Notebook Paper

Brno University of Technology at TRECVid 2013
Interactive Surveillance Event Detection

Petr Chmelar, Jozef Mlich, Martin Pesek, Tomas Volf, Pavel Zemcik, Jaroslav Zendulka

IT4Innovations Centre of Excellence
Brno University of Technology
Faculty of Information Technology
Bozetechova 2
Brno, 612 66
Czech Republic

Abstract

In the paper, we describe our experiments in the interactive surveillance event detection pilot (SED) of the 2013 TRECVid evaluation [13]. Our approach inherits functionality of the Surveillance Network Augmented by Retrieval Event Detection (SUNAR-ED) system, which is an information retrieval based wide area (video) surveillance system being developed at Faculty of Information Technology, Brno University of Technology. It contains both standard and experimental techniques evaluated at the AVSS 2009/10 Multi-Camera Tracking Challenge and 2012 SED. We have deployed active learning functionality (Bayesian, SVM and HMM) based on moving objects' trajectory statistics and shape classification using Video Terror Application Programming Interface (VTApi), which was created to unify and accelerate the intelligent vision applications development. We have focused mainly on Hidden Markov Models experiments in 2013.

The paper is organized as follows. Section 1 provides a motivation and a brief introduction. Details on the technology is presented in Section 2. The section is dedicated also to the active learning approach and to the setup of experiments. Section 3 shows some experimental results achieved during the training. Finally, Section 4 discusses achieved results and concludes the paper. Evaluation results are attached at the end of the paper.

1. We have submitted the following SED runs:
   - BrnoUT_2013_retroED_EVAL13_ENG_s-camera_p-SUNAR-SVM_1 – contains 2000 “best” shots classified by SVM for each event based on the object shape and trajectory using active-learning.
   - BrnoUT_2013_retroED_EVAL13_ENG_s-camera_c-SUNAR-HMM_1 – contains 2000 best trajectories classified by HMM (including object shapes).
   - SED13_BrnoUT_2013_interactiveED_EVAL13_ENG_s-camera_p-SUNAR-121_1 – includes only shots annotated during the interactive period.

2. The major difference between the runs is the training method (SVM versus HMM) and the active learning step based on 25 min. annotating of results of the retrospective task. The retrospective tasks both maximized recall, while the interactive task maximized precision.

3. The mayor contribution was the semi-automatic annotation using active-learning, classification of object description using trajectory and shape features and the tracker able to handle multiple occlusions as well as novel HMM implementation.

4. The challenge of the TRECVid SED pilot and the video surveillance event detection in general is the ability to learn from annotations provided and to improve the classifiers by fusion of different methods of learning – SVM, HMM and active learning.

Acknowledgements

This work has been supported by the research project Security-Oriented Research in Information Technology CEZ MSM0021630528, grant VG20102015006 of the Ministry of the Interior of the Czech Republic, the European Regional Development Fund in the IT4Innovations Centre of Excellence (CZ.1.05/1.1.00/02.0070). Special thanks go to Janicka Chmelarova.

1. Introduction

In 2006, we have started to develop an information retrieval based multi-camera tracking system intended to be at the top of the state of the art. The idea was to create an automated system for visual features detection, indexing and analysis, which can reduce the burden of continuous concentration on monitoring and increase the effectiveness of information reuse by security, police, emergency, firemen or armed services.

Brno University of Technology, Faculty of Information Technology has taken part at TRECVid since 2007. In the past, we have taken part in various tasks, but SED in 2008 and 2012 only. Our first attempt was based on advanced masking, background subtractions and extracted trajectories. Later, we have avoided the masking approach focused more on other moving object based features and active learning.

The challenge of the TRECVid pilot and a better video surveillance event detection in general are high-quality annotations. There is only temporal localization ground-truthed by the University of Pennsylvania Linguistic Data Consortium. And this fact does not help for surveillance task of this kind much. The manual labeling of objects taking part in the event annotated is really expensive in such amounts of video (a hour of video takes about 5 hours of burden work), therefore we needed a more optimal strategy. For that purpose, we have developed two modes of annotation interfaces – the first one uses the output of our vision module (so that annotating 1 hour of video takes about 1 hour).

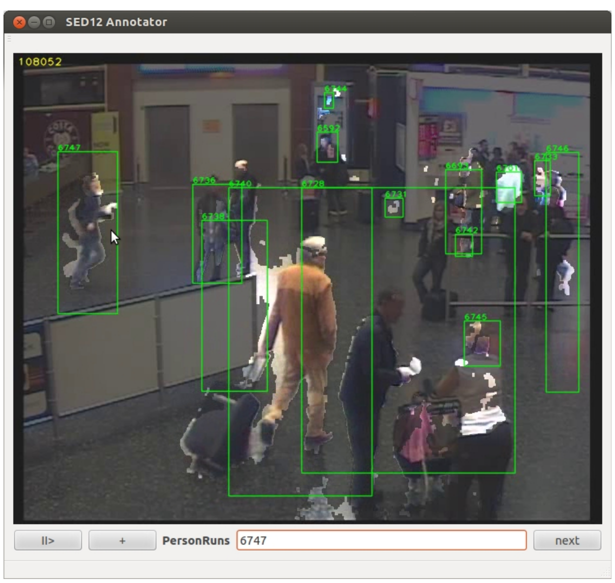


Fig. 1. SED Annotator in the initial Round #1 mode. A human annotator is supposed to click the running object or type the proper number when occluded.

