An Annotated Video Database for Abandoned-Object Detection in a Cluttered Environment

Allan F. da Silva, Lucas A. Thomaz, Gustavo Carvalho, Mateus T. Nakahata, Eric Jardim,
Jose F. L. de Oliveira, Eduardo A. B. da Silva, Sergio L. Netto, Gustavo Freitas and Ramon R. Costa
PEE/COPPE, Federal University of Rio de Janeiro, RJ, Brazil.
{allan.freitas, lucas.thomaz, gustavo.carvalho, mateus.nakahata, eric.jardim, jose.leite, eduardo, sergioln}@smt.ufrj.br
{gfreitas, ramon}@coep.ufrj.br

***Abstract*—A video database is described for testing video surveillance systems for the automatic detection of abandoned objects in a cluttered environment. The database requirements and development are included along with a full description of its contents. The complete database comprises a total 6 multi-object, 56 single-object and 4 no-object (for reference purposes) footages, acquired with two different cameras and two different light conditions, yielding an approximate total of 8.2 hours of video. Database annotation was performed with a specialized tool, developed in the context of this particular work, that provides a bounding box for each object (or its several parts in case of partial occlusion) in every video frame. The complete dataset constitutes a valuable tool for designing and testing automatic video-based systems for abandoned-object detection.**

I. Introduction

The automatic detection of abandoned objects is a challenging problem in the field of computer vision with wide applications in surveillance systems. The problem is usually addressed by comparing a newly acquired video, also known as the target video, to a reference video considered free of abandoned objects. In this way, a video anomaly, which may be associated to an abandoned object, is detected whenever and wherever the target and reference video differ to a significant amount.

The degree of difficulty in this problem is largely increased when a moving camera is employed (to increase the area covered by the camera) and when the environment at hand is cluttered. In such cases, precise time and geometric alignments between the target and reference videos become crucial stages to a successful detection process.

An important step for designing and evaluating a practical detection system is the development of a video database addressing all the requirements imposed on the detection system. A useful database for the problem at hand also requires a frame-by-frame annotation of all objects of interest in the target videos (that is, the abandoned objects yet to be detected), which is a cumbersome and tedious process on its own.

To introduce a video database for abandoned-object detection, the remainder of this article is organized as follows: Section II discusses the problem of abandoned-object detection, including a literature review and the description of a newly proposed system using a camera on a moving robotic platform; Section III describes the design and recording processes of the proposed video database. Its annotation process, including a newly developed tool for this particular purpose, is the main focus of Section IV. Finally, Section V closes the paper summarizing its main technical contributions.

II. Abandoned Object Detection with a Moving Camera

A. Previous Solutions

The technical literature on automatic detection in surveillance applications is quite extensive. Generally speaking, video event detection using a static camera can be performed by three main techniques: (i) A simple solution is to perform a temporal subtraction between consecutive frames in search of motion areas, which are classified as anomalous objects or events [1]; (ii) Another possibility is to create a statistical model for the reference frame [2]–[6], to which each new frame is compared in order to detect a possible event or object of interest; (iii) Optical-flow based techniques [7] calculate the apparent motion of objects, surfaces and edges and detect regions that have anomalous movements.

A surveillance system based solely on static cameras, however, may not be efficient in cases where a wide area must be supervised or expensive specialized cameras (e.g. infrared or hydrocarbon-detecting cameras) are employed. A possible solution in some of these situations is the use of pan-tilt-zoom (PTZ) or panoramic cameras, which add some range flexibility to the target area. Surveillance systems using PTZ cameras often try to adapt the problem to the static-camera case [8]–[10] using any of the three approaches listed above. An alternative solution, particularly suited for specialized cameras, is the use of a moving platform to increase the surveillance scope. Such a solution, however, brings new challenges as the camera movement must be properly compensated in time and space before any sort of comparison between the target and reference videos can be made.

