UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

XANDER CORTEZ PINZAS

Background Subtraction Based on Robust Subspace Tracking

Thesis presented in partial fulfillment
of the requirements for the degree of
Master of Computer Science

Advisor: Prof. Dr. Jacob Scharcanski

Porto Alegre
July 2023

CIP — CATALOGING-IN-PUBLICATION

Cortez Pinzas, Xander

Background Subtraction Based on Robust Subspace Tracking / Xander Cortez Pinzas. – Porto Alegre: PPGC da UFRGS, 2023.

69 f.: il.

Thesis (Master) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2023. Advisor: Jacob Scharcanski.

1. Subspace learning. 2. Robust principal component analysis. 3. Robust Subspace Tracking. 4. Salient detection. 5. Background Subtraction. 6. Compressive Sensing. 7. Computer Vision. I. Scharcanski, Jacob. II. Título.

“Failure teaches what success hides.”

— ENRIQUE ROJAS

ACKNOWLEDGMENT

I would like to express my gratitude to all those who contributed to complete this work. Primarily, to my parents and siblings for their unconditional support and comprehension during all stages of my academic life. Moreover, I sincerely thank my advisor Dr. Jacob Scharcanski (UFRGS) for his guidance throughout the master's process. Also, I would like to thank to all my professors and colleagues at UFRGS for the knowledge transmitted during my graduate studies. Finally, I would like to thank Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) for the financial support.

Subtração de Fundo baseado em Rastreamento Robusto de Subespaço

RESUMO

Este trabalho propõe um método robusto de subtração de fundo online, que detecta o fundo e o primeiro plano em sequências de vídeo. Os métodos de subtração de fundo existentes tendem a ter um desempenho limitado em cenários de ambiente complexo. Neste trabalho, um método de rastreamento de subespaço é usado para atualizar continuamente, ao longo do sequências de vídeo, as estimativas de plano fundo e primeiro plano. O primeiro plano estimado é atualizado continuamente ao reestimar suas características salientes e descartando elementos que potencialmente fazem parte do fundo. Os resultados experimentais sugerem que a abordagem proposta tende a ser eficaz em vídeos com uma ampla gama de cenários complexos e demonstram que nosso método supera algoritmos de subtração de fundo on-line de última geração.

Palavras-chave: Aprendizagem de subespaço, Análise robusta de componentes principais, Rastreamento de Subespaço Robusto, Detecção de Saliency, Subtração de fundo, Sensoriamento Compressivo, Visão Computacional.

ABSTRACT

This work proposes a robust online background subtraction method, which detects the background and the foreground in video sequences. Existing background subtraction methods tend to have a limited performance in complex environment scenarios. In this work, a subspace tracking method is used to update continuously, along the video sequences, the background and foreground estimates. The estimated foreground is updated by continuously re-estimating its salient features, and discarding elements that potentially are part of the background. The experimental results suggest that the proposed approach tends to be effective in videos with a wide range of complex backgrounds, and demonstrate that our method outperforms state-of-the-art online background subtraction algorithms.

Keywords: Subspace learning. Robust principal component analysis. Robust Subspace Tracking. Salient detection. Background Subtraction. Compressive Sensing. Computer Vision.

LIST OF ABBREVIATIONS AND ACRONYMS

BS-RST	Background Subtraction Based on Robust Subspace Tracking
SL	Subspace Learning
LRMF	Low Rank Matrix Factorization
RPCA	Robust Principal Component Analysis
GRASTA	Grassmannian Robust Adaptive Subspace Tracking algorithm
ReProCS	Recursive Projected Compress Sensing
RPCA-STOC	Online Robust PCA via Stochastic Optimization
OMWRPCA-CP	Online Robust Principal Component Analysis with Change Point Detection
MoG	Mixture of Gaussian
OBSL1	Efficient Low-Rank Matrix Factorization Based on $l_{1,\epsilon}$-Norm for Online Background Subtraction
PCA	Principal Component Analysis
PCs	Principal Components
SVT	Singular Value Thresholding
RST	Robust Subspace Tracking
s-ReProCS	simple-ReProCS
NORST	Nearly Optimal Robust Subspace Tracking
SVD	Singular Value Decomposition
OMoGMF	Robust Online Matrix Factorization for Dynamic Background Subtraction
RPCA-MoG	Robust Principal Component Analysis with Complex Noise
SM	Salient Map
LS	Least Squares

LIST OF FIGURES

LIST OF TABLES

CONTENTS

1 INTRODUCTION

The successive progress of computer capacity has made it possible to process, analyze, and interpret images and videos, enabling the continuous development of research fields in computer vision, pattern recognition, and image processing, whose main objective is to process and analyze images and videos. These advancements have benefited various application areas such as remote sensing, medical diagnosis, human-computer interaction, video compression, intelligent video surveillance, traffic monitoring, visual analysis of animals and insects, and optical motion capture. In most of these applications, a minimum increase in precision is essential. Therefore, in recent decades, these research areas have aroused great interest in the scientific community and industry (BOUWMANS, 2014), (GARCIA-GARCIA; BOUWMANS; Rosales Silva, 2020).

In most video-based applications, a fundamental and critical step is detecting the foreground (moving objects) in a scene. Each scene comprises a background (generally understood as the more static part or, the slower changing portion of video scenes) and a foreground. Being able to discriminate the background from the foreground can be critical in several video analysis tasks, such as in video surveillance and anomaly detection, to name a few. The process of separating the foreground and background of a video frame is often known as background subtraction (BOUWMANS, 2014). It is also called foreground segmentation or layering video. Background subtraction is a widely used approach for this task. Also, this method seeks an adequate solution that balances computation time and detection quality. The focus of the present work is on the *Background subtraction* task.

To ensure good foreground detection, the videos must be generated using a static camera, constant illumination, and a static background. However, in real applications, it is not easy to have these conditions, and a wide variety of factors present different challenges and different types of interest in moving foreground objects. In (KALSOTRA; ARORA, 2019), these factors are classified according to the challenges present in the background, foreground, and camera; Background: Illumination changes, dynamic background, shadows, challenging weather, bootstrapping, moved background, and objects. Foreground: Shadows, Camouflage, intermittent object motion, occlusion, foreground aperture, and sleeping foreground. Camera: Video noise, moving camera, and camera jitter. Thus, the development of new background subtraction methods has become necessary. These methods should be able to adapt to previously mentioned significant changes and be robust in

the presence of noise. Additionally, with technological advancements, devices are becoming increasingly cheaper, such as computers, image scanners, digital cameras, and mobile phones generating a large amount of data. New background subtraction methods must consider these large data volumes to process and analyze them effectively and efficiently.

Different approaches to the background subtraction method have been developed to deal with these challenges present in videos. These methods can be based on fundamentals, statistical information, subspace learning, and deep learning: Some examples of methods based on fundamentals are mean subtraction (LEE; HEDLEY, 2002), median subtraction (GRASZKA, 2014) and histograms (ROY; GHOSH, 2017). However, these methods may fail in practical situations. For example, when there is a camera jitter, changes in illumination, and dynamic backgrounds. The methods based on statistical information may consider a mixture of Gaussians (WREN et al., 1997) (ELGUEBALY; BOUGUILA, 2013) (MUKHERJEE; JONATHANWU, 2012), the Student t-distribution (GUO; DU, 2012), the Dirichlet distribution (MUKHERJEE; JONATHANWU, 2012) (HAINES; XIANG, 2012), and the Poisson distribution (ZIN et al., 2014), to name a few. The statistical methods can handle dynamic backgrounds in video sequences. However, prior knowledge of the background often is required and may miss spatially adjacent objects in complex backgrounds.

In Subspace Learning (SL) methods (VASWANI et al., 2018), it is common practice to project each video frame into a low-rank subspace. The difference between the original frame and its projection onto the subspace allows us to obtain the foreground. Some examples of SL methods are (DIANA; BOUWMANS, 2010), (FARCAS; MARGHES; BOUWMANS, 2012), (MARGHES; BOUWMANS; VASIU, 2012). The SL methods tend to be robust to local changes (e.g. illumination). Still, they are sensitive to noise and outliers, and missing data that often are present during the video acquisition. These limitations motivated the development SL approaches such as robust principal component analysis (RPCA) (CANDèS et al., 2011), (MCCOY; TROPP, 2011), (ZHANG; LERMAN, 2014), and low-rank matrix factorization (LRMF). The LRMF methods express a data matrix as a product of two smaller matrices, namely, the low-rank matrix and the coefficient matrix, and can be categorized into three groups based on how noise is measured (error norm). The L_2-LRMF methods (TORRE; BLACK, 2003) (GABRIEL; ZAMIR, 1979) (SREBRO; JAAKKOLA, 2003) use the L_2 error norm, but tend to be sensitive the outliers. The L_1-LRMF methods (WANG et al., 2012) (ZHENG et al., 2012) are more robust to outliers. However, it may be affected by non-Gaussian noise, and the optimal so-

lution finding can be challenging since most L_1-LRMF methods have a non-differentiable cost function. Finally, the probabilistic LRMF methods can handle complex noise (DING; HE; CARIN, 2011) (BABACAN et al., 2012) (ZHAO et al., 2014) (CAO et al., 2015) using a combination of distributions. So, these methods tend to be adaptive to different types of noise that occur in practical scenarios.

In Deep Learning (DL) methods (BOUWMANS et al., 2019), it is common to model either the background or foreground using deep neural networks (DNNs). Some examples of DL methods are (SULTANA et al., 2019) (LIM; Yalim Keles, 2018) (LIM; KELES, 2018b) (LIM; KELES, 2018a) (GAO; LI; LU, 2021), (RAHMON et al., 2021) and (LIANG et al., 2023). These methods have demonstrated high performance compared to the mentioned methods, but they are supervised approaches that require a large amount of data for training and generalization.

The mentioned methods often operate in batches (i.e., require all data to be available before processing), so applying these methods to online processing or to long data sequences can be challenging. Robust Principal Component Analysis (RPCA) based methods often use a fixed low-dimensional subspace to represent the background, which can lead to high dimensional subspace when processing large data sets. Nevertheless, there are online versions of RPCA and LRMF methods that can help dealing with some of the above mentioned issues. Some examples of online RPCA mehods are Grassmannian Robust Adaptive Subspace Tracking algorithm (GRASTA, (He; Balzano; Lui, 2011)), Recursive Projected Compress Sensing (ReProCS, (GUO; QIU; VASWANI, 2014), (QIU et al., 2014), (NARAYANAMURTHY; VASWANI, 2018a), (NARAYANAMURTHY; VASWANI, 2018b), (NARAYANAMURTHY; VASWANI, 2019)), Online Robust PCA via Stochastic Optimization (RPCA-STOC,(FENG; XU; YAN, 2013)), and Online Robust Principal Component Analysis with Change Point Detection (OMWRPCA-CP,(XIAO et al., 2020)). GRASTA and ReProCS assume that a low dimensional subspace used for representing the background can slowly change in time, while RPCA-STOC assumes a fixed subspace, and OMWEPCA-CP assumes that the subspace used for representing the background can change substantially at any time. Some examples of online LRMF are Robust Online Matrix Factorization for Dynamic Background Subtraction (OMoGMF (YONG et al., 2018)), and Efficient Low-Rank Matrix Factorization Based on $l_{1,\epsilon}$-Norm for Online Background Subtraction (OBSL1 (LIU; LI, 2022)). The approach adopted by OMoGMF is to model the foreground by a mixture of Gaussians (MoG), which is dynamically updated online. So, the foreground and the background are detected adaptively. On the

other hand, OBSL1 is based on the $l_{1,\epsilon}$-Norm, which approximates the L_1 norm with the advantage of being differentiable, which allows to search for optimal and sub-optimal solutions. It shall be observed that most of the online foreground detection methods mentioned above (except for OMoGMF) do not consider the video (or image) structural information or the relationship that may exist among groups of pixels (or local features). In other words, the above-mentioned methods do not take into account prior information that may be available about the foreground and its features. However, in practice, neighboring foreground features tend to cluster in the image and the feature space since the foreground features persist in time and show identifiable structural properties. Therefore, these above mentioned methods tend to have limitations when processing real data.

In the present work, we propose background subtraction for online foreground detection called *Background Subtraction Based on Robust Subspace Tracking* (BS-RST). It includes the robust estimation of foreground detection and utilizes Robust Subspace Tracking as the backbone.

In the proposed BS-RST method, we replace the commonly used L_1 norm, which is used to quantify the foreground in most of the methods mentioned above, with a salience detection method called spectral residual (HOU; ZHANG, 2007). Spectral residual focuses on the available structures in the foreground, unlike the L_1 norm, which does not consider the structure or relationships that may exist between groups of pixels in the foreground. The BS-RST method operates online, after obtaining an initial subspace estimate (ZHAO et al., 2014). This initial subspace estimation allows for the estimation of the foreground and background, with dynamic updates over time, as previously mentioned. Thus, the BS-RST method is capable of processing a large amount of data. Consequently, we can address the limitations commonly found in other representative methods in the state-of-the-art. The BS-RST method improves on s-ReProCS to estimate the foreground and the background online in videos by: (i) calculating the initial subspace using a Bayesian-RPCA approach; and (ii) estimating the foreground via a modified s-ReProCS-like scheme, wherein the spectral residual is used to detect the salient visual features that most likely belong to the foreground and to keep track of these salient foreground features. The proposed model applied for the foreground segmentation achieved an accuracy of 68.02% in the I2R public dataset (LI et al., 2004).

2 FUNDAMENTAL CONCEPTS AND BACKGROUND

This chapter describes the definitions needed for a better understanding of this work.

2.1 Background Subtraction (BS)

A challenging task in computer vision is detecting and identifying moving objects, which is a relatively simple task for humans. This task can be tackled using three different approaches; background subtraction detects moving objects (foreground) by the difference between the current image frame and a reference image (background model); temporal difference computing the difference between two consecutive frames and Optical flow analysis all the patterns related to the movement of objects (MADDALENA; PETROSINO, 2007). Of these approaches, background subtraction is the most efficient and widely adopted approach for differentiating the background and foreground of image sequences captured by a stationary camera. So, the last two decades have seen significant publications on background subtraction, which has allowed it to be one of the most active research fields of artificial vision (BOUWMANS et al., 2019). Thus, we can see that a quick background subtraction search in IEEE Xplore shows 3022 conference publications, 569 journal publications, 14 magazines, and one book in the last twenty-three years (2000-2023).