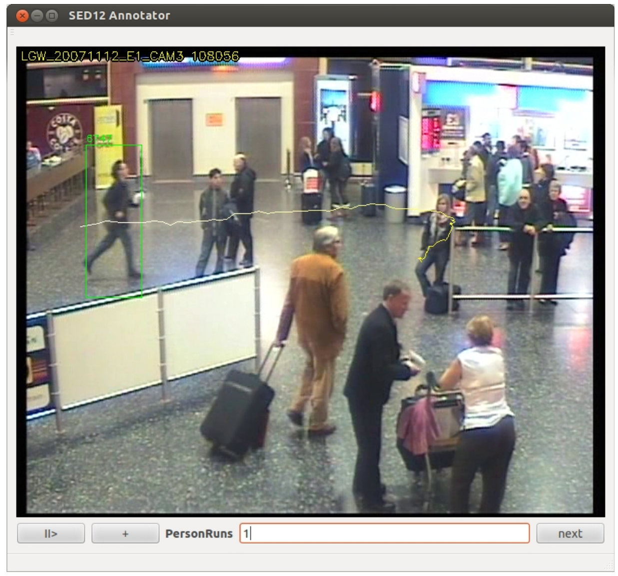


Fig. 2. Same scene as in figure 1 at the Round #2 of active learning process – a human annotator is supposed press “1” if the object highlighted is really running.

The second approach sorts the classified outputs and only two keys (of a keybord) are necessary to mark the event shown – positive (1+) or negative (0, Enter) and thus the annotation process may be further optimized. This is accomplished by active learning, as described further.

3. The technology

The technology is created to perform tracking, object and event detection. The SUNAR-ED (event detection) was based on Surveillance Network Augmented by Retrieval (SUNAR), upon our knowledge, the first implementation of a multi-camera surveillance system whose functionality is based on querying. The queries are of two types – online ones are used mainly for identity preservation; and offline to query the metadata of the camera records in the wide area when an accident, crime, a natural or human disaster occurs. The 2013 SUNAR-ED is mainly based on VTApi, which inherited most of the technology.

3.1 VTApi

Video Terror Application Programming Interface is an open source application programming interface designed to fulfill the needs of specific distributed computer vision data and metadata management and analytic systems and to unify and accelerate their development. It is oriented towards processing and efficient management of image and video data and related metadata for their retrieval, analysis and mining with the special emphasis on their spatio-temporal nature in real-world conditions.

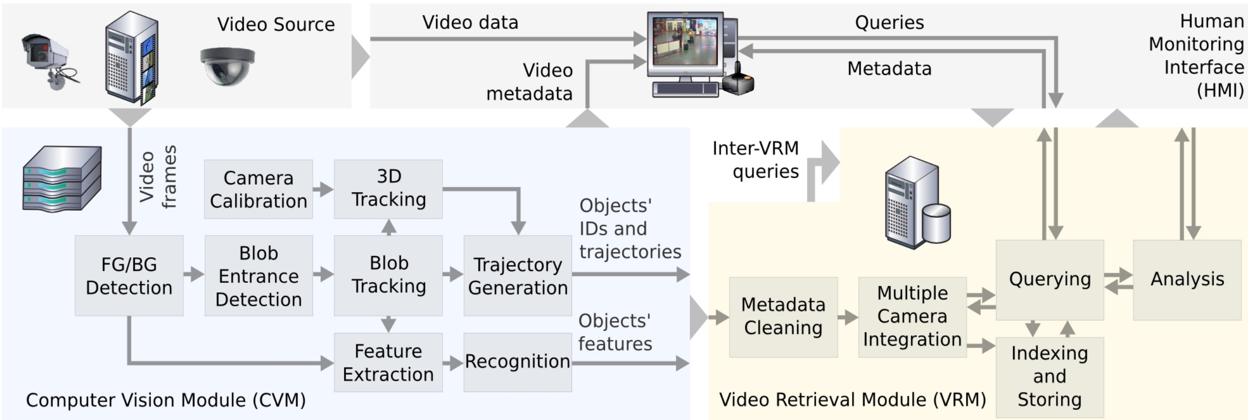


Fig. 3. Illustration of the SUNAR architecture.

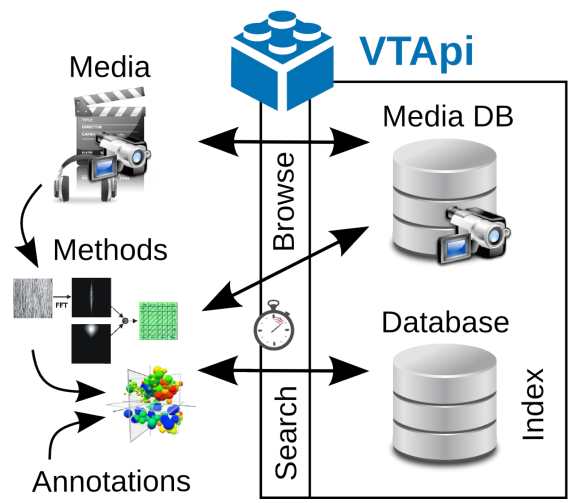


Fig. 4. The illustration of a position of the VTApi and a concept of methods' chaining.

```
Mat samples; // cv::Mat of training feature vectors
VTApi* vtapi = new VTApi(argc, argv);
Dataset* dataset = vtapi->newDataset("train"); // training dataset
dataset->next();
Sequence* sequence = dataset->newSequence();
 while (sequence->next()) { // for each video
    Interval* track = new Interval(*sequence, "tracks");
    while (track->next()) { // for each trajectory
        Mat sample; // cv::Mat for feature vector of trajectory
        float feature = track->getFloat("feature");
        // ... read features and fill feature vector
        samples.push_back(sample);
    }
}
CvEM model, labels; // GMM-EM model and cluster labels
CvEMParams params; // EM parameters
// ... set EM parameters including number of clusters
model.train(samples, Mat(), params, labels);
// ... choose dataset and sequences according to sample code above
while (track->next()) {
    Mat sample;
    // ... read features and fill feature vector
    int cluster = (int) model.predict(sample); // get cluster label
    track->setInt("cluster", cluster); // store cluster label
}
```