In [11], for instance, the space-alignment stage uses frame key points, detected by the scale-invariant feature transform (SIFT) algorithm [12], to compute an affine transformation between the given target and reference frames. In order to detect abandoned objects on a road, the work described in [13] uses a reference video (without any objects of interest) recorded by a camera on the top of a car following a predetermined path. The new videos having objects to be detected are recorded using the same setup and following the same path. Video synchronization is achieved using a GPS device set in the vehicle. Video registration is performed by a homography transformation between the target and reference frames, using the key points determined by the SIFT algorithm assisted by the random sample consensus (RANSAC) [13], [14] algorithm, which removes possible outliers. After proper time and space alignments, the two target and reference frames are compared using the normalized cross correlation (NCC) function, which generates a binary mask that indicates the presence or absence of objects in that particular target frame.

B. Proposed System

In [15], a surveillance system is described employing a high-definition camera mounted on a robotic iRobot Roomba© platform performing a linear back-and-forth motion on a track, as depicted in Fig. 1. This system should be capable of detecting abandoned objects in real time within an industrial location. The proposed algorithm is designed to operate in locations of difficult access, like an offshore oil platform, where the use of automated remote inspection can minimize labor risks and reduce the cost involved in staff transportation.

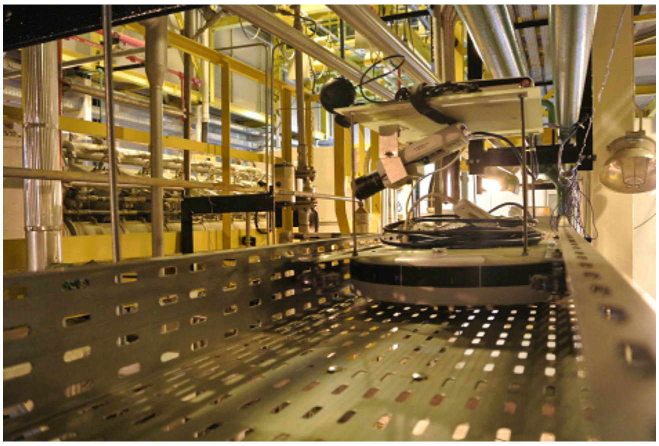


Figure 1. General setup of abandoned-object detector and database recording using a moving camera on a robotic platform.

Similar to [13], this work uses a reference video containing only the background, as validated by a system operator. Temporal alignment is performed without any external cues, using the maximum likelihood criterion to compare the robot's horizontal movement along the target and reference videos, where the change-of-direction frames are used as the movement origin in each direction. The frame key points in this case are obtained with the speed-up robust features (SURF) algorithm [16], which requires less computational operations than the SIFT algorithm. As the camera only moves along a straight line, any key-point correspondences between the target and reference videos forming an angle higher than 1 ° with the horizontal line are discarded, thus saving part of the computation required by the RANSAC algorithm. The comparison between the time- and space-aligned frames is made by the NCC function, and a detection threshold is applied to produce a binary mask, which indicates possible detected objects within the target frame. To avoid false detections, while also reducing the resulting computational complexity, the NCC is determined only in areas where the absolute values of the frame differences are higher than a given threshold. Finally, the binary mask undergoes a temporal filtering using a simple voting procedure, where each pixel is considered to belong to an object if it is detected a minimum number of times within a given interval.

III. Database Development

A. Database Requirements

In order to allow a systematic performance evaluation of the developed method, a versatile database must be deployed having the following requirements:

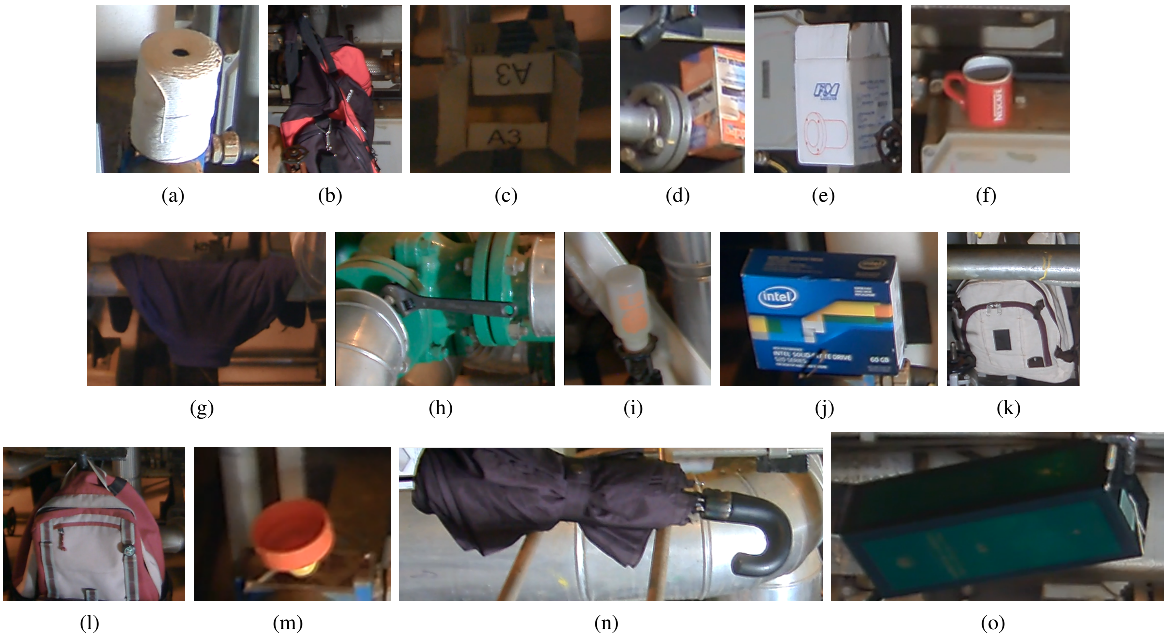
- Videos should include several changes in the movement direction to allow a proper assessment of the time-alignment algorithm;
- Objects of different colors, textures, shapes and sizes must be considered;
- The positions and numbers of objects within a frame must change in different video footages;
- Different levels of luminosity must be considered, not only in different videos but also within a same video footage.

As far as we know, there was no database available that could fulfill those requirements, which apply for a very specific case. Therefore a new database had to be designed and recorded so the application could be tested.

B. Database Design

Based on the requirements listed above, the proposed database was devised in the following way:

- Use of two different cameras, both set to a resolution of 1280 x 720 pixels and a rate of 24 frames per second, with quite distinct light/color characteristics;
- Use of a spotlight in half of the footages, making much brighter recordings, whereas natural light was employed on the other half of the recordings;
- Recording of 6 multiple-object and 2 no-object (reference) videos with one camera (Axis P1346) and 54 single-object and 2 no-object (reference) videos with the other camera (Dlink DCS-3717);
- All the multiple-object recordings included 6 full passages (or 5 direction changes) of the camera, where the single-object videos only have one passage in each direction;
- Use of 15 distinct objects for the multiple-object videos (see Fig. 2), and 9 other objects in the single-object videos (see Fig. 3). Some objects are transparent or reflective, and some are made of a material similar to the environment, so that there is a significant probability that they pass undetected by the human eye;
- The 6 multiple-object videos were recorded with the same 15 objects placed in 3 different positions, with the spotlight or natural light, making completely distinct arrangements. The different positions also caused some objects to change size in the footages, as they may be farther or closer to the camera. The 54 single-object videos work in a similar way, as each of the 9 objects was recorded in 3 different positions and with/without the use of a spotlight;
- All multiple-object videos were devised in a way that most of the frames include at least 2 objects.

