ABANDONED OBJECT DETECTION USING OPERATOR-SPACE PURSUIT

Lucas Arrabal Thomaz

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Elétrica.

Orientadores: Eduardo Antônio Barros da Silva
Sergio Lima Netto

Rio de Janeiro
Março de 2015

ABANDONED OBJECT DETECTION USING OPERATOR-SPACE PURSUIT

Lucas Arrabal Thomaz

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Examinada por:

Prof. Sergio Lima Netto, Ph.D.

Prof. Marcello Luiz Rodrigues de Campos, Ph.D.

Prof. Helio Côrtes Vieira Lopes, D.Sc

RIO DE JANEIRO, RJ – BRASIL
MARÇO DE 2015

Thomaz, Lucas Arrabal

Abandoned Object Detection Using Operator-Space Pursuit/Lucas Arrabal Thomaz. – Rio de Janeiro: UFRJ/COPPE, 2015.

XVI, 56 p.: il.; 29, 7cm.

Orientadores: Eduardo Antônio Barros da Silva
Sergio Lima Netto

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2015.

Referências Bibliográficas: p. 52 – 55.

1. Object Detection. 2. Operator Space. 3. Cluttered Environment. 4. Moving Camera. I. Silva, Eduardo Antônio Barros da *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

Night gathers, and now my watch begins.
It shall not end until my death.
I shall take no wife, hold no lands, father no children.
I shall wear no crowns and win no glory.
I shall live and die at my post.
I am the sword in the darkness.
I am the watcher on the walls.
I am the fire that burns against the cold,
the light that brings the dawn,
the horn that wakes the sleepers,
the shield that guards the realms of men.
I pledge my life and honor to the Night's Watch,
for this night and all the nights to come.
– Night's Watch Vows, in A Song of Ice and Fire book series –

Aos meus avós.

Agradecimentos

Primeiramente gostaria de agradecer a Deus, por ter guiado, através de tantas ações, os meus passos que culminaram no fim dessa jornada maravilhosa.

Em seguida, agradeço aos meus pais, que são responsáveis pela pessoa que sou hoje. Sem a ajuda deles a conclusão desse trabalho seria impossível. Obrigado pelo carinho e afeto em todos os momentos, pelos conselhos quando eu precisava, pelas críticas e elogios que me motivaram a seguir em frente, por sempre acreditarem em mim, mesmo quando eu não acreditava. Aproveito para agradecer ainda por terem me dado a minha irmã, que sempre foi uma grande companheira e amiga.

Agradeço aos meus orientadores, seu trabalho, sempre com muito empenho e atenção, me mostrou que era possível ir mais longe. Obrigado pelos votos de confiança, pelas inspirações, pelo rigor do trabalho e por se dedicarem a formar novos profissionais todos os dias. Obrigado ainda pelos desafios propostos, como fazer essa dissertação em inglês num tempo muito restrito. Enfim, por moldarem um jovem engenheiro num mestre.

Agradeço ao restante da minha família e padrinhos, que foram fundamentais para a minha formação e crescimento até aqui.

Agradeço aos meus amigos, aqueles com quem convivi na universidade, com quem muito aprendi e a quem pude ensinar um pouco. Essa jornada teria sido muito mais complexa sem a companhia de vocês. Que nossos caminhos continuem se cruzando dentro e fora da vida profissional. Em especial agradeço àqueles que participaram ativamente da confecção desse trabalho, seja com ideias ou até mesmo sendo cobaias de apresentações e sugerindo mudanças.

Meu muito obrigado aos Bonobos, não os símios, mas àqueles com quem divido meu tempo livre e ajudam a afastar a mente dos assuntos sérios quando é momento de relaxar. Desde as mesas de RPG e boardgames até as mesas de bar e quadras de futebol a companhia de vocês sempre engrandece os meus dias.

Agradeço à FAPERJ pela concessão da Bolsa Nota 10, que recebi durante a execução desse trabalho.

Por fim, a todos, mencionados aqui ou não, que contribuíram de alguma forma para a execussão desse trabalho, meus agradecimentos.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

DETECÇÃO DE OBJETOS ABANDONADOS USANDO BUSCA DE ESPAÇO DE OPERADORES

Lucas Arrabal Thomaz

Março/2015

Orientadores: Eduardo Antônio Barros da Silva
Sergio Lima Netto

Programa: Engenharia Elétrica

Este trabalho apresenta uma metodologia para ser utilizado na detecção de objetos abandonados e outros eventos de vídeo num ambiente visualmente poluído usando uma câmera móvel. No método proposto um vídeo alvo, que pode ter objetos que desejamos detectar, é comparado a um vídeo de referência previamente adquirido, o qual se assume não ter objetos ou eventos de interesse. A comparação é realizada através de operadores otimizados, gerados pelo vídeo de referência, que produzem saídas gaussianas quando a entrada correta é aplicada. Qualquer anomalia de interesse nos vídeos alvo gera uma saída não gaussiana. O método funciona sem que haja a necessidade dos vídeos de referência e alvo estarem sincronizados ou precisamente registrados, sendo robusto a rotações e translações entre os quadros dos vídeos. Os experimentos realizados mostram a boa performance do método proposto.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

ABANDONED OBJECT DETECTION USING OPERATOR-SPACE PURSUIT

Lucas Arrabal Thomaz

March/2015

Advisors: Eduardo Antônio Barros da Silva
Sergio Lima Netto

Department: Electrical Engineering

This work presents a framework to be used in the detection of abandoned objects and other video events in a cluttered environment with a moving camera. In the proposed method a target video, that may have features we would like to detect, is compared with a pre-acquired reference video, which is assumed to have no objects nor video events of interest. The comparison is carried out by way of the achieved optimized operators, generated from the reference video, that produce Gaussian outputs when applied to it. Any anomaly of interest in the target video leads to a non-Gaussian output. The method dispenses with the target and reference videos being either synchronized or precisely registered, being robust to rotations and translations between the frames. Experiments show its good performance in the proposed environment.

Contents

List of Figures

List of Tables

List of Symbols

$\mathbb{R}$ Real Number Set, p. 14

List of Abbreviations

3D	Three-dimensional, p. 8
DFT	Discrete Fourier Transform, p. 12
FFT	Fast Fourier Transform, p. 12
FoV	Filed-of-View, p. 3
GMM	Gaussian Mixture Models, p. 4
GPS	Global Positioning Systems, p. 1
IDFT	Inverse Discrete Fourier Transform, p. 21
MMFM	Multiframe Monocular Fundamental Matrix, p. 9
MRF	Markov Random Fields, p. 5
NCC	Normalized Cross-Correlation, p. 7
OOSP	Optimal Operator Space Pursuit, p. 1
PET	Positron Emission Tomography, p. 11
PTZ	Pan-Tilt-Zoom, p. 6
RANSAC	Random Sample Consensus, p. 7
RoSuRe	Robust Subspace Recovery, p. 9
VDAO	Video Database of Abandoned Objects in a Cluttered Industrial Environment, p. 28
pixel	Picture Element, p. 11

Chapter 1

Introduction

The work presented in this thesis intends to show a novel algorithm to be used in the detection of abandoned objects with a moving camera. The proposed method uses the Optimal Operator Space Pursuit (OOSP) [1] framework to represent the video sequences in a different domain that allows the comparison between a reference and a target frame to be performed in a robust way.

One of the most active problems in the field of computer vision is the automatic detection of abandoned or missing objects using video cameras. There are several approaches to the abandoned object detection problem, most of them rely on the detection of changes in the value of pixels in a frame through time. Some do it by comparing continuously the value of the pixels and detecting when there is a significant change in it. Others, such as the method we propose, try to relate matching segments of two video sources and detect whether the same region in both videos have the same approximate pixel values.

The most common way to detect abandoned objects is by using a single static camera aiming at a fixed region. Although this method may be considered simple, sometimes it is not be best choice for a given application. An alternative that can be applied to many cases is the detection of abandoned objects using moving cameras. This has been a trending problem in the literature and has been approached by several authors in their works over the past few years [2–8].

Among the methods for object detection using moving cameras, a common restriction is that the videos are both time-aligned and geometrically registered. These requirements pose several limitations in the type of environment one is interested in monitoring and also in the camera motion through the videos.

To perform the time alignment between the videos some methods rely on previously available information, as preset marks in the video, predicted direction changes or GPS (Global Positioning System) data to locate the camera in a path. Such information may not always be available, so a system that does not rely on it may be needed.

To perform the geometrical registration between the frames one usually has to find a group of features that are present in both frames, to determine the corresponding regions of the frames. The classical methods used to find these features usually fail in a cluttered environment and when the camera moves close to the scene in a translational movement. Therefore, performing the frame registration can be a challenging task in some scenarios.

The method proposed in this thesis works independently of the need of a fine alignment (requiring only that the video is roughly aligned) and geometrical registration, as it implements simple features that replace these steps of the traditional approach. Those characteristics allow our method to work in industrial cluttered scenarios using a horizontally moving camera close to the environment of interest, which are generally very challenging due to the restrictions of most systems. Common challenges are the detection of features, that may be complex in this kind of environment, and also the cluttered environment can sometimes hide objects of interest.

The proposed method was tested with the Video Database of Abandoned Objects [9], which shows an industrial scenario with lots of different abandoned objects. The results of this tests, shown in detail in Chapter 5, demonstrate the capacity of the method to deal with complex scenarios and detect objects with a high precision.

The remaining of this thesis is organized as follows: In Chapter 2 a review of the related techniques for both static and moving camera detection of abandoned objects is given. Chapter 3 discusses some approaches on image representation and the OOSP method [1] is reviewed in it. In Chapter 4 the fundamentals of an object detection system using the framework of OOSP are presented. The step-by-step development of the proposed method is presented in Chapter 5. Also in this chapter the experimental results for all the steps and the final implementation of the method are presented. In Chapter 6 the final remarks are made and some future works are presented. Finally in Appendix A the articles that resulted from this thesis are listed.

Chapter 2

Automatic Object Detection

Surveillance systems are ubiquitous in present days. The need of more security usually increases the area that must be covered by the cameras to allow a proper security level. With a large amount of videos to be observed and the need of a continuous operation, sometimes the human work is not efficient, since in long and repetitive tasks people usually lack attention. There may be also high costs involved in maintaining the staff and the possibility to expose the employees to labour risks.

Due to all the above reasons, minimizing human participation in surveillance tasks if often desirable. A common solution to this issue is the use of automatic surveillance systems. Such systems apply computer techniques (usually from the area of computer vision) to perform the same tasks that would be performed by a human, thus replacing or assisting a human operator in his job.

A common task that many security systems implement is the detection of abandoned or missing objects and other interesting video events that may be seen on security footages.

In this chapter we will review some approaches that have been used to deal with the challenge of automatic object detection with static cameras and with moving cameras. First some static camera object detection methods will be reviewed and later some moving camera object detection methods will be presented.

2.1 Static Camera Object Detection

The simplest and most common way to implement an automatic surveillance system is to use a static camera posed in a way to cover a large field of view (FoV), or at least an area of interest, and process the footages of this camera aiming to detect a group of pre-determined video events. In this section we will review such approaches using static cameras.

Techniques for automatic detection of abandoned objects have been long developed and studied by researchers. Many of them [10, 11] rely on the stationarity of

the background and implement some kind of background subtraction or suppression. In the earliest implementations it was assumed that the background was completely constant. Thus the simplest way to detect changes in the scene, and so find a moving object or something that was not previously in the scene, is to compare the values of the pixels and trigger an alarm if the difference of the pixel values in two consecutive frames is greater than a predefined threshold, as described in [12]. Besides the fact that this technique is both simple and fast, it does not retain any history of the scene and only the edges of the objects are detected, because the objects rarely move fast enough to cover an entire area of the scene between two frames.

Some more advanced techniques still rely on background subtraction, but in a more robust way, as they model the background with a statistical model. The most common statistical model is the Gaussian model as in [13]. In this method, each background pixel is modeled by a single Gaussian distribution, forming the background model. To classify a new pixel, its value is compared to the mean of the Gaussian and, depending of how far it is from it, the pixel is classified either as background or as foreground. If it is classified as background, the model is updated to comply with the new mean and variance. This approach allows the background to slowly change along the duration of the video. Thus it can deal with situations where the scene changes slowly, as in the case of lighting variations of an outdoor scene. The main drawback of this approach is that it considers that the background can be modeled with a single distribution and cannot deal with multimodal background as waving trees.

To deal with multimodal background and create a more robust model for the scene, some latter works applied multiple Gaussian models in as scheme that was latter known as Gaussian Mixture Models (GMM) [14]. This method models each pixel of the background as a weighted mixture of several Gaussian distributions. It uses K-means algorithm approximation to find the parameters and weights for each Gaussian distribution in each pixel. A decision of which Gaussian a new pixel belongs to is based on the mean and variance of each Gaussian.

Besides all the improvements implemented in the methods, these techniques still face problems when the image is noisy or too complex to be modeled by a single distribution or even by GMMs. Also there are major problems when the background is not exactly fixed, which may happen due to camera trepidation, jitter, wind, or other common causes.

Some other techniques, as the one called “Behavior Subtraction” [15], intend to extend the background subtraction techniques and solve some of the issues that are common to this approach. Whereas in background subtraction the pixel values are considered to be the variable for the background model, in behavior subtraction the stationary scene dynamics is used as a “background” activity with which the

observed scene dynamics are compared. This method does not rely neither on the computation of motion nor in the object tracking. Instead it considers the concept of an *event*, that is defined as short-term scene dynamics captured over a time window at a specific spatial location in the camera FoV. The events are computed by time-aggregation motion labels and suitable object descriptors (e.g. object shape, object size). The events are then probabilistically characterized as random variables that are independent and identically distributed (iid) in time. Behavior maps are designed for each frame based on the evolving event model and finally a subtraction is made between the behavior model of the scene and the one of the frame of interest, resulting in the detection of anomalous events. This framework solves some of the problems such as dealing with a dynamic cluttered background and jitter in the camera.

Although less common, some works [16] address the detection of abandoned objects mixing it with the object tracking problem. This approach particularly aims at the problem of car detection in a parking lot. The detection of foreground and background uses the traditional method of GMMs, but to avoid false positives due to moving shadows, noise, animal and people walking, the movement of the object is tracked. As the assumption is that there is a pattern in the trajectory of cars parking and leaving a parking lot, it is easy to discard any detection of objects whose trajectory does not correspond to the predicted car movement pattern in the scene.

There are also other methods [17, 18] that use statistical information of the video sequences to perform the detection of the objects without necessarily separating the foreground from the background. Such methods usually detect abnormal activity in the video by computing the statistics of the pixels of the frames and trying to find values that are out of the observed pattern.

Some approaches do not apply Gaussian models and separate the background from the foreground using different statistical models like compound Markov random fields (MRF). The method presented in [19] is one of those who use MRF to model the background. Such method implements two kinds of segmentation in the image: one is a spatio-temporal spatial segmentation and the other is a temporal segmentation. Spatial segmentation helps to determine the boundary of the regions in the scene accurately, and temporal segmentation helps to determine the foreground and the background parts of it.

Although very distinct in their approaches, all the above methods, as well as many others, have the limitation to work only with a stationary, or almost stationary background. This restricts their application to the case of static cameras. When a large surveillance area is needed it is necessary to install several cameras and then the costs of installation and maintenance of the system multiply. An alternative to

such situations is to install a moving camera system to cover a larger area processing a single video sequence.

2.2 Moving Camera Object Detection

In many situations, the use of a single moving camera can be of great benefit for a surveillance system. Not only it is able to cover a large area without the need of dealing with multiple cameras installed, but also allows the system to show in more detail distinct areas of interest. Also it allows the FoV to be changed following the will of an operator, or adopt a preset path showing important areas. Along with the benefits of having a moving camera in the surveillance system come several challenges. The use of moving cameras in security systems is far less studied than that of static cameras. The moving background, inherent in the use of moving cameras, usually hinders the application of traditional techniques such as background subtraction.

A common type of moving camera is the Pan-Tilt-Zoom (PTZ) camera. This genre of devices allows the FoV to be changed during the operation of the camera. In most cases the camera allows a pan of nearly 360º, tilt of 90º and some optical zoom. This kind of camera can be used in automatic surveillance systems, but traditional techniques of object detection cannot be used.

In [2] a method to use a PTZ camera with background subtraction is shown. This framework constructs a mosaic to create a complex background model compensating the camera movement. As the camera moves around a fixed center, some of the points that were displayed before the rotation can still be found in the new image, so it is possible to find correspondences between those points and then find a homography matrix [20] that is able to transform the common part of both frames from the state before the rotation to the new state. Doing this process it is possible to create an interactive background that may be used in a background subtraction using GMMs to separate the foreground. Doing so, one can find the abandoned object in the scene.