Fig. 5. The illustration of a position of the VTApi and a concept of methods' chaining (a). Sample code of reading trajectories, preparing training samples, GMM training and storing cluster labels into the database (b).

In addition to the technology, we also target usual aspects of the vision research – to unify and accelerate it by choosing an appropriate design methodology and architectural framework to enable the development of a complex computer vision applications at a reduced cost in terms of time and money.

The basic requirements include image and video feature extraction, storage and indexing to enable (content-based) retrieval, summarization and data mining in the meaning of object detection and activity recognition in an interactive and iterative process.

The VT methodology is based on the fact, that most methods of the same purpose have similar types of inputs and outputs, so there may be chains of them. Moreover, the input of a process (a running instance of a method) can be seen as another process's output (e.g., annotation, feature extraction, classification) including media data creation.

We have selected, integrated and extended a set of progressive and robust open source tools to be efficient for multimedia data and related metadata storage, indexing, retrieval and analysis. The system uses the best from (post)relational databases, it offers alternative storages and data structures we need to manage (e.g. vectors or matrices) to make the data access more efficient, especially for rapidly changing geography/ spatio-temporal data of a very complex nature in the binary form that can be now processed both on VTApi clients and in the database.

The SUNAR-ED uses the support for trajectory clustering, classification, object recognition, outliers detection and so on. The following example shows a clustering of trajectories using VTApi and an OpenCV implementation of Expectation-maximization (EM) algorithm, which estimates parameters of a Gaussian mixture model (GMM) [6].

Fig. 6. Examples of trajectory clustering results obtained by EM algorithm on trajectories from the first camera.

First, feature vectors representing trajectories are read from the database and training samples for the EM algorithm are prepared (see figure 6). Suppose that trajectories are stored in “Tracks” in this example. Second, GMM is trained by the EM algorithm and appropriate cluster labels are stored in the database. Similar approach was used for detection of most events.

3.1 SUNAR-ED

In brief, SUNAR-ED is composed of three basic modules – video processing CVM, retrieval VRM, the monitoring interface HMI and a video source. Computer Vision Modules are based on the OpenCV Library for object tracking extended by feature extraction and network communication capability similar to MPEG-7. Information about objects and the area under surveillance is cleaned, integrated, indexed and stored in Video Retrieval Modules. They are based on the PostgreSQL database extended to be capable of similarity and spatio-temporal information retrieval, which we have used to complement the learning.

Computer vision

The input of the Computer Vision Module (CVM) is a video stream. We use OpenCV [4] for tracking and 3D calibration especially (if feasible). We have extended the OpenCV Blobtrack demo to be capable of feature extraction, object (and event) recognition and IP based client-server video stream capture.

Objet tracking approach is is based mainly on proved methods of object tracking implemented in the Open Computer Vision Library [4]. These methods are illustrated in figures 1 and 2; and the schema is in figure 3. Discussed approach [5] is a complex problem in parameter configuration and learning, especially with real (crowded) scenes as illustrated in figure 2.

Foreground is derived from background, which is modeled using Gaussian Mixture Models (GMM, [6]) as an average value of color in each pixel of video and the foreground is a value different to the background. We have also been inspired by the approach developed by Carmona et al. [7], which is based on segmentation of the color in RGB color space into background, foreground and noise (reflection, shadow, ghost and fluctuation) using a color difference cone with vertex located in the beginning of the RGB coordinate system. In this way, the illumination can be separated from the color more easily. However the selection of appropriate parameters is a burden task, which is usual in unsupervised learning [8].

The other two modules – blob entrance and tracking are customized OpenCV Blobtrack [4] functions with appropriate parameters and sensitivity. Blob entrance detection is done by tracking connected components of the foreground mask. The Blob tracking algorithm is based again on a combination of tracking connected components and Particle filtering based on Means-shift resolver for collisions. We have extended the entrance algorithm by searching “in the future”, because it didn't track the object from its early appearance. There was used also the trajectory refinement using the (inverted) Kalman filter as described in [9].

The trajectory generation module has been completely rewritten to add the feature extraction as described below and TCP/IP network communication capability. The protocol is based on SQL, rather than XML (in previous versions) similarly to MPEG-7 [10].

The output of the CVM module is metadata of objects and the environment. It includes local identification of objects, its spatio-temporal location and its changes (speed) in the monitored area and a description of such objects – its dimensions, shape, color, texture or other special features (e.g. state plate or face descriptor) similarly to MPEG-7 [10]. The shape descriptor is based on normalized image moments selected most suitable from OpenCV's implementation.

The description is complemented with recognition of basic object classes (e.g. body parts, people, groups or luggage) and events (opposing flow, left luggage) based mainly on their trajectories and description as described below.

