Ricardo Eugenio González Valenzuela

"Linear Dimensionality Reduction Applied to SIFT and SURF Feature Descriptors"

"*Redução Linear de Dimensionalidade Aplicada aos Descritores de Características SIFT e SURF*"

CAMPINAS
2014

University of Campinas
Institute of Computing

Universidade Estadual de Campinas
Instituto de Computação

Ricardo Eugenio González Valenzuela

"Linear Dimensionality Reduction Applied to SIFT and SURF Feature Descriptors"

Supervisor: / *Orientador:* **Prof. Dr. Hélio Pedrini**

Co-Supervisor: / *Co-orientador:* **Prof. Dr. William Robson Schwartz**

"Redução Linear de Dimensionalidade Aplicada aos Descritores de Características SIFT e SURF"

MSc Dissertation presented to the Post Graduate Program of the Institute of Computing of the University of Campinas to obtain a Mestre degree in Computer Science.

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Computação da Universidade Estadual de Campinas para obtenção do título de Mestre em Ciência da Computação.

This volume corresponds to the final version of the Dissertation defended by Ricardo Eugenio González Valenzuela, under the supervision of Prof. Dr. Hélio Pedrini.

Este exemplar corresponde à versão final da Dissertação defendida por Ricardo Eugenio González Valenzuela, sob orientação de Prof. Dr. Hélio Pedrini.

Hélio Pedrini

Supervisor's signature / *Assinatura do Orientador*

CAMPINAS
2014

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Maria Fabiana Bezerra Muller - CRB 8/6162

G589L González Valenzuela, Ricardo Eugenio, 1984-
Linear dimensionality reduction applied to SIFT and SURF feature descriptors / Ricardo Eugenio González Valenzuela. – Campinas, SP : [s.n.], 2014.

Orientador: Hélio Pedrini.
Coorientador: William Robson Schwartz.
Dissertação (mestrado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Processamento de imagens - Técnicas digitais. 2. Descritores. 3. Compressão de dados (Computação). I. Pedrini, Hélio,1963-. II. Schwartz, William Robson. III. Universidade Estadual de Campinas. Instituto de Computação. IV. Título.

Informações para Biblioteca Digital

Título em outro idioma: Redução linear de dimensionalidade aplicada aos descritores de características SIFT e SURF
Palavras-chave em inglês:
Image processing - Digital techniques
Descriptors
Data compression (Computer science)
Área de concentração: Ciência da Computação
Titulação: Mestre em Ciência da Computação
Banca examinadora:
Hélio Pedrini [Orientador]
Anderson de Rezende Rocha
Moacir Pereira Ponti Junior
Data de defesa: 20-02-2014
Programa de Pós-Graduação: Ciência da Computação

TERMO DE APROVAÇÃO

Defesa de Dissertação de Mestrado apresentada por Ricardo Eugenio González Valenzuela – RA 100613, Aprovada em 20 de fevereiro de 2014, pela Banca examinadora composta pelos Professores Doutores:

Moacir Ponti

Prof. Dr. Moacir Pereira Ponti Junior
ICMC/USP

Anderson de Rezende Rocha

Prof. Dr. Anderson de Rezende Rocha
IC / UNICAMP

Hélio Pedrini

Prof. Dr. Hélio Pedrini
IC / UNICAMP

Institute of Computing / *Instituto de Computação*
University of Campinas / *Universidade Estadual de Campinas*

Linear Dimensionality Reduction Applied to SIFT and SURF Feature Descriptors

Ricardo Eugenio González Valenzuela[1]

February 20, 2014

Examiner Board/*Banca Examinadora*:

- Prof. Dr. Hélio Pedrini (Supervisor/*Orientador*)
- Prof. Dr. Moacir Pereira Ponti Junior
  ICMC - USP
- Prof. Dr. Ricardo da Silva Torres
  IC - UNICAMP
- Prof. Dr. Luis Gustavo Nonato
  ICMC - USP (Substitute/*Suplente*)
- Prof. Dr. Anderson de Rezende Rocha
  IC - UNICAMP (Substitute/*Suplente*)