The approach of background subtraction is also used to deal with more complex kinds of movement, like those in a PTZ camera mounted on a moving platform through a fixed track. In [3], for example, a Bayesian approach is used to determine whether the background has yet been uncovered by moving foreground objects. This will allow the system to determine if the background model is available to the pixels of interest or if the background model should be acquired, possibly in the presence of moving foreground objects.

Interesting approaches are those where the movement of the camera is not predetermined, as that of a camera mounted on a car driving in a road. Some work has

been done in this sense, usually relying on the existence of a reference video, where the camera has followed the same path as the present footage, and both sequences are compared to determine whether an object is present or missing in the scene, or if the scene has not changed.

Reference [21] presents an approach using a camera moving with a car on a road. To detect the objects, the system compares the videos from a reference sequence, which is assumed to have no abandoned nor missing objects, to another sequence which may have abandoned or missing objects. As the proposed framework is designed to be used in a road scenario, where the camera is placed on the windshield of a car, it is assumed that no objects will be found above the horizon line, so only the inferior part of the frame is considered. To deal with the correspondences between reference and target frames it is important that both videos are time-aligned (via Global Positioning System (GPS in this case), meaning that the n^{th} frame in the reference video should correspond to the n^{th} frame in the video being analysed. Even though the videos are time-aligned it is possible that may be a difference between the images caused by some misplacement or jitter in the camera, so it is necessary to have a geometrical registration between the frames. To do so, keypoints must be acquired from both frames and and homography computed with the aid of random sample consensus (RANSAC) [22] algorithm. After the process of geometrical registration a normalized cross-correlation (NCC) measure is taken between small, correspondent windows of the frames to asses the similarity of each region of the image. The regions where the NCC points a low correlation generate an alarm of potential abandoned or missing object. Finally a temporal filtering is applied to verify if the alarm is triggered to the same region in consecutive frames, indication there is an object in the area.

A latter work [4] applies several ideas presented in [21] to create a more robust abandoned object system without the need of external trigger to align the videos (such as the GPS signal used in the previous work) and aims different scenarios where the environment of interest is a cluttered industrial plant. This work presents a novel alignment method which relies on the integration of all horizontal components of the homographies between consecutive frames to determine the direction of the camera movement. Since, in their application, the camera moves horizontally through a fixed track changing its direction whenever it reaches the end of the track, the integration of the horizontal displacement of the homographies will show the places where the reference and target videos changed the direction making the alignment a pattern fitting problem.

Several other works like [5, 23] use the homography to create spatial and temporal alignment between video sequences. In most cases the trajectory of the camera can be considered to be free as long as the scene may be considered planar or the

movement is a rotation around a fixed point and the scene may contain points in different planes. If this requirements are not fulfilled there may occur occlusions between objects in the scene and the observed part of them may change, making the use of homography matrices very difficult.

Prior to computing homographies to provide both temporal and geometric alignment between frames to perform the comparison, it is necessary to acquire the description of the frames through image descriptors such as SIFT [24], SURF [25], BRISK [26] and FREAK [27]. Those algorithms perform the detection and selection of keypoints in an image, describing them in such way that they could be identified in some other image where they are shown up to an affine transformation [20]. This description allow correspondences from different frames to be taken, thus allowing the homographies to be created. The computation of the homography matrices relies heavily on the detection of those keypoints, but it is well known [20] that the detection of those points depends on the existence of points in different planes in the three-dimensional (3D) world. It is also known that if the movement of the camera is not a rotation around a fixed point it will create occlusions during the movement of the camera if the image is not purely planar, jeopardizing the detection of the objects. The most common way to deal with this issue is to consider that the whole image is sufficiently far from the camera so it can be considered planar.

Another issue related to the detection of keypoints is that it may have a high computational cost when compared with the whole object detection algorithm. In a previous work of ours [28], we assessed the main object detectors in the literature [24–27] to determine most suitable for an object detection system. Based on the fact that the objects from whom the keypoints are taken do not change sizes between the reference and target videos, in this work we also presented a modification to the FREAK, called SD-FREAK, which is scale dependent, lessening the number of false matches in the correspondences between two frames.

When the image cannot be considered planar from the camera perspective, different approaches must be tried. The computation of the Fundamental Matrix [20] is a common alternative. The fundamental matrix is the algebraic representation of the epipolar geometry. If two images acquired by two non-ambiguous cameras show at least a minimum number of common 3D points in it (that is, there are correspondences between those points in the two images) it is possible to determine the relation between the two cameras and thus infer the characteristics of one camera from the characteristics of the other. Also, if one knows the fundamental matrix between two camera poses it is simple to transform an image from one of them in an image from the other since they follow the property of the fundamental matrix which says that $x'^{\mathrm{T}}\mathbf{F}x = 0$, where x and x' are the homogeneous coordinates of correspondent points in the two images. That characteristic may be used to replace

the homography in the cases where the scene is not planar and the images are not related by an affine transformation.

Further methods extrapolate the concept of the fundamental matrix and propose some even more complex concepts. In [6] a Multiframe Monocular Fundamental Matrix (MMFM) is proposed to be a dynamic fundamental matrix. The main goal of this method is to find moving objects in a non-planar background with a moving camera. As the camera moves freely, without just rotating around a fixed point and the background is non-planar, the use of homographies is ruled out. The method defines an evolving epipolar plane between the initial camera center, its subsequent centers and the static 3D point of interest. If it is assumed that the evolution of the camera parameters can be represented by polynomial functions of time, this monocular multi-frame fundamental matrix can also be represented by polynomial functions of time, assuming that inter-frame rotation is small. In light of that, it can be determined whether the points belong to either a moving or static object, even if the background changes. By doing so, it defines an optical flow [29] and is able to observe the difference between static and moving objects.

Although the use of the fundamental matrix instead of the homography to relate frames allows more general applications, its use may not always be the best option. In [7] the authors highlight the fact that, for dense correspondences, the calculation of the fundamental matrix may entail some unnecessary computational effort. Moreover, estimating the fundamental matrix is susceptible to errors and moving cameras may increase this uncertainty. With that in mind, it is desired an alternative to deal with the detection of abandoned and missing objects without the need of geometric registration, nor keypoints detection.

In a recent work [8], the authors propose an abandoned object detection framework that runs based on reference-target video comparison but obviates the need of video alignment or geometric registration. It works under de assumption that the whole reference video, without any abandoned objects, can be expressed as a low-rank representation added to an error term. This representation is sufficient to describe the target video, except for the parts where the frames contain objects that where not present in the reference video, characterizing the presence of abandoned or missing objects. To create the model that will epitomize the reference video in a low-rank representation the Robust Subspace Recovery (RoSuRe) [30] algorithm is used. That creates a matrix X_r where each column represents a frame of the reference video and then it is decomposed as $X_r = L_r W_r + E_r$, where L_r is the low-rank linear part of the reference video and $E_r = X_r - L_r$ is its non-linear complement, which is a sparse error signal. Assuming that the target video can be represented from the low-rank representation of the reference the X_t matrix, where each column would represent a frame from the target video, and can be decomposed

as $X_t = L_r W_t + E_t$. The assumption is that all data that could not be described from L_r is contained in E_t, this data is mostly composed by either high-frequency information or information that could not be stored in a low-rank representation and the possible abandoned object. As the high-frequency data is supposed to be common between target and reference video another decomposition is performed in E_t to allow the high frequency data to be separated from the abandoned object data. So after performing the decomposition $E_t = E_r W + E$, E contains all the object-related information.

2.3 Conclusions

As seen in this chapter the abandoned and missing object detection is a interesting problem and has been a trendy topic over the last several years. The several distinct approaches intending to solve the static camera setup problem have provided a solid basis to the development of algorithms for abandoned object detection using mobile cameras, as many solutions to the later are adaptations from the existing solutions of the first.

Despite the fact that many original solutions to the moving camera problem have been proposed, there are still issues related to the detection of abandoned objects in more complex scenarios, as such camera close to the environment. This precludes the planar image assumption and cameras performing a translational movement. This does not allow the use of homography relations between consecutive frames. Also, the acquisition of keypoints to be used in the computation of homgraphies and fundamental matrices may be a challenging task in cluttered environments.

In the remainder of this work a new approach to the problem of abandoned object detection with a moving camera will be presented. This approach intends to avoid the main problems commonly found on other methods as listed above.

Chapter 3

Approaches for Image Representation

One of the main issues related to image processing is the representation of the images. In this sense, when choosing the way an image will be represented, one is concerned with the characterization of the quantity that each picture element (pixel) represents. Images are able to characterize several distinct properties as the luminance of a scene (often acquired by a regular video or photographic camera), the heat that is reflected or irradiated by a body (when a thermal image is acquired), the absorption characteristics of an observed tissue (as in x-rays and positron emission tomography (PET) scans), among others things. In fact, any two-dimensional function that carries information can be considered to be an image [31].

Distinct forms of representation serve for different purposes. Depending of the type of image one wants to represent, a specific form of representation is more or less indicated. The key to a good selection of an image representation is the fidelity and intelligibility of a certain image characteristic a given representation entails. In distinct representations color, contrast, spatial frequency, image resolution and many other aspects of the image may be represented differently. Thus, the selection of the image representation is a major part of image processing.

Nowadays, many applications require digital processing of an image. Therefore, one of the main requirements of a representation is that the images are both sampled and quantized.However, some images are available in analog format and require further conversion to the a proper format via sampling and quantization. To preserve the useful information of the image, the sampling rate (amount of pixels per area unit) must be large enough. The sampling rate is generally determined to comply with the bandwidth limit and to keep the useful information in the representation.

Sometimes, when the image is to be processed, an alternative representation is used as the image is replaced by a weighted sum of images called *basis images*, as shown in figure 3.1. For sampled images, the basis images $a_{k,l}(m,n)$ can be deter-

mined from a complete set of orthogonal unitary matrices called *image transforms* that satisfy the following properties [31]:

$$\text{Orthonormality} : \sum_{m,n=0}^{N-1} \sum a_{k,l}(m,n) a^*_{k',l'}(m,n) = \delta(k-k', l-l') \tag{3.1}$$

$$\text{Completeness} : \sum_{k,l=0}^{N-1} \sum a_{k,l}(m,n) a^*_{k',l'}(m,n) = \delta(m-m', n-n') \tag{3.2}$$

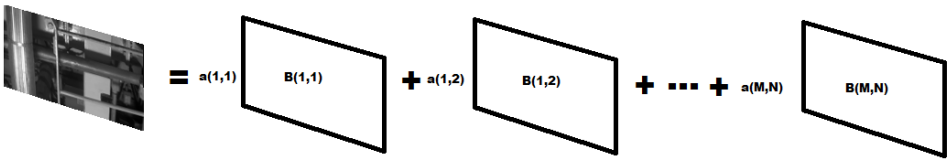


Figure 3.1: Decomposition of an image in image basis B(m,n).

An example of image transforms is shown in figure 3.2 for 8x8 cosine transform [31].

One of the most common image transforms is the two-dimensional discrete Fourier transform (DFT), which can be achieved with the fast algorithm ($N \log_2 N$) called two-dimensional fast Fourier transform (FFT). Depending of which image transform is selected to represent the image, some characteristics of it are more easily observed. For example, when the two-dimensional Fourier transform is applied to the image, observing the frequency characteristics is straightforward.

Images are often described as an ensemble rather than individually. Statistical models are applied to represent certain traits of the collective of images and allow some processing to be designed to the ensemble of images.

Sometimes we would like to represent a series of images as an ensemble so that it would store the discriminative information about each image in a low-rank representation so all information we would need to identify the image, or compare it with another, is comprised in a sub-space that is less complex and easier to deal with than the original space of images.

In the following section we will follow the development of an operator space proposed in [1], for image representation.

3.1 Optimal Operator Space Pursuit (OOSP)

High dimensional data processing is a common problem that often arises in video and image applications. The large amount of data acquired in many systems, like

Figure 3.2: Image basis functions for 8x8 cosine transform.

surveillance frameworks, generally conceals the important information from being discovered. In many image and video processing applications, the first step is to detect and extract the key information or to reduce the dimensionality of the data without loosing the information that one is interested in. This allows further processing to deal with a reduced version of the data stream where finding the key information is easier. One interesting approach is to find a low-dimensional space where the key information of the data sequence lies in. Although this is an attractive strategy it usually requires high computational power. The representation of image and video sequences in a low-dimensional space may be useful for processing a large amount of data in a way that interesting information is easily available.

In [1] the authors propose a novel framework for high dimensional data analysis. In that work the authors propose to describe image sequences through a formalism of fiber bundles and the construction of an operator space **H** which is homeomorphic to the manifold of hidden states of image sequences. The operators on the data space have a targeted output, which is known (may be a Gaussian image for instance), and eases the evaluation of the operator space. In this way, instead of working with the image space, where the key information is mingled with irrelevant data, the operator H can be used to categorize the image sequences by first developing an algorithm to find the optimal low-dimensional space where the discriminating information is compactly stored.

3.1.1 Geometric Space of Image Sequences

First let us define a metric space [32] by an ordered pair (M, d), where M is a set and d is a metric on M. That is, we have a function $d : M \times M \to \mathbb{R}$, being $\mathbb{R}$ the real number set , such that for any $x, y, z \in M$, the following properties hold:

$$d(x, y) \geq 0 \text{ (non-negative)}, \tag{3.3}$$

$$d(x, y) = 0, \text{ iff } x = y \text{ (identity of indiscernibles)}, \tag{3.4}$$

$$d(x, y) = d(y, x) \text{ (symmetry)}, \tag{3.5}$$

$$d(x, z) \leq d(x, y) + d(y, z) \text{ (triangle inequality)}, \tag{3.6}$$

where iff stands for "if and only if".

Let E be the space of all images, so each image sequence may be viewed as a sampled curve in E. E is a metric space, which implies that there is no guarantee that curves from the same class (for example, curves for the event of a human walking on a street) will be close to each other, since for a given class of image sequences, they may have different realizations. If E represents human activity videos, given a single activity different videos may show it in different ways, as people may behave

differently between the videos, even if it is the same person in the video. The difference may be found in details as gestures, clothes, speed, appearance, etc. Even if the videos show the same activity it may not be simple enough to cluster E to find that. It is needed that the activities are presented in a more friendly way, so a representation other than E is needed.

Let each frame in a video sequence be an observation of a hidden state lying on a manifold $\mathbf{B}$, which is embedded in E. These hidden states may be control variables to some activity description. We can invoke the formalism of fiber bundles [33] to describe the data set, as depicted in figure 3.3. E is the global space of all existing images. If the images are $m \times n$ grayscale (between 0 and 1) matrices, E is the space of all $m \times n$ matrices, such that each entry $x_{i,j}$ obeys $0 < x_{i,j} < 1$; $\mathbf{B}$ is the base space for the bundle containing all the control variables; fiber F over $P \in B$ is the space for different realizations of a control variable p; and $\pi : E \to B$ is a continuous surjection such that for a neighbourhood $U \in B$, $\pi^{-1}(U)$ is homeomorphic to the product space $U \times F$.

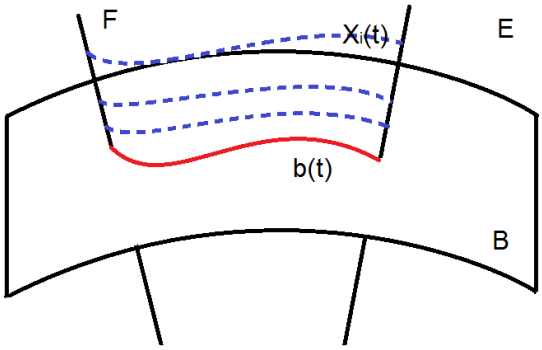


Figure 3.3: Fiber bundle structure to data space.

In this representation each image sequence is a high dimensional curve in E, which corresponds to a low-dimensional curve in $\mathbf{B}$. This way different realizations of the same activity may vary in E, but follow the same trajectory in $\mathbf{B}$, up to a noise term.

3.1.2 Operator Space Construction

Generally in video analysis frameworks we neither have the knowledge of the base manifold $\mathbf{B}$ in explicit form nor have information to infer it from the available data. One way to deal with this is, by using given samples of E, to construct an operator space $\mathbf{H}$ which is homeomorphic to $\mathbf{B}$ and work upon $\mathbf{H}$ instead of the unknown manifold $\mathbf{B}$. To do this it is necessary to create a bijection between $\mathbf{H}$ and $\mathbf{B}$.