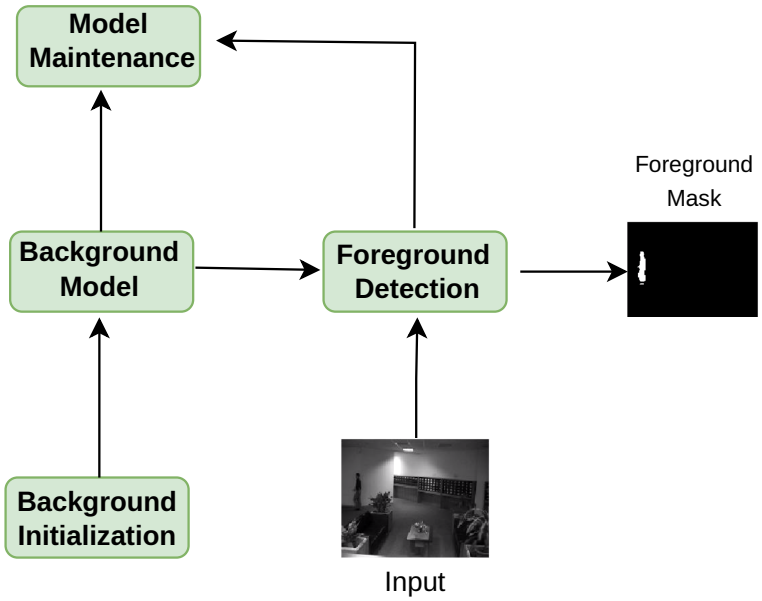
Background subtraction aims to separate foreground regions such as people, vehicles, and animals between others from the background (stationary objects such as doors, walls, rugs, or moving objects such as waves, rain. swaying three, escalators, Etc.) of a video sequence. So, this approach is considered a critical and fundamental step to simplify many computer vision and video analysis tasks, for example:

- Intelligent visual surveillance: The main focus is to detect moving objects or abandoned objects. Then, these objects are analyzed to detect possible incidents or calculate statistics on roads, airports, offices, buildings, stores, Etc.
- Intelligent visual observation of animals and insects: The main focus is to analyze the behavior (interaction within their group and their environment) of animals and insects by detecting and tracking their movements through a video-based system that would be the most appropriate.

- Optical motion capture: The objective is to detect the movements in real life and take them to the virtual world (Three dimensions). Different views from various cameras accomplish this.
- Human-machine interaction: Different applications require human-machine interaction, such as games, arts, and ludo.
- Content-based video coding: These applications are based on coding the background and foreground of static and dynamic environments separately. So, a suitable method can be applied for video coding.

Figure 2.1 shows a diagram of the background subtraction step: The first step is background initialization (also called background generation), which consists of obtaining a background model (also called background image) without moving objects through a frames sequence. Then the second step is foreground detection, which consists of labeling pixels as background or foreground pixels by comparing each incoming frame with the built background model. Thus, we can acquire a binary mask distinguishing foreground and background pixels. The last step is Model maintenance, which consists of adapting the background model to the present changes of a scene generated by different factors over time.

Figure 2.1: Background subtraction steps



2.2 Fourier Transform

For defining the Fourier transform of an integrable function $f : \mathbb{R} \to \mathbb{C}$, denoted $\mathfrak{F}\{\ f(x)\}$, is defined by the equation:

$$\mathfrak{F}\{\ f(x)\} = \int_{-\infty}^{\infty} f(x)e^{-i2\pi\mu x}dx, \forall\ \mu \in \mathbb{R} \tag{2.1}$$

Where μ indicates the frequency and we can see that $\mathfrak{F}\{\ f(x)\}$ is a function that only depends on μ then we can express the Fourier transform as $\mathfrak{F}\{\ f(x)\}$ = F(μ); That is.

$$F(\mu) = \int_{-\infty}^{\infty} f(x)e^{-i2\pi\mu x}dx, \forall\ \mu \in \mathbb{R} \tag{2.2}$$

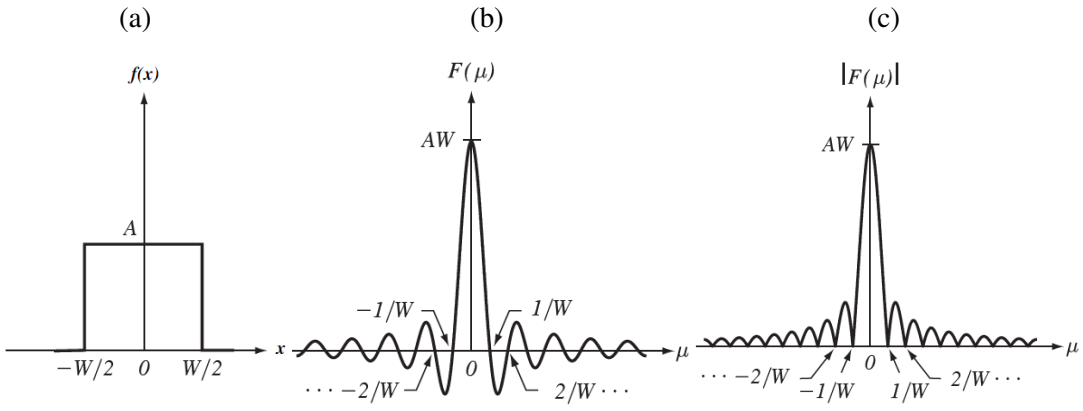
Since F(μ) is a complex number, we can calculate the Amplitude (Spectrum) and phase as follows:

$$Amplitude = |F(\mu)| \tag{2.3}$$

$$Phase = \arctan\left(\frac{\Im(F(\mu))}{\Re(F(\mu))}\right) \tag{2.4}$$

Then, we can say that the Fourier transform consists of transforming a function ***f(x)*** in the spatial domain into the spectral domain. It is used in a wide range of applications in signal processing.

Figure 2.2: (a) A simple function; b, its Fourier transform; and c, the spectrum. Figure taken from (GONZALEZ; WOODS, 2008)



2.3 Inverse Fourier Transform

Conversely, given F(μ) in spectral domain, we can obtain $f(x)$ back in spatial domain using the inverse Fourier transform, $f(x) = \mathfrak{F}^{-1}\{F(\mu)\}$, written as

$$f(x) = \int_{\infty}^{\infty} F(\mu) e^{i2\pi\mu x} d\mu, \forall\, x \in \mathbb{R} \tag{2.5}$$

2.4 Probability

To be able to define probability formally, we will give some basic concepts:

- Random experiment: Process of acquiring data related to a phenomenon that presents uncertainty in the outcome that will be obtained.
- Sample space: Set of all possible outcomes of a random experiment.
- Event: It is a subset of the sample space.

Then we can define the probability of the occurrence of an event of interest A generated by a random experiment, with a sample space Ω, as the ratio between the number of favorable cases of the occurrence of that event and the total number of cases in the sample space:

$$p(A) = \frac{A}{\Omega} \tag{2.6}$$

where A is the event that represents the number of favorable cases, and Ω represents the sample space

Mathematically, every probability of an event A must fulfill three axioms (FRONT. . . , 2016):

- **Axiom 1**: The probability of event A must be between 0 and 1.

$$0 \leq p(A) \leq 1 \tag{2.7}$$

- **Axiom 2**: The probability of occurrence of the sample space Ω is equal to one.

$$p(\Omega) = 1 \tag{2.8}$$

- **Axiom 3**: For any sequence of independent events A_i and A_j, i.e, for $A_i \cap A_j = \emptyset$

with $i \neq j$, we have:

$$p\left(\bigcup_{i=1}^{k} A_i\right) = \sum_{i=1}^{k} p(A_i) \tag{2.9}$$

2.4.1 Fundamental rules

1. **Probability of a union of two events**. Given two events, A and B, The probability of occurrence of event A or B or both is defined as:

$$p\left(A \cup B\right) = p(A) + p(B) - p(A \cap B) \tag{2.10}$$

2. **Joint probability**. Probability of simultaneous occurrence of two or more events. Considering events A and B, the joint probability of these events will be written as:

$$p(A, B) = p(A \cap B) = p(A/B)p(B) \tag{2.11}$$

   Here $p(A/B)$ is a conditional probability and is verbalized as "the probability of A given B". If events A and B are independent, we can write the joint probability as:

$$p(A, B) = p(A)p(B) \tag{2.12}$$

3. **Conditional Probability**. It is the probability that a given event will occur once a second event has occurred; $p(A|B)$ is the probability that event A will happen, given that event B has occurred.

$$p(A|B) = \frac{p(A, B)}{p(B)}, if\ p(B) > 0 \tag{2.13}$$

2.4.2 Bayes Theorem

From the conditional probability and the joint probability, we obtain Bayes Theorem:

$$p(A/B) = \frac{p(B/A)p(A)}{p(B)} \tag{2.14}$$

Where: $p(B/A)$ is called the likelihood function, $p(B)$ is called prior, and $p(A/B)$ is called Posterior. This theorem quantifies event A's occurrence based on event B's knowl-

edge.

2.5 Probability Distribution

Given a sample space Ω, a random variable X is defined as a function that takes elements in Ω and maps them to the real numbers, that is, $X : \Omega \rightarrow \Re$. The probability density function is the mapping between the real number space and the probability space $p : \Re \rightarrow [0, 1]$.

A probability distribution must satisfy the normalization conditions and not assume negative values, that is:

$$p(X_i = x_i) \geq 0 \quad i = 1, ..., N \tag{2.15}$$

For discrete random variables, it must satisfy the following:

$$\sum_{i=1}^{N} p(X_i = x_i) = 1 \tag{2.16}$$

For continuous random variables, it must satisfy the following:

$$\int_{-\infty}^{\infty} p(x)dx = 1 \tag{2.17}$$

In general, we will use $p(x_i)$ instead of $p(X_i = x_i)$ for notation simplification. The expected value or mathematical expectation $E[.]$ of a sequence of random variables x_i with $i = 1, ..., N$ and their respective probability density function $p(x_i)$ is determined for the discrete case as:

$$E[x] = \sum_{i=1}^{N} x_i p(x_i) \tag{2.18}$$

And for the continuous case, it is:

$$E[x] = \int_{-\infty}^{\infty} xp(x)dx \tag{2.19}$$

In a more general way, we can define the expected value of a sequence of random variables determined by a function $f(x_i)$ as:

$$E[f(x)] = \int_{-\infty}^{\infty} f(x)p(x)dx \tag{2.20}$$

For the discrete case, it is similar to the continuous case.

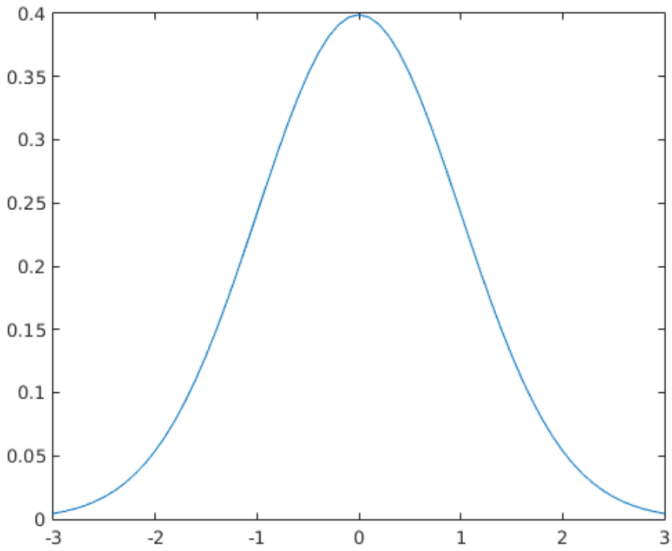
2.5.1 Gaussian Distribution

It is also known as the normal distribution and is a widely used model in statistics, machine learning, signal, and image processing. It models continuous variable distributions. In the case of a single variable x, its probability density function (pdf) is given by

$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x-\mu)} \tag{2.21}$$

Where $\mu = E[x]$ is the mean (and mode), and $\sigma^2 = var[x]$ is the variance. $\sqrt{2\pi\sigma^2}$ is the normalization constant needed to ensure the density integrates to 1. Figure 2.3 shows an example of the pdf of a Gaussian distribution with parameters $\mu = 0$ and $\sigma = 1$.

Figure 2.3: Pdf of a Gaussian distribution with $\mu = 0$ and $\sigma = 1$.



For an N-dimensional vector x, the multivariate Gauss distribution takes the form

$$\mathcal{N}(x|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{N/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2}(x-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(x-\boldsymbol{\mu})} \tag{2.22}$$

where $\boldsymbol{\mu}$ is an N-dimensional mean vector, $\boldsymbol{\Sigma}$ is a N×N covariance matrix, and $|\boldsymbol{\Sigma}|$ denotes the determinant of $\boldsymbol{\Sigma}$.

We write $X \sim \mathcal{N}(\mu,\sigma^2)$ to denote that $p(X = x) = \mathcal{N}(x|\mu,\sigma^2)$. Also, the **precision** of a Gaussian is represented by $\lambda = 1/\sigma^2$.

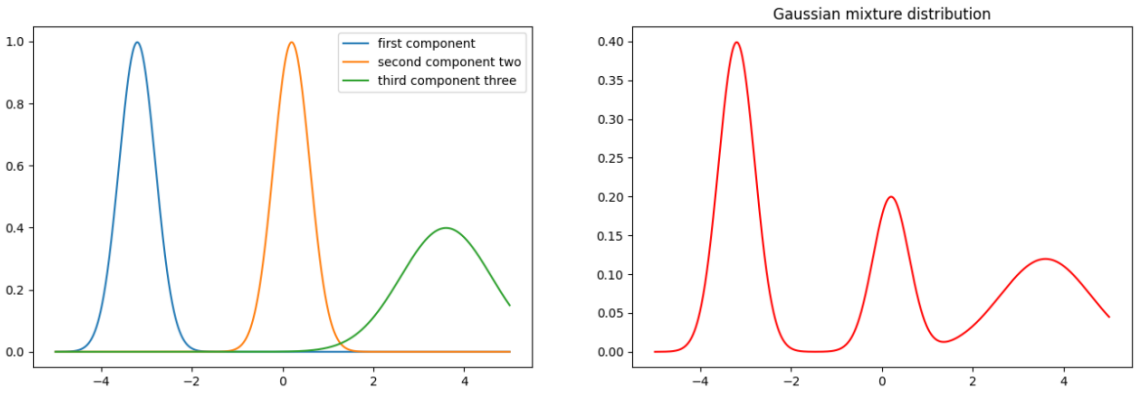
2.5.2 Gaussian mixture Distribution

The Gaussian distribution is limited when trying to model real data sets. We consider the example in Figure 2.5. This synthetic data was generated by datasets of the sklearn library that create a constant block diagonal structure array forming two groups. We can appreciate that the data set forms two dominant clumps and that a single Gaussian distribution cannot capture this structure. In contrast, a linear superposition of two Gaussian gives a better characterization of the data set. Therefore, we can define a mixture of Gaussian distribution as a superposition of K Gaussian densities, that is

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\boldsymbol{\mu_k}, \boldsymbol{\Sigma_k}) \tag{2.23}$$

where, the parameters π_k in (2.23) are called *mixing coefficients*, and each Gaussian

Figure 2.4: Example of a Gaussian mixture distribution in one dimension showing three Gaussians in blue, orange, green, and their sum in red.