Video retrieval

The most functionlity of the SUNAR-ED system is implemented in the Video Retrieval Module (VRM). The input of the module is metadata produced by CVMs. This metadata is cleaned and normalized in time and space (lighting, color bias and estimated 3D parameters) and stored in the PostgreSQL database. The primary function of the SUNAR's VRM was the object identification – to integrate identifiers (IDs) of objects in the wide area, based on the previous occurrence of the object and its appearance. Although, this functionality was omitted for the purpose of SED 2013 evaluation, we have used the feature extraction and classification capabilities of VTApi to perform the event classification.

Analysis and classification

The VRM's analysis submodule is quite complex – it uses OLAP–based functionality for providing statistics on different granularities and views and it supports many machine-learning methods as Bayes classifiers, SVM [11], EM/GMM [6], HMM some other and time-series variants, frequent pattern analysis and various clustering algorithms. More detailed information can be found in [12].

For the purpose of SED 2013, we have employed parameter selection search using 5-fold cross-validation SVM [11] based on transformed features extracted from the moving objects and their trajectories. We refer this classification scheme as "Track". It contains features:

1. Camera (1-5).
2. Position – trajectory start (x_1, y_1), end (x_2, y_2), mean (μ_x, μ_y), standard deviation (σ_x, σ_y) and sum (Σdx, Σdy).
3. Trajectory duration (t).
4. Speed at trajectory start (dx_1, dy_1, v_1) end (dx_2, dx_2, v_2) mean (μ_{dx}, μ_{dy}, μ_v) and standard deviation (σ_{dx}, σ_{dy}, σ_v).
5. Object size at first occurence (w_1, h_1), last one (w_2, h_2), mean (μ_w, μ_h) and standard deviation (σ_w, σ_h).
6. Average color (layout) based on JPEG compression technique of 8x8 pixel object resampled in Y'CbCr color space, from which are zig-zag extracted DCT coefficients. We use 15 (Y) + 2*10 (Cb and Cr) coefficients ($c_{1..35}$).
7. Object shape using central normalized moments up to the third order (η_{20}, η_{11}, η_{02}, η_{30}, η_{21}, η_{12}, η_{03}) computed [4] from segmented image (alpha channel).

Because the shape moments do not give good classification results when aggregated (average), we have created a separate training case. The trajectory is split into 4 segments and their border shapes are extracted and concatenated into a feature vector. We refer this classification scheme as "Shape".

In case of HMMs, we have created an own implementation based on Viterbi and Baum-Welch algorithms as described below. The input of HMMs were trajectories and features described as above. The main disadvantage of HMMs is that both training and testing consumes incomparably more time than using aggregated Track and Shape features and SVM – about 100 times.

Because the VRM Analysis module uses various classifiers, we have adopted the fully-probabilistic combination of their results. For SED 2013, we used naïve Bayes combination of SVM and HMM as a first input to the interactive analysis. However, HMMs aren't able to be used in the interactive task because of their complexity.

Hidden Markov Models

The need of knowledge discovery in the trajectory data leads to the linear dynamic an Example d Markov models for the data classification. The presented approach is based on supervised learning and classification using HMMs of classes of behavior are created upon some annotated trajectories. A hidden Markov model (HMM, [6]) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states, similarly to dynamic Bayesian network. Example of an HMM is shown in figure 7. The sample model is described as a graph with four internal and two marginal states connected by (oriented) transitions. Moreover, there are six output vectors associated in the figure 7.

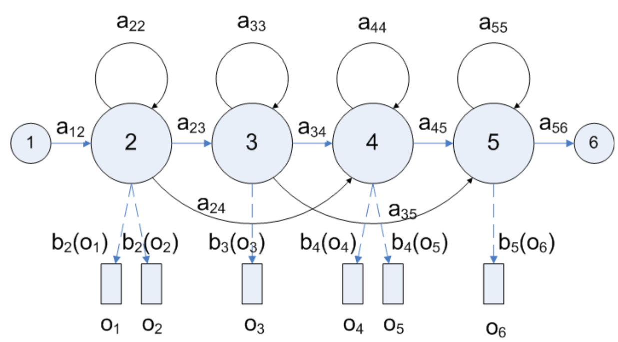


Fig. 7. Example configuration of a Hidden Markov Model [Mlich and Chmelar, 2008].

The trajectory classification problem can be formulated as to identify the class c_i ($i = 1..N$) to which belongs the trajectory state sequence. The basic formulation of the problem is given by maximization of a conditional probability:

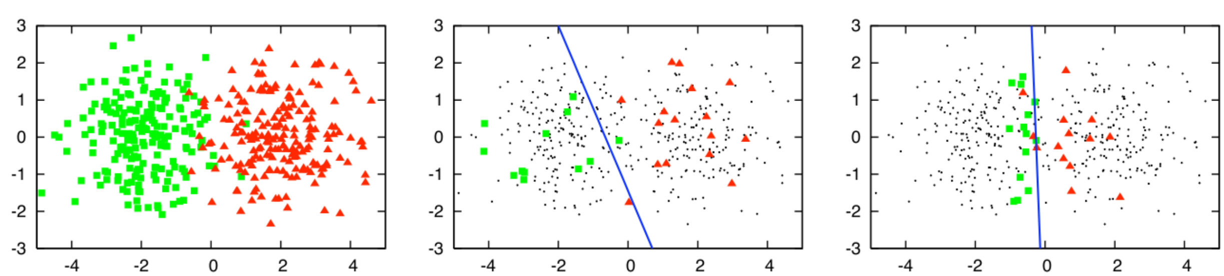


Fig. 8: Illustration of a pool-based active learning. It shows the advantage to the learning performance when annotated the same amount of samples in (b) and (c) out of (a) [Settles, 2009].

