Evaluation of Robust Covariance Estimation for Object Detection

Andres Felipe Tamayo-Arango
aftamayoa@eafit.edu.co

David Plazas Escudero
dplazas@eafit.edu.co

Juan Pablo Vidal-Correa
jpvidalc@eafit.edu.co

Juan Sebastián Cárdenas-Rodríguez
jscardenar@eafit.edu.co

Mathematical Engineering
Universidad EAFIT

April 7, 2021

Abstract

This work presents an initial approach to the evaluation of robust covariance estimation for object detection (localization) using the "region covariance" technique from the literature. The covariance estimation is performed using the Comedian, Kendall, Spearman and Ledoit and Wolf robust approaches for covariance, and the procedure wasalso compared using two different matrix norms for estimating dissimilarity. The performance was measured quantitatively using linear regression and Pareto boundaries, yielding the Ledoit and Wolf estimation with best overall performance in object detection in normal and noisy images.

Keywords: Region covariance, robust estimation, object detection, Pareto boundaries, image features.

The code implemented for this paper can be found in this link.

1 Introduction

The object detection and classification in digital images is yet a difficult challenge. This topic is a branch of a what is known as "computer vision" [1]. This interdisciplinary field addresses the challenge of how computers can understand and gain insight from digital images and videos, similar to how human visual system works [2, 11, 1, 26]. Computer vision is widely used nowadays, in application fields such as medical imaging, automotive safety, surveillance, bio-metrics, face detection, autonomous navigation, industrial inspection and beyond [11, 26].

More specifically, object detection refers to "anything from identifying a location to identifying and registering components of a particular object class at various levels of detail" [1]. Object detection is a necessary requirement for the further "recognition" of the object or for more general tasks, such as "tracking" in dynamic scenarios (see [16] for an excellent review in tracking methodologies). Some general object detection methodologies can be found, for example, [1, 26] and the references therein; for more recent developments involving deep learning, refer to [12, 36] and the references within; finally, for an statistical approach, check [17].

This paper aims to evaluate the effectiveness of robust covariance estimations for object detection using the well-known “region covariance” scheme proposed in [29]. This methodology has been widely used by different authors in different research fields [33], such as object classification/recognition [10, 24, 27], human detection [19, 28, 35, 9], face recognition [20, 8], image smoothing [13], and object tracking [23, 32, 34, 31].

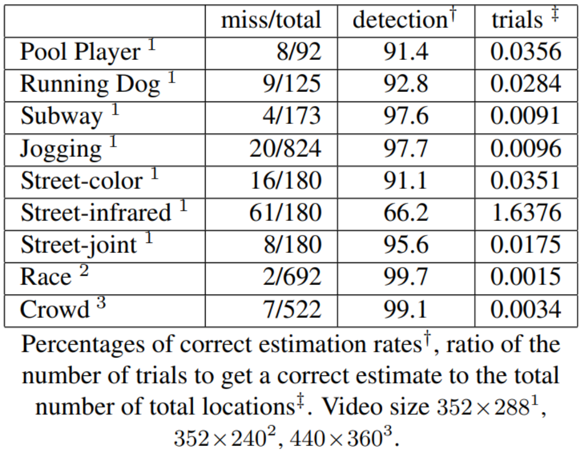
This work is organized as follows: Section 2 presents a brief, but rather specific, description of three main applications of the Region Covariance approach for object detection, which serve as main justification to the purpose of this work. On other hand, Section 3 presents the general scheme for the CRISP-DM methodology for projects in analytics and describes how it is applied to the present research. Furthermore, Section 4 presents the generalities on the Region Covariance method for object localization. Additionally, Section 5 presents the obtained results for different covariance matrix estimation and different evaluation objects. The concluding remarks and future work are discussed in Section 6.

2 Justification

This section briefly describes some of the most remarkable applications of the general methodology used in this work, as main justification for this exploratory work.

One of the most important applications of the methodology proposed in [29] is in Object Tracking, which is a natural consequence of the original localization scheme. The work in [23] and [33] show outstanding results for object tracking, using real-time model update. The first, uses a model update method based on Lie algebra and Lie groups to estimate the intrinsic mean of the covariance matrices through frames. The symmetric positive definite matrices $\mathrm{Sym}^+(n,\mathbb{R})$ have a Lie group structure and, hence, this approach. The table presented in Fig. 1 shows the obtained results by the authors in different scenarios, where almost all presented good detection of the object.

Figure 1: Results for experiments (taken from [23]).



	miss/total	detection†	trials ‡
Pool Player ^{1}	8/92	91.4	0.0356
Running Dog ^{1}	9/125	92.8	0.0284
Subway ^{1}	4/173	97.6	0.0091
Jogging ^{1}	20/824	97.7	0.0096
Street-color ^{1}	16/180	91.1	0.0351
Street-infrared ^{1}	61/180	66.2	1.6376
Street-joint ^{1}	8/180	95.6	0.0175
Race ^{2}	2/692	99.7	0.0015
Crowd ^{3}	7/522	99.1	0.0034

Percentages of correct estimation rates†, ratio of the number of trials to get a correct estimate to the total number of total locations‡. Video size 352×288^1, 352×240^2, 440×360^3.

The second, uses a model update approach divided in two stages: a probabilistic Bayesian inference for covariance tracking – via Maximum A Posteriori (MAP) estimation –, and an Incremental Covariance Tensor Learning (ICTL). The former propagates the sample distribution over time and the latter learns a low-dimensional covariance model online, as time progresses. The general scheme of this approach is presented in Fig. 2. The authors claim to have achieved real-time performance using state-of-the-art data structures for images.

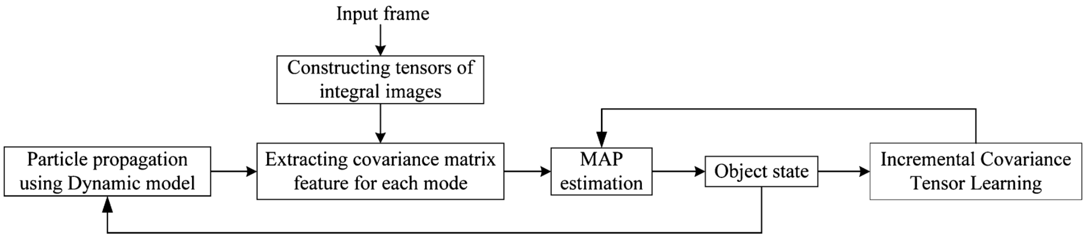
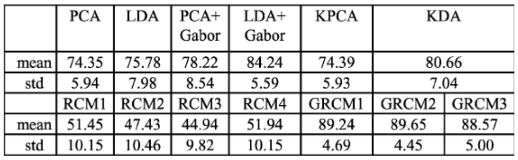

Input frame
Constructing tensors of integral images
Particle propagation using Dynamic model
Extracting covariance matrix feature for each mode
MAP estimation
Object state
Incremental Covariance Tensor Learning

Figure 2: General scheme (taken from [33]).

The work presented in [20] shows a newly developed feature descriptor framework, deeply related with Region Covariance Matrices (RCMs) to perform face recognition. The use of RCMs provides a natural method of fusing multiple features in addition to the fact that RCM has a low dimension, and it is independent from scale or illumination. The table presented in Fig. 3 shows the obtained results by the authors comparing the accuracy of face recognition between RCM-based methods and classical methods such as principal component analysis (PCA), linear discriminant analysis (LDA), kernel PCA (KPCAP) and kernel LDA (KLDA). Note that, although the normal RCM performs poorly, the Gabor-based RCM outperforms the standard methods, giving better accuracy.



