Autonomous Learning of a Robust Background Model for Change Detection *

Helmut Grabner Peter M. Roth Michael Grabner
Horst Bischof
Graz University of Technology
Institute for Computer Graphics and Vision
Inffeldgasse 16/II, 8010 Graz, Austria

Abstract

We propose a framework for observing static scenes that can be used to detect unknown objects (i.e., left luggage or lost cargo) as well as objects that were removed or changed (i.e., theft or vandalism). The core of the method is a robust background model based on on-line AdaBoost which is able to adapt to a large variety of appearance changes (e.g., blinking lights, illumination changes). However, a natural scene contains foreground objects (e.g., persons, cars). Thus, a detector for these foreground objects is automatically trained and a tracker is initialized for two purposes: (1) to prevent that a foreground object is included into the background model and (2) to analyze the scene. For efficiency reasons it is important that all components of the framework are using the same efficient data structure. We demonstrate and evaluate the developed method on the PETS 2006 sequences as well as on own sequences of surveillance cameras.

1. Introduction

For most video surveillance systems a foreground/background segmentation is needed (at least in the very beginning). Usually this segmentation is obtained by first estimating a robust background model and second by thresholding the difference image between the current frame and the background model. Such methods are referred as background subtraction methods. Let $\mathbf{B}_t$ be the current estimated background image, $\mathbf{I}_t$ the current input image and θ a threshold, then a pixel is classified as foreground if

$$|\mathbf{B}_t(x,y) - \mathbf{I}_t(x,y)| > \theta. \quad (1)$$

*This work has been supported by the Austrian Federal Ministry of Transport, Innovation and Technology under P-Nr. I2-2-26p VITUS2 and by the Austrian Joint Research Project Cognitive Vision under projects S9103-N04 and S9104-N04. In addition this work has been sponsored by the MISTRAL Project which is financed by the Austrian Research Promotion Agency (www.ffg.at) and the EU FP6-507752 NoE MUSCLE IST project.

(a) (b)

Figure 1: A background model is used for detecting foreground objects in a static scene (colored boxes in the right image). The automatic learning of allowed objects (yellow) allows the detection of unknown objects (red).

As for realistic applications different environmental conditions (e.g., changing lightening conditions or foreground models moving to background and vica versa) have to be handled. Therefore, several adaptive methods for estimating a background model $\mathbf{B}_t$ have been proposed that update the existing model with respect to the current input frame $\mathbf{I}_t$ (e.g., running average [8], temporal median filter [10], approximate median filter [11]). A more detailed discussion of these different background models can be found in [1,6,15].

But all of these methods have two drawbacks: First, as the foreground objects may have a similar color as the background, these objects can not be detected by thresholding. Second, these methods are only slowly adapting to slightly changing environmental conditions. Thus, faster changes as a flashlight signal or fluttering leaves in the wind can not be modeled. To overcome this problem a multi-modal model such as Mixture of Gaussian [20], Eigenbackgrounds [13] or more efficiently by analyzing of foreground models [21] can be applied.

For left luggage detection this simple segmentation process has to be extended. We are not only interested in a foreground/background segmentation but we are also interested in differentiating between *known objects* (e.g., persons) and *unknown objects* (e.g., left luggage). Thus, we propose a

framework that combines a background model and a known object identifier. An object detector is trained and the obtained detections are eliminated from an already computed foreground/background segmentation. As a result we obtain regions that can not be modeled and may represent unknown objects. An example is depicted in Figure 1. The small red and blue squares represent regions that can not be explained by the background model. By using a person detector (yellow bounding boxes) the regions according to the blue boxes are recognized as parts of a person. All other regions (red boxes) can not be explained and are therefore unknown objects.

The background model is learned by observing a scene from a static camera and by learning everything that is present in the scene by on-line AdaBoost [4]. Thus, all dynamic changes (even moving objects) are assumed to be normal and are therefore learned as an allowed mode. The major benefit of such a background model is its capability to adapt to any background and its ability to model even complex scenes (e.g., containing dynamic background such as blinking lights). Furthermore, it allows to adapt to continuous changes (e.g., illumination changes in outdoor scenes) while observing the scene.

The object detector is trained by Conservative Learning [17,18] and is applied for two purposes. First, the detection results are used to distinguish between relevant changes (e.g., left luggage, lost cargo, etc.) and natural occurrences (e.g., walking persons). Second, as the background model can be updated on-line it is used to define an update policy. Thus, image areas where a foreground object was detected can be excluded from the update process and a background model can be estimated even if foreground objects are present during the learning stage. To increase the stability of the detector a tracker is used and an object is even detected if there are larger changes in its appearance.