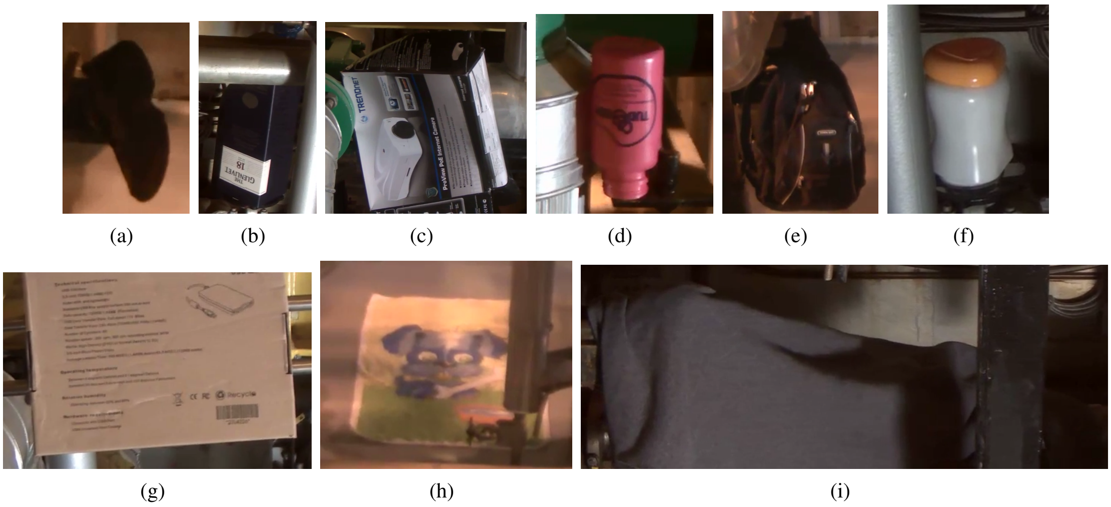


(a) (b) (c) (d) (e) (f)

(g) (h) (i) (j) (k)

(l) (m) (n) (o)

Figure 2. Objects used in the multiple-object videos (scales have been changed for a better presentation): (a) string roll; (b) bag; (c) white box; (d) lamp-bulb box; (e) spotlight box; (f) mug; (g) blue coat; (h) wrench; (i) bottle; (j) blue box; (k) backpack; (l) pink backpack; (m) bottle cap; (n) umbrella; (o) green box.



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

(g) (h) (i)

Figure 3. Objects used in the single-object videos (scales have been changed for a better presentation): (a) shoe; (b) dark blue box; (c) camera box; (d) pink bottle; (e) black backpack; (f) white jar; (g) brown box; (h) towel; (i) black coat.

C. Database Recording

The database recording was performed with the camera mounted on an iRobot Roomba© platform passing over a hanging rail at a height of approximately 2.5-m. The camera was pointed to a cluttered environment comprising several pipes and valves simulating a scene of interest inside an offshore facility.

Slight differences in illumination, video durations and robot speeds can be identified within the database, as the recordings were performed in several sessions comprising a total period of about 7 months. Such differences are interesting in the sense that they allow a more robust evaluation of the proposed system with respect to these characteristics.

Due to the cluttered characteristic of the environment, many object occlusions occur during the videos. An example is when part of the scenario (or of a foreground object) passes in front of (another) object of interest. This feature is also a desired characteristic of the database, since the surveillance system should be able to detect an abandoned object even if it is occluded during a small period of time.

One (out of the 6) multiple-object videos has only 5 camera passings instead of 6. The single-object videos have an average duration of 6 minutes, where the multiple-object videos have an average duration of 18 minutes (except the one with 5 passages that is about 12-minute long).

Another interesting feature of the database is that the recorded videos include unwanted images of the objects, due to the many reflexive surfaces of the cluttered environment, and shadows casted by the objects or the scenario itself. These effects may impair the performance of the surveillance system, by causing false-positive or false-negative situations.

IV. Database Annotation

A crucial step to turn the proposed database into a useful tool for the performance assessment of object-detecting systems is to identify all objects within its video footages. In our case, we opted for a manual annotation process of each object, since the human ability is considered the gold standard for the application at hand. Due to the large quantity of frames to be marked (in the order of 7.1×10^5 frames), it became clear that would be necessary the support of a specific software for this purpose with the following characteristics:

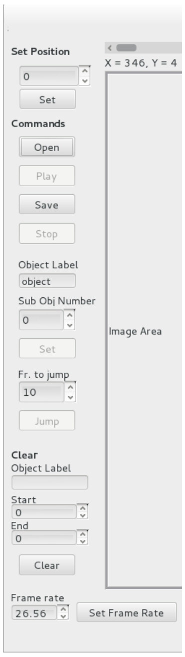
- Object mark consisting of an outline of easy identification, preferably a simple bounding box;
- Marks inserted quickly, via mouse, due to the large number of frames to be considered;
- Input commands via a GUI interface with a minimum number of intuitive commands;
- Ability to mark multiple objects in the same frame;
- Possibility to identify and associate several parts to a single object due to occasional partial occlusions;
- Generation of an output file with the labels and corresponding coordinates of all objects in each frame.