	PCA	LDA	PCA+ Gabor	LDA+ Gabor	KPCA	KDA	
mean	74.35	75.78	78.22	84.24	74.39	80.66	
std	5.94	7.98	8.54	5.59	5.93	7.04	
	RCM1	RCM2	RCM3	RCM4	GRCM1	GRCM2	GRCM3
mean	51.45	47.43	44.94	51.94	89.24	89.65	88.57
std	10.15	10.46	9.82	10.15	4.69	4.45	5.00

Figure 3: Results of accuracy (taken from [20])

Finally, the work developed in [22] shows a remarkable application for detection of license plates using the covariance as region descriptor and then it is flatten into an input vector to a multi-layer perceptron (faster than calculating the dissimilarities between the matrices). The authors claim that this approach is robust against noises, illumination distortions and rotations. In Fig. 4, an example of covariance matrix estimation from a 7-feature vector per pixel. Furthermore, Fig. 5 shows the estimated ROC curve for 4 different experiments, using a different method and different feature vectors.

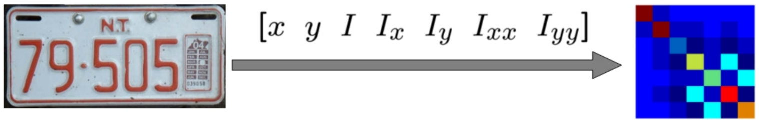

N.T.
79·505
$[x \;\; y \;\; I \;\; I_x \;\; I_y \;\; I_{xx} \;\; I_{yy}]$

Figure 4: Example of covariance matrix from features (taken from [22]).

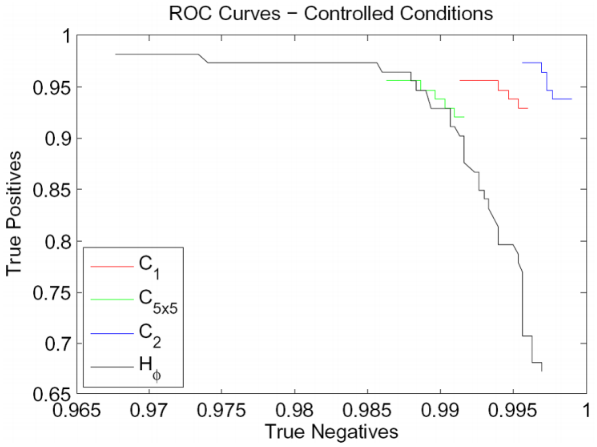
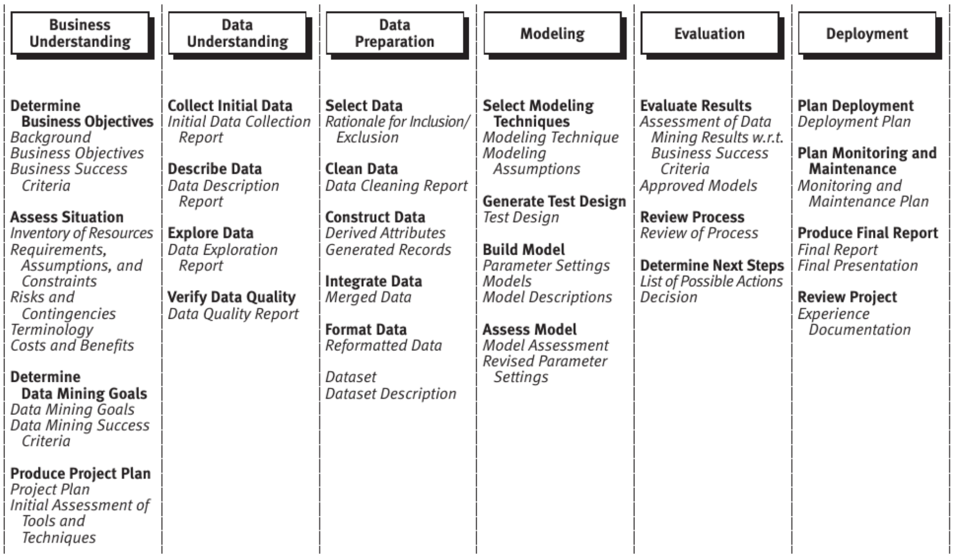


Figure 5: ROC curve for different feature vectors (taken from [22]).

3 CRISP-DM Methodology

The CRISP-DM (Cross Industry Standard Process for Data Mining) methodology splits the data-mining endeavor into six phases: business understanding, data understanding, data preparation, modeling, evaluation and deployment [3].



Business Understanding

Determine Business Objectives
Background
Business Objectives
Business Success Criteria

Assess Situation
Inventory of Resources
Requirements, Assumptions, and Constraints
Risks and Contingencies
Terminology
Costs and Benefits

Determine Data Mining Goals
Data Mining Goals
Data Mining Success Criteria

Produce Project Plan
Project Plan
Initial Assessment of Tools and Techniques

Data Understanding

Collect Initial Data
Initial Data Collection Report

Describe Data
Data Description Report

Explore Data
Data Exploration Report

Verify Data Quality
Data Quality Report

Data Preparation

Select Data
Rationale for Inclusion/ Exclusion

Clean Data
Data Cleaning Report

Construct Data
Derived Attributes
Generated Records

Integrate Data
Merged Data

Format Data
Reformatted Data

Dataset
Dataset Description

Modeling

Select Modeling Techniques
Modeling Technique
Modeling Assumptions

Generate Test Design
Test Design

Build Model
Parameter Settings
Models
Model Descriptions

Assess Model
Model Assessment
Revised Parameter Settings

Evaluation

Evaluate Results
Assessment of Data Mining Results w.r.t. Business Success Criteria
Approved Models

Review Process
Review of Process

Determine Next Steps
List of Possible Actions
Decision

Deployment

Plan Deployment
Deployment Plan

Plan Monitoring and Maintenance
Monitoring and Maintenance Plan

Produce Final Report
Final Report
Final Presentation

Review Project
Experience Documentation

Figure 6: CRISP-DM Methodology (taken from [3]).

Business Understanding: *The first objective of the data analyst is to thoroughly understand, from a business perspective, what the customer really wants to accomplish. Often the customer has many competing objectives and constraints that must be properly balanced. The analyst's goal is to uncover important factors, at the beginning, that can influence the outcome of the project.*

For the phase of Business Understanding see sections 1 and 2.

Data Understanding: *Acquire the data (or access to the data) listed in the project resources. This initial collection includes data loading, if necessary for data understanding.*

The acquire data for the project were personal photos for the following reasons. The input object was required to have different scale, orientation, location, and lighting characteristics in order to evaluate performance with the different metrics and covariance estimates. On the other hand, it is necessary to be able to manually locate the real object to assess the correct location of the object (supervise focus).

Data Preparation: *These are the dataset(s) produced by the data preparation phase, which will be used for modeling or the major analysis work of the project.*

For current project the data preparation phase was developed as follows. In the first step we take six photos of an object, one of this is considered as the input of the program and the other five were considered as test cases. This last photo was cropped in order to become into a single image. On the other hand, all images were scaled so that their major axis had 70 pixels. Finally, real objects were extracted from the five test images for further evaluation of the program.