Considering each frame of the image sequence as a sampled curve in E and no data directly acquired from $\mathbf{B}$, $\mathbf{B}$ is set as an equivalent class of elements on fiber $\pi^{-1}(p) = \{F \,\text{over point}\, p \in E\} : B = \{[x] : x \in \pi^{-1}(p)\}$. To map all points in a given fiber F_p using the ideas presented in [34], it is possible to empirically establish a homeomorphism between the operator space $\mathbf{H}$ and $\mathbf{B}$ as follows:

$$h^* = \arg\min_h \sum_i |x_i \odot h - g|, \tag{3.7}$$

where x_i, $i = 1, ..., m$ are samples in the fiber $F_p = \pi^{-1}(p)$ and g is a fixed function.

Since the object function is designed to be convex, we have a unique solution to $h\{F_p\}$, for each fiber F_p. This way it is possible to see that there is a one-to-one correspondence between the operator space $\mathbf{H} = \{h\{F_p\}\}$ and the manifold $\mathbf{B}$. With that in mind we can now work with the sequence of operators H, which is known and has a predicted behavior, rather than with $\mathbf{B}$ or E. Figure 3.4 shows the homeomorphism between $\mathbf{H}$ and $\mathbf{B}$.

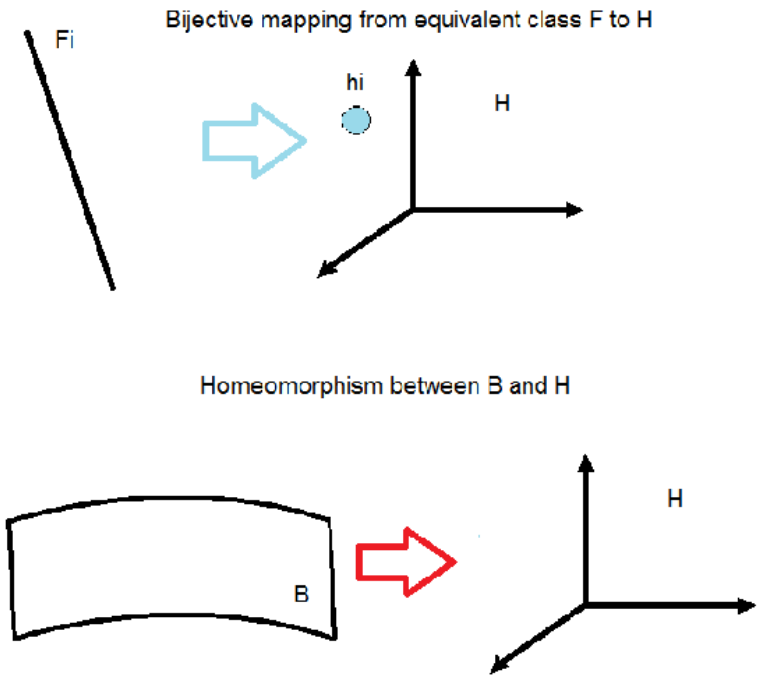


Figure 3.4: Homeomorphism between $\mathbf{H}$ and $\mathbf{B}$.

3.1.3 Optimal Operator Space Pursuit

Being $\mathbf{B}$ the base manifold of the image sequence, as stated before, we assume that $\mathbf{H}$ lies in a low dimensional sub-space of $\mathbf{B}$. We wish that $\mathbf{H}$ has the lowest dimension

possible. We then have to solve the constrained dimension minimization problem described as

$$\min \dim(H) \text{ s.t. } \|h_i(X_i) - g\|_2 \leq C, \;\; h_i \in \mathbf{H}, \tag{3.8}$$

where X_i, $i = 1, \ldots, m$ are frames of a given image sequence, C a constant and g is specific target function, that will be from now on selected as a two-dimensional Gaussian function.

Performing this minimization corresponds to finding the lowest rank matrix $H = \begin{bmatrix} h_1 \cdots h_m \end{bmatrix}$ under the constraints of equation (3.8). However, this minimization is an NP-hard problem as shown in [35]. In [1] this problem is substituted by a constrained nuclear norm minimization, which may be seen as a convex relaxation of equation (3.8). In the Fourier domain we have,

$$\min \|H\|_* \text{ s.t. } \|X_i h_i - g\|_2 \leq C, \;\; H = \begin{bmatrix} h_1 \cdots h_m \end{bmatrix}, \tag{3.9}$$

where X_i, for $i = 1, 2, \ldots, m$, are diagonally structured matrices with Fourier coefficients of each frame on the diagonal [1], h_i are the Fourier transforms of the corresponding filters, and once again C a constant and g is a two-dimensional Gaussian function.

One can use a more general form for equation (3.9):

$$\min \|H\|_* \text{ s.t. } \|A_i(H) - g\|_2 \leq C, \text{ for } i = 1, \ldots, m \tag{3.10}$$

where $A(\cdot) : \mathbb{R}^n \to \mathbb{R}^n$ is a linear operator.

For a matrix X, the singular-value threshold operator is defined as

$$D_\tau(X) = U S_\tau(\Sigma) V^*, S_\tau(\Sigma) = \text{diag}\{(\sigma_i - \tau)_+\}, \tag{3.11}$$

where σ_i are the singular values of X and $u_+ = \max(0, u)$. This operator satisfies the following theorem, obtained from [35]:

Theorem 1: For each $\tau \geq 0$ and $Y \in \mathbb{R}^{m \times n}$, the singular-value threshold operator is the solution to

$$D_\tau(Y) = \arg\min_X \frac{1}{2}\|X - Y\|_F^2 + \tau \|X\|_*, \tag{3.12}$$

where $\|X\|_F$ is the Frobenius norm of X [36].

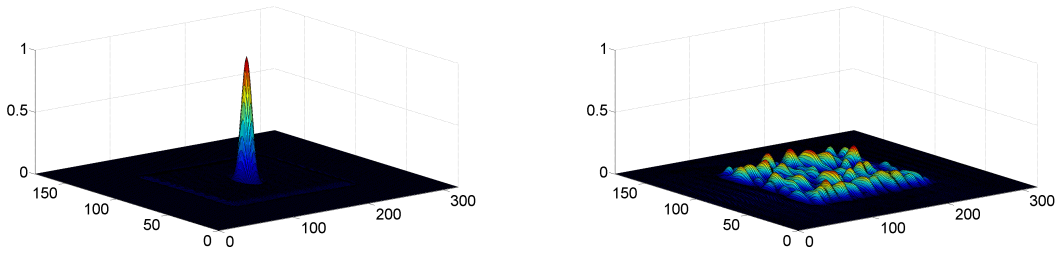
The authors in [1] developed a modified version of the singular value thresholding algorithm adapted to equation (3.10). Using some techniques that may be found in [35] it is shown in [1] that the iteration that leads to the optimum H is given by

$$\begin{cases} H^k = D_\tau \left(\sum_i A_i^*(y_i^{k-1}) \right) \\ \begin{bmatrix} y^k \\ s^k \end{bmatrix} = P \left(\begin{bmatrix} y^{k-1} \\ s^{k-1} \end{bmatrix} + \begin{bmatrix} b - A(X^k) \\ -\epsilon \end{bmatrix} \right) \end{cases} \tag{3.13}$$

with b and ϵ constants and P being a projection operator given by

$$P(y,s) = \begin{cases} (y,s), & \|y\| \leq s \\ \frac{\|y\|+s}{2\|y\|}(y,s), & -\|y\| \leq s\|y\| \\ (0,0), & s \leq -\|y\| \end{cases} \tag{3.14}$$

After the completion of the minimization process, **H** is a sub-space spanned by matched-filters (H) that can represent the images that form the original space of sequences. In this alternative representation, each filter is linked to an image of the sequence in the sense that if an image is used as input to the corresponding matched-filter the output is the Gaussian function g. Figure 3.5 shows two possible outputs for matched and unmatched entries in the filter. In 3.5(a) the output of the filter is a Gaussian image, as the entry of the filter was similar to the image that generated the filter. In 3.5 (b) the output is not a Gaussian image, as the entry of the filter was not similar to the image that created the filter. In that sense, the output of the filter is more similar to the predicted output (a Gaussian image in our case) when the entry of the filter is more close to the image used to create the filter.



(a) (b)

Figure 3.5: Possible outputs for matched filter: (a) corresponding image; (b) erroneous image.

3.1.4 Video Sequence Similarity Measure

In [1] the authors propose the creation of a similarity measure for two distinct video sequences based on the principles exposed in this section. The development of the similarity measure uses the approach described next. One of the sequences is chosen as the reference (X^q) and is used to generate the subspace of filters H^q. Then, each

frame X_i^p of the other sequence is tested to measure its distance to the reference sequence, yielding the so-called *frame-to-sequence distance*.

Definition 1 *(Frame-to-sequence distance): For two sequences X^p, $p = 1...n$ and X^q, $q = 1...m$, for any frame $X_i^p \in X^p$*

$$d(X_i^p, X^q) = \min_{j=1...m} \|X_i^p h_j^q - g\|_2, \tag{3.15}$$

where g is the predicted Gaussian output to the matched input in the filter h^q.

Intuitively the best operator is searched to give the minimum deviation from the ideal output g. It means that this is the best operator, among all h_j^q, to represent the given frame X_i^p. Also, to determine the similarity of two distinct sequences, the mean deviation between both sequences is assessed and named *sequence-to-sequence distance*.

Definition 2 *(Sequence-to-sequence distance): For two sequences X^p, $p = 1...n$ and X^q, $q = 1...m$*

$$d(X^p, X^q) = \max(\text{mean}\{d(X_i^p, X^q)\}, \text{mean}\{d(X_i^q, X^p)\}), \tag{3.16}$$

The maximum between the two mean measures is taken to deal with the case where one sequence in contained in the other, when the two distance measures would be very different.

3.2 Conclusions

In this chapter we discussed forms of image representation for image and video processing. We showed some traditional approaches to the usual image examples and discussed their use. Also, some methods to achieve the representations were shown. Latter, in the second part of the chapter, a recent approach for high-dimensional data (which may comprise image and video sequences) representation was reviewed.

In this efficient high-dimensional data representation model, a mathematical model for mapping the high-dimensional data in a low-dimensional sub-space whose behavior is known is presented. This operator space allows several characteristics of the data to be stored and latter compared without the need to know the original representation of the image (or video) and stands without the need of keeping all the image information.

In what follows we use the optimal subspace representation, shown above, to develop an abandoned-object detection method in videos acquired from moving cameras. In the proposed framework, equation (3.15) is employed to measure the distance among frames from a reference and a target video, which may help to determine the existence of interesting video events in the target sequence.

Chapter 4

Object Detection: Optimal Operator-Space Pursuit Approach

As shown in the previous chapter, the optimal sub-space representation of images proposed in [1] is a powerful tool for projecting a high-dimensional data in a low-dimensional space, where all discriminating information is kept. Also, due to inherent characteristics of the representation it is easy to compute some form of a distance between two image sequences, if at least one of them is stored using the representation. In light of that, we propose to use such representation to compare the reference and target sequences obtained by an automatic video-surveillance system in order to detect some video event of interest. In fact, major differences between the two video sequences can be interpreted as an abandoned or missing object.

In this chapter we will discuss the usage of the optimal operator space pursuit framework of [1] as a part of an abandoned object detection system that compares two image sequences (reference and target) in order to detect whether there is an object in the frame or not. We will highlight some traits that may be particularly useful to the proposed method.

4.1 Image Transform Domain Based Similarity Comparison

During the development of the object detection algorithm, the first idea was to compare each frame of the reference and target videos in the image transform domain. It would be done by acquiring both reference and target images, then measuring their similarity via phase correlation.

To perform this measurement, we first compute the complex conjugate of the DFT of one frame of the reference sequence. Later the DFT of a frame from the other sequence (target sequence in our case) was found and multiplied with the

complex conjugate of the former frame. Finally, the inverse DFT (IDFT) of the multiplication was obtained. As the measure of similarity between the frames the maximum value of the IDFT of the previous multpiplication was used, as can be seen in algorithm 1.

Algorithm 1 Image Transform Domain Comparison

1: **for all** X **do**
2: $X\_trans(i) \leftarrow DFT(X(i))$
3: $X\_trans\_conj(i) \leftarrow conj(X\_trans(i))$
4: $Y\_trans(i) \leftarrow DFT(Y(i))$
5: $measure \leftarrow IDFT(X\_trans\_conj(i) * Y\_trans)$
6: $final\_measure \leftarrow \max(measure)$
7: **end for**

where $X(i)$ is the i^{th} frame of the reference sequence and $Y(i)$ is the i^{th} frame of the target sequence.

After some experiments using a set of simple binary images, that can be seen in figure 4.1, we found out that this method entails a small difference between equal images and those with artifacts (that could be considered abandoned objects).

The correlation between reference and reference(autocorrelation) images and the correlation between reference and target images is shown in figure 4.2. There are artifacts in target sequence frames 21 to 31. It is easy to see, from the plot, which frames have an object, even with the low difference between correlation values.

As the artifacts in figure 4.1 (e-h) were artificially inserted and the images are completely planar we later performed more realistic tests. In order to do so, we applied homography transforms, obtained from a real translational moving camera with some trepidations, to all reference and target images resulting in the images on 4.1 (i-p) . Later, we applied the same method to detect the correlation between the untransformed reference images and the transformed reference images (shown in figure 4.3 in red) and also between the untransformed reference and the transformed target (shown in figure 4.3 in blue.)

In figure 4.3 it is clear that small trepidation between reference and target images create a large decrease of the correlation between the images, and this decrease is far greater than the one observed when a simple artifact is present in the image. Thus, the use of this metric can be ruled out in the detection of abandoned object using moving cameras, because it is very susceptible to small differences between reference and target camera poses.

The main reason that may explain why this method does not work for the proposed application is that it does not handle well changes in the image. The exceptions are the circular shifts, since the phase correlation method is invariant to them. Figure 4.4 depicts the distortion on the correlation when the homography transfor-

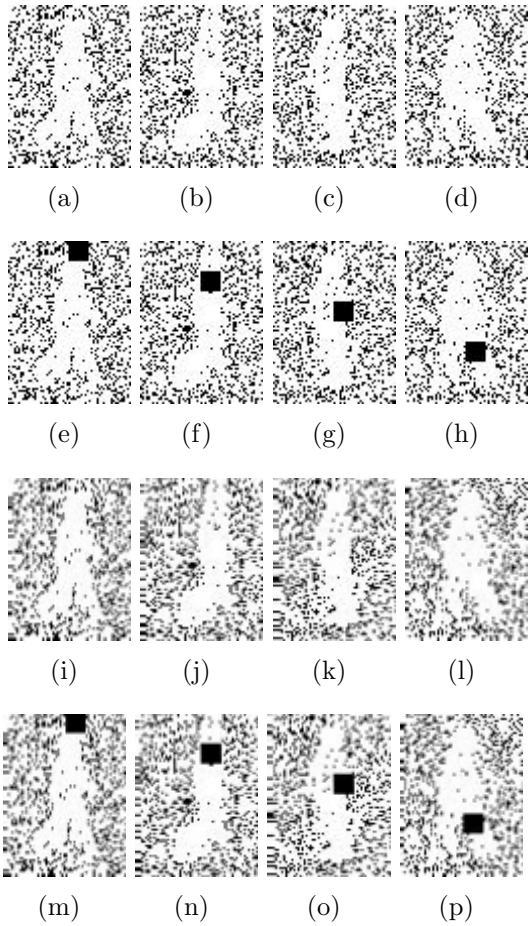


Figure 4.1: (a-d) Reference binary images; (e-h) Target binary images; (i-l) Transformed reference images; (m-p) Transformed target images;

mation is applied, lowering the maximum correlation and distorting the shape of the correlation.

4.2 Optimal Operator-Space Pursuit Approach

Since the comparison in the image transform domain did not work well for our application, lowering the correlation more in the case where camera misalignments happen than in the existence of video artifacts, we developed a different approach. Our goal was to employ the framework presented in [1], using the optimal operator space pursuit, to detect differences between a reference and a target videos that could help us to detect the occurrence of abandoned objects and other video events.