The proposed framework can be applied to detect (dynamic as well as non-dynamic) changes in a static scene. Thus, unknown objects (i.e., left luggage or lost cargo) as well as objects that were removed or changed (i.e., theft or vandalism) can be detected. As the known-object detector can be trained without any user interaction directly from video data and the background model is estimated automatically we have a fully automatic framework.

The outline of the paper is as follows: First, the framework is introduced and the different modules are described in Section 2. Next, experiments and results are shown in Section 3. Finally, the paper is summarized in Section 4.

2. Video Surveillance Framework

The main components of the framework (see Figure 2) are a robust block based background model and a known-object identifier (detector and tracker). First, a foreground object detector is trained for "known objects" by Conservative Learning [17, 18]. To be more robust additionally a tracker [5] is initialized when a known object is detected in the scene for the first time. Next, the background model is estimated by observing the scene assuming that all input frames contain only (even changing) background. When a first model was estimated new input frames are evaluated. All non-background regions are detected and verified by the detection results. Thus, only regions are reported that can not be explained by any of the models. Finally, in a post-processing step only detections are considered that are stable over time.

In addition, the background model is updated for every new frame. To avoid that foreground objects are included into the background model a special update policy is defined (detection results are used to define areas where no updates should be performed for some time).

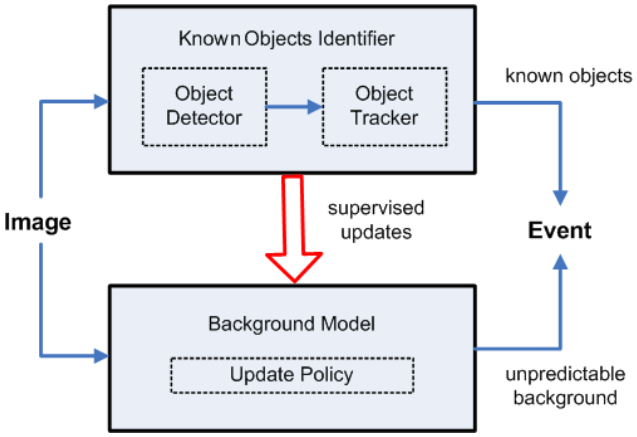


Figure 2: Overview of the proposed framework.

All modules (background model, detector and tracker) are based on the same type of classifier that is trained using the same features. In particular we apply Haar-like features [23], orientation histograms (with 16 bins) similar to [2, 9] and a simplified version (4-th neighborhood) of local binary patterns (LBP) [12]. Using integral data structures the features can be estimated very efficiently [16] because this data structure has to be computed only once for all modules. To have an efficient system we also need an estimate of the ground plane which can be automatically done by, e.g., [14].

2.1. Background Model

For estimating the change detection we apply a new *classifier-based* background model [4] that is based on the idea of a block based background model [7]. Thus, the gray-scale image is partitioned into a grid of small (overlapping) rectangular blocks (patches). For each of them a separate classifier is computed by combining the image features that

were described in the Section 2. For training the classifiers we use boosting for feature selection from this highly over complete representation (each feature prototype can appear at different positions and scales). The overall principle is depicted in Figure 3.

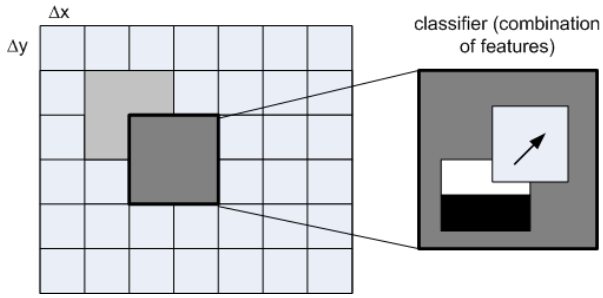


Figure 3: The background model is formed by a grid of regular aligned classifiers with an overlap of $\Delta x = \Delta y = 50\%$. Each cell is represented by a strong classifier obtained by boosting which combines several weak classifiers based on visual features. Efficient computation is achieved by using fast computable features via integral structures.