Modeling: *As the first step in modeling, select the actual modeling technique that is to be used. Although you may have already selected a tool during the Business Understanding phase, this task refers to the specific modeling technique, e.g., decision-tree building with 5.0, or neural network generation with back propagation. If multiple techniques are applied, perform this task separately for each technique.*

For modeling phase, see section 4.

Evaluation: *This step assesses the degree to which the model meets the business objectives and seeks to determine if there is some business reason why this model is deficient. Another option is to test the model(s) on test applications in the real application, if time and budget constraints permit*

For Evaluation phase, see section 5

Deployment: *This task takes the evaluation results and determines a strategy for deployment. If a general procedure has been identified to create the relevant model(s), this procedure is documented here for later deployment.*

Deployment phase can be found in this link.

4 Methodology

4.1 Covariance as a Region Descriptor

Descriptors are set of numbers produced to describe a given shape in a quantifiable measure [30]. The shape may not be entirely reconstructed from the descriptors, but these measures for different shapes should be different enough that the shapes can be discriminated [4]. A region descriptor describes the object within based on the pixel distribution in this 2-D array.

The covariance as a region descriptor, proposed in [29], will now be presented. For an image of width W and height H, let $\mathcal{W} = \{1, \ldots, W\}$ and $\mathcal{H} = \{1, \ldots, H\}$. The image is then mapped into a feature space for each pixel

$$F : \mathcal{W} \times \mathcal{H} \rightarrow \mathbb{R}^d$$

yielding a tensor $\mathcal{A} \subset \mathbb{R}^{W \times H \times d}$, as depicted in Fig. 7.

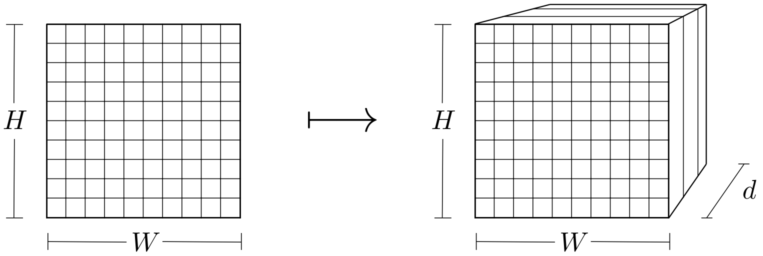


Figure 7: Feature extraction.

Let W' and H' the width and height of region $R \subset \mathcal{A}$. The covariance matrix of R is estimated by flattening the region into a $(W' \cdot H') \times d$ data matrix, yielding $\mathbf{C}_R \in \mathbb{R}^{d \times d}$. This process is depicted in Fig. 8.

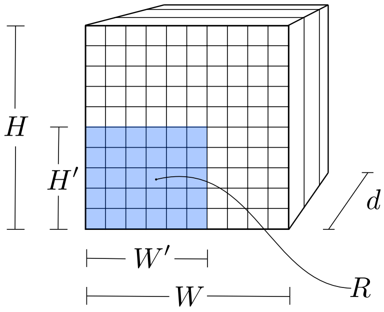


Figure 8: Region tensor.

There are multiple advantages for using the covariance as a region descriptor. Usually, a single covariance matrix extracted from a region is enough to match the region in different views, since the covariance of a distribution is enough to discriminate it from other distributions. Moreover, the covariance matrix allows to observe features that might be correlated and filter out noise corrupting samples during the computation.

Additionally, the covariance matrices are low-dimensional, compared to other region descriptors and due to symmetry $\mathbf{C}_R$ has only $\left(d^2 + d\right)/2$ different values. This method also allows to discriminate orientation, scale and illumination features from an image, since the covariance descriptor is not invariant regarding the orientation of the points.

4.2 Metrics

Definition 4.1. The matrix norm is a function $\|\cdot\| : \mathbf{K}^{m \times n} \to \mathbb{R}$ where $\mathbf{K}$ is the set of either real or complex numbers and $\mathbf{K}^{m \times n}$ is the vector space of all matrices of size $m \times n$. This function must satisfies the following five axioms:

1. $\|\mathbf{A}\| \geq 0$
2. $\|\mathbf{A}\| = 0$ if and only if $\mathbf{A} = 0$
3. $\|c\mathbf{A}\| = |c|\|\mathbf{A}\|$
4. $\|\mathbf{A} + \mathbf{B}\| \leq \|\mathbf{A}\| + \|\mathbf{B}\|$
5. $\|\mathbf{AB}\| \leq \|\mathbf{A}\|\|\mathbf{B}\|$

Example 1. The l_1 norm is defined for $\mathbf{A} \in \mathbf{K}^{m \times n}$ by

$$\|\mathbf{A}\|_1 = \sum_{i,j=1}^{n} |a_{ij}|$$

Example 2. The l_∞ norm is defined for $\mathbf{A} \in \mathbf{K}^{m \times n}$ by

$$\|\mathbf{A}\|_\infty = \max_i \sum_{j=1}^{n} |a_{ij}|$$

4.2.1 Frobenius

An $m \times n$ matrix $\mathbf{A}$ can be consider as a particular kind of vector $x = \mathbf{A} \in \mathbb{R}^{(m \cdot n)}$, and its norm is any function that maps $\mathbf{A}$ to a real number $\|\mathbf{A}\|$. In this order of ideas, if we treat the $m \times n$ elements of $\mathbf{A}$ as the elements of an $(m \cdot n)-$dimensional vector, then the $p-$norm of this vector can be used as the $p-$norm of $\mathbf{A}$:

$$\|\mathbf{A}\|_p = \left\{ \sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^p \right\}^{1/p}$$

If we consider the case when $p = 2$ we got the Frobenius norm $\|\mathbf{A}\|_F$

$$\|\mathbf{A}\|_2 = \|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2} = \sqrt{\operatorname{tr}(\mathbf{A}^T \mathbf{A})}$$

4.2.2 Dissimilarity of Two Covariance Matrices

It is well known that the covariance matrix lies on a non-euclidean space, therefore, in order to measure the dissimilarity of two covariance matrices, the distance measure proposed in [7] is used:

$$\rho\left(\mathbf{C}_1, \mathbf{C}_2\right) = \sqrt{\sum_{i=1}^{n} \ln^2 \lambda_i \left(\mathbf{C}_1, \mathbf{C}_2\right)} \tag{1}$$

where $\left\{\lambda_i \left(\mathbf{C}_1, \mathbf{C}_2\right)\right\}_{i=1}^{n}$ are the generalized eigenvalues of $\mathbf{C}_1$ and $\mathbf{C}_2$, computed from

$$\lambda_i \mathbf{C}_1 \mathbf{x}_i - \mathbf{C}_2 \mathbf{x}_i = 0, \quad i = 1 \ldots d$$

and $\mathbf{x}_i \neq 0$ are the **generalized eigenvectors**. The distance measure ρ satisfies the metric axioms for positive definite symmetric matrices (see [29]). The equation (1) can be computed with a $O\left(d^3\right)$ arithmetic operations using numerical methods.

4.3 Covariance Computation

The methods for estimating the covariance matrix of each image region will be now presented:

4.3.1 Maximum Likelihood Estimation

The maximum likelihood estimation (with the bias correction factor) of the covariance matrix is given by

$$\mathbf{C}_R = \frac{1}{n-1} \sum_{k=1}^{n} \left(\mathbf{z}_k - \boldsymbol{\mu}\right) \left(\mathbf{z}_k - \boldsymbol{\mu}\right)^T, \tag{2}$$

where $\boldsymbol{\mu}$ is the mean vector of the features inside the region.