A new marking software attending all these requirements was developed in C++ [17], using the free version of the QT Creator 2.81 application [18], due to its friendly interface, extensive documentation available and portability between Windows and Linux (Fedora 19) operating systems. The program also used the free and multiplatform OpenCV library [19] for video manipulations. The main screen of the developed annotation tool is shown in Fig. 4, whose main commands are indicated on the left side (from top to bottom):

1. Set video position (in frames) by a slider (gross adjustment) or frame number (fine adjustment);
2. Open video file to be marked;
3. Play video (with marked objects if a corresponding markup file exists);
4. Save the markup text file;
5. Set the object name and sub-index;
6. Skip frames without manual mark (interpolation is performed, as described below);
7. Clear specific object from the output file;
8. Set video frame rate.

The marking process of a given video starts by opening the desired file and setting it to the desired frame position. When first marking a video, the frame position should be set to 1; a different frame number is used when marking a video which has already been partially marked. The play-video function can be used to determine the video portion which has already been marked.

For marking an object, a rectangular bounding box was chosen for its simplicity. For that purpose, the mouse must be positioned at any bounding-box corner and dragged to the opposite corner with the left button pressed. When the left mouse button is released a box is drawn around the object, which is then validated with the "set" button.



Set Position: 0

X = 346, Y = 4

Object Label: object

Sub Obj Number: 0

Fr. to jump: 10

Start: 0

End: 0

Frame rate: 26.56

Figure 4. Main menu of video annotation tool showing all marking features.

The name of the object and its sub-index should be informed by the user. In the proposed syntax, full objects are marked with sub-index zero. In case an object is obstructed by another and it seems to be divided into several parts, each part receives a sub-index number varying from 1 to the number of parts.

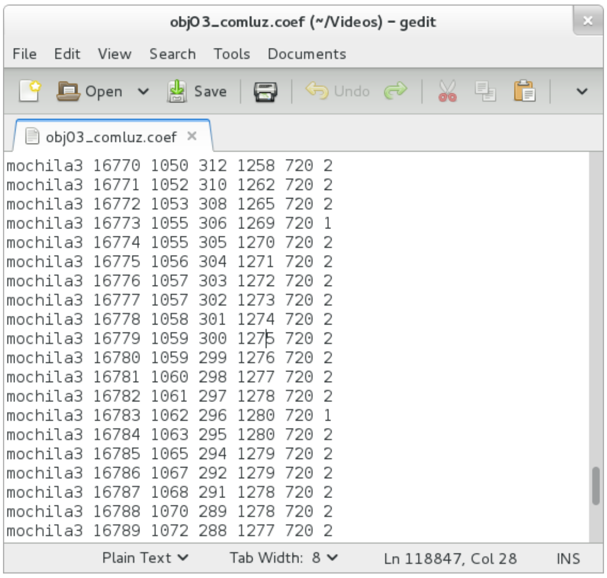
To expedite the marking process, the user can skip a given number of frames and have the annotation tool to generate bounding boxes at interpolated positions (supposing they are at a constant speed). Although such interpolation process can lead to some marking error, at high frame rates (e.g., 30 frames/s), the variation between close frames (about 10 frames apart) is quite small and so is the inserted error.

The entire annotation process is summarized in a text file containing, in each line, the label of the object, the frame number and the coordinates of the upper-left and bottom-right corners of the bounding box for each object or sub-object, as shown in Fig.5.

A visual example of the results provided by the annotation and detection processes is provided in Fig. 6 for a specific multi-object video frame. In this figure, we notice the presence of the 'spotlight box', 'blue coat' and 'bottle cap' objects (shown in Figs. 2e, 2g and 2m, respectively), with the 'blue coat' being partially occluded by a large pipe in the foreground, as properly identified in Figs. 6a and 6b.