In general Boosting (see [3] for a good introduction) is a widely used technique in machine learning for improving the accuracy of any given learning algorithm. In fact, boosting converts a weak learning algorithm into a strong one. Therefore, for an input vector $\mathbf{x}$ a strong classifier $h^{strong}(\mathbf{x})$ is computed as linear combination of a set of N weak classifiers $h_n^{weak}(\mathbf{x})$:

$$h^{strong}(\mathbf{x}) = \text{sign}(conf(\mathbf{x})) \tag{2}$$

$$conf(\mathbf{x}) = \frac{\sum_{n=1}^{N} \alpha_n \cdot h_n^{weak}(\mathbf{x})}{\sum_{n=1}^{N} \alpha_n} \tag{3}$$

A weak classifiers is a classifier that has to perform only slightly better than random guessing, i.e., for a binary decision task, the error rate must be less than 50%. As $conf(\mathbf{x})$ is bounded by $[-1, 1]$ it can be interpreted as a confidence measure (which is related to the margin). The higher the absolute value, the more confident is the result.

Boosting for feature selection was first introduced by Tieu and Viola [22] and has been widely used for different applications (e.g., face detection [23]). The main idea is that each feature corresponds to a single weak classifier and boosting selects from these features. Given a set of possible features in each iteration step n all features are evaluated. The best one is selected and forms the weak hypothesis h_n^{weak} which is added to the final strong classifier h^{strong}.

This process as described above works off-line. Thus, all training samples must be given in advance. But for learning a dynamic background model we need an on-line learning method that can adept to changing environmental conditions as new frames arrive. Thus, we apply an on-line version of boosting for feature selection [4]. The main idea is to introduce "selectors" and to perform boosting on these selectors and not directly on the weak classifiers. Each selector $h^{sel}(\mathbf{x})$ holds a set of M weak classifiers $\{h_1^{weak}(\mathbf{x}), \ldots, h_M^{weak}(\mathbf{x})\}$ and selects one of them according to an optimization criterion based on the estimated error e_i of the classifier h_i^{weak}:

$$h^{sel}(\mathbf{x}) = h_m^{weak}(\mathbf{x}), \quad m = \arg\min_i(e_i) \tag{4}$$

Moreover, the importance/difficulty of a sample is estimated by propagating it through the set of selectors. For more details see [4]. But since the creation of weak classifiers is very important for our specific task this step is described explicitly in this paper.

For on-line learning a weak classifier h_j^{weak} for a feature j we first build a model by estimating the probability $P(1|f_j(\mathbf{x})) \sim \mathcal{N}(\mu^+, \sigma^+)$ for positive labeled samples and $P(-1|f_j(\mathbf{x})) \sim \mathcal{N}(\mu^-, \sigma^-)$ for negative labeled samples, where $f_j(\mathbf{x})$ evaluates this feature on the image $\mathbf{x}$. The mean and variance are incrementally estimated by applying a Kalman-filtering technique. Next, to estimate the hypothesis for the Haar-Wavelets we use either simple thresholding

$$h_j^{weak}(\mathbf{x}) = p_j \cdot \text{sign}(f_j(\mathbf{x}) - \theta_j), \tag{5}$$

where

$$\theta_j = |\mu^+ + \mu^-|/2, \quad p_j = \text{sign}(\mu^+ - \mu^-) \tag{6}$$

or a Bayesian decision criterion

$$\begin{aligned} h_j^{weak}(\mathbf{x}) &= \text{sign}(P(1|f_j(\mathbf{x})) - P(-1|f_j(\mathbf{x}))) \\ &\approx \text{sign}(g(f_j(\mathbf{x}|\mu^+, \sigma^+) - g(f_j(\mathbf{x})|\mu^-, \sigma^-)), \end{aligned} \tag{7}$$

where $g(x|\mu, \sigma)$ is a Gaussian probability density function. For histogram features (orientation histograms and LBPs), we use nearest neighbor learning with a distance function D (e.g., Euclidean):

$$h_j^{weak}(\mathbf{x}) = \text{sign}(D(f_j(\mathbf{x}), \mathbf{p}_j) - D(f_j(\mathbf{x}), \mathbf{n}_j)) \tag{8}$$

The cluster centers for positive $\mathbf{p}_j$ and negative $\mathbf{n}_j$ samples are learned by estimating the mean and the variance for each bin separately.