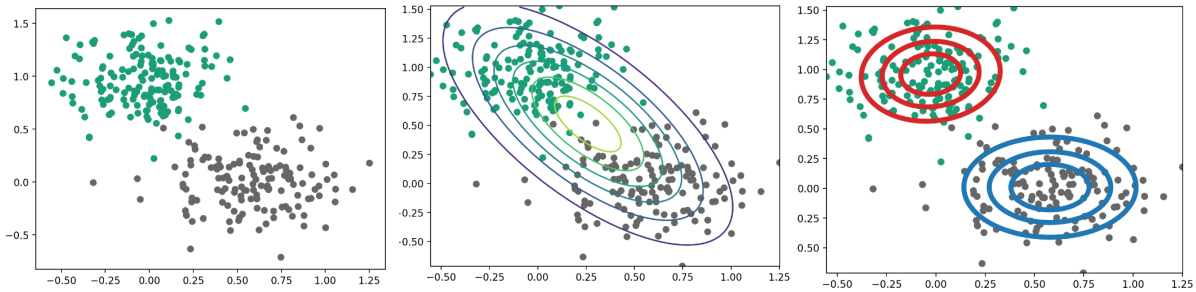


density $\mathcal{N}(x|\boldsymbol{\mu_k}, \boldsymbol{\Sigma_k})$ is called a *component of the mixture* and has its own mean $\boldsymbol{\mu_k}$ and covariance $\boldsymbol{\Sigma_k}$). Figure 2.4 shows a linear combination of Gaussian generated by three Gaussians.

These distributions are suitable in applications such as clustering, where the data is scattered but forming different groups. For example, Figure 2.5 shows how each Gaussian distribution is fitted or associated with each group, and the combination represents the mixed Gaussian model that fits the data.

Figure 2.5: Plots of the synthetic datasets generated by $sklearn.datasets.makebiclusters$. In (a) show of spread of data, in (b) show contours of constant probability density (single Gaussian distribution)

(a) (b) (c)



that fitted to the data. We can note that this distribution fails to capture the two clumps in the data and indeed places much of its probability mass in the central region between the clumps where the data are relatively sparse, and in (c) show the distribution is given by linear combination of two Gaussian which fitted better to the data.

2.6 Subspace Learning

Subspace learning models are powerful tools for analyzing high-dimensional data. Which consists of performing a dimensionality reduction of this data through a projection in a new intrinsic subspace of lower dimension that allows us to analyze, interpret and visualize data.

In the context of dimensionality reduction, Principal Component Analysis (PCA) is one of the most widely used techniques. PCA finds a small number of orthogonal basis vectors, called principal components (PCs) while retaining as much as possible of the variation present in the data. PCs are the basis of intrinsic lower dimension subspace.

Given a set of data points $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_d$ where $\boldsymbol{y}_i \in \mathbb{R}^n$, for $i = 1, 2, \ldots, d$ and let $\boldsymbol{b}$ be a transformation vector. We aim to obtain a low-dimensional compressed representation $\boldsymbol{z}_i = \boldsymbol{b}^T \boldsymbol{y}_i$ of $\boldsymbol{y}_i$by means of its variance:

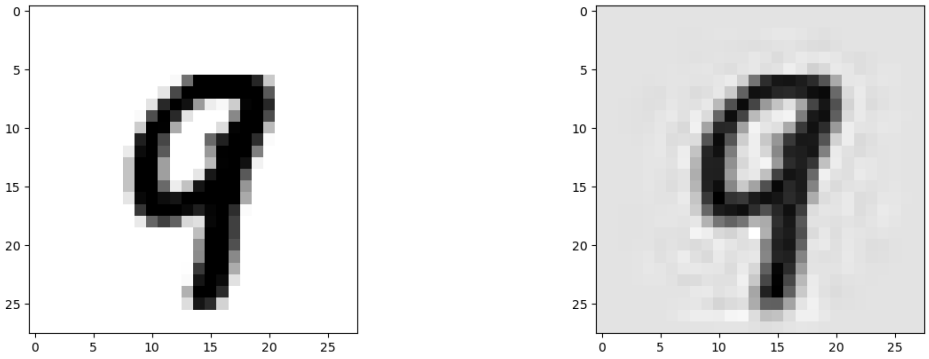
$$\begin{aligned} \mathbb{V}[\boldsymbol{z_i}] &= \frac{1}{d} \sum_{i=1}^{d} \boldsymbol{z}_i^2 \\ &= \frac{1}{d} \sum_{i=1}^{d} (\boldsymbol{b}^T \boldsymbol{y}_i)^2 = \boldsymbol{b}^T \boldsymbol{S} \boldsymbol{b} \end{aligned} \tag{2.24}$$

Then, we maximize the variance of $\boldsymbol{z}_i$ and can obtain $\boldsymbol{b}$ by means of:

$$\boldsymbol{b}_{opt} = \arg\max_{\boldsymbol{b}} \boldsymbol{b}^T \boldsymbol{S} \boldsymbol{b} \tag{2.25}$$

Figure 2.6: Image compression using the PCA method. In (a) show the digit 9 of 784 dimensions, and in (b) show the digit 9 of 184 dimensions

(a) (b)



where $\boldsymbol{S}$ is the data covariance matrix. PCs also called basis functions of PCA are the eigenvectors of the data covariance matrix associated with the largest eigenvalues. Figure 2.6 shows an example of dimensionality reduction for 784-dimensional images. We use the MNIST[1] dataset of handwritten digits to obtain the images of the digits zero to nine. Then we apply PCA on all the images and we obtain 184 principal components. Thus, we can represent any images of 784 dimensions for one of 184 dimensions.

2.6.1 Robust Principal Components Analysis

PCA tends to be sensitive to outliers, which can affect the finding of the PCs. So, Robust Principal Component (RPCA) arises to address these limitations. Robust Principal Component (RPCA) is the method that decomposes a given data matrix into the sum of a low-rank matrix that represents true data (it has no noise) and a sparse matrix that represents outliers of the data matrix. Therefore, the RPCA method was developed to estimate the PCs of a data matrix emphasizing the uncontaminated data elements, which also allows to estimate the spurious (contaminated) elements of the data matrix (WRIGHT et al., 2009).

Given a data matrix $\boldsymbol{Y}$, the RPCA method decomposes $\boldsymbol{Y}$ into a sum of two uncorrelated matrices, namely, a low-rank matrix $\boldsymbol{L}$, and a sparse matrix $\boldsymbol{X}$:

$$\boldsymbol{Y} = \boldsymbol{L} + \boldsymbol{X} \tag{2.26}$$

[1] http://yann.lecun.com/exdb/mnist/

Figure 2.7: Background modeling from Lobby video. In (a) show video frame 300, in (b) show the background, and in (c) show the foreground.

(a) (b) (c)

where the columns (i.e. the column subspace) of the low-rank matrix $\boldsymbol{L}$ are the PCs estimated putting higher emphasis on the uncontaminated data.

This decomposition of the RPCA method is obtained by solving the following constrained convex minimization problem (CANDèS et al., 2011):

$$\min_{\boldsymbol{L},\boldsymbol{X}} \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{X}\|_1 \, s.t \;\; \boldsymbol{Y} = \boldsymbol{L} + \boldsymbol{X}. \tag{2.27}$$

where $\|\boldsymbol{L}\|_*$ denotes the nuclear norm of $\boldsymbol{L}$, $\|\boldsymbol{X}\|_1$ denotes L_1-norm of $\boldsymbol{X}$, and λ is a scalar used for balancing the weights of $\|\boldsymbol{L}\|_*$ and $\|\boldsymbol{X}\|_1$. The desired solution in terms of $\boldsymbol{L}$ and $\boldsymbol{X}$ minimizes the rank of $\boldsymbol{L}$ and the norm-1 of $\boldsymbol{X}$. The minimization problem in Eq. 2.27 can be solved using the principal components pursuit (PCP) algorithm(CANDèS et al., 2011), which allows to estimate $\boldsymbol{L}$ and $\boldsymbol{X}$ with high probability. However, other algorithms also could be used to solve RPCA minimization problem in Eq. 2.27, such as singular value thresholding (SVT) (CAI; CANDES; SHEN, 2008), accelerated proximal gradient (LIN; CHEN; MA, 2010) and augmented lagrangian multiplier (LIN; CHEN; MA, 2010). Figure 2.7 shows the decomposition of a video frame by means of the RPCA method. We take 400 frames from the video Lobby of the I2R dataset (LI et al., 2004) and generate a matrix where each column represents a video frame. We decompose the given matrix Y into two matrices: L and X. Next, we select column 300 from each matrix (Y, L, and X) and create an image based on these columns.

2.6.2 Generalized Robust Principal Components Analysis

The RPCA approach assumes that $\boldsymbol{L}$ is exactly low-rank and $\boldsymbol{X}$ is sparse. Consequently, $\boldsymbol{Y} = \boldsymbol{L} + \boldsymbol{X}$, and $\boldsymbol{L}$ and $\boldsymbol{X}$ can be estimated exactly using different algorithms, as mentioned above.

However, the data matrix $\boldsymbol{Y}$ may be noisy and/or contain outliers [2]. Let us consider the general case of a noisy data matrix $\boldsymbol{Y}$, and the following representation for the data matrix decomposition problem :

$$\boldsymbol{Y} = \boldsymbol{L} + \boldsymbol{X} + \boldsymbol{V} \tag{2.28}$$

where $\boldsymbol{L}$ is always low rank, $\boldsymbol{X}$ is always sparse, and $\boldsymbol{V}$ only contains noise that is small in magnitude as compared to the $\boldsymbol{X}$ element magnitudes. Actually, the low rank component can be obtained as $\boldsymbol{L} = BB^T\boldsymbol{Y}$, where the columns of the matrix B are the eigenvectors spanning the lower dimensional space where $\boldsymbol{L}$ lies, and $\boldsymbol{Y} - \boldsymbol{L} = \boldsymbol{X} + \boldsymbol{V}$. If the low rank approximation of $\boldsymbol{Y}$ can be calculated exactly (e.g., if $\boldsymbol{Y}$ is not noisy), then its low-rank approximation is $\boldsymbol{L}$, and $\boldsymbol{X}$ is sparse, and $\boldsymbol{V} = 0$. However, if the low rank approximation of $\boldsymbol{Y}$ can not be calculated exactly (e.g., when $\boldsymbol{Y}$ is noisy), its low rank approximation still is $\boldsymbol{L}$, and $\boldsymbol{X}$ still is sparse, and $\boldsymbol{V} \neq 0$ is the residual $\boldsymbol{V} = \boldsymbol{Y} - \boldsymbol{L} - \boldsymbol{X}$.

The discussion above leads to the more general version of the RPCA, which can be expressed by Equation 2.28, where $\boldsymbol{L}$, $\boldsymbol{X}$ are defined above and $\boldsymbol{V}$ is a residual approximation error $\boldsymbol{Y} - \boldsymbol{L} - \boldsymbol{X}$ (i.e. small unstructured noise).

2.6.3 Robust Subspace Tracking

Robust Subspace Tracking (RST), also known as dynamic RPCA, is an online variant of the Generalized RPCA method. It operates online after an initial subspace estimation While RPCA methods assume a single fixed subspace for the true data, RST considers that the true data is in a low-dimensional subspace that can change every so often. Therefore, given the estimated initial subspace, it can track and updates the subspace that can change over time in the presence of sparse outliers. This model is more appropriate for long data sequences, e.g., surveillance videos or long dynamic social network connectivity data sequences.

Problem definition. The measurement vector at time t, $\boldsymbol{y}_t$, is an n-dimensional vector which can be decomposed as

$$\boldsymbol{y}_t := \boldsymbol{\ell}_t + \boldsymbol{x}_t + \boldsymbol{\nu}_t, \text{ for } t = 1, 2, \ldots, d \tag{2.29}$$

[2] Outliers are assumed to have a magnitude that is larger or equal to the smallest magnitude element of $\boldsymbol{X}$ (ZHAO et al., 2014).

Here d is the number of observed data, $\boldsymbol{x}_t \in \mathbb{R}^n$ is a sparse outlier vector, $\boldsymbol{\nu}_t \in \mathbb{R}^n$ is small unstructured noise, and $\boldsymbol{\ell}_t \in \mathbb{R}^n$ is the true data vector that lies in a changing low-dimensional subspace of $\mathbb{R}^n$ that over time t.

2.6.4 Low-Rank Matrix Factorization

Low-Rank Matrix Factorization (LRMF) is a frequently employed method for subspace learning in the context of background subtraction. Its primary objective is to extract the low-rank approximation of a data matrix by multiplying two smaller matrices, which represent the basis matrix and coefficient matrix. The general form LRMF problem can be expressed as:

$$\min_{\boldsymbol{B},\boldsymbol{A}} \left\| \boldsymbol{W} \odot (\boldsymbol{Y} - \boldsymbol{B}\boldsymbol{A}^T) \right\|_{L_p} \tag{2.30}$$

where $\boldsymbol{B} \in \mathbb{R}^{n \times r}$ and $\boldsymbol{A} \in \mathbb{R}^{d \times r}$ denote the basis and coefficient matrices, with $r \ll min(n, d)$, implying the low-rank property of $\boldsymbol{B}\boldsymbol{A}^T$. $\boldsymbol{W}$ is the indicator matrix of the same size as $\boldsymbol{Y}$, with $w_{ij} = 0$ if y_{ij} is missing and 1 otherwise. $\|.\|_{L_p}$ denotes the $p-th$ power of an L_p norm. Equation 2.30 can also be equivalently understood under the maximum likelihood estimation (MLE) framework as:

$$y_{ij} = \boldsymbol{b}_i^T \boldsymbol{a}_j + e_{ij} \tag{2.31}$$

where $\boldsymbol{b}_i$, $\boldsymbol{a}_j \in \mathbb{R}^r$ are the $i-th$ and $j-th$ row vectors of $\boldsymbol{B}$ and $\boldsymbol{A}$, respectively, and e_{ij} denotes the noise element embedded in y_{ij}.