4.3.2 Comedian Estimation

The first robust calculation of the distance is based on a robust estimation of the covariance matrix following the ideas from [5], using the following definition.

Let X and Y be two random vectors. The comedian between X and Y is defined as

$$\mathrm{Com}(X, Y) = \mathrm{Med}[(X - \mathrm{Med}(X))(Y - \mathrm{Med}(Y))].$$

The covariance matrix is then estimated by applying the comedian to each feature of the data (comedian matrix).

4.3.3 Kendall Estimation

This method uses the Kendall rank correlation coefficient, usually known as Kendall's τ coefficient, originally proposed in [14]. In order to define the Kendall's τ coefficient we have to give a notion of concordance.

Definition 4.2. Let (x_j, y_j) and (x_k, y_k) be two elements of a sample $\{(x_i, y_i)\}_{i=1}^{n}$ from a bi-variate population. One says that (x_j, y_j) and (x_k, y_k) are *concordant* if

$$x_j < x_k \quad \text{and} \quad y_j < y_k$$

or if

$$x_j > x_k \quad \text{and} \quad y_j > y_k$$

On the other hand, (x_j, y_j) and (x_k, y_k) are *discordant* if

$$x_j < x_k \quad \text{and} \quad y_j > y_k$$

or if

$$x_j > x_k \quad \text{and} \quad y_j < y_k$$

Given this definition and establishing that the number of distinct pairs of observations in the sample is given by $\binom{n}{2}$ and each pair is either concordant or discordant one can denote as S the number of concordant pairs minus the number of discordant pairs and give the definition of the Kendall's τ for the sample as

$$\tau = \frac{S}{\binom{n}{2}} = \frac{2S}{n(n-1)}$$

With this coefficient, each entry of the covariance matrix is estimated using

$$\mathrm{Cov}(x, y) = \rho_k S_x S_y$$

where ρ_k is Kendall's τ coefficient, and S_x and S_y are the respective standard deviations.

4.3.4 Spearman Estimation

This estimation was performed similarly to Kendall's. The covariance matrix is estimated using

$$\mathrm{Cov}(x, y) = \rho_s S_x S_y$$

where ρ_s is Spearman's correlation coefficient (see [25]), and S_x and S_y are the respective standard deviations. Spearman's coefficient can be defined as follows (see [6]). Suppose that there are n pairs of associated rankings

$$u_1, u_2, \ldots, u_n \quad \text{and} \quad v_1, v_2, \ldots, v_n$$

where the integers u_i $(i = 1, 2, \ldots, n)$ may be taken in ascending order $1, 2, \ldots, n$ and the v_i are a permutation of these integers. The measure of correlation between these rankings given by the Spearman's coefficient is simply the product moment correlation coefficient of u_i, v_i and may be computed from the sum of squared differences

$$S_s = \sum_{i=1}^{n} (u_i - v_i)^2$$

Then, the coefficient is given by

$$\rho_s = 1 - \frac{6S_s}{n^3 - n}$$

4.3.5 Ledoit and Wolf Estimation

This method, proposed in [15], uses the shrinkage constant δ, to 'shrunk' the sample covariance matrix towards the structured estimator. The covariance matrix is estimated using

$$\mathrm{Cov}(x, y) = \delta F + (1 - \delta)\, S$$

where S is the sample covariance matrix, F is a structured estimator and δ is a number between 0 and 1.

4.4 Object Detection

Based on the methodology proposed in [29] for object detection, the following approach is used to locate an object image in an arbitrary image after a nonrigid transformation.

Initially, the location of a pixel in the target image is defined by its coordinates (x, y). The axis coordinates define the pixel location as an array in multi-dimensional space. Each image axis has a length, in pixels, so that the image coordinates run between 1 and the length of the axis [18]. Then, each pixel of the image is converted to a nine-dimensional feature vector

$$F(x, y) = \left[x,\ y,\ R(x, y),\ G(x, y),\ B(x, y),\ \left| \frac{\partial I(x, y)}{\partial x} \right|,\ \left| \frac{\partial I(x, y)}{\partial y} \right|,\ \left| \frac{\partial^2 I(x, y)}{\partial x^2} \right|,\ \left| \frac{\partial^2 I(x, y)}{\partial y^2} \right| \right]^T \tag{3}$$

with RGB color values, and intensity I. The first and second order derivatives are calculated through Sobel filters.

The first step is to estimate the covariance matrix of the input target object T. A brute force search is performed to find matching regions, analyzing nine different scales (four smaller, four larger). Instead of scaling the target image, the size of the search window is varied with a 15% scaling factor between two consecutive scales.

After the search, the best matching 1000 locations are kept and each region is divided into five different sub-regions to evaluate different covariances of the region; this division is better depicted in Fig. 9.

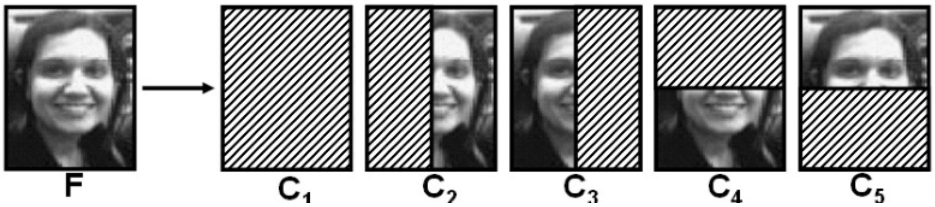


Figure 9: Region subdivision, image taken from [29].

Finally, the objective function is evaluated to find the matching region. The idea behind this function is find the region R that minimizes the dissimilarity of itself with all the remaining:

$$\lambda(R,T) = \min_j \left[\sum_{i=1}^{5} \rho\left(\mathbf{C}_i^R, \mathbf{C}_i^T\right) - \rho\left(\mathbf{C}_j^R, \mathbf{C}_j^T\right) \right]$$

where $\mathbf{C}_j^R$ is the j-th covariance matrix from Fig. 9 for the region R and $\mathbf{C}_j^T$ is the j-th covariance matrix of the target object.

4.5 Performance Measures

For each of the test cases the expected result was extracted to measure the performance of the algorithms. In this manner, two performance measures of each of the test cases were calculated, the value of the objective function and the distance between the covariance matrix of the expected result and the algorithm result. This measures allowed to understand how close the algorithm was to obtaining the correct result and how well the optimization problem was solved.

It is important to remark than metrics, normally, have a wide range of scales between them. This generated difficulties in the way to compare them. To solve this issue, to compare each of the dissimilarity functions the distance to the original object for each of the test cases is were used. The idea was to see if, normally, one of the dissimilarity function outperformed (had a smaller distance to the original) than the other one. Hence, a linear regression of this distances were extracted using the desired metrics. If the linear regression is significant, this would allow to see which metric had a more close result to the expected one. The distance explained in Subsection 4.2.2 was used for calculating the distance between the expected and the algorithm result.

On the other hand, for comparing each of the covariances a similar strategy was used. Let n be the number of test cases. For each covariance matrix i and distance j a matrix $M_{ij} \in \mathbb{R}^{n\times 2}$ was constructed containing the value of the objective function and the distance to the expected result. Then a scatter plot was constructed for each of distance j to see which covariance matrix, generally, minimized both of the performance metrics. As both of the performance measures are desired to be minimized, a Pareto curve is obtained in order to solve the multi-objective optimization problem.