For the application of background modeling we *do not have negative examples*. But, we can treat the problem as an one-class classification problem and use only positive samples for updating. The key idea is to calculate the negative distribution for each feature directly without learning. We model the gray value of each pixel as uniformly distributed

with mean 128 and variance $\frac{256^2}{12}$ (for an 8 bit image). Applying standard statistics the parameters of the negative distribution μ^- and σ^- of Haar features can be easily computed. For orientation histogram features the negative cluster $\mathbf{n}_j$ consists of equally distributed orientations. The characteristic negative cluster for a 16 bin LBP-feature is given by $\mathbf{n}_j = \frac{1}{50} \cdot [6, 4, 4, 1, 4, 1, 1, 4, 4, 1, 1, 4, 1, 4, 4, 6]$ for the binary patterns $[0000_2, 0001_2, 00010_2..., 1111_2]$[1].

Thus, we are able to compute the weak classifiers and use them for predicting the background (including statistical predictable changes). In the initial learning stage a separate classifier is built for all image patches assuming that all input images are positive examples. Later on, new input images are analyzed and the background model is updated according to a given policy.

Evaluation

A region is labeled as foreground if it can not be modeled by the classifier, i.e., the obtained confidence of the classifier is below a certain threshold:

$$conf(\mathbf{x}) < \theta^{eval} \tag{9}$$

Update

For updating the classifiers we adopt the following very simple policy. We update a classifier if its confidence response is within a certain interval:

$$\theta_{lower}^{update} < conf(\mathbf{x}) \leq \theta_{upper}^{update} \tag{10}$$

Usually $\theta_{lower}^{update} = \theta^{eval}$ and the upper threshold is set to avoid over-fitting. In addition, in the post-processing steps several regions (known objects) are excluded from updating for a certain time.

Due to the specific type of features used for training the classifiers the background model is high sensitive and even small changes can be detected. Moreover, the proposed background model is capable of modeling dynamically changing backgrounds (e.g., flashing lights, moving leaves in the wind, flag waving, etc.). Since an efficient data structure (integral image) is used the evaluation can be implemented very efficiently

2.2. Known-object Identifier

Object Detector

For automatically learning a person model we apply Conservative Learning [17, 18]. Starting with motion detection an initial set of positive examples is obtained by analyzing the geometry (aspect ratio) of the motion blobs. If a blob fulfills the restrictions the corresponding patch is selected. Negative examples are obtained from images where no motion was detected. Using these data sets a first discriminative classifier is trained using an on-line version of AdaBoost [4]. In fact, by applying this classifier all persons are detected (a general model was estimated) but there is a great amount of false positives. Thus, as a generative classifier robust PCA [19] is applied to verify the obtained detections and to decide if a detected patch represents a person or not. The detected false positives are fed back into the discriminative classifier as negative examples and the true positives as positive examples. As a huge amount of data (video stream) is available very conservative thresholds can be used for these decisions. Thus, most of the patches are not considered at all. Applying these update rules an incrementally better classifier is obtained. Moreover, an already trained classifier can be re-trained on-line and can therefore easily be adapted to a completely different scene. As the framework is very general we can apply it to learn a person detector as well as to learn a model for cars. For the whole procedure no user interaction is needed.

Object Tracker

After a target object was successfully detected a tracker is initialized. In particular we apply a tracker [5] that is based on on-line boosting [4] similar to the background classifier. First, to initialize the tracker, a detected image region is assumed to be a positive image sample. At the same time negative examples are extracted by taking regions of the same size as the target window from the surrounding background. Using these samples several iterations of the on-line boosting algorithm are carried out. Thus, the classifier adapts to the specific target object and at the same time it is discriminating against its surrounding background. The tracking step is based on the approach of template tracking. We evaluate the current classifier on a region of interest and obtain a confidence value for each sub-patch. We analyze the confidence map and shift the target window to the (new) location where the confidence value is maximized. Once the object has been tracked the classifier has to be updated in order to adjust to possible changes in appearance of the target object and its background. The current target region is used as a positive update of the classifier while the surrounding regions represent the negative samples. As new frames arrive the whole procedure is repeated. The classifier is therefore able to adapt to changes in appearance and in addition it becomes robust against background clutter. Note that the classifier focuses on the current target object while at the same time it attempts to distinguish the target from its surrounding. Moreover, tracking of multiple objects is feasible just by initializing a separate classifier for each target object.

[1] These numbers are obtained by a lengthly calculation assuming equal probability of all patches.