We use Bayes theorem, because we cannot evaluate $P(c_i | O)$ directly. Assuming we know prior probabilities $P(c_i)$ and $P(O)$, we are about to compute the likelihood $P(O | c_i)$; the probability of the sequence O knowing the class c_i. To compute this, we should have a model M for class c_i. The model is a finite state automaton with K states generating sequence O. There are transition probabilities $a_{k,j}$ between the states. Except the first and the last state, states are emitting or generating output probability density function $b_j(o(t))$, as illustrated in figure 7.

In the figure, there is a sample configuration of $A = [a_{k,j}]$ (k, $j = 1..K$), the transition matrix, which defines the probability of transition to the next state for each combination of HMM states. The corresponding sample HMM sequence or path through the model is $X = \{1, 2, 2, 3, 4, 4, 5, 6\}$. However, this information is from the view of the trajectory state sequence hidden. The probability of passing an object O through a model M by a way X is defined by:

$$P(O, X|M) = a_{x(o)x(1)} \prod_{t=1}^{T} b_{x(t)}(o_t) a_{x(t)x(t+1)}$$

Viterbi algorithm finds the most probable way through the model:

$$P^*(O|M) = \max_{\{X\}} P(O, X|M)$$

The algorithm is used to evaluate the model by maximizing probability of correspondence with a trajectory class. For training the model M_i, corresponding to the trajectory class c_i, the Baum-Welch algorithm is used. It is a generalized expectation-maximization algorithm defined by equation that modifies weights of transitions and statistics of the models [6]:

$$P(O|M) = \sum_{\{X\}} P(O, X|M)$$

Before the training process, the initial probabilities and number of states are chosen. The training itself is affected mainly by training step size. For the improvement of the classification performance, several initial setups were evaluated.

3.1 Active learning

We have adopted the principles of Active Learning (AL) in 2012. The systems attempt to overcome the labeling bottleneck by asking queries in the form of unlabeled instances to be labeled by an oracle (e.g. a human annotator). In this way, the active learner aims to achieve high accuracy using as few labeled instances as possible, thereby minimizing the cost of obtaining labeled data. Active learning is an optimal experimental design strategy. The hypothesis is that if the learning algorithm is allowed to choose the data from which it learns, it will perform better with less training [1].

An example of active learning is the pool-based active learning cycle. A learner may begin with a small number of instances in the labeled training set L and request labels for one or more carefully selected instances, learn from the query results, and then leverage its new knowledge to choose which instances to query next. An illustration of such process is in figure 2.

Several approaches to the active learning exists. One of the first active learning scenarios to be investigated is learning with membership queries. In this setting, the learner may request labels for any unlabeled instance in the input space, including (and typically assuming) queries that the learner generates de novo rather than those sampled from some underlying natural distribution [1]. The idea of synthesizing queries has also been extended to regression learning tasks, which is similar to the stream-based selective sampling. Other query strategies aim to the metric that should be minimized (or maximized) by the learner. For instance, it is entropy, expected model change, density weight, error rate and variance. For more detailed information see [1].

The approaches of AL may thus iterate to achieve a higher learner performance. Moreover, it can be supplemented by an unsupervised (clustering) or semi-supervised learner [2]. In this way, the annotator can mark only well discriminative centers of clusters (making sense) according to the requirements for instance. A survey on other semi-supervised learning methods can be found in [3].

The interactive interface

Human Monitoring Interface (HMI) of SUNAR-ED is based on active learning. It is capable of displaying events in addition to simple monitoring the area, but also querying monitored object(s) based on its previous occurrences, visual properties and behavior. The behavior is either a detected event or (statistical) analysis of the objects' spatio-temporal properties in the global context, such as who met who, where was a subject when something happened or some other nontrivial analysis based on statistics and data mining using VRM.

For SED 2013 we have even more simplified the user experience to make the annotations as simple as possible – a user can return to an event or see it in a context. We have created three types of the SUNAR-ED's Annotator as illustrated in figures 1 and 2. Moreover for sed 2013 we have created a validation interface, similar to the interactive one.

The first mode (annotator) was used for Round #1 annotations. The GUI shows the output of blob-track algorithm and cuts the shots where an event is expected accordingly to the LDC's annotations. The goal of the #1 is to match the event and the exact object for the learner. It can be done in two ways – either a human annotator can click the objects (subjects) that are concerned in the event or type their trajectory numbers when occluded.

Most events, however need just a single number, but Embrace, ObjectPut, PeopleMeet and PeopleSplitUp have eventually two or more objects involved. Thus, there are less cases of the event when there are multiple trajectories involved than shown in table 1.

Around 1000 events were annotated during Round #1 in 2012 – see table 1 for details. This took about 20 hours. We have performed the learning and classification of other objects within intervals specified in LDC's annotations. Accordingly to their probability (and grouped by videos for performance reasons) the HMI presented probable shots to the human annotator in the second mode. In this mode (figure 2), a human annotator is supposed press "1" (or more) if the event belongs to the object highlighted or "0" (or Enter) else. We have annotated about 1400 events in less than 6 hours.

Because of the simplicity, Round #2 annotations/validations were considered "extremely boring" in contrast to "just boring" Round #1 annotation. For the reason, the round #3 was split in four days and four rounds technically. The problem of the active learning in this case is that HMI it cannot present many positive samples as there are not enough in the training data.