5 Results

This work only presents some localization results that the authors considered valuable for the discussion and concluding remarks. The complete output for all possible 500 combination of covariance matrices, type of input and metric can be found in this link. This zipped file contains several folders, one for each input image with its respective test cases. All files follow an standardized name structure: `noise-cov-dist-test_caseX.jpg` where

`noise`:	0 → Impulse	`cov`:	0 → MLE	`dist`:	0 → Author
	1 → No noise		1 → Comedian		1 → Frobenius
			2 → Spearman		
			3 → Kendall		
			4 → LW		

5.1 Implementation

This section briefly describes some details of the implementation of the algorithm. First, we calculate the first and second order derivatives of all pixels in both the input and target images, instead of calculating on each region considered; this allows the implementation to access the derivatives for each region, used in the features, in constant time. Second, the whole methodology was tested using RGB color scheme. Third, an additional step was included in the detection process: the "transpose" of the region is also considered in search for a matching region; this allows the algorithm to detect possible rotations of the image or object in question by switching the width and height of the searching region.

5.2 Implementations

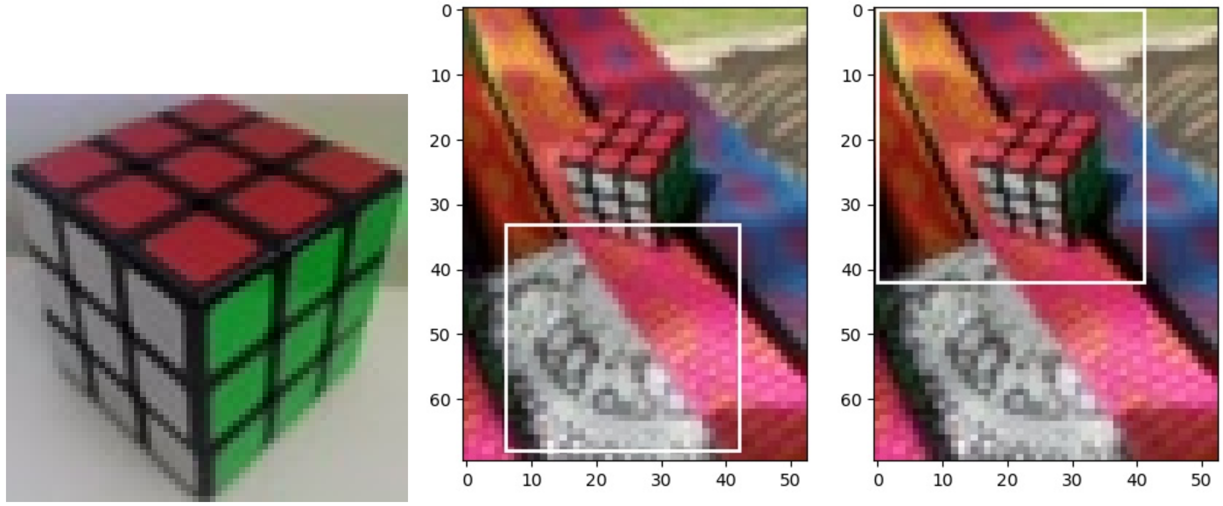
In Table 1, the execution time in seconds and the mega-bytes consumed by the implementations can be found. The "naïve" implementation used `for`-loops and suggested commands to find each of the estimators. The "ours" implementation used only matrix operations. This was tested with a uniform random matrix of size 1000 × 500 once, as one run is enough to compare the speeds as calculating a similar sized matrix rarely varies in execution time.

	Spearman	Kendall	Comedian
Naïve (s)	1.44	162.13	369.78
Ours (s)	**0.30**	**4.11**	**3.66**
Naïve (MB)	**109**	**108**	**104**
Ours (MB)	115	1977	2020

Table 1: Tests with random matrix in $\mathbb{R}^{1000\times500}$.

5.3 Image Results with Kendall Matrix

Some examples of the object localization results, for different input objects, are presented in Figs. 10-13. These results only present locations obtained with the Kendall covariance matrix, varying the metric and for input with and without noise.

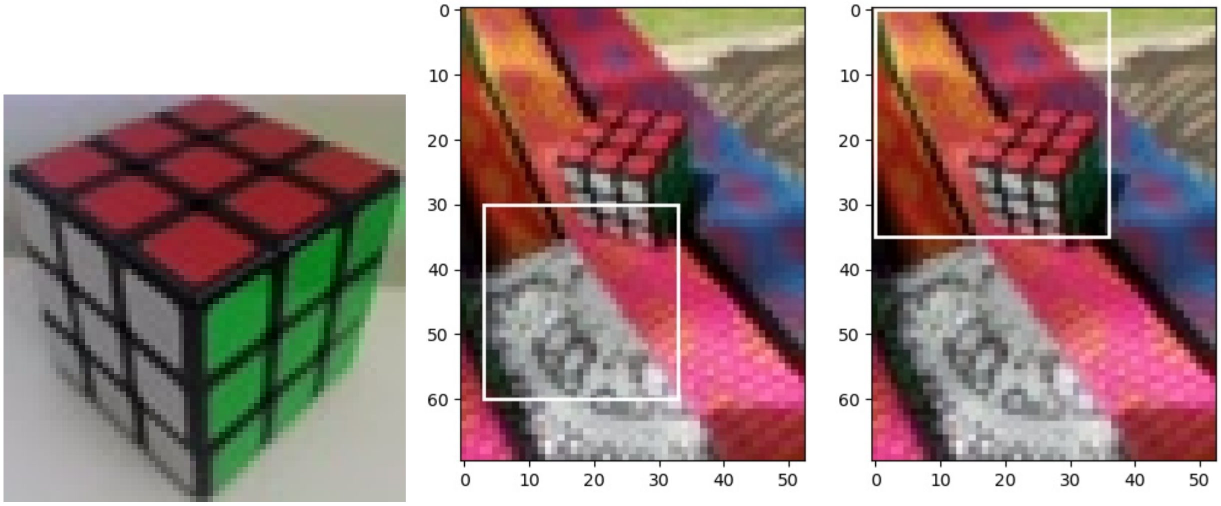


(a) Input.

(b) With noise.

(c) Without noise.

Figure 10: Tests for Rubik's cube with authors' metric.