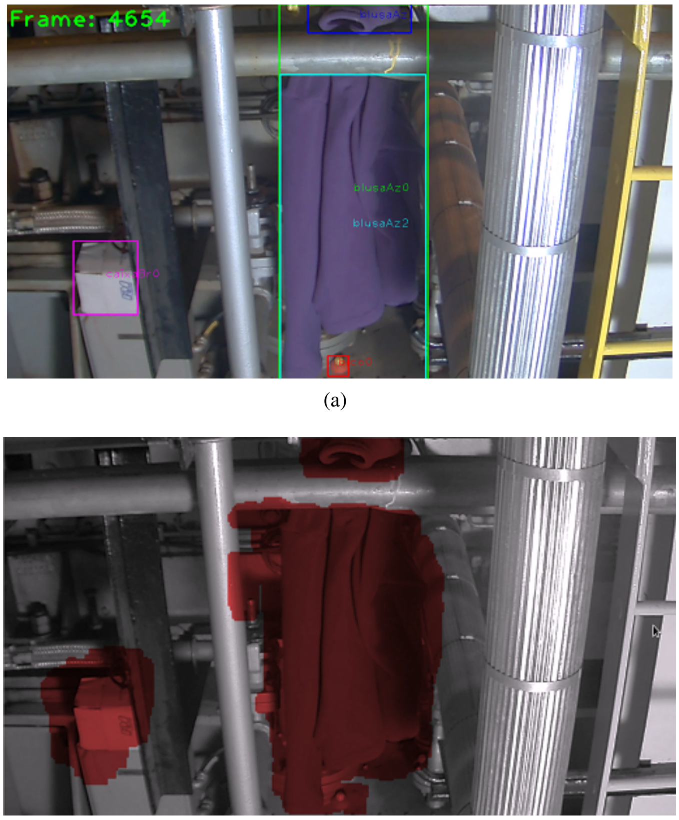
V. Conclusion

A video database for training and validation of an automatic detector of abandoned objects using a moving camera was described. The database is composed of 56 videos featuring a single object as well as 6 videos containing multiple objects simultaneously shown. Other 4 videos were acquired with no objects for reference purposes. The requirements and characteristics of the database were described along with



```
mochila3 16770 1050 312 1258 720 2
mochila3 16771 1052 310 1262 720 2
mochila3 16772 1053 308 1265 720 2
mochila3 16773 1055 306 1269 720 1
mochila3 16774 1055 305 1270 720 2
mochila3 16775 1056 304 1271 720 2
mochila3 16776 1057 303 1272 720 2
mochila3 16777 1057 302 1273 720 2
mochila3 16778 1058 301 1274 720 2
mochila3 16779 1059 300 1275 720 2
mochila3 16780 1059 299 1276 720 2
mochila3 16781 1060 298 1277 720 2
mochila3 16782 1061 297 1278 720 2
mochila3 16783 1062 296 1280 720 1
mochila3 16784 1063 295 1280 720 2
mochila3 16785 1065 294 1279 720 2
mochila3 16786 1067 292 1279 720 2
mochila3 16787 1068 291 1278 720 2
mochila3 16788 1070 289 1278 720 2
mochila3 16789 1072 288 1277 720 2
```

Figure 5. Example of a text file excerpt generated by the annotation process.



(a)

(b)

Figure 6. Example of a multiple-object video frame: (a) annotation result; (b) detection result.

the motivation of each feature. Following this framework, two different high-resolution cameras and two different light conditions were considered. Also, a software tool that assisted the annotation process (through bounding-box identification of all objects in all frames) of the entire 8.2-hour database was presented. The development of this tool was explained and its use clarified. The main use of the database is to assist the in the design of a moving-camera abandoned-object detector, and allow a proper assessment of its performance.

Acknowledgments

This work was developed with the support of Statoil Brazil, Petrobras, ANP, FAPERJ, and CAPES with project *Pró-Defesa* under process number 23038009094/2013-83.

References

[1] Lipton, A. J., Fujiyoshi, H., Patil, R. S., *Moving Target Classification and Tracking from Real-time Video*, In: Proc. IEEE Workshop on Applications of Computer Vision, pp. 8-14, 1998.