In this new framework, the reference sequence is used as a system input to form the optimal sub-space, generating the filters that will later be used to assess the similarity with the target-sequence frames. Although the process of finding the matched filters given by equations (3.13) and (3.14) can be computationally expensive, in a surveillance system the reference video may be available long before

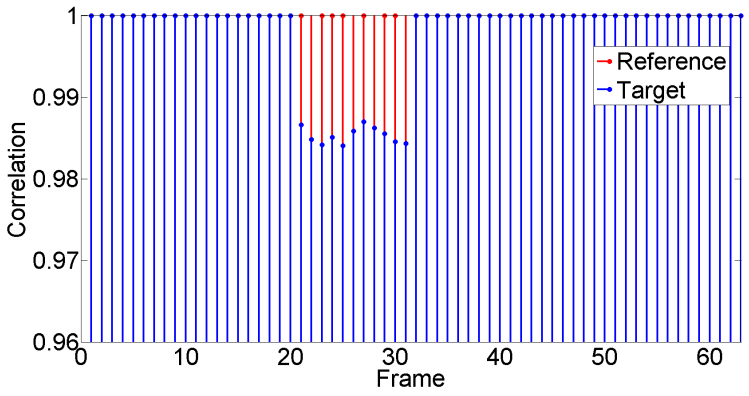


Figure 4.2: Correlation between sequences. Red points are the correlation between the reference sequence and itself. Blue points are the correlation between reference and target sequences.

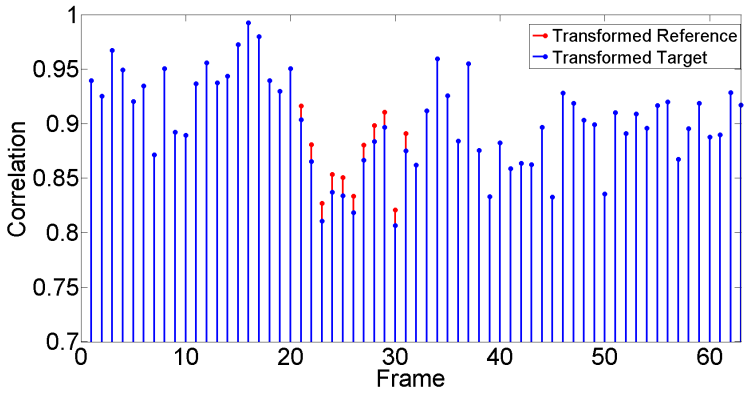


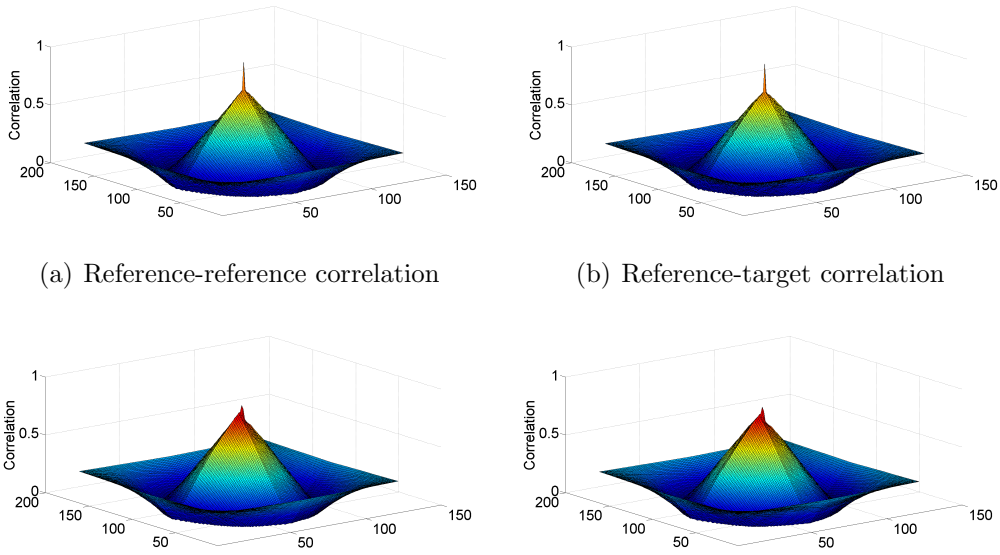
Figure 4.3: Correlation between sequences. Red points are the correlation between the untransformed reference sequence and transformed reference. Blue points are the correlation between untransformed reference and transformed target sequences.

the target one. Then, the process of finding the optimal sub-space can usually be done as an off-line task.

Also, in automatic surveillance systems, one can assume that the reference and target sequences are at least roughly temporally aligned. If this is the case, it is possible to simplify the measure of distance presented in equation (3.15), since there is no need for searching for the best correspondence in the whole reference sequence, which can be a very costly task in long videos, as usually surveillance videos are. In this sense equation (3.15) can be simplified to

$$d(X_i^p, X^q) = \min_{j=i-K\ldots i+K} \|X_i^p h_j^q - g\|_2, \tag{4.1}$$

where K is a pre-defined vicinity, where it is reasonable to find corresponding frames in the target video.



(a) Reference-reference correlation

(b) Reference-target correlation

(c) Reference-transformed reference correlation

(d) Reference-transformed target correlation

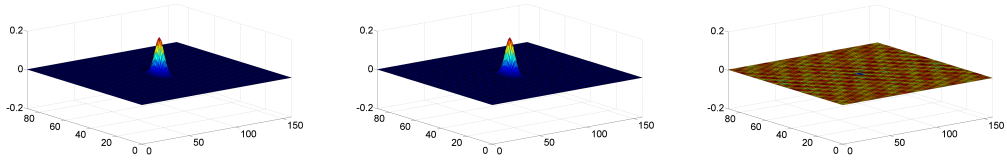
Figure 4.4: Correlation measures from different sequences.

Another recurrent requirement in the scope of automatic surveillance is the geometric registration among the sequences. This is specially important when moving cameras are employed, since they are susceptible to vibration that is uncorrelated to the trajectory. Such vibration may generate frames where the position of the camera is not the same as it was in the reference video, creating differences in the frame view. In this work we propose to avoid the need for registration by effecting comparisons of not only the expected output Gaussian function of a given filter, but also with shifted and rotated versions of this function, emulating small variations in the camera position.

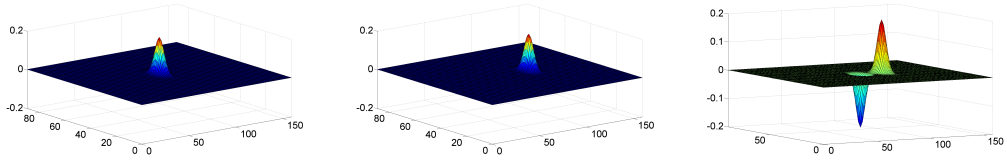
This has been implemented and we verified that the comparison with multiple versions of the expected output function solves the problem related to small shifts in the image, as can be seen in figure 4.5.

This method can fix the image shift problem. However, it does not deal properly with variations in the shape of the filter output, which can be caused by problems other than objects in the scene, like non-circular shifts in the image, as can be seen in figure 4.6. Also the exhaustive search for the perfect output shift to perform the subtraction entails a large extra computational cost.

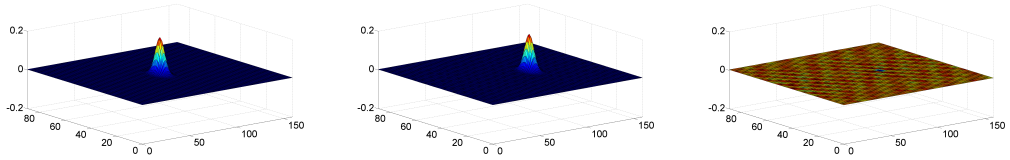
To deal with both of the problems presented above a distinct distance/similarity evaluation process was tested. Instead of performing a simple subtraction between both outputs a normalized correlation could be used, similar to that applied in the previous section, but applied to the filter output and the expected output of the filter. As final similarity measure the maximum value of the correlation between the



(a) Expected output (b) Real output (c) Real minus expected output

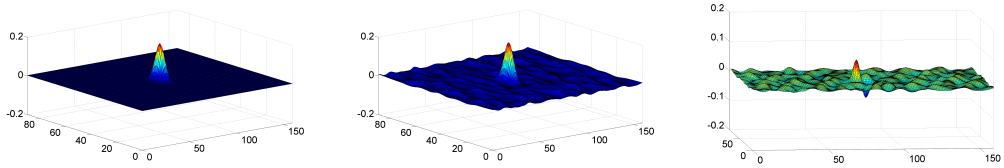


(d) Expected output (e) Real output (f) Real minus expected output



(g) Expected output (h) Real output (i) Real minus expected output

Figure 4.5: Subtraction solution: (a-c) Ideal scenario ; (d-f) Problematic scenario (image shift); (g-i) Exhaustive search solution;

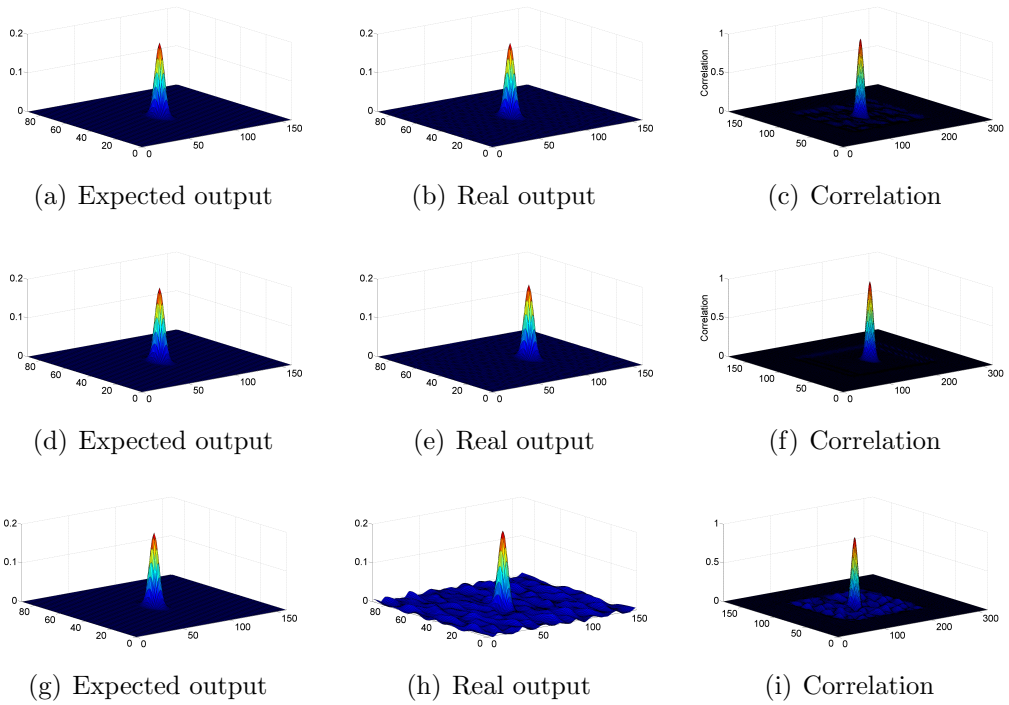


(a) Expected output (b) Real output (c) Real minus expected output

Figure 4.6: Subtraction solution: non-shift problem

outputs should be kept. Such a maximum correlation is invariant to shifts and thus avoids the exhaustive search if it is computed in the transform domain. Some cases of the correlation measure between filter outputs is shown in figure 4.7.

By the results shown in figure 4.7 it is possible to see that the proposed modified framework deals properly with some of the problems that can happen with a moving camera object detection application. Even in the face of image shifts and other undesired artifacts the method still conserves the output form and shows a high correlation between the frames. These tests suggested that it was worth to further investigate it for the application in the abandoned object detection system using moving cameras.



(a) Expected output (b) Real output (c) Correlation

(d) Expected output (e) Real output (f) Correlation

(g) Expected output (h) Real output (i) Correlation

Figure 4.7: Correlation solution: (a-c) Ideal scenario ; (d-f) Problematic scenario (image shift); (g-i) Non-shift problem;

4.3 Conclusions

In this chapter we presented two approaches to deal with the abandoned objects detection problem using a single moving camera which records a reference video, with no abandoned or missing objects, and compares it with a target video to find the anomalies that can be the objects of interest.

The first approach is not robust to variations in the camera pose and did not show significant quantitative differences between the frames with and without objects, even though the tests were made with artificially inserted artifacts that are simple to detect.

The second approach is is based in the framework presented in chapter 3, with the Optimal Operator Space Pursuit. This approach is much more robust to small errors in the camera pose and frame alignment. This is an indication that it is more suited to be used in a moving camera application.

Some changes in the proposed framework were presented to deal with problems that could come up when a the algorithm would face a real scenario.

In the next chapter a method is proposed based on the approach shown here. The method will be presented as it was developed showing the steps and results of each implementation.

Chapter 5

Proposed Method

In the previous chapter we presented a framework to be used in the abandoned objects detection with a moving camera problem. This framework is based on the method first shown in [1] and discussed in chapter 3. Some modifications were proposed to the original method so it could deal with some problems that commonly arise when moving cameras are employed in a surveillance task.

In this chapter a new method will be proposed to address the same problem using the ideas presented in chapter 3. The proposed method intends to detect abandoned objects in a cluttered environment without the need of a precise time-alignment between the videos, only needing the videos to be roughly aligned. Also, it should not be required the videos to be geometrically registered, which avoids most of the common implementation problems that appear when comparing two videos.

The proposed framework uses two video sequences. The first, called reference, represents the normal state of the observed area. It is considered to have no abnormalities, abandoned or missing objects nor any other video event that would cause the system to trigger an alarm. This video should be verified by a human operator to guarantee its normal state. The second video, called target video, is not verified by any operator and may have objects we would like the system to detect.

In the algorithm the target and reference videos are compared in order to detect abandoned or missing objects. Differently of the methods presented in [4] and [8] the proposed method does not find the position of the detected object inside a frame, but it shows instead in which frames there are anomalies that could be considered abandoned objects.

In this chapter we will present the algorithm in the chronology it was developed. We will justify every change based in the results of the previous step and in the end show the final form of the algorithm.

5.1 Database

To evaluate the quality of each step of the development, and also to verify the final quality of the system, the Video Database of Abandoned Objects in a Cluttered Industrial Environment (VDAO), developed in a previous work of ours, was used. The database description can be found in [9] and it can be download from [37].

This database is composed by 85 videos, being 6 multiple-object videos, 2 no-object (reference for the multiple-object videos) videos, 66 single-object and 11 no-object (reference for the single-object videos) videos. Since, in this work, the main purpose is to identify in which frames there are abandoned or missing objects, not caring if there are multiple objects in the scene, only the single-object videos were used.

The single object videos were acquired with the resolution of 1280×720 pixels, and a rate of 24 frames per second. Also, there are videos acquired with extra illumination and others with regular natural illumination. In the single-object videos, 9 different objects are used. The videos were recorded putting the 9 different objects in 3 positions each, which causes the objects to change size between videos. Some of the scenarios (position of the object in the environment) had to be recorded multiple times due to excessive trepidation of the camera in the first attempt, all versions are available in the database. The videos were acquired using a fixed color camera mounted over an iRobot Roomba©, in a cluttered industrial environment. In each video the robot passes twice through the whole scene following a fixed path that goes back and forth the environment. Each video has about 350 seconds (8300 frames) in average.

In figure 5.1 all objects presented in the single-object videos are shown. The scales of the objects have been changed, for better visualization.

In figure 5.2 the variation of size of the objects in the database due to changing position of the object is shown.

Figure 5.3 depicts the general setup of database recording using a moving camera on a robotic platform.

The videos in this database are only roughly aligned, that is a requirement for this framework. Also, as the cameras in reference and target videos have slightly different motions and speed, there can be misalignments during playback of the videos.

Another problem that usually arises in the kind of problems that this method aims to deal with, is the geometric mismatch between the videos. It is usually caused by a difference in camera pose, that usually can be modeled by a rotation around an arbitrary axis. An example of this type of camera mismatch is shown in figure 5.4. This kind of mismatch causes the image to show some regions of the environment

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

(g) (h) (i)

Figure 5.1: Objects used in the single-object videos (scales have been changed for a better visualization): (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.

(a) Position 1 (b) Position 2

Figure 5.2: Object size changing between videos in database.

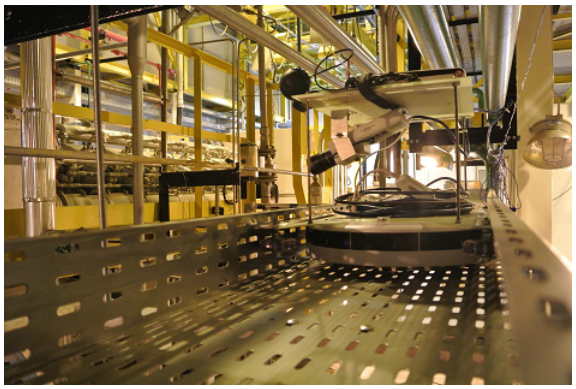


Figure 5.3: General setup of database recording using a moving camera on a robotic platform.

that are not shown in the original reference frame, which creates some innovations that usually jeopardize the detection of anomalies in the video. In this database these common video problems are present. Thus, its use which allows us to develop

solutions to these problems in a real scenario.