Based on the values that p can take in the loss term utilized to measure the approximation represent in Equation 2.30. The LRMF methods can be mainly categorized into three classes: L_2-LRMF methods, L_1-LRMF methods, and probabilistic models (it encodes the noise as a parametric probabilistic model).

3 RELATED WORK

In this section, we provide a review of related works. Our analysis is primarily centered on solutions that utilize robust subspace tracking and low-rank matrix factorization for online background subtraction. We have chosen this focus due to the fact that Deep Learning-based methods often necessitate extensive data, are frequently supervised, and primarily operate in a batch processing mode. Additionally, they require significant computational resources. This restriction aligns with our proposed solution, which explores these two approaches.

3.1 Methods-Based Robust Subspace Tracking

The RST-based methods for the background subtraction task have a common formulation represented by sparse+low-rank (S+LR). This formulation means that any vector can be decomposed as the sum of a vector that lies in changing low rank subspace and a sparse vector, as represented in Equation 2.29

Below, we elucidate the methods relevant to this section. Furthermore, we provide a more detailed explanation of the s-ReProCS method, as it offers fundamental steps that are crucial to our proposed solution.

The **s-ReProCS** (VASWANI et al., 2018) method (An RST-based method) considers the simplest model on the change of subspace. In each change only one direction must change and the others remain the same, i.e., the subspace at time t spanned by $\boldsymbol{B}_t$ contains all the vectors lying in the previous subspace spanned by $\boldsymbol{B}_{t-1}$. So, the differences in terms of the data representation in the spaces spanned by $\boldsymbol{B}_{t-1}$ and $\boldsymbol{B}_t$ tend to be small. In other words, ($\left\|(\mathbf{I} - \boldsymbol{B}_{t-1}\boldsymbol{B}_{t-1}')\boldsymbol{B}_t\right\| \ll \|\boldsymbol{B}_t\| = 1$), where $\boldsymbol{B}_t \in \mathbb{R}^{n\times r}$ is a basis matrix ($\boldsymbol{B}_t'\boldsymbol{B}_t = I$), $\boldsymbol{B}_t$ is r-dimensional, $r \ll$n, and the support sets $\mathcal{T}_t$ of $\boldsymbol{x}_t$ is defined as $\{i : (\boldsymbol{x}_t)_i \neq 0\}$.

To ensure that each $\boldsymbol{\ell}_t = \boldsymbol{B}_t\boldsymbol{a}_t$ is not a sparse vector, each $\boldsymbol{x}_t$'s is not dense, and the subspace spanned by $\boldsymbol{B}_t$ be of low rank, the following conditions must be true:

(i) $\frac{|\hat{\mathcal{T}}_t|}{n}$ is upper bounded.

(ii) $\hat{\mathcal{T}}_t$ changes enough over time so that any one index is not part of the outlier support for too long.

(iii) The columns of $\boldsymbol{B}_t$ are dense (not sparse).

(iv) The subspace coefficient $\boldsymbol{a}_t$ are element-wise bounded, mutually independent., zero mean, have identical and diagonal covariance matrices, and are independent of the outlier supports $\hat{\mathcal{T}}_t$.

(v) The subspace span($\boldsymbol{B}_t$) is piecewise constant with time, i.e., $\boldsymbol{B}_t = \boldsymbol{B}_{t_j}$, for all $t \in [t_j, t_{j+1})$, $j = 1, \ldots, J$ and to lower bound $t_{j+1} - t_j$, with $t_0 = 1$ and $t_{J+1} = d$.

With i) and ii) it is ensured that the $\boldsymbol{x}_t$'s are sparse vectors, iii) and iv) guarantee that the $\boldsymbol{\ell}_t$'s are dense, and with v) it is ensured that the subspace generated by the $\boldsymbol{B}_t$ is of low rank. Based on these conditions, it is possible to recover $\boldsymbol{x}_t$, $\boldsymbol{l}_t$, update $\boldsymbol{B}_t$ with a short delay. Then, it is possible to recover $\boldsymbol{x}_t$, $\boldsymbol{l}_t$, update $\boldsymbol{B}_t$ with a short delay.

Thus, with the formulation in Equation 2.29, we can decompose t-th frame of a video as the sum of a background frame and a foreground frame as can be seen in Figure 3.1. The background frame changes slowly and the changes are usually dense. It is therefore well modeled as a dense vector that lies in a low-dimensional subspace that can change over time, albeit gradually, in the original space. The foreground frame usually consists of one or more moving objects and is correctly modeled as the sparse outlier (VASWANI; NARAYANAMURTHY, 2018). Thus, under this approach, we review the following RST-based papers: Provable Dynamic Robust PCA or Robust Subspace Tracking (s-ReProCS, (VASWANI et al., 2018)), Nearly Optimal Robust subspace tracking (NORST, (NARAYANAMURTHY; VASWANI, 2018b)), and Online Robust Principal Component Analysis with Change Point Detection (OMWEPCA-CP, (XIAO et al., 2020)). Also, We highlight that they have code available, thus allowing us to perform comparisons.

Figure 3.1: Background subtraction based on the model of equation 2.29: The t-th frame is separated by a background component and a foreground component.

s-ReProCS starts with a good estimate of the initial subspace, $\hat{\boldsymbol{B}}_0$ by means of AltProj (NETRAPALLI et al., 2014) applied to the first t_{train} data frames $\boldsymbol{Y}_{[1,t_{train}]} =$

$[y_1, y_2, ..., y_{t_{train}}]$, i.e.,

$$\begin{aligned} [\hat{\boldsymbol{L}}_0, \hat{\boldsymbol{X}}_0] &= AltProj(\boldsymbol{Y}_{[1,t_{train}]}) \\ \hat{\boldsymbol{B}}_0 &= SVD(\hat{\boldsymbol{L}}_0) \end{aligned} \tag{3.1}$$

where $\hat{\boldsymbol{L}}_0$, $\hat{\boldsymbol{X}}_0$ are low-rank matrix and sparse matrix of the first t_{train} data frames ($\boldsymbol{Y}_{train}$) and SVD represent to the Singular Value Decomposition algorithm (QIU et al., 2014). At time t, we have the observed vector $\boldsymbol{y}_t$, which can be represented by Equation 2.29, and $\hat{\boldsymbol{B}}_{t-1}$ which is a good estimate of the previous subspace, span($\boldsymbol{B}_{t-1}$), with $\hat{\boldsymbol{B}}_{t-1}$ we construct the projection function $\boldsymbol{\Psi} = \mathbf{I} - \hat{\boldsymbol{B}}_{\mathbf{t-1}}\hat{\boldsymbol{B}}'_{\mathbf{t-1}}$. So, we estimate $\boldsymbol{x}_t$ by means of:

$$\tilde{\boldsymbol{y}}_t = \boldsymbol{\Psi}\boldsymbol{x}_{\mathbf{t}} + \mathbf{b_t} \tag{3.2}$$

where $\tilde{\boldsymbol{y}}_t = \boldsymbol{\Psi}\boldsymbol{y}_{\mathbf{t}}$, $\boldsymbol{y}_t := \boldsymbol{\ell}_t + \boldsymbol{x}_t + \boldsymbol{\nu}_t$, $\boldsymbol{b}_t = \boldsymbol{\Psi}(\boldsymbol{\ell}_{\mathbf{t}} + \boldsymbol{\nu}_{\mathbf{t}})$, with $\|\boldsymbol{b}_t\|$ and $\|\boldsymbol{\nu}_t|$ are small. Under s-ReProCS conditions, it is possible to recover $\boldsymbol{x}_t$ from the Equation 3.2 by Projected comprehensive sensing (ProjCS) method (CANDèS, 2008), i.e.,

$$\hat{\boldsymbol{x}}_{t,cs} = \arg\min_{\tilde{\boldsymbol{x}}_t} \|\tilde{\boldsymbol{x}}_t\|_1 \ s.t \ \|\tilde{\boldsymbol{y}}_t - \boldsymbol{\Psi}\tilde{\boldsymbol{x}}_{\mathbf{t}}\|_1 \leq \xi \tag{3.3}$$

then by thresholding ω_{supp} we get $\hat{\mathcal{T}}_t$, since $\boldsymbol{x}_t$ is lower bounded.

$$\hat{\mathcal{T}}_t = \{i : |\hat{\boldsymbol{x}}_{t,cs}| > \omega_{supp}\} \tag{3.4}$$

Finally, to get better estimate of $\boldsymbol{x}_t$ we use Least Squares based debiasing on $\hat{\mathcal{T}}_t$:

$$\hat{\boldsymbol{x}}_t = \mathbf{I}_{\hat{\mathcal{T}}_t}(\boldsymbol{\Psi}'_{\hat{\mathcal{T}}_{\mathbf{t}}}\boldsymbol{\Psi}_{\hat{\mathcal{T}}_{\mathbf{t}}})^{-1}\boldsymbol{\Psi}'_{\hat{\mathcal{T}}_{\mathbf{t}}}\tilde{\boldsymbol{y}}_{\mathbf{t}}. \tag{3.5}$$

Then, we estimate $\boldsymbol{\ell}_t$ as:

$$\hat{\boldsymbol{\ell}}_t = \boldsymbol{y}_t - \hat{\boldsymbol{x}}_t. \tag{3.6}$$

The $\boldsymbol{\ell}_t$'s are used for: i) detecting the t_j, which represents the time when the subspace change ii) obtaining improved versions of the addresses that change in $\boldsymbol{B}_t$ by K steps of projection SVD (QIU; VASWANI; HOGBEN, 2013), each done with a new set α frames of $\hat{\boldsymbol{\ell}}_t$, and iii) Updating $\boldsymbol{B}_t$ by a simple SVD, done with another new set α frames of $\hat{\boldsymbol{\ell}}_t$.

The **NORST** method has the same procedure as s-ReProCS to obtain $\boldsymbol{x}_t$, $\boldsymbol{l}_t$, and

update $\boldsymbol{B}_t$ with a short delay, but it assumes that r directions of the subspace can change in time. Which is less restrictive than s-ReProCS (only one subspace direction can change at each change time) and in order to update the subspace instead of using the projection-SVD method, it uses a simple SVD.

s-ReProCS and NORST assume that any standard RPCA algorithms can estimate accurate initial subspace. But, they are not robust to the different scenarios present in a video (ZHAO et al., 2014). For example, dynamic background, Bootstrap, and campus sequences (LI et al., 2004). So, the approximation error increases to estimate $\boldsymbol{B}_0$. Also, the ProjCS method used to detect sparse outlier vector does not consider any structure or relationship between subsets of entries (l_1 norm treats each entry (pixel) independently). However, in foreground detection, moving parts (outliers) in a video often have the structural properties of spatial contiguity and locality (LIU et al., 2015).

The **OMWRPCA** technique is proposed for effectively tracking both slowly changing and abruptly changing subspaces. Initially, the RPCA-STOC algorithm (FENG; XU; YAN, 2013) is employed to estimate all $\hat{\boldsymbol{\ell}}_t$ and $\hat{\boldsymbol{x}}_t$ from 1 to t. Instead of updating the subspace $\boldsymbol{B}_t$ with all the estimated vectors, the update process only considers the vectors from $t - n_{win} + 1$ to t, where n_{win} represents the user-defined number of most recent samples. Subsequently, hypothesis testing is integrated to detect change points, which indicate when the subspace should change. This heuristic is based on the observation of an unusually large magnitude of the estimated $|\hat{\mathcal{T}}_t|$ derived from $\hat{\boldsymbol{x}}_t$. This phenomenon occurs due to the significant difference between the current and previous subspaces, resulting in poor modeling of $\hat{\boldsymbol{y}}_t$ by the subspace and reflected in the estimation of $\hat{\boldsymbol{x}}_t$, which exhibits a relatively higher number of non-zero elements compared to those calculated until $t - 1$. Furthermore, this algorithm dynamically and automatically estimates the dimension of each subspace, distinguishing it from previously mentioned methods.

Overall, the OMWRPCA technique demonstrates its effectiveness in tracking changing subspaces, adapting to various change patterns, and dynamically estimating subspace dimensions. The integration of the RPCA-STOC algorithm and hypothesis testing enables accurate tracking and change point detection, making it suitable for diverse applications requiring subspace analysis.

3.2 Methods-Based Low-Rank Matrix Factorization

We address the last two classes of the LRMF category because they are more robust in the presence of outliers compared to the L_2-norm. All of these algorithms are online versions of LRMF methods.

The **OBSL1** method combines the LRMF approach and the $L_{1,\epsilon}$ norm to address the underlying problem of online background subtraction. It uses the LRMF method because, unlike RPCA-based methods, it does not require any adjustment of the penalty parameters, it changes the norm L_1 to an approximate norm denoted by $L_{1,\epsilon}$-norm in the loss function, in order to deal with the complexity of finding the optimal solution and to reduce the processing time of L_1-norm. This approximate norm, being differentiable and locally convergent, greatly facilitates finding the solution and reduces processing time. The $L_{1,\epsilon}$-norm is defined as:

$$\|Y\|_{1,\epsilon}^{1} = \sum_{i=1}^{n}\sum_{j=1}^{d}((y_{i,j}^{2} + \epsilon^{2})^{1/2} - \epsilon) \tag{3.7}$$

Where $\epsilon > 0$, ensure the smoothness of the norm.

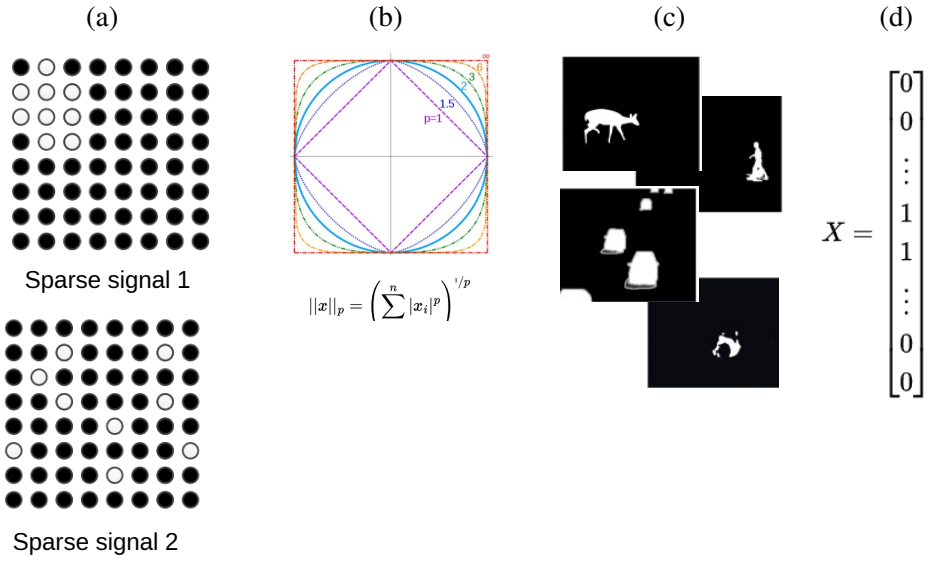
The **MoGMF** method has goals to address online background subtraction problems. Instead of using a fixed noise distribution for all frames, this method models the noise and foreground of each frame separately using a mixture of Gaussian (MoG) distribution. The MoG distribution's good approximation capability enables the method to adapt to complex video foreground variations, even when the video noises have dynamic complex structures. Additionally, an affine transformation operator is employed for each video frame. This allows the method to handle real-time background transformations caused by a moving camera, resulting in improved robustness compared to previous approaches. Furthermore, the proposed method exhibits potential for extending to other subspace alignment tasks, including image alignment and video stabilization applications.