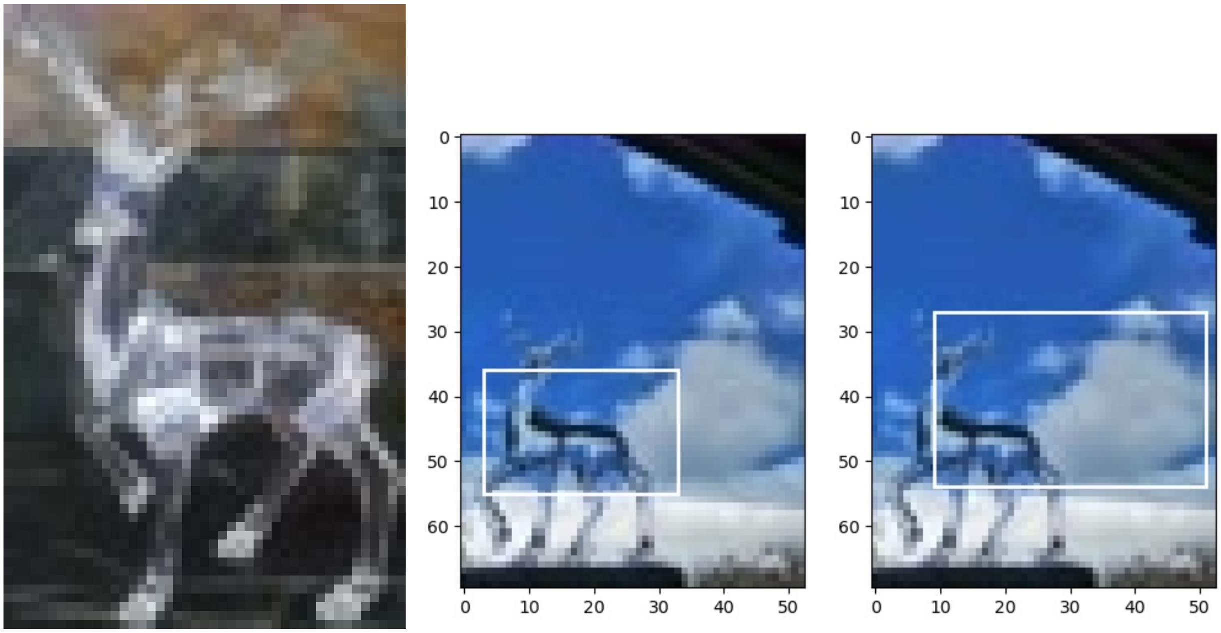


(a) Input.

(b) With noise.

(c) Without noise.

Figure 11: Tests for Rubik's cube with Frobenius metric.

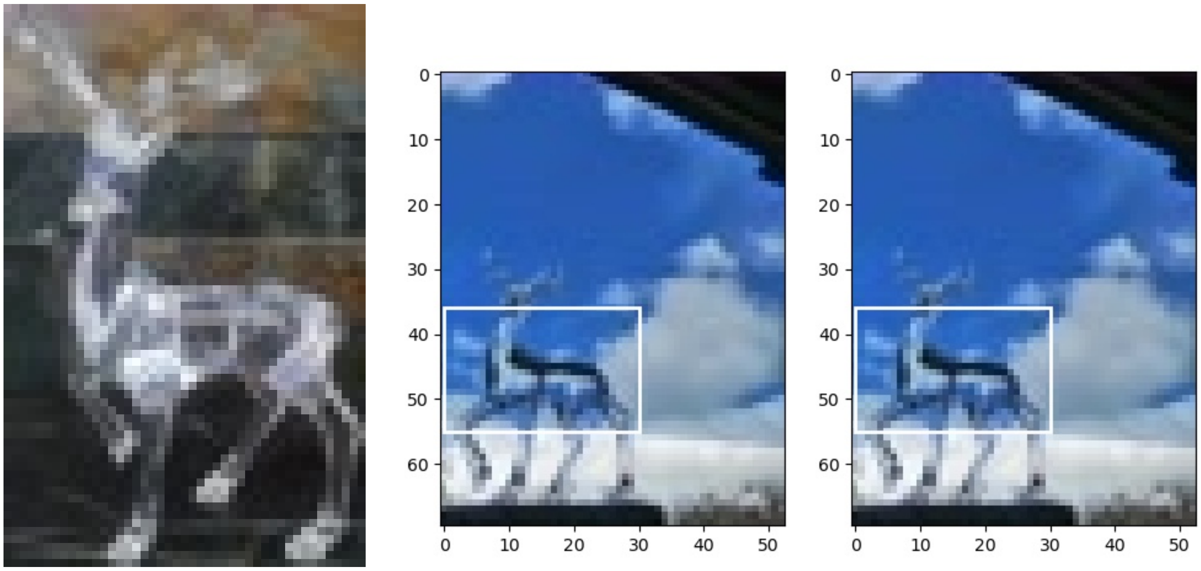


(a) Input.

(b) With noise.

(c) Without noise.

Figure 12: Tests for glass deer with authors' metric.

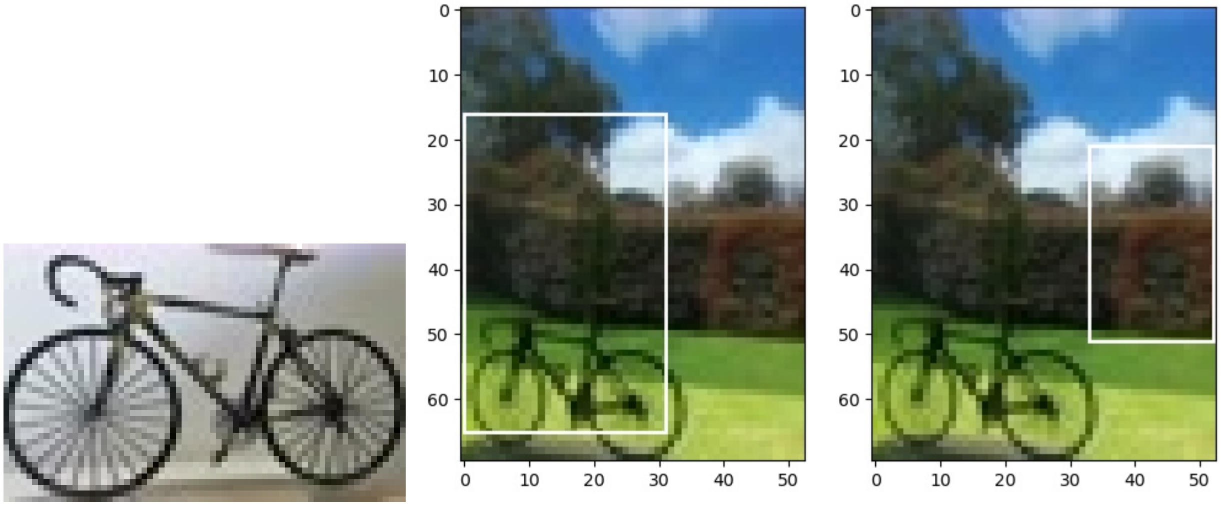


(a) Input.   (b) With noise.   (c) Without noise.

Figure 13: Tests for glass deer with Frobenius metric.

5.4 Image Results with LW

In Fig. 14, the localization of noisy input images is presented, using the Ledoit and Wolf covariance matrix estimation. Note that the authors metric shows better results.



(a) Input.   (b) Authors' metric.   (c) Frobenius metric.

Figure 14: Tests for bike with noise.

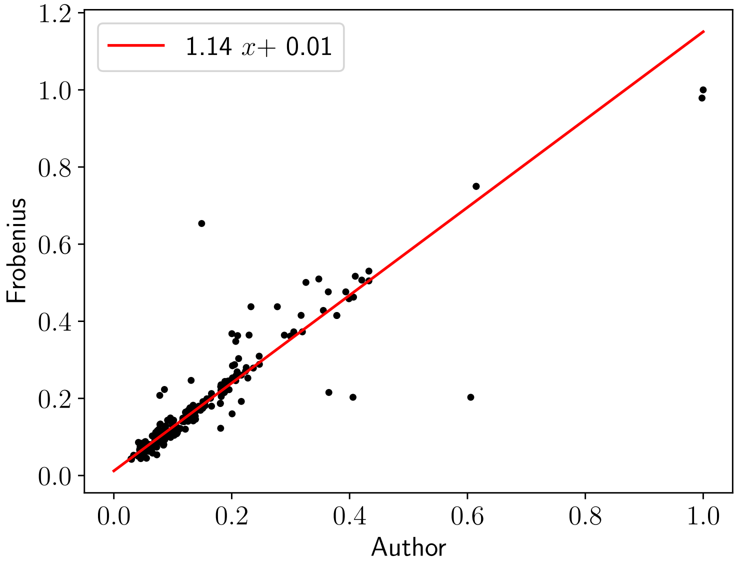


Figure 15: Linear regression comparing metrics.