3. Experimental Results

To show the power of the approach we applied the proposed framework on three different scenarios. The first experiment was carried out on the PETS 2006 Benchmark Data that is publicly available[2]. But as this paper is mainly focused on change detection (left objects as well as objects that were removed) by using a robust background model the detection of left luggage (briefcase, bag, etc.) is not limited to the luggage that was left by a certain person. In addition, we have created various sequences showing a corridor in a public building and a tunnel. Each of the sequences demonstrates a special difficulty that can be handled by our framework.

To obtain comparable results we have used the same parameter settings for all experiments. For training the classifiers (background, detector) the size of the pool of weak classifiers was limited to 250 for all modules. The classifier for the tracker and the background model was estimated from a linear combination of 30 weak classifiers (selectors) whereas for the more exact detector 50 weak classifiers where used. The search region for the tracker was set twice as large as the current object dimension. For estimating the background model patches of 20×20 pixels with 50% overlapp were defined. The evaluation threshold was set to $\theta^{eval} = 0$ and for the update policy the parameters $\theta^{update}_{lower} = 0$ and $\theta^{update}_{upper} = 0.5$ were used. For our experiments we achieve a frame-rate of 5 to 10 frames per second on an 1.6 GHz PC with 1 GB RAM.

3.1. PETS 2006 Benchmark Data

First, we demonstrate our approach on the publicly available PETS 2006 Benchmark Data with supplied ground truth. Therefore, we have selected two sequences from different camera positions[3]. We have chosen these special sequences to show that we are not limited to a certain camera position or camera geometry. In addition, we can demonstrate that our background model can handle the occurrence of shadows and reflections.

The first sequence was taken from a "side view". People are walking around and a man is entering the scene and leaves a suitcase behind. Typical frames of this scene are shown in Figure 4(a). Thus, the defined task was to detect the suitcase. Therefore, first the suitcase as well as the persons are detected as non-background objects (Figure 4(b)). Second, the "known-object identifier" (detector/tracker) detects all known objects, i.e., the persons walking or standing around (Figure 4(c)). Finally, when combining the results from the background model and the detections obtained by the tracker only the suitcase is detected (Figure 4(d)). In addition to the "known-object identifier" a region growing algorithm is applied. All non-background patches that are within the same region as a detected person are assumed to be part of the person. Thus, an outstretched arm or a drawn suitcase (see Figure 4, second row) are recognized as a part of a person and are therefore not labeled as unknown objects. Since the current implementation of the person detector can not detect partial persons (person entering or leaving the scene) such patches were not considered at all.

The second sequence taken from a "semi frontal view" is more difficult for our method because persons and other unknown objects (luggage) are present in the scene from the very beginning (see Figure 5). As can be seen in Figure 5(b-d) (second row) persons are not a problem. If a person was detected the corresponding region is not used for updating the background model. Thus, the person in not included in the background. When combining the results of both modules the person is not detected any more. In contrast to persons unknown objects are modeled as background. The same applies for persons that are not detected (e.g., children may not detected due to the restriction from the calibrated ground plane). But all other objects, i.e., the ski-sack near to the glass wall, the bag and even the newspaper on the bench were detected as left luggage!

For evaluation we analyzed the detections of the suitcase (Sequence 1) and of the ski-bag (Sequence 2). Thus, the true positives and the false positives were counted starting with the first occurrence of the left object (Sequence 1: frame 1210/3400, Sequence 2: frame 1962/2400). The results are summarized in Table 1.