4 PROPOSED METHOD FOR SUBSPACE TRACKING AND ONLINE BACKGROUND SUBTRACTION

In this section, we propose a new online background subtraction method called Background Subtraction Based on Robust Subspace Tracking (BS-RST) to generate a more robust estimate of the foreground in complex environmental scenarios compared to state-of-the-art methods. BS-RST relies on a robust initial subspace estimate and readjusts the foreground by filtering out spurious elements. Based on the foreground, the background is estimated, and it is used to update the subspace using the subspace tracking method, as detailed next.

A robust estimate of the initial subspace is obtained using *RPCA with complex noise* (RPCA-MoG) (ZHAO et al., 2014) (see Section 4.1). This allows for a better approximation of the background model at the initial stage than most RPCA-based methods, as shown in Figure 4.2. The *Saliency Detection* method (HOU; ZHANG, 2007) is used as a filter for the foreground (see details in Section 4.2). This method addresses the limitations of most algorithms that use the L_1-norm to quantify the foreground, as it is known that this norm does not take into account the structure or relationships that may exist between groups of pixels in the foreground. In Figure 4.1, we can see two sparse signals (zeros and ones), and if we use the L_1-norm to quantize the signal, we will obtain the same value, which makes it difficult to consider structures in the foreground. The proposed method is detailed in Section 4.3.

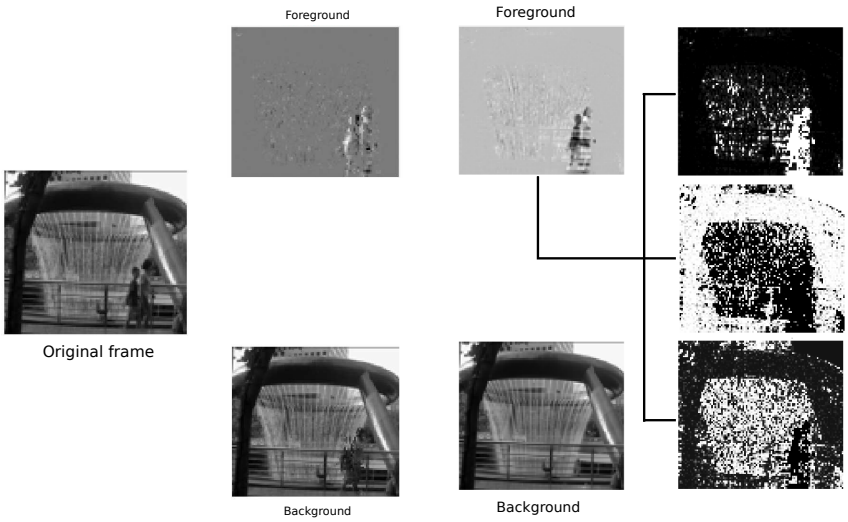
Figure 4.1: (a) Sparse distribution; (b) the L_1-norm; (c) the foreground; and (d) the foreground vector



4.1 Robust Principal Component Analysis with Complex Noise

As mentioned before, the standard RPCA tries to decompose $\boldsymbol{Y}$ into $\boldsymbol{L}$ and $\boldsymbol{X}$. But, the structure of the sparse outlier vectors $\boldsymbol{X}$ are not taken into consideration since the L_1-norm is used to characterize the error term as shown in the Equation 2.27, and it is known that the L_1-norm treats each element (pixel) independently. Thus, to overcome this limitation, the Robust Principal Component Analysis with Complex Noise (RPCA-MoG) approach is used in this work (ZHAO et al., 2014). It works in batch, and assumes that each input $x_{i,j}$ in $\boldsymbol{X}$ is generated from a mixture of Gaussians (MoG) distribution. This MoG can accommodate different types of noise, such as Laplacian, Gaussian, and Sparse noise and/or their combinations. A MoG component will group elements $x_{i,j}$, obtaining the sparse outlier vector $\boldsymbol{X}$, while the other components will group different types of noise present in the data matrix $\boldsymbol{Y}$. This scheme has a clear advantage over RPCA because of its ability to recover $\boldsymbol{L}$ and $\boldsymbol{X}$. Figure 4.2 shows a comparison of RPCA and RPCA-MoG methods in the task background subtraction.

Figure 4.2: Example illustrating the difference between RPCA (CANDèS et al., 2011) and MoG-RPCA (ZHAO et al., 2014) : (a) shown a 200^{th} frame of the fountain video; (b) shows the foreground and the background obtained by RPCA; and (c) shows the foreground and the background obtained by MoG-RPCA.



4.2 Saliency Detection

In this work, we utilize the Spectral Residual Model (HOU; ZHANG, 2007), a straightforward, rapid, and unsupervised algorithm for saliency detection (detecting moving objects). Unlike methods such as (CUI; LIU; METAXAS, 2009) or any deep learning-based approaches (ULLAH et al., 2020) (WANG et al., 2022) (SINGH et al., 2022), the Spectral Residual Model does not necessitate training data. It consists of analyzing the log-spectrum of an image to obtain the spectral components that are associated with frequently occurring features (image redundancy). Afterward, a saliency map (SM) is built in the space domain using Inverse Fourier Transform and a convolution operation with a Gaussian filter.

Given an image $I(x)$, its $SM$is obtained by Algorithm 1, where $A(f)$, $P(f)$, $L(f)$, and $R(f)$ denote the amplitude, the phase spectrum, the log spectrum, and the spectral residual of $I(x)$, respectively; g_σ is a Gaussian filter with $\sigma = 8$, and $\mathfrak{F}$ and $\mathfrak{F}^-1$ denote the Fourier Transform and Inverse Fourier Transform, respectively.

Algorithm 1: Saliency map

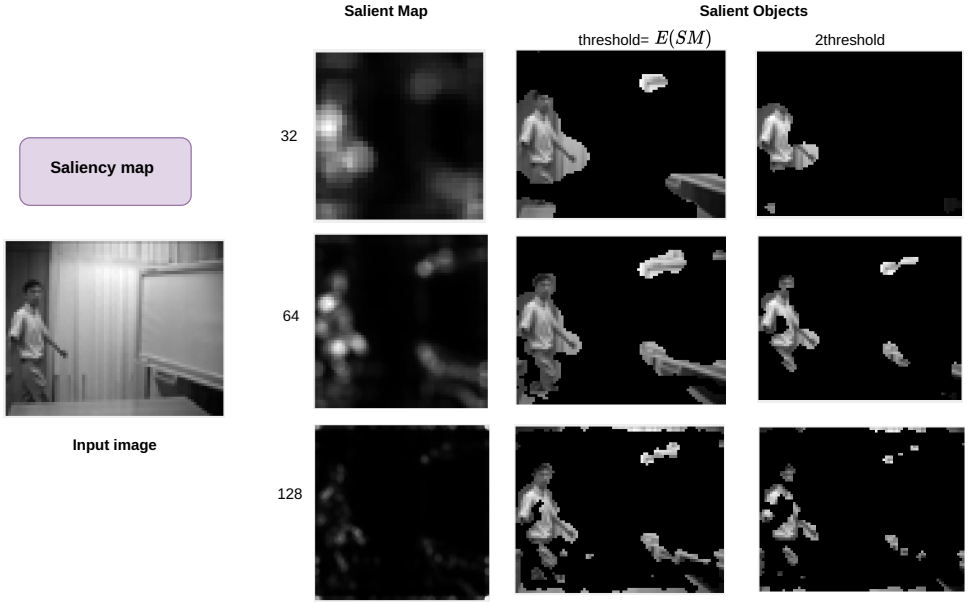
Input: Original image $I(x)$
Output: Saliency Map $S(x)$
1 **function** $SM\ (I(x))$
2 $\quad A(f) \leftarrow \Re\left(\mathfrak{F}[(I(x))]\right)$
3 $\quad P(f) \leftarrow \Im(\mathfrak{F}[I(x)])$
4 $\quad L(f) \leftarrow \log(A(f))$
5 $\quad R(f) \leftarrow L(f) - h_n(f) * L(f)$
6 $\quad S(x) \leftarrow \boldsymbol{g}_\sigma * \mathfrak{F}^{-1}[exp(R(f) + P(f))]^2$
7 $\quad$ **return** $S(x)$;
8 **end function**

Algorithm 1 takes an image as input and transforms it into the frequency domain using the Fourier transform. In this domain, we take the real part of the signal and eliminate redundant elements from the image using the spectral residual obtained from line 5 of Algorithm 1. Then, we reconstruct the signal by replacing the real part with the exponential of the spectral residual while maintaining the same phase spectrum $(P(f))$ as the original signal. Next, we bring the signal back to the spatial domain using the inverse Fourier transform, and finally, we apply a convolution operation to obtain the saliency map of the image.

In Figure 4.3, we present the input image, which is then resized into three images of sizes 32x32, 64x64, and 128x128. Subsequently, we apply Algorithm 1 to each image

and obtain their Spectral Residual Maps (SM). For each SM, we calculate a threshold by taking the average value of the SM. Using this threshold, we create a binary image (object map) where all pixels in the SM smaller than the threshold are set to zero, and all pixels greater than the threshold are set to one. Finally, we multiply the binary image with the original image and resize it back to its original size. When directly applying Algorithm 1, the results may not meet our expectations, but we can observe that we manage to identify a significant portion of the moving object of interest.

Figure 4.3: Detection of salient objects



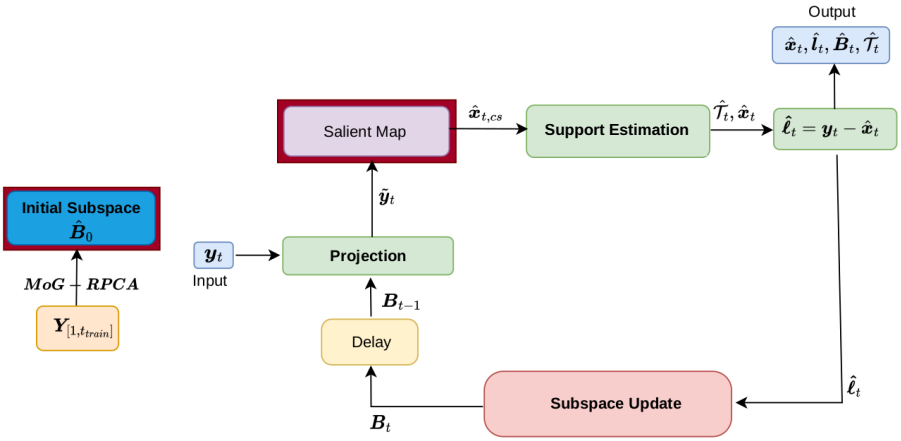
4.3 Background Subtraction Based on Robust Subspace Tracking

As mentioned before, a new method called Background Subtraction Based on Robust Subspace Tracking (BS-RST) is proposed in this work to deal with the limitations of ReProCS. The BS-RST approach combines s-ReProCS (NARAYANAMURTHY; VASWANI, 2019), RPCA-MoG (ZHAO et al., 2014), and saliency detection (HOU; ZHANG, 2007) algorithm to obtain robust estimates of the background and foreground in video sequences.

The BS-RST starts by estimating $\boldsymbol{B}_0$ using the RPCA-MoG algorithm, next at time t, $\boldsymbol{y}_t$ is projected by $\boldsymbol{\Psi}$ to get $\tilde{\boldsymbol{y}}_t = \boldsymbol{\Psi}\boldsymbol{y}_\mathbf{t}$, and saliency detection is applied to $\tilde{\boldsymbol{y}}_t$ to obtain $\hat{\boldsymbol{x}}_{t,s} = SM(\tilde{\boldsymbol{y}}_t)$ and $\hat{\mathcal{T}}_t = \{i : (\hat{\boldsymbol{x}}_{t,s})_i > \omega_s\}$, where SM denotes the saliency map of $\tilde{\boldsymbol{y}}_t$, and the threshold $\omega_s = E(\hat{\boldsymbol{x}}_{t,s}) * 3$ denotes the average intensity of the saliency map. Next, the sparse outlier vector ($\hat{\boldsymbol{x}}_t$) is estimated using Least Squares (LS) Based

debiasing on $\hat{\mathcal{T}}_t$: $\hat{\boldsymbol{x}}_t = \mathbf{I}_{\hat{\mathcal{T}}_t}(\boldsymbol{\Psi}'_{\hat{\mathcal{T}}_t}\boldsymbol{\Psi}_{\hat{\mathcal{T}}_t})^{-1}\boldsymbol{\Psi}'_{\hat{\mathcal{T}}_t}\tilde{\boldsymbol{y}}_t$. Figure 4.5 shows the procedure to obtain the foreground of a video frame. Finally, $\boldsymbol{\ell}_t$ is estimated as $\hat{\boldsymbol{\ell}}_t = \boldsymbol{y}_t - \hat{\boldsymbol{x}}_t$. Then, each α frames of $\hat{\boldsymbol{\ell}}_t$ will be used to update the subspace $\hat{\boldsymbol{B}}_t$ using the simple SVD method. In Figure 4.4, we present the proposed solution to address the problem of background subtraction.