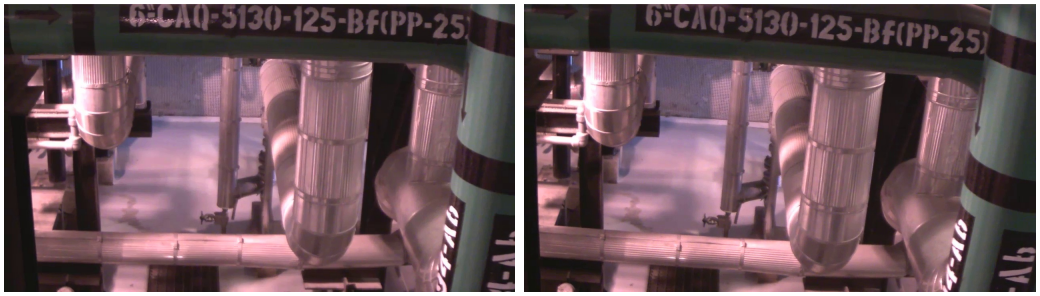


Figure 5.4: Example of frame mismatch due to camera rotation: (a) reference frame; (b) corresponding target frame.

5.2 First Implementation - Simple Detection

The first implementation of the method was the straightforward application of the original approach discussed in chapter 4. First the reference and target video sequences can be spatially downsampled, as the crucial information is usually still available in a smaller version of the image. This is possible as long as, given the thresholds, the dimensions of the smaller object you would like to detect are sufficiently large. In our case, considering the database's [9] image resolution of 1280 $\times$ 720 pixels, we downsapled it by a factor of 8 in each dimension, having a final frame of 160 $\times$ 90 pixels. This factor was chosen due to the limitation of available memory in the computer we used to test it and the amount of frames in each video sequence.

After the video is properly downsampled, the reference video sequence is used to find the ensemble of matched filters using the OOSP method [1]. This task is performed using equations (3.13) and (3.14). As said before, even though this part of the algorithm can be computationally expensive it can be performed offline, since the reference video may be available long before the target one.

With all the matched filters available each frame from the target sequence is filtered by the corresponding filter of the reference sequence. As the videos are, at least, roughly time-aligned, each frame from the target sequence is supposed to correspond to the filter of the same number, as given by

$$Y_i = \text{IDFT}(H_i \times \text{DFT}(\text{target}_i)), \tag{5.1}$$

where H_i is the Fourier transform of the matched filter corresponding to the i^{th} frame of the reference sequence, and target_i is the i^{th} frame of the target sequence.

For each output of the filtering step, the normalized cross-correlation with the

expected output of the filter was computed. As the final similarity measure, the maximum of each normalized cross-correlation was kept.

After performing the similarity measure through the normalized cross-correlation, a threshold is set. Any frame with a similarity measure lower than the pre-set threshold triggers an alarm, meaning there is an abandoned or missing object in the frame. Figure 5.5 summarizes the first implementation.

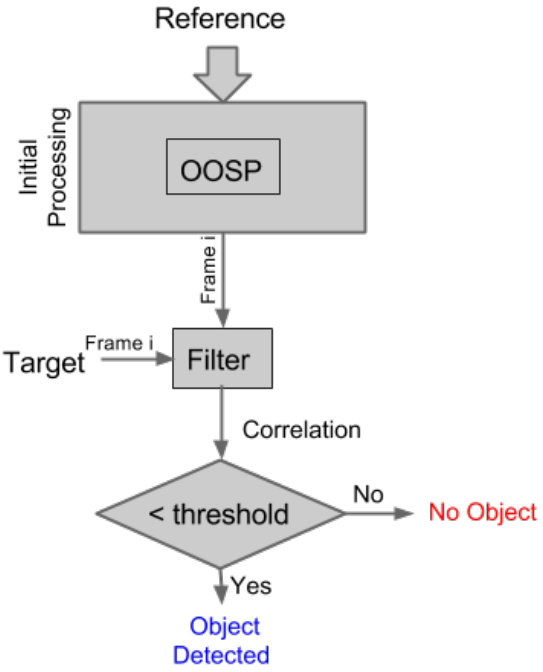


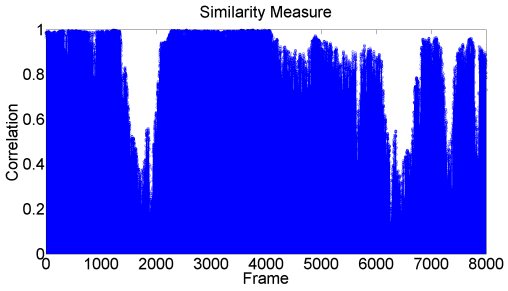
Figure 5.5: Block diagram of the first implementation of the object-detection system using operator-space approach.

Examples of the similarity measure using this implementation are shown in figure 5.6.

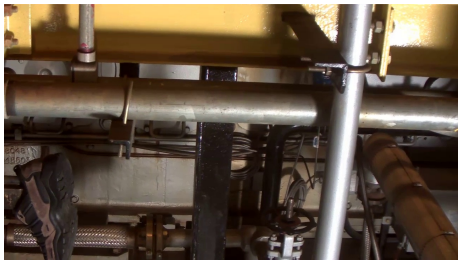
In the first sequence shown in figure 5.6 (a-b) the object is present between frames 1080 and 1820 and between frames 5820 and 6570. Outside these intervals there are no objects or video events of interest. In the sequence shown in figure 5.6 (c-d) the object is present between frames 730 and 1150, 1320 and 1680, 7100 and 7450, 7620 and 8050. Outside these intervals there are no objects or video events of interest. For both video sequences it is clear that the correlation between the frames decreases after the first half of the sequence. This happens mainly because in the end of the first half the camera changes its direction, but usually takes more time to do it in one of the videos, and that creates a misalignment between both sequences.

It is clear from figure 5.6 that this implementation has several problems, as it does not show very high similarity measures between frames from reference and target videos that have no abandoned objects. In some cases it shows such a low level of similarity that it seems to be a large object in the scene.

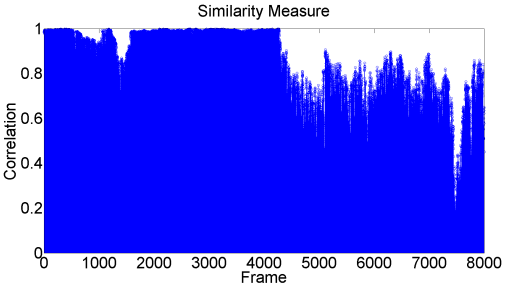
(a) Large object (blue box)



(b) Similarity Measure



(c) Small object (shoe)



(d) Similarity Measure

Figure 5.6: Framewise similarity measure between two sets of reference and target videos from the VDAO database. Low correlation values indicate the presence of the abandoned object in the target video.

5.3 Second Implementation - Aligned Detection

The second implementation of the algorithm was created to solve the main problem that was found in the last implementation. When the camera changes its direction it is common that it takes some time in a still position in the end of the track. The amount of time that it takes to begin to move again in the opposite direction can vary a lot, which causes the videos to lose their time-alignment. It is also possible that for other reasons, like slightly distinct speeds between the cameras, the alignment between the videos be progressively lost.

To solve the problem presented above and allow the system to work even if the sequences are not perfectly aligned or with those whose alignment change in time, we developed a method to create a fine alignment between the sequences during the detection process. The fine-alignment process works by searching in a vicinity around the reference frame what is the best correspondence to the target frame.

To check the best match between the reference and target sequence a vicinity of K frames is searched around the original target frame. The similarity measure is assessed for each of those frames and the pair with the best similarity measure is kept. This process is illustrated in figure 5.7.

The new implementation works in the following way. After loading and down-sampling the reference and target sequences, the reference video is used to find the

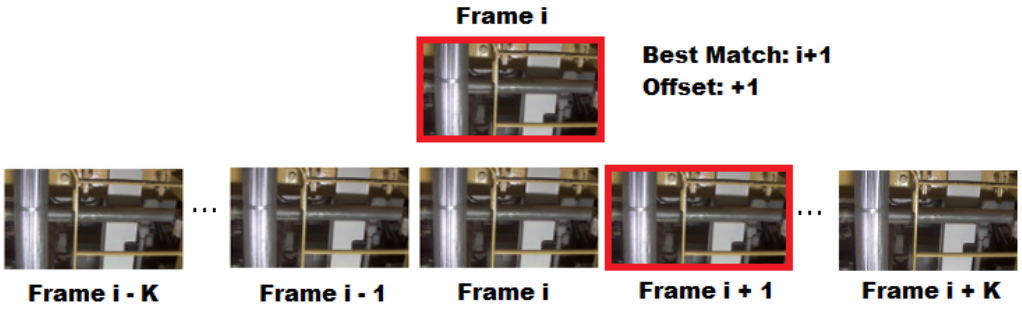


Figure 5.7: Example of the fine temporal alignment.

matched filters using the OOSP method [1], as in the previous implementation. Then, the correlation measure is assessed between all target frames and the corresponding reference frames using the fine-alignment method described before. After the best correspondence between each frame is taken, the similarity measure is kept as the maximum of each normalized cross-correlation between the best matched frames. Figure 5.8 summarizes this second implementation of the method.

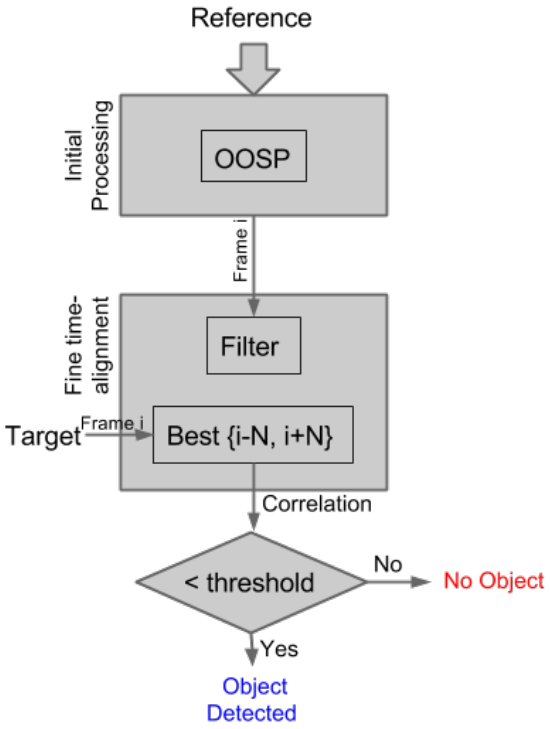
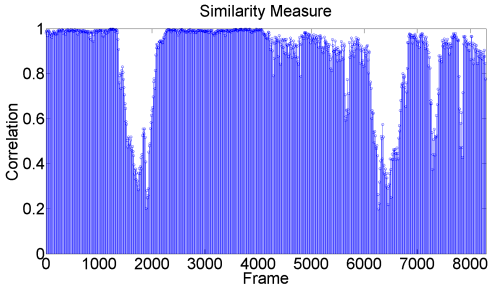


Figure 5.8: Block diagram of the second implementation of the object-detection system using the operator-space approach with the fine alignment system.

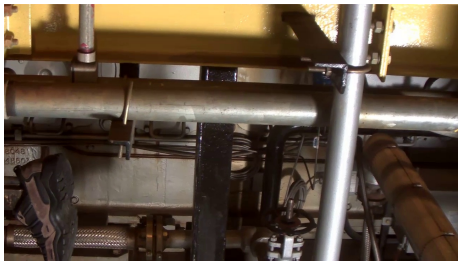
In our tests the variable K was chosen as 10, meaning that 21 frame correspondences were tested to find the best correspondence between the sequences: 10 frames to the left of the original corresponding frame, the original corresponding frame and 10 frames to the right of the original corresponding frame. This value of K was chosen based on a small portion of the database videos we used to do preliminary

tests of the algorithm. With this value most of misalignment in time could be solved with a reasonable increase in the computational cost. Also, due to the increase of computational effort in this new implementation we downsampled the target video for a factor of 10 in the time, yielding a frame rate of 2.4 frames per second in opposition to the 24 frames per second of the original video sequence. Examples of the similarity measure using this implementation are shown in figure 5.9.

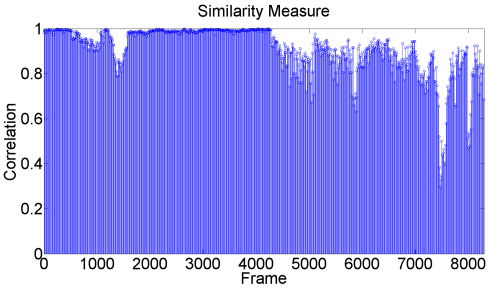
(a) Large object (blue box)



(b) Similarity Measure



(c) Small object (shoe)



(d) Similarity Measure

Figure 5.9: Framewise similarity measure between two sets of reference and target videos from VDAO database. Low correlation values indicate the presence of the abandoned object in the target video.

Again, in the first sequence shown in figure 5.9 (a-b) the object is present between frames 1080 and 1820 and between frames 5820 and 6570. Outside these intervals there are no objects or video events of interest. In the sequence shown in figure 5.9 (c-d) the object is present between frames 730 and 1150, 1320 and 1680, 7100 and 7450, 7620 and 8050. Outside these intervals there are no objects or video events of interest.

Although the modification applied in this implementation improves the results making the correlation between frames that have neither abandoned nor missing objects show a higher similarity measure, it is clear that it does not correct completely the problems with the correlation measure. The main reason that the improvement does not appear to be good is that the temporal misalignment between the sequences is not constant. Then setting a constant vicinity to the local search is not the best solution, unless the vicinity covers the entire reference video, which would entail a huge computational cost.

5.4 Third Implementation - Adaptive Alignment

The third implementation of the algorithm was divised to solve the alignment problem for the videos whose misalignment was greater than K frames, or where the misalignment was variable with the position of the camera. In the previous implementation it was clear that the misalignment for some videos was smaller in the first half of the duration and after that a vicinity of K frames would no longer solve the problem. In this implementation we propose to create an adaptive method that could solve the fine alignment problem in a more general scenario improving the similarity measure to the whole duration of the videos.

To enhance the fine-alignment method we propose an improvement in the method presented in the previous section. In the original algorithm, a fixed vicinity around the expected frame is evaluated, regardless of the previous offset between frames. We now propose that the offset chosen for the previous frame is is used as base for the next frame correspondence, as can be seen in figure 5.10 and is described in algorithm 2.

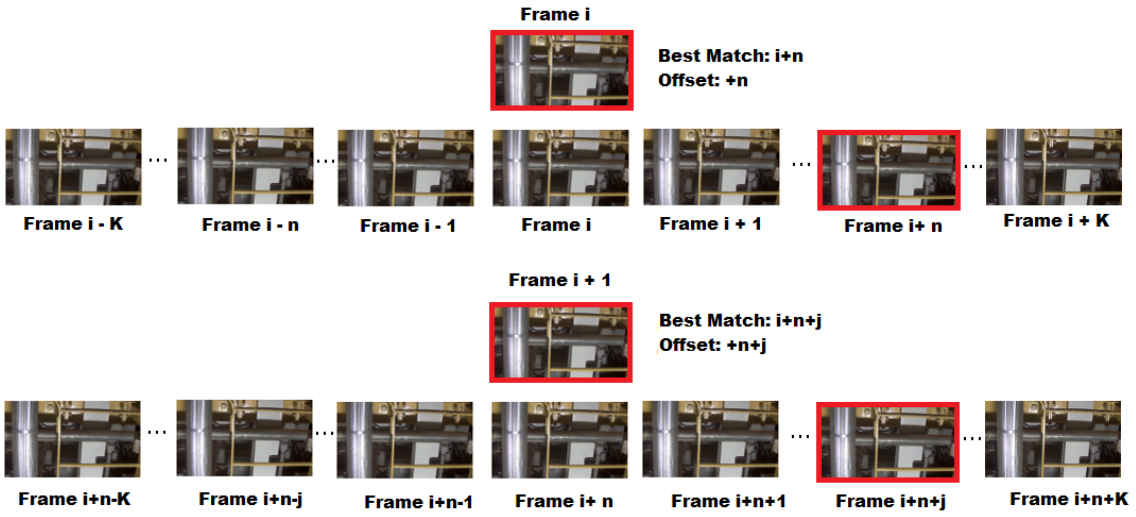


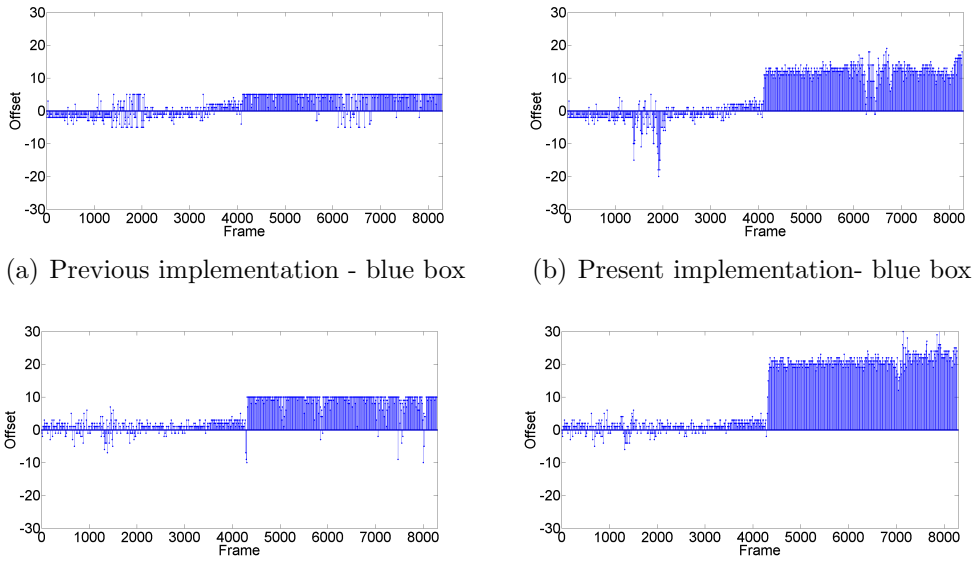
Figure 5.10: Example of the adaptive version of the fine temporal alignment.