5.5 Metrics

As can be appreciated in the Fig. 15, the metrics were compared by setting a scatter plot of the distances to the real covariance matrix, using the Frobenius and the authors' dissimilarity measures:

5.6 Covariance Matrices

In Figs. 16 and 17, the Pareto boundaries for covariance matrices using the author's metric and the Frobenius metric, are presented. Note that both cases present a boundary determined only by points obtained using Kendall's covariance matrix, suggesting that this estimation is overall better.

5.7 Noise

Fig. 18 shows the Pareto boundaries for inputs with noise and, same as the results in previous section, the points obtained by Kendall's matrix are non-dominated solutions on the estimated Pareto boundary, which also confirms that this estimation performs better than the other ones.

6 Conclusions

In conclusion, in first place, faster implementations for the Kendall, Spearman and Comedian matrix estimator were successfully created. These implementations were much faster than the naïve implementations with a big downgrade in memory management. This is deemed by the authors as a worthy trade-off as memory can be improved with hardware; on the other hand, implementation speeds cannot be massively improved by the same logic. Furthermore, it is important to remark that

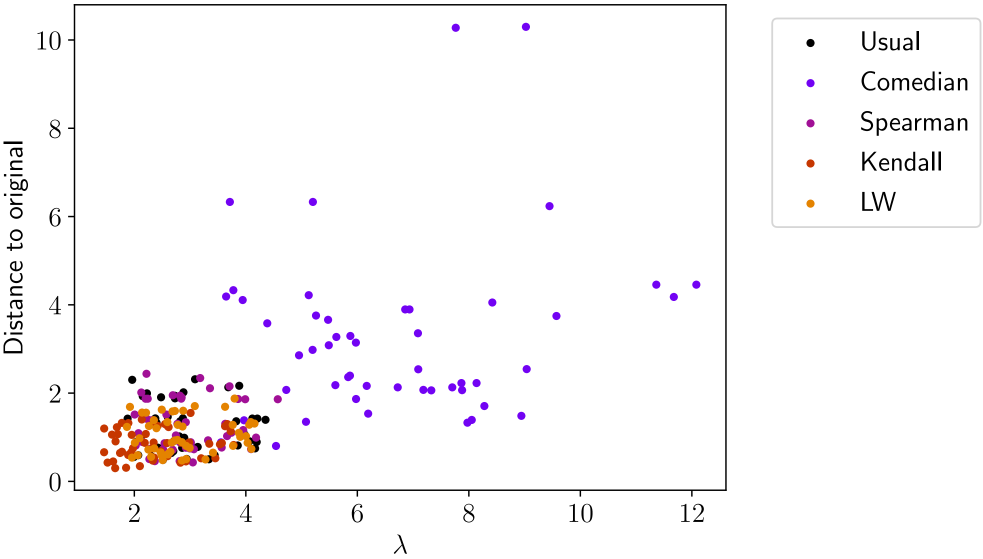


Figure 16: Pareto boundaries for covariance matrices (authors' metric).

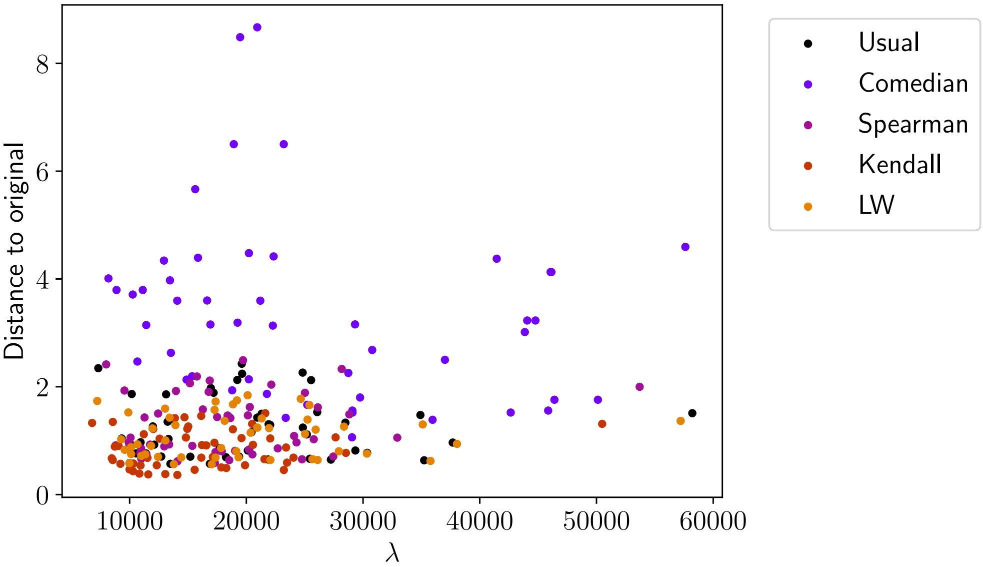


Figure 17: Pareto boundaries for covariance matrices (Frobenius metric).

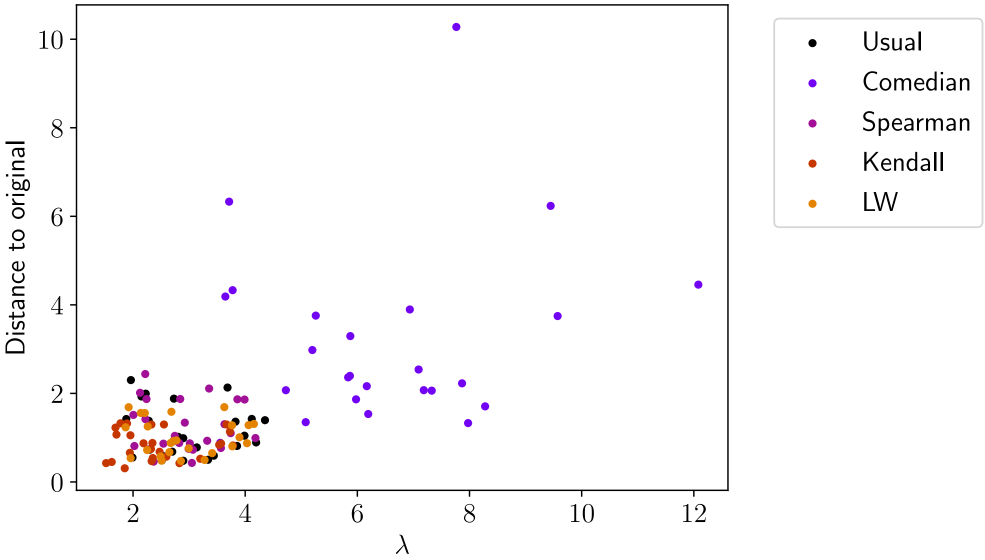


Figure 18: Pareto boundaries for inputs with noise.

this implementations are significantly faster because `for`-loops where avoided and only operations given by the `numpy` package where used.

