115

[3] Yusuf Hakan Habiboglu, Osman Gunay, A. Enis Cetin. *Flame Detection Method In Video Using Covariance Descriptors* // Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference . Prague, 2010, P. 1817 - 1820

[4] DongKeun Kim, Yuan-Fang Wang. *Smoke Detection in Video* // 53rd IEEE International Midwest Symposium on Circuits and Systems, 2010, P. 383 - 387

[5] Damir Krstinić, Darko Stipaničev, Toni Jakovčević. *Histogram-Based Smoke Segmentation in Forest Fire Detection System* // Information Technology and Control. 2009. 38. N. 3.

[6] Nicholas True. *Computer Vision Based Fire Detection*

[7] Steven Verstockt et al. *Wavelet-Based Multi-Modal Fire Detection* // Signal Processing Conference, 2011 19th European . Barcelona: IEEE, 2011, P. 903 – 907.

[8] Steven Verstockt et al. *FireCube: A Multi-View Localization Framework for 3D Fire Analysis* // Fire Safety Journal. 2011. 46. N 5. P. 262-275

[9] Hongcheng Wang et al. *Spatial-Temporal Structural and Dynamics Features for Video Fire Detection* // Applications of Computer Vision (WACV), 2013 IEEE Workshop on . Tampa, FL: IEEE, 2013, P. 513 – 519.