Thus, for the purpose of evaluations - the 25 min. "interactive" annotations, the evaluation videos were played faster (150%) and because there was just one object marked including the whole trajectory, it was still well decidable (she considered it "high-dynamically boring"). See the "interactiveED" attachment for details.

We performed the trajectory clustering on a set of trajectories extracted from the the i-LIDS dataset of five cameras at the LGW airport. An example of visualization of some obtained results is shown in figure 6. Different colors of trajectories refer to different clusters. There is a result of clustering trajectories from the first camera using the EM algorithm. We have prepared also an outliers analysis within the Video Terror project.

4. Experiments

We have performed three rounds of the active learning process during the development and training using primarily SVM to be used as the model for retrospective runs, which was extended by HMM. Finally, we have performed the interactive run.

The table 1 presents the numbers of theoretical (LDC), Round #1 to #3 annotations after five to 25 hours burden for each task, which is the reason we haven't used all the annotations suggested (coping with the unsatisfactory tracker results of heavily occluded objects and the overall quality of data, because some events take just a few pixels of the screen).

Table 1. Numbers of annotated objects (active learning).

Event	#LDC	#1	#2	#3
CellToEar	828	80	120	270
ElevatorNoEntry	12	4	5	13
Embrace	940	75	138	530
ObjectPut	3172	181	422	1312
OpposingFlow	34	1	4	9
PeopleMeet	2718	282	717	2007
PeopleSplitUp	1571	122	441	1007
PersonRuns	673	59	153	398
Pointing	4095	235	478	1334
TakePicture	30	0	0	3
Sum (distinct)	< 14073	944	2280	6194

Table 2 presents the SVM – based classification accuracy of optimized classification schemes "Track" and "Shape" as described in section 3.1 - Analysis and classification. Note, that Round #3 classification data is of about 1GB and the whole database is about 20GB (compared to 300GB video data). A single learning process of an average classification model is about 20 seconds. We performed 5 (fold) * 100 (parameter selection) learning processes, which takes about 30 minutes, performed 9 times in parallel. We have considered 9 distinct events – omitting the TakePicture, because we were unable to asses who is taking the picture in the devel recordings.

In case of HMMs, we haven't used the aggregated values for them, neither we used them for active learning, because the training took about 46 hours in contrast to 30 minutes of SVM. Their performance can be compared in the attachment and it seems that SVM was little over-trained in comparison to HMMs, because they performed little better.

Table 2. Prediction accuracy of 5-fold cross-validation on training data.

	SVM		HMM
Event	Tracks	Shapes	Both
CellToEar	91.52	91.53	46.27
ElevatorNoEntry	99.79	99.68	98.15
Embrace	92.05	92.06	59.88
ObjectPut	81.25	81.03	45.86
OpposingFlow			99.86
PeopleMeet	71.93	70.55	49.65
PeopleSplitUp	87.29	87.08	35.49
PersonRuns	94.28	93.75	58.02
Pointing	76.27	75.21	60.99
TakePicture			93.20
Average	86.80	86.36	64.74

5. Conclusions

In the paper, we presented an open source multi-camera computer vision based surveillance event detection system SUNAR-ED (see sourceforge.net/p/sunar-ed). We have selected, integrated and extended a set of a state of the art progressive and robust tools efficient for multimedia data and related metadata storage, indexing, retrieval and analysis. In 2013 year we have focused on Hidden Markov Models and classifier fusion mainly.

SUNAR-ED is composed of three basic modules – video processing, retrieval and the monitoring interface. Computer Vision Module is based on the OpenCV library for moving object discovery and single camera tracking with collision resolving. It is based on custom background subtraction and it provides color and shape descriptors similarly to MPEG-7 in addition to the trajectory data of objects and subjects.

The information about objects and the area under surveillance is cleaned, integrated, indexed and stored in Video Retrieval Modules. They are based on the PostgreSQL database extended to be capable of similarity and spatio-temporal information retrieval. For the purpose of analysis, we used machine-learning methods as SVM, EM/GMM, HMM and Bayes classifiers for their fusion and prepared frequent pattern analysis and various clustering algorithms for experiments.

Similarly to 2012, we have used active learning and semi-automatic annotation generation including localization for future evaluations using SVM and HMM as learners and their fusion.

We have improved a simple (yet boring) user interface, which can reduce the burden of continuous concentration on monitoring and increase the effectivity.

Together with SUNAR-ED, we offer to the public the data and metadata management framework – VTApi (application programming interface, see gitorious.org/vtapi). The main advantages of the API is the reduction of effort and time to produce quality intelligent vision applications by unified and reusable both methods and data sets of video, image, metadata and features on all levels. Using VTApi, we have developed tools to be (re)used in the future to unify and accelerate vision research.

We also offer data, methods and methodology to be used by researchers and developers of both academic and commercial sectors to collaborate and chain their efforts, especially to other SED participants.

We have to thank all the people of NIST and groups providing data, annotations, evaluation metrics and all the human power that encourages our research and development [13].

References

[1] B. Settles, *Active Learning*. Morgan & Claypool Publishers, 2012.

[2] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques: Concepts and Techniques*. Elsevier, 2011.

[3] X. Zhu and A. B. Goldberg, “Introduction to Semi-Supervised Learning,” *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 3, no. 1, pp. 1–130, Jan. 2009.

[4] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*, 1st ed. O’Reilly Media, 2008.

[5] W. Hu, T. Tan, L. Wang, and S. Maybank, “A survey on visual surveillance of object motion and behaviors,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 34, no. 3, pp. 334 –352, Aug. 2004.