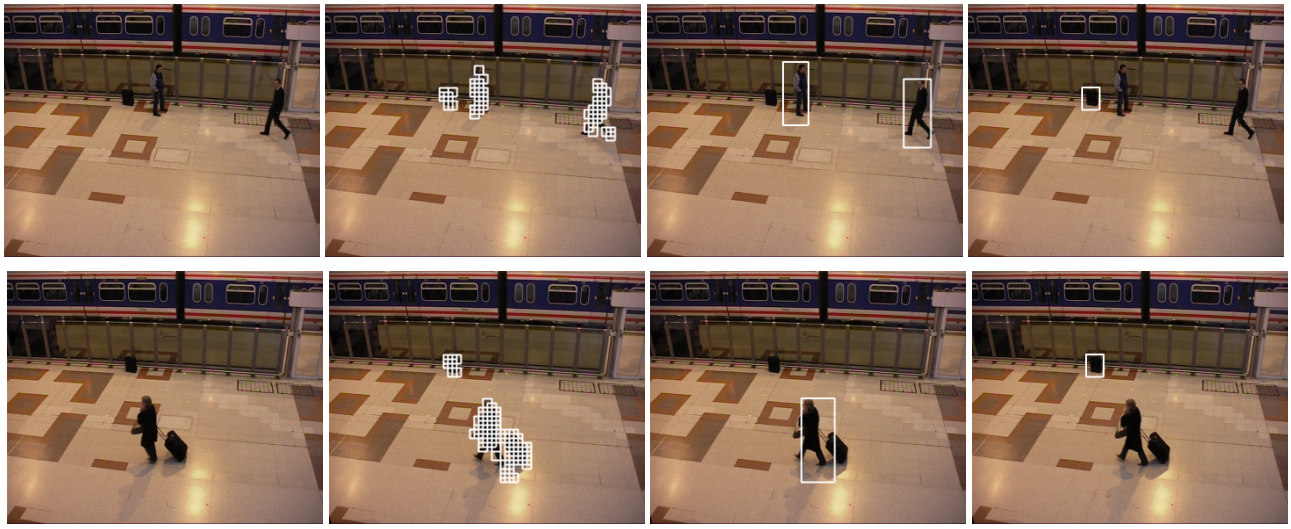
sequence	true pos.	false pos.
PETS Seq. 1	91.5%	3.4%
PETS Seq. 2	96.6%	1.9%

Table 1: Evaluation Results for the PETS 2006 Sequences.

For both sequences we obtain a detection rate of more than 90%, where most of the misses arise from non-background regions growing together (e.g., a person is passing by a left object very closely) which can be easily avoided by temporal filters. Thus, the detection rate for the second sequence is much better compared to the first sequence since less persons are passing by the left object very close. The false positives are the result of temporary unstable detections (e.g., a thrown suitcase or a cast shadow is detected as left luggage) or may be caused by changes in background that were not learned during the training stage. But by using simple logic and time constraints the number of false alarms can be reduced.

[2] http://www.pets2006.net, April 25th, 2006

[3] Dataset S7 (Take 6-B), Camera 3 and Dataset S5 (Take 1-G), Camera 4.



(a) (b) (c) (d)

Figure 4: PETS 2006 Dataset - Sequence 1: (a) original image, (b) background patches that can not be explained by the background model, (c) detected (tracked) persons, (d) finally detected the suitcase.

(a) (b) (c) (d)

Figure 5: PETS 2006 Dataset - Sequence 2: (a) original image, (b) background patches that can not be explained by the background model, (c) detected (tracked) persons, (d) finally detected the ski-sack (and in addition the bag and the newspaper on the bench).

(a) (b) (c) (d) (e)

Figure 6: CoffeeCam / Larceny Scenario - Changes in the background model: (d) the missing poster (object removed) and (e) the poster lying on the floor (object added) are detected.

(a) (b) (c) (d)

Figure 7: Tunnel Safety - Objects that were thrown out of the car are detected: (a) no object, (b) chock, (c) car tire, (d) safety cone.

3.2. CoffeeCam

Next, we demonstrate that our framework can cope with dynamic backgrounds. Therefore, we have taken several sequences showing a corridor in a public building near to a coffee dispenser. The dynamic background was simulated by using a flashlight. Several typical scenarios including left luggage and the larceny of paintings were defined and evaluated.

In the following the results obtained for the larceny of paintings scenario are presented. After a background model was learned from dynamically changing background images (blinking alarm light) the corridor was kept under surveillance. To simulate the larceny the poster was removed by one person and thrown down to the floor. Figure 6 shows the detection results of five consecutive frames. In the beginning, persons are walking around and nothing suspicious is detected (Figure 6(a-b)). Then, the poster is removed but the changed background area is occluded by the person (Figure 6(c)). Finally, the missing poster (Figure 6(d)) as well as the poster lying on the floor (Figure 6(e)) are detected.

3.3. Tunnel Safety

Finally, to show the generality of our approach we demonstrate the method on a tunnel safety task. In addition, we show that we can also detect objects of low contrast that would not be detected by using a standard approach. Figure 8 shows the first (a) and the last (b) frame of a test sequence. It is even hard for a human to detect all three objects that were thrown out of the car! Moreover, each of the objects (a chock, a car tire and a safety cone) has a size of only a few pixels. Due to lights of cars, warning lights of trucks etc. the background is changing over time; a dynamic multi-modal background model is needed to robustly detect changes. In fact, our framework handles both, the dynamic background and the low contrast video data. Thus, detection results of four subsequent frames are shown in Figure 7.