Algorithm 2 Adaptive fine-alignment

```
for all X do
    offset ← 0
    for j=-K:K do
        aux(j) ← similarity_measure(X(i + offset + j), Y(i))
    end for
    [measure, local_offset] ← max(aux)
    offset ← offset + local_offset
end for
```

where $X(i)$ is the i^{th} frame of the reference sequence and $Y(i)$ is the i^{th} frame of the target sequence.

Figure 5.11 shows the difference of alignment offset between the previous and present implementations. It is clear from the plots that the original K-frame fixed window was not enough to deal with the misalignment, specially after the first half of the video sequence. In the new implementation the target video is better tracked and the difference in speed between the reference and target video is better compensated. In this implementation we kept the value of $K = 10$ from the previous one. This value was mantained due to the computer effort already calculated before and because, after evaluating a larger part of the database videos, we detected that with the new method this value could solve almost all misalignment problems.



(a) Previous implementation - blue box

(b) Present implementation- blue box

(c) Previous implementation - shoe

(d) Present implementation - shoe

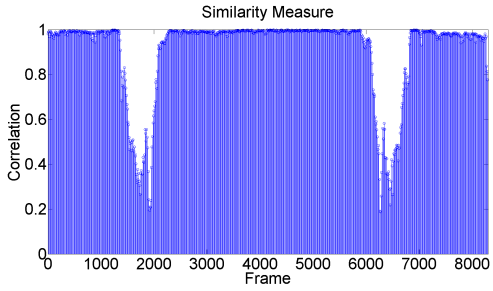
Figure 5.11: Difference between offsets in former and present implementations of fine-alignment algorithm.

Examples of the similarity measure using this implementation are shown in figure 5.12.

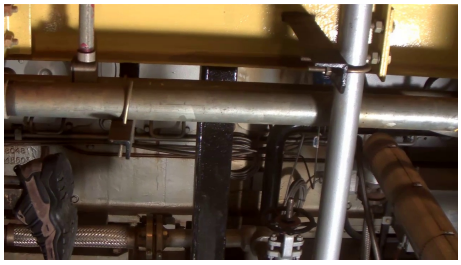
Again, in the first sequence shown in figure 5.12 (a-b) the object is present between frames 1080 and 1820 and between frames 5820 and 6570. Outside these intervals there are no objects or video events of interest. In the sequence shown in figure 5.12 (c-d) the object is present between frames 730 and 1150, 1320 and 1680, 7100 and 7450, 7620 and 8050. Outside these intervals there are no objects or video events of interest.

As can be seen in figure 5.12 the present implementation solves the problem it was intended to solve. But by the time we had it implemented, another kind of problem surfaced. Sometimes, the frames are not only time-misaligned, but also geometrically

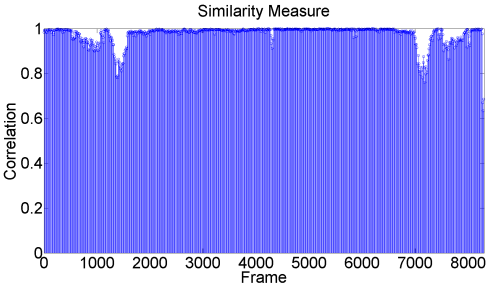
(a) Large object (blue box)



(b) Similarity Measure



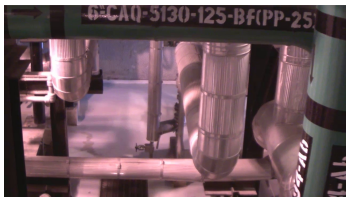
(c) Small object (shoe)



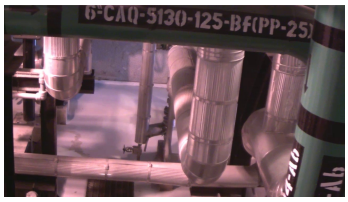
(d) Similarity Measure

Figure 5.12: Framewise similarity measure between two sets of reference and target videos from VDAO database when adaptive time-alignment is used. Low correlation values indicate the presence of the abandoned object in the target video.

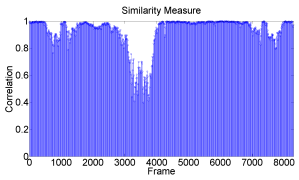
mismatched. This kind of problem cannot be solved by time-alignment, and it will be dealt with it in a latter implementation. An example of geometrical mismatch and the similarity measure with the present implementation is shown in figure 5.13



(a) Reference frame



(b) Target frame



(c) Similarity measure

Figure 5.13: Geometrical mismatch between frames and similarity measure with present method.

In figure 5.13 the object is present between frames 550 and 990, 1150 and 1520, 6950 and 7300, 7470 and 7910. Outside these intervals there are no objects or video events of interest. The low correlation outside these intervals is due to geometrical mismatch between the frames.

5.5 Final Implementation - Local Registration

The fourth and final implementation was designed to solve the geometrical mismatch between corresponding frames that could not be solved with time-alignment. If the similarity measure, after the time-alignment, is below a given threshold value, then that frame possibly contains an abandoned object or an observable video event of interest. However, low similarity values may also be caused by frame mismatches due to geometric transformations that cannot be accounted for by a simple cross-correlation between the filter output and the predicted Gaussian function. To solve this problem we would like to perform some kind of local registration between the frames. The objective is to perform this registration without using the traditional methods, such as the ones that employ detection of keypoints or image features in the process.

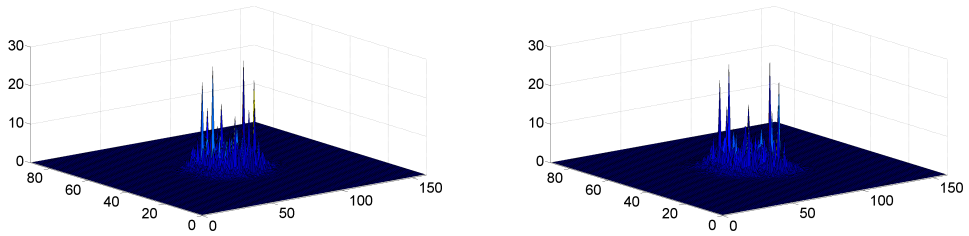
The remaining mismatch between the frames was, usually, caused by different camera positions when passing through the corresponding region. These mismatches can be modeled as rotations around arbitrary axes, as depicted in figure 5.4.

It is well known that a rotation around an arbitrary axis can be modeled by a rotation followed by a translation [20]. The classical way to address this problem would be to perform a registration between the frames by finding keypoints common to both frames and computing the geometric transformations between them [21].

To avoid such a computationally-intensive strategy, one way to deal with the above mentioned problem would be to search through all small rotations and translations of the current frame in the cases that a low correlation value is obtained. In the present case, this search can be made in the matched-filter domain, exploring the shift and rotation-invariant properties of the filter. In the transform domain, if a filter H is multiplied by an image X yielding an image Y, the filter H rotated by θ, when multiplied by the image X rotated by θ, outputs the image Y rotated by θ but up to some border effects [38], as may be seen in figure 5.4. The equivalence between rotation in the transform and image domain will be shown later in this chapter.

The first idea was to rotate the filters several times to find out which rotation yielded the best match between the frames and then perform the similarity measure. To perform this filter rotation, in the transform domain, we first have to convert the filter to the continuous domain, by interpolating it. Then we perform the rotation of the filter, resample and crop it to fit the same size as before. This process could cause the filter to have border effects due to information lost by the resampling and cropping procedures. But since the filter is concentrated around a central area, there will be no problems related to border effect, as after resampling and cropping it the filter will loose no important information. In figure 5.14 the filter is shown,

and one can easily see that rotating and cropping it would not really distort it.



(a) Original filter (b) Rotated filter

Figure 5.14: Filter rotation without perceived loss

After performing some tests with this idea the results were not very promising. In the database experiments for small variations of rotation angle the similarity measure between the rotated reference and the target frame decreases only after an interval bigger than the usual relative rotation between the frames. Also, the rotation of the filter does not enhance the similarity measure even if we use the exact rotation angle. The relation between the rotation angle and the similarity measure is depicted in figure 5.15. In this example the optimal rotation would be of about 2 degrees.

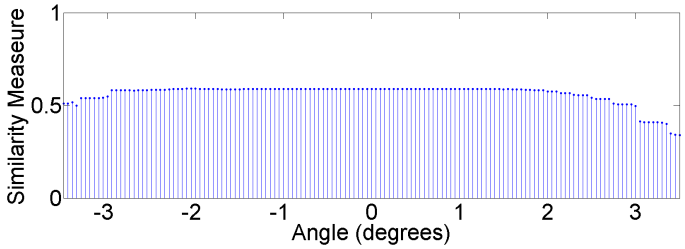


Figure 5.15: Relation between the rotation angle and the similarity measure.

After this first failed attempt to solve the geometrical registration problem, a different approach was used. Instead of performing the rotation in the filter domain the target frame is rotated and compared with the reference matched-filter. Both approaches are equivalent since, by the stretch theorem [39] a rotation in the spatial domain corresponds to an identical rotation in the frequency domain, as can be seen in the following equations

$$X(\vec{\omega}) = \sum_{\vec{n}} X(\vec{n}) \exp^{-j\vec{\omega}^{\mathrm{T}}\vec{n}} \tag{5.2}$$

$$\vec{m} = H\vec{n} \tag{5.3}$$

$$\vec{\omega'}^{\mathrm{T}}\vec{m} = \vec{\omega}^{\mathrm{T}}\vec{n} \tag{5.4}$$

$$\vec{\omega'}^{\mathrm{T}}H\vec{n} = \vec{\omega}^{\mathrm{T}}\vec{n} \tag{5.5}$$

$$\vec{\omega'}^{\mathrm{T}} H = \vec{\omega}^{\mathrm{T}} \tag{5.6}$$

$$\vec{\omega'}^{\mathrm{T}} = \vec{\omega}^{\mathrm{T}} H^{-1} \tag{5.7}$$

$$\vec{\omega'} = H^{-\mathrm{T}} \vec{\omega} \tag{5.8}$$

Being H is a rotation matrix,$H^{-1} = H^{\mathrm{T}}$ and $H^{-\mathrm{T}} = H$, then finally

$$X(H\vec{\omega}) = \sum_{\vec{n}} X(\vec{n}) \exp^{-j\vec{\omega}^{\mathrm{T}} H \vec{n}} \tag{5.9}$$

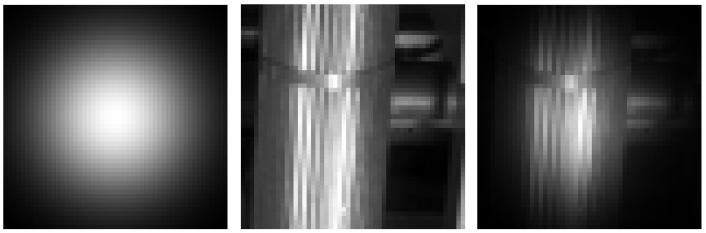
Since trying many rotation angles would require a great computational cost offset, we used an $N \times N$ window to detect which rotation would be the ideal match between the filters. Also, as we verified before, since the rotation around an arbitrary axis can be decomposed as a rotation followed by a translation, along with the rotation we will perform shifts in the window to find the best match.

Border effects reduce the normalized cross-correlation even in the case of accurate computation of the rotation, and leads to false detections. To deal with them, we decided to apply an $N \times N$ Gaussian window to the center of the reference frames before computing the corresponding matched filters. In our implementation, we used $N = 21$ pixels. Next the normalized cross-correlation is computed between the filter's output to translated and rotated versions of a target-frame input and the desired Gaussian function. In this search, we chose a maximum rotation angle of 3.5^{o} with steps of 0.25^{o} and a maximum translation movement of 20 pixels. The highest correlation value for all these rotation and translation values indicates the proper rotation-translation combination for the given frame, and these values are used to initialize the search for the subsequent target frame. Figure 5.16 illustrates the idea behind the method.

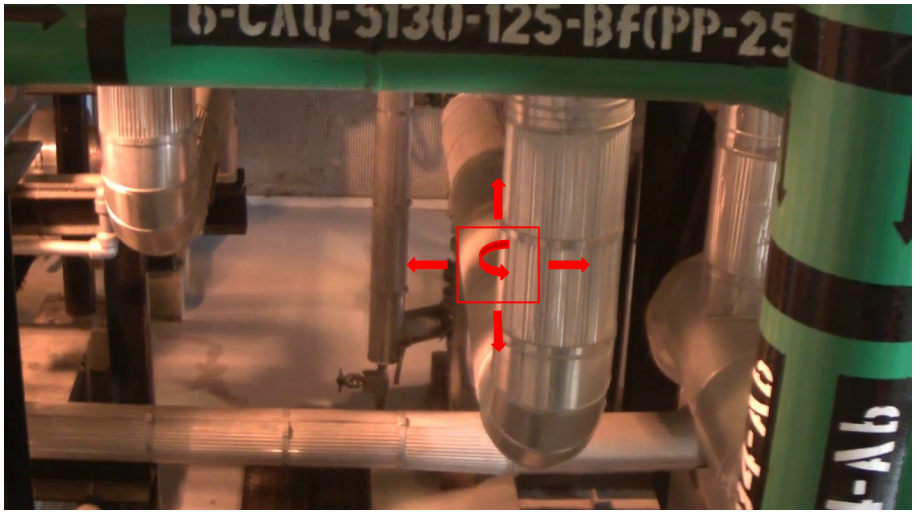
The size of the window was chosen as, in the tested videos of the database, it is the smaller window that could retain enough information about the image to perform this local registration. The values of the maximum horizontal and vertical translations and the rotation angles and steps were chosen based on the typical mismatches between reference and target frames in the evaluated videos of the database. Usually a 20 pixel shift is enough to contain the best match and the rotation angles rarely are greater than 3.5 degrees.

Similarly to the calculation of the matched-filter, that can be done prior to the acquisition of the target sequence, the calculation of the matched-filters of the $N \times N$ windowed sample of the center of each reference frame can also be performed offline, lowering the computational effort during the calculation of the local registration.

After this process, another Gaussian window is applied to the whole reference image to construct a final filter (Figure 5.17a). The same window is applied to a shifted and rotated version of the target frame (Figure 5.17b), according to the best



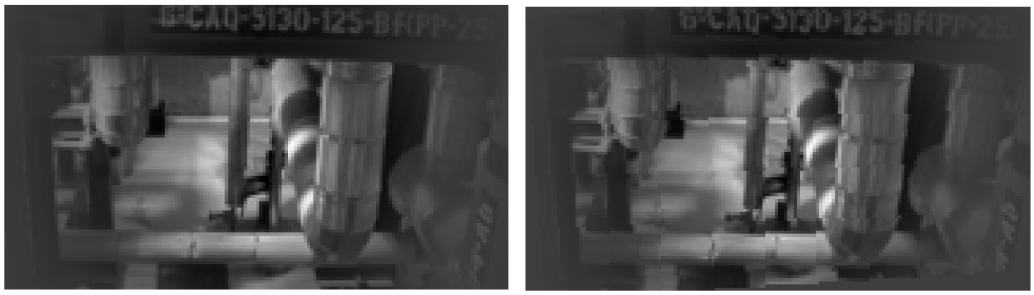
(a) Gaussian Window (b) Frame sample (c) Windowed sample



(d) Window movement

Figure 5.16: Example of local frame registration.

result in the last step, and the final correlation is assessed. If this value is lower than a certain preset threshold, it is considered to be an abandoned (or missing) object or a video event in the frame. Otherwise it is considered to be a similar frame with no observable events.