[6] C. M. Bishop, *Pattern Recognition and Machine Learning*, 1st ed. Springer, 2007.

[7] E. J. Carmona, J. Martínez-Cantos, and J. Mira, “A new video segmentation method of moving objects based on blob-level knowledge,” *Pattern Recogn. Lett.*, vol. 29, no. 3, pp. 272–285, Feb. 2008.

[8] P. Chmelar, I. Rudolfova, and J. Zendulka, “Clustering for Video Retrieval,” in *Data Warehousing and Knowledge Discovery*, 2009, pp. 390–401.

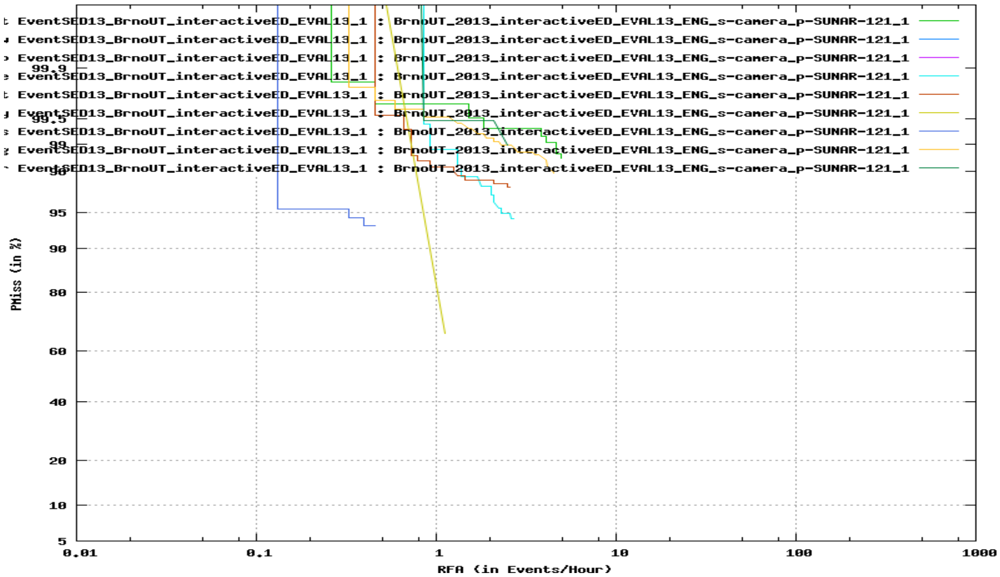
[9] P. Chmelar and J. Zendulka, “Visual Surveillance Metadata Management,” in *Eighteenth International Workshop on Database and Expert Systems Applications*, 2007, pp. 79–83.

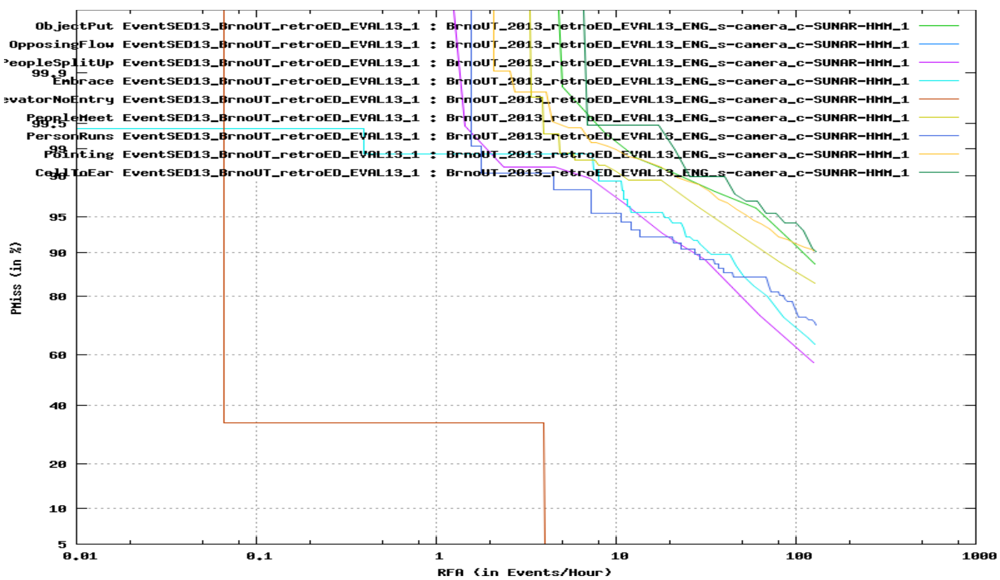
[10] B. S. Manjunath, P. Salembier, and T. Sikora, *Introduction to MPEG-7: Multimedia Content Description Interface*. John Wiley & Sons, 2002.