[2] Cheng, L., Gong, M., Schuurmans, D., et al., *Real-Time Discriminative Background Subtraction*, IEEE Trans. on Image Processing, vol. 20, no. 5, pp. 1401–1414, New Jersey, USA, May 2011.

[3] Yu, Y., Zhou, C., Huang, L., et al., *A Moving Target Detection Algorithm Based on the Dynamic Background*, In: Int. Conf. Computational Intelligence and Software Engineering, pp. 1–5, Wuhan, China, December 2009.

[4] Peijiang, C., *Moving Object Detection Based on Background Extraction*, In: Int. Symp. Computer Network and Multimedia Technology, pp. 1–4, Wuhan, China, January 2009.

[5] Niu, L., Jiang, N., *A Moving Objects Detection Algorithm Based on Improved Background Subtraction*, In: Int. Conf. Intelligent Systems Design and Applications, pp. 604–607, Kaohsiung, Taiwan, November 2008.

[6] Li, Q.-Z., He, D.-X., Wang, B., *Effective Moving Objects Detection Based on Clustering Background Model for Video Surveillance*, In: Congress on Image and Signal Processing, pp. 656–660, Sanya, China, May 2008.

[7] Zhou, D., Zhang, H., *Modified GMM Background Modeling and Optical Flow for Detection of Moving Objects*. In: IEEE Int. Conf. Systems, Man and Cybernetics, pp. 2224–2229, Waikoloa, USA, October 2005.

[8] Suhr, J. K., Jung, H. G., Li, G., Noh, S.-I., Kim, J., *Background Compensation for Pan-Tilt-Zoom Cameras Using 1-D Feature Matching and Outlier Rejection*, In: IEEE Trans. Circuits and Systems for Video Technology, vol. 21, no. 3, pp. 371–377, March 2011.

[9] Xue, K., Ogunmakin, G., Liu, Y., Vela, P. A., Wang, Y., *PTZ Camera-Based Adaptive Panoramic and Multi-Layered Background Model*, In: IEEE Int. Conf. Image Processing, Brussels, Belgium, September 2011.

[10] Hayman, E., Eklundh, J.-O., *Statistical Background Subtraction for a Mobile Observer*, In: Proceedings ICCV, pp. 67–74, Nice, France, 2003.

[11] Zhou, D., Wang, L., Cai, X., et al., *Detection of Moving Targets with a Moving Camera*. In: IEEE Int. Conf. Robotics and Biomimetics, pp. 677–681, Guilin, China, 2009.

[12] Lowe, D. G., *Distinctive Image Features from Scale-Invariant Keypoints*, In: Int. Journal of Computer Vision, New Jersey, USA, 2004.

[13] Kong, H., Aaudibert, J.-Y., Ponce, J., *Detecting Abandoned Objects with a Moving Camera*, In: IEEE Trans. Image Processing, vol. 19, no. 8, pp. 2201–2210, August 2010.

[14] Hartley, R., Zisserman, A., *Multiple View Geometry in Computer Vision*, 2nd edition, Cambridge University Press, Cambridge, U.K, 2003.

[15] Carvalho, G., de Oliveira, J. F. L., da Silva, E. A. B., Netto, S. L, et al., *Um Sistema de Monitoramento para Detecção de Objetos em Tempo Real Empregando Câmera em Movimento*, In: Simpósio Brasileiro de Telecomunicações, Fortaleza, Brazil, September 2013.

[16] Bay, H., Ess, A., Tuytelaars, T., et al. *Speeded-Up Robust Features (SURF)*, In: Computer Vision and Image Understanding, vol. 110, no. 3, pp. 346–359, 2008.

[17] Deitel, H. M., Deitel, P. J., *C++ Como Programar*, 3rd edition, Bookman, Porto Alegre, Brazil, 2001.

[18] Blanchette, J., Summerfield, M., *C++GUI Programming with Qt 4*, Prentice-Hall, Massachusetts, USA, 2006.

[19] Laganière, R., *OpenCV 2 Computer Vision Application Programming Cookbook*, PacktPuv, Birmingham, UK, 2011.