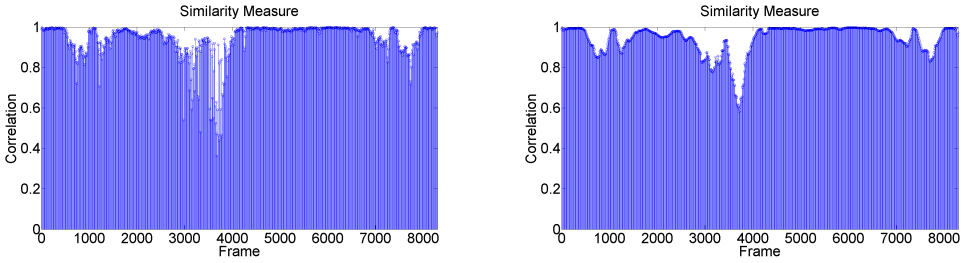


(a) Windowed reference frame (b) Registered target frame

Figure 5.17: Example of final frame comparison

Finally, after the similarity measure is taken for all frames in a video sequence, a smoothing filter is applied over the results to avoid wrong object detection due to short duration errors in the original similarity measure. The comparison between the detection with and without the smoothing filter is shown in figure 5.18. In our

application the filter replaces the value of the central point with the mean of the 5 point vicinity around it.



(a) Original measure (b) Smoothed measure

Figure 5.18: Similarity measure smoothing comparison.

Figure 5.19 shows the block diagram of the final implementation of the abandoned object detection algorithm using moving cameras based on the OOSP method and algorithm 3 describes it.

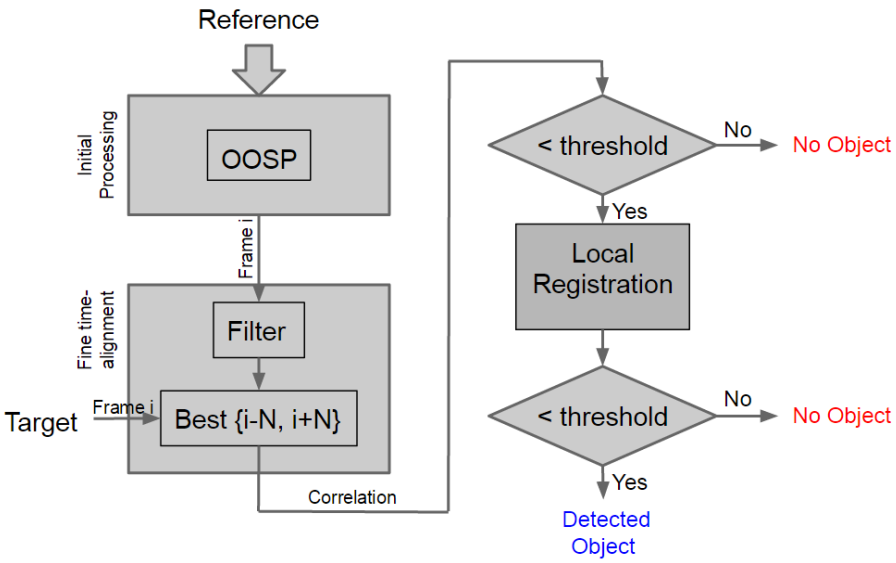


Figure 5.19: Block diagram of the final implementation of the object-detection system using operator-space approach.

After the implementation of the above mentioned enhancements we had results like those shown in figure 5.20.

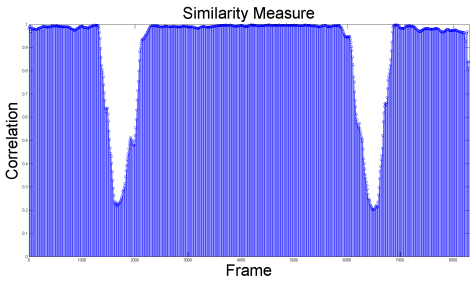
Once again, in the first sequence shown in figure 5.20 (a-b) the object is present between frames 1080 and 1820 and between frames 5820 and 6570. Outside these intervals there are no objects or video events of interest. In the sequence shown in figure 5.20 (c-d) the object is present between frames 730 and 1150, 1320 and 1680, 7100 and 7450, 7620 and 8050. Outside these intervals there are no objects or video events of interest.

A comparison between the results for the video shown in figure 5.13 (where the geometrical mismatch problem is a great issue) for former and new methods is

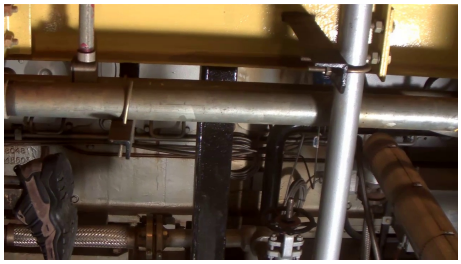
Algorithm 3 Final Implementation

```
X ← target_sequence
Y ← reference_sequence
H ← OOSP(Y)
Win = gaussian_window(Y, N × N)
H_win ← OOSP(Win)
for all X do
    offset ← 0
    for j=-K:K do
        aux(j) ← similarity_measure(X(i), H(i + offset_frame + j))
    end for
    [measure, local_offset] ← max(aux)
    offset_frame ← offset_frame + local_offset
end for
for all X do
    index ← 0
    offset_horizontal ← 0
    offset_vertical ← 0
    offset_angle ← 0
    if similarity_measure < threshold_1 then
        for j=-max_shift:max_shift do
            for k=-max_shift:max_shift do
                for angle =-max_angle:step:max_angle do
                    index ← index + 1
                    Z = gaussian_window(X(i), N × N)
                    Z = translate(Z, offset_horizontal + j, offset_vertical + k)
                    Z = rotate(Z, offset_angle + angle)
                    aux(index) ← similarity_measure(Z(i), H_win(i+offset_frame))
                end for
            end for
        end for
        [measure, offset_horizontal, offset_vertical, offset_angle] ← max(aux)
        W = gaussian_window(Y(i + offset))
        X_win = gaussian_window(X(i))
        W = translate(W, offset_horizontal + j, offset_vertical + k)
        H_W ← OOSP(W)
        [final_measure] ← similarity_measure(Y_win, H_W)
    end if
end for
```

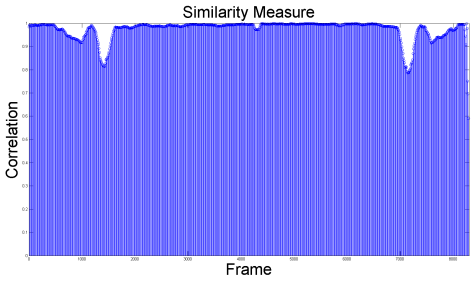
(a) Large object (blue box)



(b) Similarity Measure



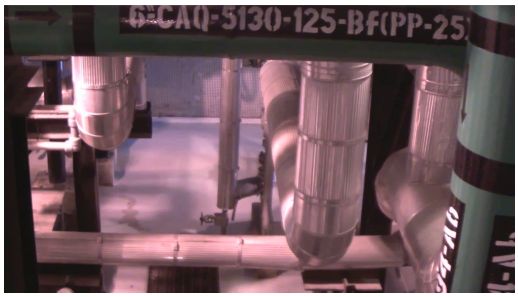
(c) Small object (shoe)



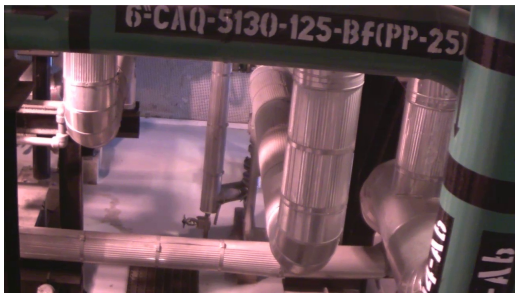
(d) Similarity Measure

Figure 5.20: Framewise similarity measure between two sets of reference and target videos from VDAO database. Low correlation values indicate the presence of the abandoned object in the target video.

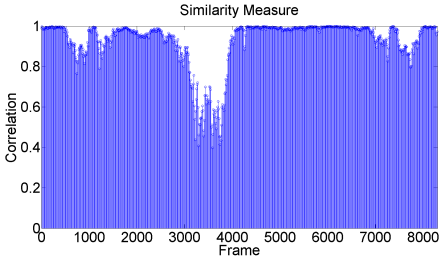
depicted in figure 5.21. The plots clearly show the increase of the similarity measure in the regions where there are no objects of interest and the maintenance of the low values in the regions where there are objects.



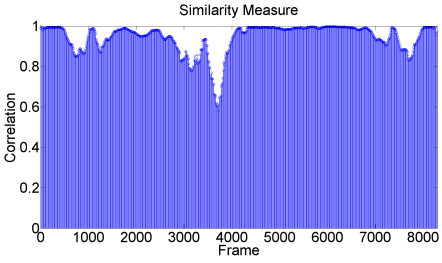
(a) Reference frame



(b) Target frame



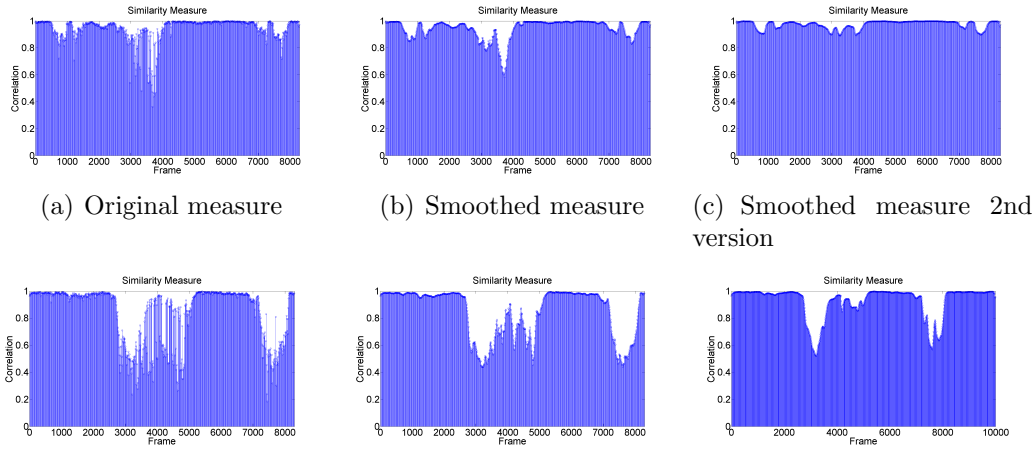
(c) Similarity measure former method



(d) Similarity measure present method

Figure 5.21: Geometrical mismatch between frames and similarity measure with former and present method.

Later we tried a different method of post-processing to smooth the output of the similarity measure, by replacing a central point by the maximum in a 5 point vicinity around it prior to the application of the mean filter. This method was tried due to the great amount of videos where the similarity measure showed erroneous low correlation points between correct high correlation point that were lowered by the previous smoothing filter. This attempt did not interfere with the correct detections in our experiments but removed some false detections found with the previous method. The results are shown in figure 5.22 comparing the original results, the first smoothing method and the new smoothing method for this final implementation.



(a) Original measure (b) Smoothed measure (c) Smoothed measure 2nd version

(d) Original measure (e) Smoothed measure (f) Smoothed measure 2nd version

Figure 5.22: Similarity measure smoothing comparison.

In these examples the objects are, in the first sequence (represented by figure 5.22(a-c)), between frames 550 and 990, 1150 and 1520, 6950 and 7300, 7470 and 7910. In the second sequence (represented by figure 5.22(d-f)) are between frames 2730 and 3690 and between frames 7190 and 8140. Outside these intervals there are no objects or video events of interest. The low correlation outside these intervals is due to geometrical mismatch between the frames. In both examples shown here it is easy to see the improvements of both smoothing methods, as the regions where there are objects become continuous and clear. The second smoothing method increases the value of the similarity measure for the region where there are no objects while keep the value of the regions where there are objects still low enough to be detected. Even though the improvement does not correct all false positive detections it still shows a better result than the previous version.

5.6 Experimental Results

The final implementation was tested with about 15 of the single-object videos from VDAO database [9]. The characteristics of this database are explained in section 5.1.

To perform these tests the two thresholds of the final implementation had to be set. The first one for the preliminar no-object detection and the second threshold for the locally-registered frames. These thresholds were set depending on the size of the smaller object to be detected. In the end, a value of 0.7 was sufficient to detect larger objects, 0.8 for medium objects, and 0.95 for very small objects.

Tests were made for different object sizes and shapes and the different illumination levels covered in the VDAO database. In some of the VDAO videos employed, there are intervals where the reference and target videos poorly match due to camera rotation and translation between the frames, as seen in figure 5.4. There is also a great deal of camera shake due to imperfections on the track. These characteristics allow the algorithm to be tested in situations where registration and salient-point detection would be required.

The experiments were designed to detect the target frames with a given abandoned object. In that sense, the detection performance was assessed by the number of true-positive detections (frames with object properly detected), number of true-negative detections (frames without objects correctly undetected), number of false-positive detections (frames without objects incorrectly detected) and number of false negative detections (frames with objects improperly undetected). Table 5.1 shows the results of the object detection for 15 videos from the VDAO database, with about 350 seconds (8300 frames) per video in average.

Table 5.1: Experimental results (frames) for proposed object-detection system.

	Positive	**Negative**
True	23540/26900 (87.51%)	86430/91000 (94.98%)
False	4570/91000(5.02%)	3360/26900 (12.49%)

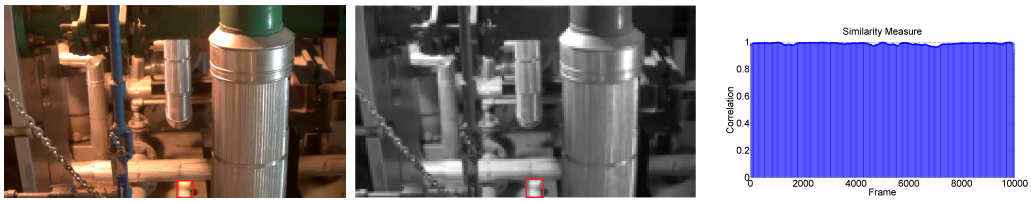
In the performed experiments, it is easy to observe that most of the frames were correctly categorized. In the case of the frames that were considered false positives, the most common cause is a mismatch between the frames due to rotation and vertical translation with larger amplitudes than those predicted in the implementation of the system. Even when the rotation or translation is greater than the pre-set search interval it can be found due to the adaptive search method that was implemented, where the ideal matching position of the previous frame is used as starting point to the search of the next one. But, sometimes, the shaking camera makes neighbour frames to have a relative rotation greater than that predicted in the system, thus not allowing the system to compensate the camera movement.

Also, the main cause of frames being considered false negatives is that, sometimes, the object appears only partially in the frame, reducing its effective size. This occurs in the cases of partial occlusion of the object and also when the object is entering or leaving a frame. Figure 5.23 shows this situation occurring.

(a) Object entering scene (b) Object leaving scene (c) Partial occlusion of the object

Figure 5.23: Problematic detection cases.

Another common problem is to detect objects with very small sizes and low contrast with the background. These objects tend to disappear when the image is downsampled and become very hard to detect. An example of it is shown in figure 5.24. In this sequence the object appears between frames 5130 and 5630. Outside this interval there are no objects or video events of interest.

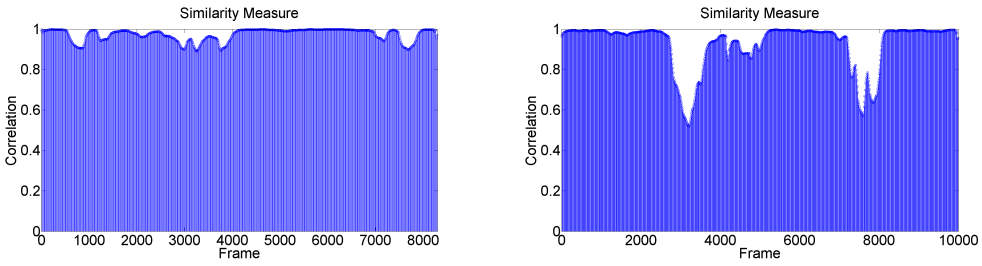


(a) Original frame (b) Downsampled frame (c) Similarity Measure

Figure 5.24: Problematic detection of very small object with low contrast.