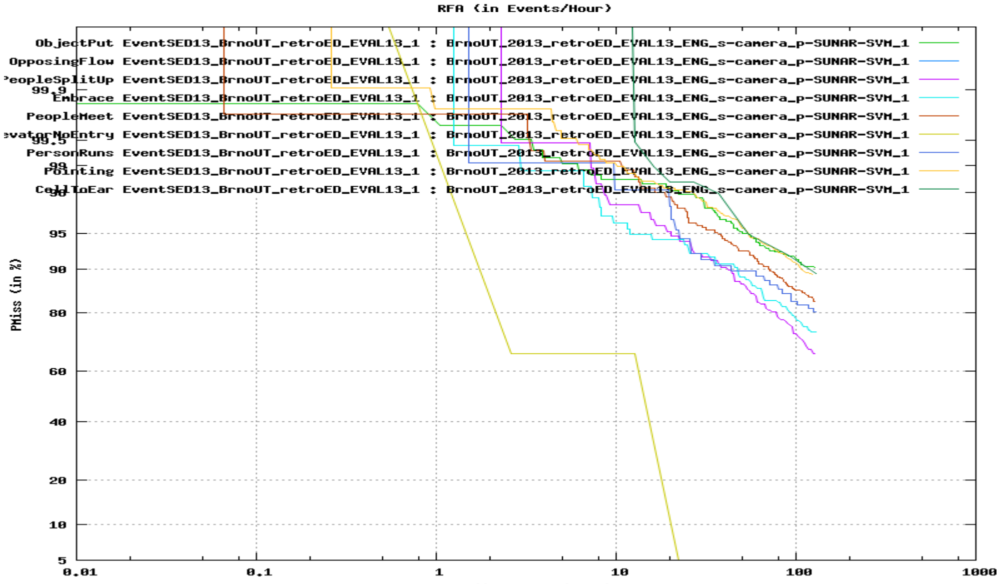
[11] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 27:1–27:27, 2011.

[12] P. Chmelar, A. Lanik, and J. Mlich, “SUNAR: Surveillance Network Augmented by Retrieval,” in *ACIVS 2010*, 2010, pp. 155–166.

[13] P. Over, G. Awad, M. Michel, J. Fiscus, G. Sanders, W. Kraaij, A. F. Smeaton, and G. Quéenot, “TRECVID 2013 – An Overview of the Goals, Tasks, Data, Evaluation Mechanisms and Metrics,” in Proceedings of TRECVID 2013, 2013.


EventSED13_BrnoUT_interactiveED_EVAL13_1 : BrnoUT_2013_interactiveED_EVAL13_ENG_s-camera_p-SUNAR-121_1
PMiss (in %)
RFA (in Events/Hour)


ObjectPut EventSED13_BrnoUT_retroED_EVAL13_1 : BrnoUT_2013_retroED_EVAL13_ENG_s-camera_c-SUNAR-HMM_1
OpposingFlow EventSED13_BrnoUT_retroED_EVAL13_1 : BrnoUT_2013_retroED_EVAL13_ENG_s-camera_c-SUNAR-HMM_1
PeopleSplitUp EventSED13_BrnoUT_retroED_EVAL13_1 : BrnoUT_2013_retroED_EVAL13_ENG_s-camera_c-SUNAR-HMM_1
Embrace EventSED13_BrnoUT_retroED_EVAL13_1 : BrnoUT_2013_retroED_EVAL13_ENG_s-camera_c-SUNAR-HMM_1
ElevatorNoEntry EventSED13_BrnoUT_retroED_EVAL13_1 : BrnoUT_2013_retroED_EVAL13_ENG_s-camera_c-SUNAR-HMM_1
PeopleMeet EventSED13_BrnoUT_retroED_EVAL13_1 : BrnoUT_2013_retroED_EVAL13_ENG_s-camera_c-SUNAR-HMM_1
PersonRuns EventSED13_BrnoUT_retroED_EVAL13_1 : BrnoUT_2013_retroED_EVAL13_ENG_s-camera_c-SUNAR-HMM_1
Pointing EventSED13_BrnoUT_retroED_EVAL13_1 : BrnoUT_2013_retroED_EVAL13_ENG_s-camera_c-SUNAR-HMM_1
CellToEar EventSED13_BrnoUT_retroED_EVAL13_1 : BrnoUT_2013_retroED_EVAL13_ENG_s-camera_c-SUNAR-HMM_1
PMiss (in %)
RFA (in Events/Hour)


ObjectPut EventSED13_BrnoUT_retroED_EVAL13_1 : BrnoUT_2013_retroED_EVAL13_ENG_s-camera_p-SUNAR-SVM_1
OpposingFlow EventSED13_BrnoUT_retroED_EVAL13_1 : BrnoUT_2013_retroED_EVAL13_ENG_s-camera_p-SUNAR-SVM_1
PeopleSplitUp EventSED13_BrnoUT_retroED_EVAL13_1 : BrnoUT_2013_retroED_EVAL13_ENG_s-camera_p-SUNAR-SVM_1
Embrace EventSED13_BrnoUT_retroED_EVAL13_1 : BrnoUT_2013_retroED_EVAL13_ENG_s-camera_p-SUNAR-SVM_1
PeopleMeet EventSED13_BrnoUT_retroED_EVAL13_1 : BrnoUT_2013_retroED_EVAL13_ENG_s-camera_p-SUNAR-SVM_1
ElevatorNoEntry EventSED13_BrnoUT_retroED_EVAL13_1 : BrnoUT_2013_retroED_EVAL13_ENG_s-camera_p-SUNAR-SVM_1
PersonRuns EventSED13_BrnoUT_retroED_EVAL13_1 : BrnoUT_2013_retroED_EVAL13_ENG_s-camera_p-SUNAR-SVM_1
Pointing EventSED13_BrnoUT_retroED_EVAL13_1 : BrnoUT_2013_retroED_EVAL13_ENG_s-camera_p-SUNAR-SVM_1
CellToEar EventSED13_BrnoUT_retroED_EVAL13_1 : BrnoUT_2013_retroED_EVAL13_ENG_s-camera_p-SUNAR-SVM_1
PMiss (in %)
RFA (in Events/Hour)