(a) (b)

Figure 8: Tunnel Safety: Due to lightening conditions and to low quality cameras the contrast is very low.

4. Summary and Conclusion

We have presented a framework for detecting changes in the background. Thus, we are able to detect unknown objects or objects that were removed. A new robust background model that is based on on-line learning feature based classifiers and an object detector (tracker) are combined. Thus, detected changes are verified by the detector and all regions that can not be explained are returned as unknown foreground objects. In addition, detected regions are excluded from updating. Thus, a background model can be learned even if (known) foreground objects are present in the scene. The proposed background model is very sensitive, i.e., even objects in low contrast images are detected. In addition, it can handle multi-modalities, i.e., dynamic changes in the background. As for all components the same data structures (integral representations) are used the whole framework can be implemented in a very efficient way. Moreover, all components run in an unsupervised manner.

References

[1] S.-C. S. Cheung and C. Kamath. Robust techniques for background subtraction in urban traffic video. In *Proc. SPIE Visual Communications and Image Processing*, pages 881–892, 2004.

[2] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 886–893, 2005.

[3] Y. Freund and R. Schapire. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 14(5):771–780, 1999.

[4] H. Grabner and H. Bischof. On-line boosting and vision. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2006. (in press).

[5] M. Grabner, H. Grabner, and H. Bischof. Real-time tracking with on-line feature selection. In *Video Proc. for IEEE Conf. on Computer Vision and Pattern Recognition*, 2006. (in press).

[6] D. Hall, J. Nascimento, P. C. R. E. Andrade, P. Moreno, S. P. T. List, R. Emonent, R. B. Fisher, J. Santos-Victor, and J. L. Crowley. Comparision of target detection algorithms using adaptive background models. In *Proc. IEEE Workshop on VS-PETS*, pages 113–120, 2005.

[7] M. Heikkilä and M. Pietikäinen. A texture-based method for modeling the background and detecting moving objects. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28:657 – 662, 2006.

[8] D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B.Rao, and S. Russell. Towards robust automatic traffic scene analysis in real-time. In *Proc. Intern. Conf. on Pattern Recognition*, volume I, pages 126–131, 1994.

[9] K. Levi and Y. Weiss. Learning object detection from a small number of examples: The importance of good features. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 53–60, 2004.

[10] B. Lo and S. A. Velastin. Automatic congestion detection system for underground platforms. In *Proc. IEEE Intern. Symposium on Intelligent Multimedia , Video and Speech Processing*, pages 158–161, 2001.

[11] N. J. McFarlane and C. P. Schofield. Segmentation and tracking of piglets. *Machine Vision and Applications*, 8(3):187–193, 1995.

[12] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[13] N. M. Oliver, B. Rosario, and A. Pentland. A bayesian computer vision system for modeling human interactions. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):831–843, 2000.

[14] R. Pflugfelder and H. Bischof. Online auto-calibration in man-made worlds. In *Digital Image Computing: Technqiues and Applications*, pages 519 – 526, 2005.

[15] M. Piccardi. Background subtraction techniques: a review. In *Proc. IEEE Intern. Conf. on Systems, Man and Cybernetics*, volume 4, pages 3099–3104, 2004.

[16] F. Porikli. Integral histogram: A fast way to extract histograms in cartesian spaces. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 829–836, 2005.

[17] P. M. Roth and H. Bischof. On-line learning a person model from video data. In *Video Proc. for IEEE Conf. on Computer Vision and Pattern Recognition*, 2006. (in press).

[18] P. M. Roth, H. Grabner, D. Skočaj, H. Bischof, and A. Leonardis. On-line conservative learning for person detection. In *Proc. IEEE Workshop on VS-PETS*, pages 223–230, 2005.

[19] D. Skočaj and A. Leonardis. Weighted and robust incremental method for subspace learning. In *Proc. IEEE Intern. Conf. on Computer Vision*, volume II, pages 1494–1501, 2003.

[20] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume II, pages 246–252, 1999.

[21] Y.-L. Tian, M. Lu, and A. Hampapur. Robust and efficient foreground analysis for real-time video surveillance. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 1182 – 1187, 2005.

[22] K. Tieu and P. Viola. Boosting image retrieval. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 228–235, 2000.

[23] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume I, pages 511–518, 2001.