During the selection of the thresholds it is important to decide the size of the smaller object to be detected. This decision is a compromise between the size of the detected objects and the greatest mismatch that can exist in the sequence without being perceived as a false object. In figure 5.25 there is a comparison between two sequences. In figure 5.25 (a) the similarity measure for the region where there are objects is about the same values of the measure of the region with the geometrical mismatch. In figure 5.25 (b) the similarity measure of the region where there are objects is smaller than that of the region where there is a geometrical mismatch.

It is important to mention that at a higher level, all abandoned objects were properly detected in all VDAO videos considered.



(a) Small object (b) Large object

Figure 5.25: Difference between the value of the similarity measures for different sized objects.

5.7 Comparison With Existing Methods

In the literature, the two works that are closer to our work are [4] and [8]. Both of them aim at finding the objects inside the frame, rather than finding in which frames there are abandoned or missing objects.

The work in [4] intends to create a real-time application. In contrast our method is too complex to run in real time, as it takes around 80 seconds to process a frame that potentially has an object (that is, whose similarity measure is lower than the first threshold) and about 2 seconds to process a frame that does not have an abandoned object (that is, whose similarity measure is higher than the first threshold). These complexity figures correspond to a Matlab© implementation using an Intel©Core™ i7-3630QM processor with a clock rate of 2.4 GHz, and with 16 GB of RAM.

An interesting characteristic of our method is that it does not depend on the motion of the camera, in opposition to the method in [4]. It can deal (theoretically) with subtle changes in the camera motion, as long as the reference and target videos were shot along the same paths and the fine-alignment algorithm is able to track the movement. In [4] the camera motion has to follow a preset path with a minimum number of direction changes and the videos have to be finely aligned. In addition our method does not require any direction changes, making it possible to work on a closed camera path. Also, unlike of our method, the one in [4] makes a geometrical registration of the frames using keypoints detection prior to search of the objects.

As for the quality of object detection both methods seem to have a similar accuracy, as both can detect all objects in the scene and present problems with occlusions and objects entering or leaving the frame. Also, both methods work with different settings to detect objects of distinct sizes, needing to group the objects of interest by size to determine which configuration to use.

The work in [8] does not give many details on the experimental results, but, as it uses the same database as in our work, a comparison is possible. The results of [8]

are obtained in the multi-object videos of the VDAO database [9]. A similarity of the method in [8] with ours is that it intends to detect objects without the need of time-alignment between the reference and target videos. In a perfect scenario the method in [8] would not need any kind of time-alignment, as it uses the other frames to reconstruct the present frame. However, as it needs a considerable computational effort, the tests of performance were made only with short versions of the videos (reference and target videos with 70 and 50 frames, respectively).

Another characteristic that makes our method similar to that of [8] is that neither of them depend on pre-registered videos and have their own mechanisms to compensate the movement between the scenes. However the method in [8] may not deal well with the border effect caused by anomalous movements of the camera.

Both approaches rely on statistical methods to process the data in a more friendly environment and avoid unexpected results.

The two methods in [4] and [8] do not seem to be on their final states, and possibly will show improvements and better results in a short future. Along with this one they propose different approaches to deal with the same problem, while using distinct mathematical and engineering tools to perform their tasks.

5.8 Conclusions

This chapter presented a new method to detect abandoned objects and video events in a cluttered environment without the need of previous registration or fine temporal alignment. It performs its own fine time-alignment and local registration between the sequences, without the need for any feature or keypoint detection prior to the alignment and registration process.

The method is based on an optimized sub-space representation of frames that allows the comparison between images to be more robust. The method is able to cope with visually complex environments without the use of feature based registration, that is not a very robust procedure in this kind of environment.

The main steps taken to design and implement this method were explained and the improvements of each step were highlighted. Also, the main difficulties and problems were shown, and the method's limitations were presented.

In the next chapter the final remarks about this project will be presented along with the future steps in the improvement of this work.

Chapter 6

Conclusions and Future Work

This thesis presented a novel method to be used in the detection of abandoned objects using a moving camera. The method uses the Optimal Operator Space Pursuit framework discussed in Chapter 3 along with several modifications to fit the applications and the environment of interest.

In the proposed method, two sequences, namely reference (with no abandoned nor missing objects) and target (potentially containing abandoned or missing objects) sequences, are compared to detect whether there are abandoned or missing objects in the videos. To do so the reference sequence is used to generate an ensemble of matched-filters whereby the target frames are filtered generating outputs. These outputs are compared, through normalized cross-correlation, with the expected output of the system and the maximum of this correlation is used as a similarity measure, for the frames. The detection of the object is done through a threshold of that similarity measure, meaning that if the similarity measure is below the threshold there is an object in the frame.

Most object detection frameworks using moving cameras depend on time-aligned videos to perform the detection. The time-alignment requirement entails a complexity in the pre-processing of the videos. Also, it can be a problem in systems where the camera path is a closed loop, and does not have any direction changes in the camera movement.

Another common issue related to the detection of objects with a moving camera is the need of geometrical registration between frames of the reference and target sequences. Usually the algorithms that perform the registration are both computationally costly and their use requires the scene to satisfy some conditions. Some of these conditions are related to the type of movement of the camera or the type of scenario, and they are usually not robust in the type of environment we are interested in working with (cluttered industrial environment).

With our framework, we propose alternative solutions to deal with the time-alignment and geometrical registration problems. We perform a fine time-alignment

between the sequences (thus requiring only a priori roughly aligned video sequences) and a local registration, that works in simpler ways than the traditional video registration algorithms.

One of the major drawbacks of the proposed method is that the computational requirements can be high. The current implementation in Matlab© takes up to 80 seconds to process a single 160×90 pixel frame where there is a possibility of existing an object (passes the first threshold) and up to 2 seconds if there is no object in it (does not pass the first threshold) on a quad core Intel©Core™ i7-3630QM processor with a clock rate of 2.4 GHz and with 16 GB of RAM, as described in chapter 5. The long time needed to process a single frame makes the proposed method not viable for work in a real-time automatic detection system. A possible solution to this problem may be a more robust and fast implementation of the method, using another programming language. In the future we intend to create a different implementation using C++ programming language to evaluate the processing speed of the method with a faster framework.

Another feature of the method that needs to be observed is the maximum rotation angle and translational movement that it can correct for mismatched frames. In the current implementation the method can deal with virtually any rotation and translation between frames within certain limits. These limits cannot be extended without increasing the complexity of the algorithm. In that sense, if the system can be improved to be faster, then we can improve its capacity of correcting previously mismatching frames, thus lowering the false positive detection rates.

Some suggestions for future works that may improve the analysis found in this thesis is to perform further tests using more videos from the database we worked with and also videos from different databases. These tests may show what are the main issues that have to be dealt with and help us set guidelines to the selection of the thresholds and of other parameters of the method.

Finally, this method can, theoretically, be used to perform the detection of objects in a system using a camera performing a closed path trajectory, but this feature was not yet tested. To perform such a test it would be necessary to use a different database containing videos acquired with cameras moving in a closed path.

The results presented in this work are promising, showing that the proposed method may be viable to implementation in automatic surveillance systems using moving cameras. We have proposed new ideas that can be used in scenarios that are neither simple nor have been well explored in the literature.

Bibliography

[1] BIAN, X., KRIM, H. "Optimal Operator Space Pursuit: A Framework for Video Sequence Data Analysis". In: *Computer Vision*, v. 7725, *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 760–769, 2013.

[2] TSINKO, E. *Background Subtraction with a Pan/Tilt Camera*. Master Thesis, University of British Columbia, Vancouver, Canada, 2010.

[3] HAYMAN, E., EKLUNDH, J.-O. "Statistical Background Subtraction for a Mobile Observer". In: *Proceedings of the Ninth IEEE International Conference on Computer Vision*, pp. 67–74, 2003.

[4] CARVALHO, G., DE OLIVEIRA, J. F. L., DA SILVA, E. A. B., et al. "Um Sistema de Monitoramento para Detecção de Objetos em Tempo Real Empregando Câmera em Movimento". In: *Proceedings of Simpósio Brasileiro de Telecomunicações*, Fortaleza, Brazil, September 2013.

[5] DIEGO, F., PONSA, D., SERRAT, J., et al. "Video Alignment for Change Detection", *IEEE Transactions on Image Processing*, v. 20, pp. 1858–1869, July 2011.

[6] DEY, S., REILLY, V., SALEEMI, I., et al. "Detection of Independently Moving Objects in Non-planar Scenes via Multi-Frame Monocular Epipolar Constraint". In: *Computer Vision*, v. 7576, *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 860–873, 2012.

[7] EVANGELIDIS, G., BAUCKHAGE, C. "Efficient and Robust Alignment of Unsynchronized Video Sequences". In: *Pattern Recognition*, v. 6835, Springer Berlin Heidelberg, pp. 286–295, 2011.

[8] JARDIM, E., BIAN, X., DA SILVA, E. A., et al. "On the detection of abandoned objects with a moving camera using robust subspace recovery and sparse representation", *Accepted for presentation at IEEE International Conference on Acoustics, Speech, and Signal Processing, Brisbane, Australia*, April 2015.

[9] DA SILVA, A. F., THOMAZ, L. A., CARVALHO, G., et al. "An Annotated Video Database for Abandoned-Object Detection in a Cluttered Environment". In: *International Telecommunications Symposium*, pp. 1–5, 2014.

[10] CHENG, L., GONG, M., SCHUURMANS, D., et al. "Real-Time Discriminative Background Subtraction", *IEEE Transactions on Image Processing*, v. 20, n. 5, pp. 1401–1414, May 2011.

[11] MUKHERJEE, D., WU, Q. M. J., NGUYEN, T. M. "Multiresolution Based Gaussian Mixture Model for Background Suppression", *IEEE Transactions on Image Processing*, v. 22, n. 12, pp. 5022 – 5035, December 2013.

[12] LIPTON, A. J., FUJIYOSHI, H., PATIL, R. S. "Moving Target Classification and Tracking From Real-Time Video". In: *IEEE Workshop on Applications of Computer Vision*, v. 14, pp. 8–14, 1998.

[13] WREN, C. R., AZARBAYEJANI, A., DARRELL, T., et al. "Pfinder: Real-Time Tracking of the Human Body", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 19, n. 7, pp. 780 – 785, July 1997.

[14] STAUFFER, C., GRIMSON, W. "Adaptive Background Mixture Models for Real-Time Tracking". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, v. 2, pp. 246–252, 1999.

[15] JODOIN, P.-M., SALIGRAMA, V., KONRAD, J. "Behavior Subtraction", *IEEE Transactions on Image Processing*, v. 21, n. 9, pp. 4244–4255, September 2012.

[16] TIAN, Y., FERIS, R., LIU, H., et al. "Robust Detection of Abandoned and Removed Objects in Complex Surveillance Videos", *IEEE Transactions on Systems, Man, and Cybernetics*, v. 41, n. 5, pp. 565–576, September 2011.

[17] DORE, A., SOTO, M., REGAZZONI, C. S. "Bayesian Tracking for Video Analytics", *IEEE Signal Processing Magazine*, v. 27, n. 5, pp. 46–55, September 2010.

[18] SALIGRAMA, V., KONRAD, J., JODOIN, P.-M. "Video Anomaly Identification", *IEEE Signal Processing Magazine*, v. 27, pp. 18–33, September 2010.

[19] SUBUDHI, B. N., NANDA, P. K., GHOSH, A. "A Change Information Based Fast Algorithm for Video Object Detection and Tracking", *IEEE Transactions on Circuits and Systems for Video Technology*, v. 21, n. 7, pp. 993–1004, July 2011.

[20] HARTLEY, R., ZISSERMAN, A. *Multiple View Geometry in Computer Vision*. 2nd ed. Cambridge, UK, Cambridge University Press, 2004.

[21] KONG, H., AUDIBERT, J.-Y., PONCE, J. "Detecting Abandoned Objects with a Moving Camera", *IEEE Transactions on Image Processing*, v. 2201-2210, pp. 803–806, August 2010.

[22] FISCHLER, M. A., BOLLES, R. C. "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography", *Communications of the ACM*, v. 24, n. 6, pp. 381–395, June 1981.

[23] SERAT, J., DIEGO, F., LUMBRERAS, F., et al. "Alignment of Videos Recorded from Moving Vehicles". In: *14th International Conference on Image Analysis and Processing*, pp. 512–517, 2007.

[24] LOWE, D. "Object Recognition from Local Scale-Invariant Features". In: *The Proceedings of the 7th IEEE International Conference on Computer Vision*, v. 2, p. 1150–1157, 1999.

[25] BAY, H., ESS, A., TUYTELAARS, T. "SURF: Speeded-Up Robust Features", *Computer Vision Image Understanding*, v. 110, n. 3, pp. 346–359, 2008.

[26] LEUTENEGGER, S., CHLI, M., SIEGWART, R. "Brisk: Binary Robust Invariant Scalable Keypoints", *IEEE International Conference on Computer Vision*, p. 2548–2555, 2011.

[27] ALAHI, A., ORTIZ, R., VANDERGHEYNST, P. "Freak: Fast Retina Keypoint", *IEEE Conference on Computer Vision and Pattern Recognition*, p. 510–517, 2012.

[28] KUCHARCZAK, F., DA SILVA, A. F., THOMAZ, L. A., et al. "Comparison and Optimization of Image Descriptors for Real-Time Detection of Abandoned Objects". In: *Anais do Simpósio de Processamento de Sinais da UNICAMP*, v. 1, 2014. Available at: <`http://www.sps.fee.unicamp.br/anais/`>.

[29] BEAUCHEMIN, S. S., BARRON, J. L. "The Computation of Optical Flow", *ACM Computing Surveys*, v. 27, n. 3, pp. 433–466, September 1995.

[30] BIAN, X., KRIM, H. "Robust Subspace Recovery via Dual Sparsity Pursuit", *Computing Research Repository*, v. abs/1403.8067, 2014. Available at: <http://arxiv.org/abs/1403.8067>.

[31] JAIN, A. K. *Fundamentals of Digital Image Processing*. London, Englewood Cliffs, N. J., Prentice-Hall International, 1989.

[32] CHOUDHARY, B. "The Elements of Complex Analysis". chap. Metric Spaces, New Delhi, India, New Age International Publisher, 1992.

[33] STEENROD, N. E. *The Topology of Fiber Bundles*. New Jersy, USA, Princeton University Press, 1951.

[34] BIAN, X., KRIM, H. "Video-Based Human Activities Analysis: An Operator-Based Approach". In: *20-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, 2012.

[35] CAI, J.-F., CANDÈS, E. J., SHEN, Z. "A Singular Value Thresholding Algorithm for Matrix Completion", *SIAM Journal on Optimization*, v. 20, n. 4, pp. 1956–1982, March 2010.

[36] GOLUB, G. H., LOAN, C. F. V. *Matrix Computations*. 3rd ed. Baltimore, USA, John Hopkins, 1996.

[37] "VDAO - Video Database of Abandoned Objects in a Cluttered Industrial Environment". [Online], 2014. Available at http://www.smt.ufrj.br/~tvdigital/database/objects.

[38] DUDGEON, D. E., MERSEREAU, R. M. *Multidimensional Digital Signal Processing*. New Jersey, USA, Prentice Hall, 1990.

[39] OSGOOD, B. "The Fourier Transform and its Applications". University Lecture, 2007. Available at http://see.stanford.edu/see/materials/lsoftaee261/handouts.aspx.

Appendix A

List of Articles Derived from this Thesis

DA SILVA, A. F., THOMAZ, L. A., CARVALHO, G., et al. "An Annotated Video Database for Abandoned-Object Detection in a Cluttered Environment". In: International Telecommunications Symposium, São Paulo, Brasil, pp. 1-5, 2014.

KUCHARCZAK, F., DA SILVA, A. F., THOMAZ, L. A., et al. "Comparison and Optimization of Image Descriptors for Real-Time Detection of Abandoned Objects". In: Anais do Simpósio de Processamento de Sinais da UNICAMP, v. 1, 2014. Available at: <http://www.sps.fee.unicamp.br/anais/>.

THOMAZ, L. A., DA SILVA, A. F., DA SILVA, E. A. B., et al. "Abandoned Object Detection Using Operator-Space Pursuit". In: Submitted to International Conference on Image Processing, Quebec, Canada, 2015.