Figure 4.4: Solution Architecture



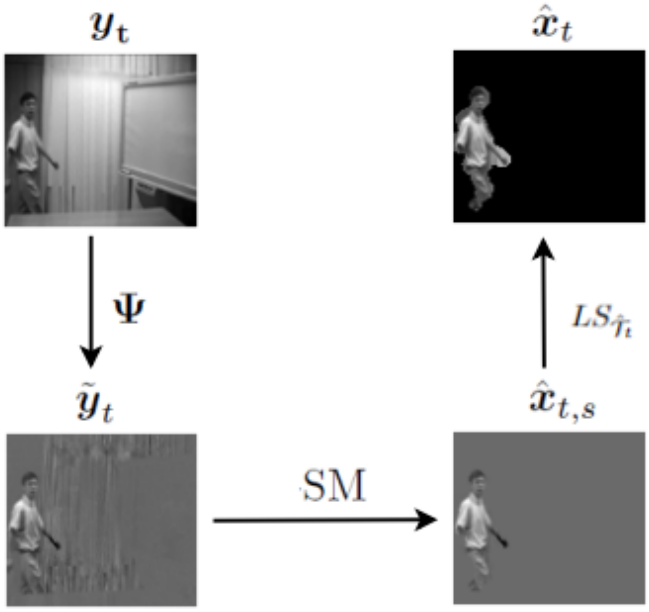
The pseudo-code of BS-RST is displayed in Algorithm 2, and its main steps are :

1. We start with the estimation of $\hat{\boldsymbol{L}}_0$ by means of MoG-RPCA on $\boldsymbol{Y}_{[1,t_{train}]}$
2. The parameters K, α, ξ, r and ω_{evals} are set as in s-ReProCS.
3. s-ReproCS states that projecting $\boldsymbol{y}_t$ onto the orthogonal complement of $\boldsymbol{B}_{t-1}$ will nullify most of ℓ_t under the assumption of fixed or slowly changing subspace. This is expressed in lines 8 and 9 of the Algorithm 2, where $\boldsymbol{\Psi}$ is projection operator and $\tilde{\boldsymbol{y}}_t$ is the projection of $\boldsymbol{y}_t$ onto $\boldsymbol{\Psi}$.
4. Since $\tilde{\boldsymbol{x}}_t$ is contained in $\tilde{\boldsymbol{y}}_t$, first, we remove most elements in $\tilde{\boldsymbol{y}}_t$ that are not part of $\tilde{\boldsymbol{x}}_t$ using the Algorithm 1. We then calculate $\hat{\mathcal{T}}_t$ based on the threshold ω_s as shown in lines 11 and 12 of the Algorithm 2. Finally, we perform an LS thresholded by $\hat{\mathcal{T}}_t$ to obtain $\tilde{\boldsymbol{x}}_t$ using the lines 13 and 14 of the Algorirthm 2.

4.4 Model Evaluation

We adopt traditional evaluation metrics to assess our model's performance for online foreground detection in videos. Thus, performance assessment was based on sensitivity (Sens, also called recall), precision (Prec, *i.e.,* positive predictive value), F1-score (*i.e.,* the harmonic mean between recall and precision).

Figure 4.5: Stages to obtain the foreground of the t-th frame $\boldsymbol{y}_t$ by means of $\boldsymbol{\Psi}$, Salient map (SM), and LS based debiasing on $\hat{\mathcal{T}}_t$.



Algorithm 2: Robust Subspace Tracking for BS-RST

Input: $\hat{\boldsymbol{L}}_0$, $\boldsymbol{y}_t$
Output: $\hat{\boldsymbol{x}}_t$, $\hat{\boldsymbol{l}}_t$ $\hat{\boldsymbol{B}}_t$, $\hat{\mathcal{T}}_t$
Parameters: $\omega_s, K, \alpha, \xi, r, \omega_{evals}$

$\hat{\boldsymbol{B}}_0$ is equal to SVD_r (top r singular vectors) of $\hat{\boldsymbol{L}}_\mathbf{0}$
$\hat{\boldsymbol{L}}_{t;\alpha} := [\hat{l}_{t-\alpha+1}, \hat{l}_{t-\alpha+2}, ..., \hat{l}_t]$
$\hat{\boldsymbol{B}}_{t_{train}} \leftarrow \hat{\boldsymbol{B}}_0$
$j \leftarrow 1$
$k \leftarrow 1$
$t \leftarrow t_{train} + 1$
Assign phase $\leftarrow$ update
for $t > t_{train}$ **do**
  $\boldsymbol{\Psi} \leftarrow \mathbf{I} - \hat{\boldsymbol{B}}_{\mathbf{t-1}}\hat{\boldsymbol{B}}'_{\mathbf{t-1}}$
  $\tilde{\boldsymbol{y}}_t \leftarrow \boldsymbol{\Psi}\boldsymbol{y}_\mathbf{t}$
  $\hat{\boldsymbol{x}}_{t,s} \leftarrow$SM$(\tilde{\boldsymbol{y}}_t)$; `// Algorithm 1`
  $\omega_s \leftarrow E(\hat{\boldsymbol{x}}_{t,s}) * 3$
  $\hat{\mathcal{T}}_t \leftarrow \{i : \hat{\boldsymbol{x}}_{t,s} > \omega_s\}$
  $\hat{\boldsymbol{x}}_t \leftarrow \mathbf{I}_{\hat{\mathcal{T}}_t}(\boldsymbol{\Psi}'_{\hat{\mathcal{T}}_\mathbf{t}}\boldsymbol{\Psi}_{\hat{\mathcal{T}}_\mathbf{t}})^{-\mathbf{1}}\boldsymbol{\Psi}'_{\hat{\mathcal{T}}_\mathbf{t}}\tilde{\boldsymbol{y}}_\mathbf{t}$
  $\hat{\boldsymbol{\ell}}_t \leftarrow \boldsymbol{y}_t - \hat{\boldsymbol{x}}_t$
  `/* Automatic Subspace Update method (NARAYANAMURTHY; VASWANI, 2019) */`
  ($\hat{\boldsymbol{B}}_t$, $\hat{\boldsymbol{B}}_j$, $\hat{t}_j$, k, j, phase) $\leftarrow$ AutoSubUp($\hat{\boldsymbol{L}}_{t;\alpha}$, $\hat{\boldsymbol{B}}_{j-1}$, t, $\hat{t}_{j-1}$, j, k, phase, $\hat{\boldsymbol{B}}_{t-1}$) ; `// Algorithm 3`
  $t \leftarrow t + 1$

Algorithm 3: Automatic Subspace Update

function `AutoSubUp`($\hat{\boldsymbol{L}}_{t;\alpha}$, $\hat{\boldsymbol{B}}_{j-1}$, t, $\hat{t}_{j-1}$, j, k, *phase*, $\hat{\boldsymbol{B}}_{t-1}$)
  if *phase* = *update* **then**
    if $t = \hat{t}_j + \alpha u$ *for* $u = 1, 2, \ldots, K+1$ **then**
      $\boldsymbol{B} \leftarrow (I - \hat{\boldsymbol{B}}_{j-1}\hat{\boldsymbol{B}}'_{j-1})\hat{\boldsymbol{L}}_{t;\alpha}$
      $\boldsymbol{B}_{j,chd,k} \leftarrow SVD_1[\boldsymbol{B}]$
      $\hat{\boldsymbol{B}}_t \leftarrow [\hat{\boldsymbol{B}}_{j-1}\boldsymbol{B}_{j,chd,k}]$
      $k \leftarrow k+1$
      if $k = K+1$ **then**
        $\hat{\boldsymbol{B}}_j \leftarrow SVD_r[\hat{\boldsymbol{L}}_{t;\alpha}]$
        $\hat{\boldsymbol{B}}_t \leftarrow \hat{\boldsymbol{B}}_j$
        $j \leftarrow j+1$
        $k \leftarrow 1$, phase $\leftarrow$ detect
    else
      $\hat{\boldsymbol{B}}_t \leftarrow \hat{\boldsymbol{B}}_{t-1}$
  $\hat{t}_{j-1,fin} \leftarrow \hat{t}_{j-1} + k\alpha + \alpha - 1$
  if *phase* = *detect and* $t = \hat{t}_{j-1,fin} + \alpha u$ **then**
    $\boldsymbol{B} \leftarrow (I - \hat{\boldsymbol{B}}_{j-1}\hat{\boldsymbol{B}}'_{j-1})\hat{\boldsymbol{L}}_{t;\alpha}$
    if $\sigma_{max}(\boldsymbol{B}) \geq \sqrt{\alpha\omega_{evals}}$ **then**
      phase $\leftarrow$ update
      $\hat{t}_j \leftarrow t$
    $\hat{\boldsymbol{B}}_t \leftarrow \hat{\boldsymbol{B}}_{t-1}$
  return $\hat{\boldsymbol{B}}_t$, $\hat{\boldsymbol{B}}_j$, $\hat{t}_j$, j, k, phase
end function

5 DESIGN CHOICES

This chapter aims to describe some of the choices that we made during the development of this study and presenting the respective justifications.

5.1 Environment for experiment

For the development of this work, we have used the following experimentation environment detailed below.

We used a computer with Intel(R) Core(TM) i5-7300HQ as CPU and GeForce GTX 1050 graphics card with 4GB of RAM; additional information is reported in Table 5.1. All experiments were implemented in the Matlab software.

Table 5.1: The experimental environment is presented in this work.

	Local enviroment
CPU	Intel(R) Core(TM) i5-7300HQ
N° cores	4
RAM	8 GB
Operating System	Ubuntu 22.04.03
Graphic card	GeForce GTX Titan
GPU RAM	4 GB
Cuda version	11.0

5.2 Dataset

The I2R dataset (LI et al., 2004), recorded by Lin and Huang, was created to address the difficulties in background modeling for detecting moving objects in complex environments. In 2004, they assessed their statistical-based background modeling algorithm and Bayesian framework for foreground detection using the I2R dataset. The dataset comprises 9 video sequences captured in both indoor and outdoor settings, encompassing various challenging factors such as bootstrapping problems, shadows, camouflage, illumination changes (both sudden and gradual), video noise, challenging weather conditions, and dynamic backgrounds. Ground truth is provided in the form of manually segmented foreground masks for 20 video frames of each sequence. Some published works, iden-

tified as (EBADI; IZQUIERDO, 2018), (HU et al., 2017), provide application examples of this dataset. The complete details of each video sequence are presented in Table 5.2. Additionally, the dataset is also referred to as the STAR dataset in certain publications (BERJóN et al., 2018), (XIANG, 2013).

Table 5.2: Characteristics of the video sequences in the I2R Dataset.

Video	Scenes	Frame number	Challenges
Airport	Indoor	3584	Shadows, Bootstrapping
Restaurant	Indoor	3055	Shadows, Bootstrapping
Shopping Mall	Indoor	1286	Shadows, Bootstrapping
Lobby	Indoor	2545	Shadows, Sudden
Subway Station	Outdoor	2634	Dynamic Background, Sudden Illumination
Curtains	Indoor	23893	Dynamic Background, Camouflage
Campus	Outdoor	1439	Dynamic Background, Shadows, Gradual Illumination Changes
Water Surfaces	Outdoor	633	Dynamic Background
Fountain	Outdoor	1523	Dynamic Background

We use the I2R Dataset because all of the compared methods use most or all of the videos from this dataset to quantify the performance of their algorithms.

5.3 Experimental Results

Our experiments were performed on the I2R dataset (LI et al., 2004). In these experiments, the first three/four letters of each sequence name are taken as an abbreviation. We select 400 frames for each video, except for the Water Surfaces video where we choose 200 frames. Then, we train and obtain the initial subspace $\boldsymbol{B}_0$.

A number of typical online state-of-the-art background subtraction methods were used as comparative methods in these experiments, such as: OMoGMF[1] (YONG et al., 2018), NORST[2] (NARAYANAMURTHY; VASWANI, 2018b), s-ReProCS[3] (NARAYANAMURTHY; VASWANI, 2019), OMWRPCA[4] (XIAO et al., 2020), and OBSL1[5] (LIU; LI, 2022). As we can see in Table 5, the proposed metshod BS-RST is the only one that does not have a restriction for modeling the foreground. It utilizes an unsupervised algorithm called Saliency detection to highlight objects in an image, which is incorporated into our proposed pipeline.

[1] http://gr.xjtu.edu.cn/web/dymeng/7
[2] https://github.com/praneethmurthy/NORST
[3] https://github.com/praneethmurthy/ReProCS
[4] https://github.com/wxiao0421/onlineRPCA
[5] https://sites.google.com/site/qiliucityu/discussion

Table 5.3: Comparing BS-RST with the state-of-the-art methods.

Method	Approach	Decompostion	Drawback	Constraint
OMoGMF	LRMF	$\boldsymbol{y}_t = \boldsymbol{B}_t\boldsymbol{a}_t + \boldsymbol{x}_t$	Prior knowledge about MoG distribution, complex solution	$x_{t,i} \sim \sum_{k=1}^{K} \pi_k \mathcal{N}(x_{t,i}\|\boldsymbol{b_i}^T\boldsymbol{a}, {\sigma_k}^2)$
s-ReproCS	RST	$\boldsymbol{y}_t := \boldsymbol{\ell}_t + \boldsymbol{x}_t + \boldsymbol{\nu}_t$	Fixed subspace, single component change	$\boldsymbol{x}_t \sim$ Laplace(1,0)
NORST	RST	$\boldsymbol{y}_t := \boldsymbol{\ell}_t + \boldsymbol{x}_t + \boldsymbol{\nu}_t$	Fixed subspace, r component changes	$\boldsymbol{x}_t \sim$ Laplace(1,0)
OBSL1	LRMF	$\boldsymbol{y}_t = \boldsymbol{B}_t\boldsymbol{a}_t + \boldsymbol{x}_t$	Fixed subspace r component changes	$\\|\boldsymbol{x}_t\\|_{1,\epsilon}^1 = \sum_{i=1}^{n}((x_{t,i}^2 + \epsilon^2)^{1/2} - \epsilon)$
OMWRPCA	RST	$\boldsymbol{y}_t := \boldsymbol{B}_t\boldsymbol{a}_t + \boldsymbol{x}_t$	Weak performance in foreground estimation	$\boldsymbol{x}_t \sim$ Laplace(1,0)
BS-RST	RST	$\boldsymbol{y}_t := \boldsymbol{\ell}_t + \boldsymbol{x}_t + \boldsymbol{\nu}_t$	Fixed subspace, single component change	No

5.4 Performance Evaluation

The F-measure was used as the quantitative metric for performance evaluation, which is calculated as follows:

$$\text{F-measure} = 2 \times \frac{precision \cdot recall}{precision + recall}$$

where $precision = \frac{|\mathbf{S}_f \bigcap \mathbf{S}_{gt}|}{|\mathbf{S}_f|}$ and $recall = \frac{|\mathbf{S}_f \bigcap \mathbf{S}_{gt}|}{|\mathbf{S}_{gt}|}$, $\mathbf{S}_f$ and $\mathbf{S}_{gt}$ denote the support sets of the foreground estimated and the ground truth, respectively.

6 RESULTS

This chapter provides a comprehensive summary of the results obtained from the conducted experiments. It is worth noting that throughout these experiments, we employed the exact settings established by the respective authors of each comparison method. By adhering to their prescribed parameters, we aimed to ensure a fair and objective evaluation of the performance of each method.