In second place, the dissimilarity functions were successfully evaluated. The experiments realized in this work showed that the proposed function by Tuzel, Porikli, and Meer [29] had much better performance than the classic Frobenius norm. This was confirmed by Fig. 15, as the linear regression had a $\beta_1 > 1$. Furthermore, this could be justified as covariance matrices do not belong to a euclidean space; hence, the Frobenius dissimilarity could not effectively separate each region.

In third place, the covariance matrices were successfully evaluated. The overall best covariance matrix was using the one based on Kendall's tau coefficient. On the other hand, using this covariance estimator yielded a significantly slower execution time (even with the improvement on the implementation). Therefore, the results also showed that the Ledoit-Wolf covariance estimator is the second best algorithm and is much faster. In this manner, both of the mentioned covariance matrices are selected by the authors as the more effective covariances.

For future work, the integral image data structure [21] could be tested to enhance the speed of the algorithm when exploring the regions of the image.

References

[1] Yali Amit. *2D Object Detection and Recognition: Models, Algorithms, and Networks*. MIT Press, 2002.

[2] Dana Harry Ballard and Christopher M Brown. *Computer Vision*. Prentice Hall, 1982.

[3] P. Chapman et al. "CRISP-DM 1.0: Step-by-step data mining guide". In: 2000.

[4] Zhen Cui et al. "Fusing Robust Face Region Descriptors Via Multiple Metric Learning for Face Recognition in the Wild". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2013, pp. 3554–3561.

[5] Michael Falk. "On MAD and Comedians". In: *Annals of the Institute of Statistical Mathematics* 49.4 (1997), pp. 615–644.

[6] E. C. Fieller, H. O. Hartley, and E. S. Pearson. "Test for Rank Correlation Coefficients". In: *Biometrika* 44.3-4 (1957), pp. 470–481.

[7] Wolfgang Förstner and Boudewijn Moonen. "A Metric for Covariance Matrices". In: *Geodesy-the Challenge of the 3rd Millennium*. Springer, 2003, pp. 299–309.

[8] Walid Hariri et al. "3D Face Recognition Using Covariance Based Descriptors". In: *Pattern Recognition Letters* 78 (2016), pp. 1–7.

[9] Martin Hirzer et al. "Person Re-Identification by Descriptive and Discriminative Classification". In: *Scandinavian conference on Image analysis*. Springer. 2011, pp. 91–102.

[10] Xiaopeng Hong et al. "Sigma Set: A Small Second Order Statistical Region Descriptor". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2009, pp. 1802–1809.

[11] Thomas Huang. "Computer Vision: Evolution and Promise". In: (1996).

[12] Xiaoyue Jiang et al. *Deep Learning in Object Detection and Recognition*. Springer, 2018.

[13] Levent Karacan, Erkut Erdem, and Aykut Erdem. "Structure-Preserving Image Smoothing Via Region Covariances". In: *ACM Transactions on Graphics (TOG)* 32.6 (2013), pp. 1–11.

[14] Maurice G Kendall. "A New Measure of Rank Correlation". In: *Biometrika* 30.1/2 (1938), pp. 81–93.

[15] Olivier Ledoit and Michael Wolf. "Honey, I Shrunk the Sample Covariance Matrix". In: *The Journal of Portfolio Management* 30.4 (2004), pp. 110–119.

[16] Xi Li et al. "A Survey of Appearance Models in Visual Object Tracking". In: *ACM transactions on Intelligent Systems and Technology (TIST)* 4.4 (2013), pp. 1–48.

[17] Yali Li et al. "Feature Representation for Statistical-Learning-Based Object Detection: A Review". In: *Pattern Recognition* 48.11 (2015), pp. 3542–3559.

[18] Edward M Mikhail, Mark L Akey, and O Robert Mitchell. "Detection and Sub-Pixel Location of Photogrammetric Targets in Digital Images". In: *Photogrammetria* 39.3 (1984), pp. 63–83.

[19] Sakrapee Paisitkriangkrai, Chunhua Shen, and Jian Zhang. "Fast Pedestrian Detection Using a Cascade of Boosted Covariance Features". In: *IEEE Transactions on Circuits and Systems for Video Technology* 18.8 (2008), pp. 1140–1151.

[20] Yanwei Pang, Yuan Yuan, and Xuelong Li. "Gabor-Based Region Covariance Matrices for Face Recognition". In: *IEEE Transactions on circuits and systems for video technology* 18.7 (2008), pp. 989–993.

[21] Fatih Porikli. "Integral histogram: A fast way to extract histograms in cartesian spaces". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. IEEE. 2005, pp. 829–836.

[22] Fatih Porikli and Tekin Kocak. “Robust license plate detection using covariance descriptor in a neural network framework”. In: *2006 IEEE International Conference on Video and Signal Based Surveillance*. IEEE. 2006, pp. 107–107.

[23] Fatih Porikli, Oncel Tuzel, and Peter Meer. “Covariance Tracking Using Model Update Based on Lie Algebra”. In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*. Vol. 1. IEEE. 2006, pp. 728–735.

[24] Ravishankar Sivalingam et al. “Tensor Sparse Coding for Region Covariances”. In: *European conference on computer vision*. Springer. 2010, pp. 722–735.

[25] Charles Spearman. “”General Intelligence” Objectively Determined and Measured.” In: (1961).

[26] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer Science & Business Media, 2010.

[27] Diego Tosato et al. “Multi-Class Classification on Riemannian Manifolds for Video Surveillance”. In: *European conference on computer vision*. Springer. 2010, pp. 378–391.

[28] Oncel Tuzel, Fatih Porikli, and Peter Meer. “Human Detection via Classification on Riemannian Manifolds”. In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2007, pp. 1–8.

[29] Oncel Tuzel, Fatih Porikli, and Peter Meer. “Region Covariance: A Fast Descriptor for Detection and Classification”. In: *European conference on computer vision*. Springer. 2006, pp. 589–600.

[30] V. Tyagi. *Content-Based Image Retrieval: Ideas, Influences, and Current Trends*. Springer Singapore, 2018. ISBN: 9789811067594.

[31] Yi Wu, Jinqiao Wang, and Hanqing Lu. “Robust Bayesian Tracking on Riemannian Manifolds via Fragments-Based Representation”. In: *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE. 2009, pp. 765–768.

[32] Yi Wu et al. “Probabilistic Tracking on Riemannian Manifolds”. In: *2008 19th International Conference on Pattern Recognition*. IEEE. 2008, pp. 1–4.

[33] Yi Wu et al. “Real-Time Probabilistic Covariance Tracking with Efficient Model Update”. In: *IEEE Transactions on Image Processing* 21.5 (2012), pp. 2824–2837.

[34] Yi Wu et al. “Real-Time Visual Tracking via Incremental Covariance Tensor Learning”. In: *2009 IEEE 12th International Conference on Computer Vision*. IEEE. 2009, pp. 1631–1638.

[35] Jian Yao and Jean-Marc Odobez. *Fast Human Detection from Videos Using Covariance Features*. Tech. rep. Idiap, 2007.

[36] Zhong-Qiu Zhao et al. “Object Detection with Deep Learning: A Review”. In: *IEEE transactions on neural networks and learning systems* 30.11 (2019), pp. 3212–3232.