Table 6.1 shows the average F-measure values that were obtained for all foreground frames computed by the various comparative methods. These F-measure values serve as a quantitative measure of the accuracy and precision of each method in delineating the foreground objects within the video sequences.

Table 6.1: F-measure (%) results for all comparative methods for the videos of the I2R dataset. Each value is averaged over all foreground-annotated frames in the corresponding video. The most right column shows the average performance of each competing method over all video sequences. The best result is highlighted in bold, and our results are underlined.

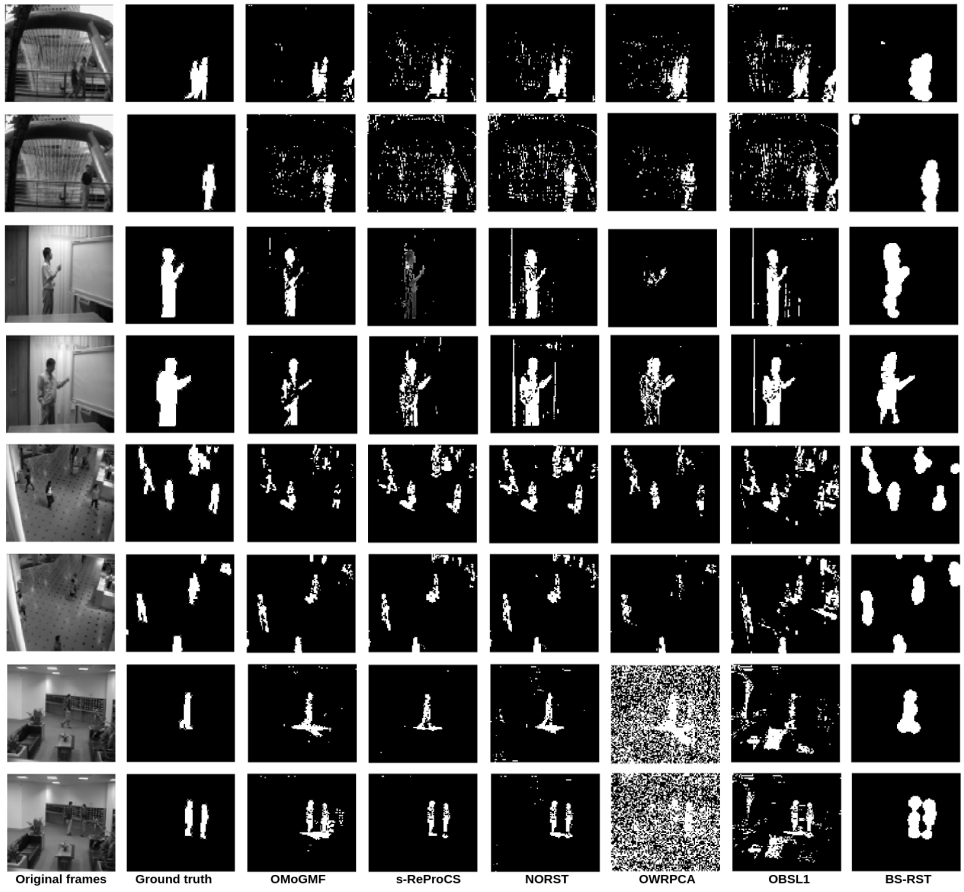
Methods	Data									
	airp.	boat.	shop.	lobb.	esca.	curt.	camp.	wate.	foun.	Average
OMoGMF	51.79	55.22	45.81	48.29	41.27	70.77	19.27	47.71	17.33	44.16
s-ReproCS	54.11	52.54	56.29	55.12	56.43	61.88	31.24	**74.01**	46.85	54.23
NORST	56.86	55.14	56.28	54.49	51.14	70.66	30.12	73.76	46.69	55.02
OMWRPCA	38.33	45.84	39.80	51.68	56.49	42.38	39.85	30.07	55.53	44.33
OBSL1	49.83	44.24	46.12	31.82	43.09	71.55	26.78	71.67	41.86	47.44
BS-RST	**63.46**	**65.16**	**71.73**	**69.26**	**67.40**	**75.68**	**61.52**	58.77	**79.21**	**68.02**

6.1 Discussion

The obtained results suggest that the proposed BS-RST method, potencially can obtain better detection of foreground in each frames of videos in comparison with methods that represent the state-of-the-art, such as OMoGMF, NORST, s-ReProCS, OMWRPCA and OBSL1.

The BS-RST method tends to provide a better performance as compared to the comparative methods, in average. Also, no image pre-processing techniques were used for these videos. The BS-RST results are shown in Fig. 6.1, which illustrates the foregrounds detected by each comparative method in typical frames of tested videos. Also, it is possible to conclude that the obtained foreground detection results obtained by the proposed BS-RST method tend to be similar to the groundtruths, which is validated by

Figure 6.1: From left to right: Typical frames from fountain, curtain, bootstrap, and lobby sequences, groundtruth foreground objects, foregrounds detected by all competing methods.

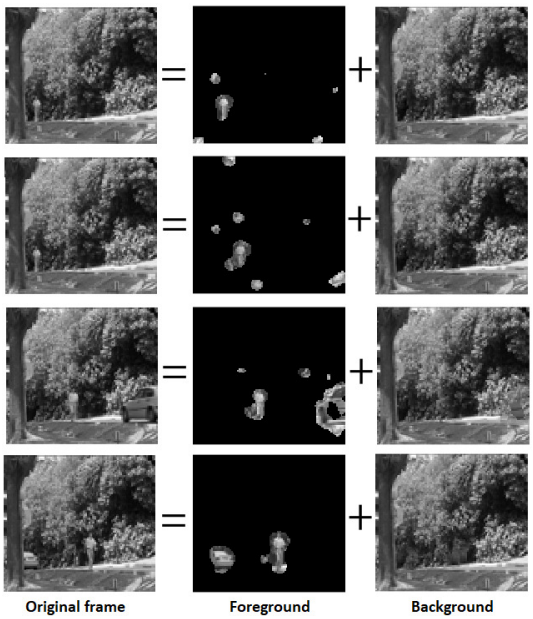


the obtained F-measure in the experiments.

The improved performance of the proposed BS-RST method may be explained by two main reasons. First, BS-RST better approximates the initial subspace by being less sensitive to outliers, while other approaches use PCA, RPCA, or SVD, which are sensitive to noise and outliers. Second, the saliency detection step provides a robust estimate of the foreground support regions, and considers the relationships between subsets of pixels that are part of the foreground, while removing unlikely foreground elements, such as noise and/or background elements. This refinement step can be seen as an additional filter after projecting the frame into the orthogonal subspace to get the foreground at each time t. It shall be mentioned that the proposed BS-RST tends to faster in foreground estimation than s-ReproCS, since s-ReProCS relies on compressive sensing, which is computationally more expensive.

It should be clarified that the proposed BS-RST scheme has limitations, as illustrated in Fig. 6.2. The original frame is decomposed into two the foreground and the background components. As can be seen in Fig. 6.2, over-segmentation occurred and

Figure 6.2: From left to right: Typical frames from campus sequences, foreground and background detected by BS-RST method.



small groups of pixels that are not part of foreground were incorrectly detected. These artifacts were generated mainly by the saliency detection algorithm, since it failed to detected coherently groups of foreground pixels. In this process, the saliency detection algorithm considered all these groups of pixels as outstanding image parts, and grouped them as parts of the foreground.

7 CONCLUSION AND FUTURE WORKS

This work proposed a novel method, namely BS-RST, for online background subtraction of video sequences. The proposed method tends to make robust estimates of the background, which helps obtaining improved foreground estimates, as compared to other methods that are representative of the state of the art.

A series of experiments were designed to evaluate the performances of the proposed method and the comparative methods. Based on these experiments, it is possible to conclude that BS-RST tends to be less sensitive to outliers, makes robust estimates of the foreground support regions, and tends to faster in foreground estimation than most of the comparative methods.

Among the possibilities for future work, we would like to explore current methods in Saliency detection. As discussed in this paper, our method utilizes a fast and simple approach for foreground detection that does not discriminate similarity, size, and distance between objects. In addition, we want to use the temporal information of the previous frames and be able to filter the elements that were considered as foreground in the estimation stage of the current foreground. Another point to investigate is the update stage of the subspace, as well as the potential reduction of certain restrictions imposed by the current model. This would lead to improved performance in foreground detection, considering that in real-world scenarios, satisfying the constraints of the current model becomes challenging.

REFERENCES

BABACAN, S. D. et al. Sparse Bayesian Methods for Low-Rank Matrix Estimation. **IEEE Transactions on Signal Processing**, v. 60, n. 8, p. 3964–3977, aug. 2012. ISSN 1941-0476.

BERJóN, D. et al. Real-time nonparametric background subtraction with tracking-based foreground update. **Pattern Recognition**, v. 74, p. 156–170, 2018. ISSN 0031-3203. Available from Internet: <https://www.sciencedirect.com/science/article/pii/S0031320317303503>.

BOUWMANS, T. Traditional and recent approaches in background modeling for foreground detection: An overview. **Computer Science Review**, v. 11-12, p. 31–66, 2014. ISSN 1574-0137. Available from Internet: <https://www.sciencedirect.com/science/article/pii/S1574013714000033>.

BOUWMANS, T. et al. Deep neural network concepts for background subtraction:a systematic review and comparative evaluation. **Neural Networks**, v. 117, p. 8–66, 2019. ISSN 0893-6080. Available from Internet: <https://www.sciencedirect.com/science/article/pii/S0893608019301303>.

CAI, J.-F.; CANDES, E. J.; SHEN, Z. **A Singular Value Thresholding Algorithm for Matrix Completion**. 2008.

CANDèS, E. J. The restricted isometry property and its implications for compressed sensing. **Comptes Rendus Mathematique**, v. 346, n. 9, p. 589–592, 2008. ISSN 1631-073X. Available from Internet: <https://www.sciencedirect.com/science/article/pii/S1631073X08000964>.

CANDèS, E. J. et al. Robust principal component analysis? **Journal of the ACM**, v. 58, n. 3, p. 1–37, may 2011. ISSN 0004-5411, 1557-735X. Available from Internet: <https://dl.acm.org/doi/10.1145/1970392.1970395>.

CAO, X. et al. Low-rank matrix factorization under general mixture noise distributions. In: **Proceedings of the IEEE international conference on computer vision**. [S.l.: s.n.], 2015. p. 1493–1501.

CUI, X.; LIU, Q.; METAXAS, D. Temporal spectral residual: Fast motion saliency detection. In: **Proceedings of the 17th ACM International Conference on Multimedia**. New York, NY, USA: Association for Computing Machinery, 2009. (MM '09), p. 617–620. ISBN 9781605586083. Available from Internet: <https://doi.org/10.1145/1631272.1631370>.

DIANA, F.; BOUWMANS, T. Background modeling via a supervised subspace learning. In: INTERNATIONAL SOCIETY FOR RESEACH IN SCIENCE AND TECHNOLOGY. **International Conference on Image, Video Processing and Computer Vision**. [S.l.], 2010. p. 1–7.

DING, X.; HE, L.; CARIN, L. Bayesian robust principal component analysis. **IEEE Transactions on Image Processing**, v. 20, p. 3419–3430, 2011.

EBADI, S. E.; IZQUIERDO, E. Foreground segmentation with tree-structured sparse rpca. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 40, n. 9, p. 2273–2280, 2018.

ELGUEBALY, T.; BOUGUILA, N. Finite asymmetric generalized gaussian mixture models learning for infrared object detection. **Computer Vision and Image Understanding**, Elsevier, v. 117, n. 12, p. 1659–1671, 2013.

FARCAS, D.; MARGHES, C.; BOUWMANS, T. Background subtraction via incremental maximum margin criterion: a discriminative subspace approach. **Machine Vision and Applications**, Springer, v. 23, n. 6, p. 1083–1101, 2012.

FENG, J.; XU, H.; YAN, S. Online robust pca via stochastic optimization. In: BURGES, C. J. C. et al. (Ed.). **Advances in Neural Information Processing Systems**. Curran Associates, Inc., 2013. v. 26. Available from Internet: <https://proceedings.neurips.cc/paper/2013/file/8f121ce07d74717e0b1f21d122e04521-Paper.pdf>.

FRONT Matter. In: PROBABILITY and Statistics with Reliability, Queuing and Computer Science Applications. John Wiley & Sons, Ltd, 2016. p. i–xix. ISBN 9781119285441. Available from Internet: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119285441.fmatter>.

GABRIEL, K. R.; ZAMIR, S. Lower rank approximation of matrices by least squares with any choice of weights. **Technometrics**, Taylor & Francis, v. 21, n. 4, p. 489–498, 1979.

GAO, F.; LI, Y.; LU, S. Extracting moving objects more accurately: A cda contour optimizer. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 31, n. 12, p. 4840–4849, 2021.

GARCIA-GARCIA, B.; BOUWMANS, T.; Rosales Silva, A. J. Background subtraction in real applications: Challenges, current models and future directions. **Computer Science Review**, v. 35, p. 100204, 2020. ISSN 1574-0137. Available from Internet: <https://www.sciencedirect.com/science/article/pii/S1574013718303101>.

GONZALEZ, R. C.; WOODS, R. E. **Digital image processing**. Upper Saddle River, N.J.: Prentice Hall, 2008. ISBN 9780131687288 013168728X 9780135052679 013505267X. Available from Internet: <http://www.amazon.com/Digital-Image-Processing-3rd-Edition/dp/013168728X>.

GRASZKA, P. Median mixture model for background–foreground segmentation in video sequences. Václav Skala-UNION Agency, 2014.

GUO, H.; QIU, C.; VASWANI, N. An online algorithm for separating sparse and low-dimensional signal sequences from their sum. **IEEE Transactions on Signal Processing**, v. 62, n. 16, p. 4284–4297, 2014.

GUO, L.; DU, M.-h. Student's t-distribution mixture background model for efficient object detection. In: IEEE. **2012 IEEE International Conference on Signal Processing, Communication and Computing (ICSPCC 2012)**. [S.l.], 2012. p. 410–414.

HAINES, T. S.; XIANG, T. Background subtraction with dirichlet processes. In: SPRINGER. **European Conference on Computer Vision**. [S.l.], 2012. p. 99–113.

He, J.; Balzano, L.; Lui, J. C. S. Online Robust Subspace Tracking from Partial Information. **arXiv e-prints**, p. arXiv:1109.3827, sep. 2011.

HOU, X.; ZHANG, L. Saliency detection: A spectral residual approach. In: **2007 IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2007. p. 1–8.

HU, W. et al. Moving object detection using tensor-based low-rank and saliently fused-sparse decomposition. **IEEE Transactions on Image Processing**, v. 26, n. 2, p. 724–737, 2017.

KALSOTRA, R.; ARORA, S. A comprehensive survey of video datasets for background subtraction. **IEEE Access**, v. 7, p. 59143–59171, 2019.

LEE, B.; HEDLEY, M. Background estimation for video surveillance. 2002.

LI, L. et al. Statistical modeling of complex backgrounds for foreground object detection. **IEEE Transactions on Image Processing**, v. 13, n. 11, p. 1459–1472, 2004.

LIANG, D. et al. Crossnet: Cross-scene background subtraction network via 3d optical flow. **IEEE Transactions on Multimedia**, p. 1–14, 2023.

LIM, L. A.; KELES, H. Foreground segmentation using a triplet convolutional neural network for multiscale feature encoding. **Pattern Recognition Letters**, v. 112, 01 2018.

LIM, L. A.; KELES, H. Y. Learning multi-scale features for foreground segmentation. **Pattern Analysis and Applications**, v. 23, p. 1369 – 1380, 2018. Available from Internet: <https://api.semanticscholar.org/CorpusID:51926299>.

LIM, L. A.; Yalim Keles, H. Foreground segmentation using convolutional neural networks for multiscale feature encoding. **Pattern Recognition Letters**, v. 112, p. 256–262, 2018. ISSN 0167-8655. Available from Internet: <https://www.sciencedirect.com/science/article/pii/S0167865518303702>.

LIN, Z.; CHEN, M.; MA, Y. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. **arXiv preprint arXiv:1009.5055**, 2010.

LIU, Q.; LI, X. Efficient low-rank matrix factorization based on l1-norm for online background subtraction. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 32, n. 7, p. 4900–4904, 2022.

LIU, X. et al. Background subtraction based on low-rank and structured sparse decomposition. **IEEE Transactions on Image Processing**, v. 24, n. 8, p. 2502–2514, 2015.

MADDALENA, L.; PETROSINO, A. Moving object detection for real-time applications. In: **14th International Conference on Image Analysis and Processing (ICIAP 2007)**. [S.l.: s.n.], 2007. p. 542–547.

MARGHES, C.; BOUWMANS, T.; VASIU, R. Background modeling and foreground detection via a reconstructive and discriminative subspace learning approach. In: **International Conference on Image Processing, Computer Vision, and Pattern Recognition, IPCV**. [S.l.: s.n.], 2012. v. 2012.

MCCOY, M. B.; TROPP, J. A. Two proposals for robust pca using semidefinite programming. **Electronic Journal of Statistics**, v. 5, p. 1123–1160, 2011.

MUKHERJEE, D.; JONATHANWU, Q. Real-timevideosegmentation using student'stmixture model. **Procedia Computer Science**, Elsevier, v. 10, p. 153–160, 2012.

NARAYANAMURTHY, P.; VASWANI, N. A fast and memory-efficient algorithm for robust pca (merop). In: **2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. [S.l.: s.n.], 2018. p. 4684–4688.

NARAYANAMURTHY, P.; VASWANI, N. Nearly optimal robust subspace tracking: A unified approach. In: **2018 IEEE Data Science Workshop (DSW)**. [S.l.: s.n.], 2018. p. 81–85.

NARAYANAMURTHY, P.; VASWANI, N. Provable dynamic robust pca or robust subspace tracking. **IEEE Transactions on Information Theory**, v. 65, n. 3, p. 1547–1577, 2019.

NETRAPALLI, P. et al. **Non-convex Robust PCA**. 2014.

QIU, C.; VASWANI, N.; HOGBEN, L. Recursive robust pca or recursive sparse recovery in large but structured noise. In: **2013 IEEE International Conference on Acoustics, Speech and Signal Processing**. [S.l.: s.n.], 2013. p. 5954–5958.

QIU, C. et al. Recursive Robust PCA or Recursive Sparse Recovery in Large but Structured Noise. **arXiv:1211.3754 [cs, math]**, aug. 2014. ArXiv: 1211.3754. Available from Internet: <http://arxiv.org/abs/1211.3754>.

RAHMON, G. et al. Motion u-net: Multi-cue encoder-decoder network for motion segmentation. In: **2020 25th International Conference on Pattern Recognition (ICPR)**. [S.l.: s.n.], 2021. p. 8125–8132.

ROY, S. M.; GHOSH, A. Real-time adaptive histogram min-max bucket (hmmb) model for background subtraction. **IEEE Transactions on Circuits and Systems for Video Technology**, IEEE, v. 28, n. 7, p. 1513–1525, 2017.

SINGH, V. K. et al. Video-based salient object detection: A survey. In: **2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)**. [S.l.: s.n.], 2022. p. 617–622.

SREBRO, N.; JAAKKOLA, T. Weighted low-rank approximations. In: **Proceedings of the 20th international conference on machine learning (ICML-03)**. [S.l.: s.n.], 2003. p. 720–727.

SULTANA, M. et al. Unsupervised deep context prediction for background estimation and foreground segmentation. **Mach. Vision Appl.**, Springer-Verlag, Berlin, Heidelberg, v. 30, n. 3, p. 375–395, apr 2019. ISSN 0932-8092. Available from Internet: <https://doi.org/10.1007/s00138-018-0993-0>.

TORRE, F. D. L.; BLACK, M. J. A framework for robust subspace learning. **Int. J. Comput. Vision**, Kluwer Academic Publishers, USA, v. 54, n. 1–3, p. 117–142, aug 2003. ISSN 0920-5691. Available from Internet: <https://doi.org/10.1023/A:1023709501986>.

ULLAH, I. et al. A brief survey of visual saliency detection. **Multimedia Tools Appl.**, Kluwer Academic Publishers, USA, v. 79, n. 45–46, p. 34605–34645, dec 2020. ISSN 1380-7501. Available from Internet: <https://doi.org/10.1007/s11042-020-08849-y>.

VASWANI, N. et al. Robust subspace learning: Robust pca, robust subspace tracking, and robust subspace recovery. **IEEE Signal Processing Magazine**, v. 35, n. 4, p. 32–55, 2018.

VASWANI, N.; NARAYANAMURTHY, P. Static and Dynamic Robust PCA and Matrix Completion: A Review. **Proceedings of the IEEE**, v. 106, n. 8, p. 1359–1379, aug. 2018. ISSN 1558-2256. Conference Name: Proceedings of the IEEE.

WANG, N. et al. A probabilistic approach to robust matrix factorization. In: SPRINGER. **European Conference on Computer Vision**. [S.l.], 2012. p. 126–139.

WANG, W. et al. Salient object detection in the deep learning era: An in-depth survey. **IEEE Transactions on Pattern Analysis & Machine Intelligence**, IEEE Computer Society, Los Alamitos, CA, USA, v. 44, n. 06, p. 3239–3259, jun 2022. ISSN 1939-3539.

WREN, C. R. et al. Pfinder: Real-time tracking of the human body. **IEEE Transactions on pattern analysis and machine intelligence**, IEEE, v. 19, n. 7, p. 780–785, 1997.

WRIGHT, J. et al. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In: BENGIO, Y. et al. (Ed.). **Advances in Neural Information Processing Systems**. Curran Associates, Inc., 2009. v. 22. Available from Internet: <https://proceedings.neurips.cc/paper/2009/file/c45147dee729311ef5b5c3003946c48f-Paper.pdf>.

XIANG, T. Background subtraction with dirichlet process mixture models. In: . [S.l.: s.n.], 2013.

XIAO, W. et al. Online Robust Principal Component Analysis With Change Point Detection. **IEEE Transactions on Multimedia**, v. 22, n. 1, p. 59–68, jan. 2020. ISSN 1941-0077. Conference Name: IEEE Transactions on Multimedia.

YONG, H. et al. Robust online matrix factorization for dynamic background subtraction. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 40, n. 7, p. 1726–1740, 2018.

ZHANG, T.; LERMAN, G. A novel m-estimator for robust pca. **The Journal of Machine Learning Research**, JMLR. org, v. 15, n. 1, p. 749–808, 2014.

ZHAO, Q. et al. Robust principal component analysis with complex noise. In: XING, E. P.; JEBARA, T. (Ed.). **Proceedings of the 31st International Conference on Machine Learning**. Bejing, China: PMLR, 2014. (Proceedings of Machine Learning Research, 2), p. 55–63. Available from Internet: <https://proceedings.mlr.press/v32/zhao14.html>.

ZHENG, Y. et al. Practical low-rank matrix approximation under robust l 1-norm. In: IEEE. **2012 IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.], 2012. p. 1410–1417.

ZIN, T. T. et al. A new background subtraction method using bivariate poisson process. In: IEEE. **2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing**. [S.l.], 2014. p. 419–422.

APPENDIX A — RESUMO EXPANDIDO EM PORTUGUÊS

Título da Dissertação de Mestrado: Subtração de Fundo baseado em Rastreamento Robusto de Subespaço.

Resumo expandido: O progresso sucessivo da capacidade dos computadores tornou possível processar, analisar e interpretar imagens e vídeos, impulsionando o desenvolvimento contínuo de campos de pesquisa em visão computacional, reconhecimento de padrões e processamento de imagens. Isso beneficiou várias áreas de aplicação, como sensoriamento remoto, diagnóstico médico, interação humano-computador, compressão de vídeo, vigilância inteligente por vídeo, monitoramento de tráfego, análise visual de animais e insetos, e captura de movimento óptico. A precisão é essencial em muitas dessas aplicações, o que levou a um grande interesse na comunidade científica e na indústria (BOUWMANS, 2014), (GARCIA-GARCIA; BOUWMANS; Rosales Silva, 2020).

Em muitas aplicações baseadas em vídeo, um passo fundamental e crítico é a detecção do primeiro plano (objetos em movimento) em uma cena, conhecida como subtração de fundo. Isso é vital para tarefas de análise de vídeo, como vigilância e detecção de anomalias.

A subtração de fundo é um método amplamente utilizado para esse fim, buscando um equilíbrio entre tempo de computação e qualidade de detecção. No entanto, condições ideais, como câmera estática, iluminação constante e fundo estático, são raras em aplicações reais, apresentando diversos desafios relacionados a fatores no fundo, primeiro plano e câmera. Portanto, novos métodos de subtração de fundo adaptativos e robustos se tornaram necessários, especialmente com o aumento da geração de grandes volumes de dados por dispositivos como computadores, câmeras digitais e celulares.

Diferentes abordagens para o método de subtração de fundo foram desenvolvidas para lidar com os desafios apresentados em vídeos, incluindo abordagens baseadas em fundamentais (LEE; HEDLEY, 2002) (GRASZKA, 2014) (ROY; GHOSH, 2017), informações estatísticas (WREN et al., 1997) (ELGUEBALY; BOUGUILA, 2013) (MUKHERJEE; JONATHANWU, 2012) (MUKHERJEE; JONATHANWU, 2012) (HAINES; XIANG, 2012), aprendizado de subespaço (VASWANI et al., 2018) (DIANA; BOUWMANS, 2010) (CANDèS et al., 2011) (MCCOY; TROPP, 2011) (GABRIEL; ZAMIR, 1979) (SREBRO; JAAKKOLA, 2003) , e aprendizado profundo (SULTANA et al., 2019) (LIM; Yalim Keles, 2018) (LIM; KELES, 2018b) (LIM; KELES, 2018a) (GAO; LI; LU, 2021).

Neste trabalho apresentamos o BS-RST, um método de subtração de fundo para detecção do primeiro plano online baseado no Rastreamento de Subespaço Robusto (Uma abordagem de aprendizado de subespaço). Comparamos nosso método com outros trabalhos, OMoGMF (YONG et al., 2018), NORST (NARAYANAMURTHY; VASWANI, 2018b), s-ReProCS (NARAYANAMURTHY; VASWANI, 2019), OMWRPCA (XIAO et al., 2020), e OBSL1 (LIU; LI, 2022). Publicamos nossos resultados avaliando a métrica F-measure em um conjunto de dados público para comparações justas com as outras abordagens.

BS-RST possui duas etapas principais, a primeira parte focado na construção do subespaço inicial e a segunda parte obtenção do primeiro plano.

1. **SUBESPAÇO INICIAL**. Para esta etapa utilizamos o método RPCA-MoG (ZHAO et al., 2014) para estimar de maniera mais robusta o subespeço inicial, atravez de um conjunto de frames $\boldsymbol{Y}_{[1,t_{train}]}$ de tamanho t_{trian} para calcular os subespaço inicial.
2. **ESTIMAÇÃO DO PRIMEIRO PLANO.** Ele consiste de duas etapas.
   - **Refinamento do primeiro plano**. Dado o subespaço inicial, começamos a estimação do t-esimo primeiro plano por meio do subespaço $t-1$ e o refinamento por meio do método SM (HOU; ZHANG, 2007), assim conseguimos remover elementos que foram erroneamente considerados como primeiro plano logrando obter uma estimação mais proxima do primeiro plano.
   - **Atualização do subespaço**. Depois de refinar o primeiro plano, estimamos o fundo correspondente a esse frame pela diferença entre o frame atual e o primeiro plano estimado, logo a atualizaçãmos o subespaço atravez do metodo (NARAYANAMURTHY; VASWANI, 2019). Na Figura 4.4, podemos visualizar todos os passos mencionados para estimar o subespaço inicial, refinar o primeiro plano, estimar o fundo e finalmente atualizar o subespaço no tempo.

No decorrer do desenvolvimento desta dissertação utilizamos o banco de dados chamado I2R dataset (LI et al., 2004). O qual foi criado para abordar as dificuldades na modelagem de fundo para detecção de objetos em movimento em ambientes complexos. O conjunto de dados compreende 9 sequências de vídeo capturadas em ambientes internos e externos, abrangendo vários fatores desafiadores, como problemas de inicialização, sombras, camuflagem, mudanças de iluminação (súbitas e graduais), ruído de vídeo, condições climáticas desafiadoras e fundos dinâmicos. O ground truth é fornecida na forma de máscaras de primeiro plano segmentadas manualmente para 20 quadros de

vídeo de cada sequência.

Além dos experimentos citados, realizamos uma discução sobre as advantagem e desvantagem de nosso metodo ao obter o primeiro plano de um video frame de maneira online. Assim, em nossos experimentos obtivimos um 68,02% de acurácia, destacando assim que nosso metodo proposto é menos sensivel aos outliers.

Por fim, apresentamos visualizações dos resultados obtivos na etapa de experimentos na tarefa de segmentação de objetos.