[1]Financial support: CNPq scholarship (process 133057/2011-9) 2011–2013
Santander Bank Scholarship Program of International Exchange Mobility - 2013

Abstract

Robust local descriptors usually consist of high dimensional feature vectors to describe distinctive characteristics of images. The high dimensionality of a feature vector incurs into considerable costs in terms of computational time and storage requirements, which affects the performance of several tasks that employ feature vectors, such as matching, image retrieval, and classification. To address these problems, it is possible to apply some dimensionality reduction techniques by building a projection matrix, which explains adequately the importance of the data in other basis. This dissertation aims at applying linear dimensionality reduction to SIFT and SURF descriptors. Its main objective is to demonstrate that, even risking to decrease the accuracy of the feature vectors, the dimensionality reduction can result in a satisfactory trade-off between computational time and storage. We perform the linear dimensionality reduction through Random Projections (RP), Independent Component Analysis (ICA), Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and Partial Least Squares (PLS) in order to create lower dimensional feature vectors. This work evaluates such reduced feature vectors in a matching application, as well as their distinctiveness in an image retrieval application. The computational time and memory usage are then measured by comparing the original and the reduced feature vectors.

Resumo

Descritores locais robustos normalmente compõem-se de vetores de características de alta dimensionalidade para descrever atributos discriminativos em imagens. A alta dimensionalidade de um vetor de características implica custos consideráveis em termos de tempo computacional e requisitos de armazenamento afetando o desempenho de várias tarefas que utilizam descritores de características, tais como correspondência, recuperação e classificação de imagens. Para resolver esses problemas, pode-se aplicar algumas técnicas de redução de dimensionalidade, essencialmente construindo-se uma matriz de projeção que explique adequadamente a importancia dos dados em outras bases. Esta dissertação visa aplicar técnicas de redução linear de dimensionalidade aos descritores SIFT e SURF. Seu principal objetivo é demonstrar que, mesmo com o risco de diminuir a precisão dos vetores de caraterísticas, a redução de dimensionalidade pode resultar em um equilíbrio adequado entre tempo computacional e recursos de armazenamento. A redução linear de dimensionalidade é realizada por meio de técnicas como projeções aleatórias (RP), análise de componentes principais (PCA), análise linear discriminante (LDA) e mínimos quadrados parciais (PLS), a fim de criar vetores de características de menor dimensão. Este trabalho avalia os vetores de características reduzidos em aplicações de correspondência e de recuperação de imagens. O tempo computacional e o uso de memória são medidos por comparações entre os vetores de características originais e reduzidos.

A DIOS quien por sobre todo siempre está conmigo.

Acknowledgements

I would like to thank all my friends and family who gave me their support over all my master's studies. Special thanks to Professor Hélio Pedrini, Lehilton Chaves Pedrosa and Junior Fabian Arteaga.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This chapter describes the problem addressed in this work, the main objectives and contributions, as well as the structure of the dissertation.

1.1 Motivation

Feature descriptors are frequently used to support the analysis of data content, since they contain information to reliably identify objects or classes of objects present in some data, which are images in our case. For this reason, descriptors are fundamental in applications such as image retrieval, image matching, object recognition, tracking, among others.

Descriptors can be global or local. The former type describes an image as a whole, whereas the latter describes, separately, different parts of the image.

Local feature descriptors are employed in the task of describing images [26, 39] by representing numerous characteristics such as color, texture and shape. Many of these local features represent simple concepts and might be inadequate to describe an image when employed individually. Therefore, robust local features group several simple local features or may even build some feature description based on the neighborhood of an interest point. In terms of quality, the more distinctive and invariant the interest points are, the more accurate the processes that use them will become. Consequently, as a large amount of characteristics could be joined to better describe an image, higher dimensional feature vectors are obtained as the result.

Local descriptors usually generate robust local feature descriptions by performing two stages. The first one is called keypoint detection, which performs some processes to find points in the images that are considered to be more distinctive. The second stage is called feature description, in which a set of data is generated in order to represent some important features such as orientation, color, shape, texture or illumination.

Several feature descriptors have been developed over the last decades [46]. Among the most remarkable state-of-the-art descriptors, we can stand out SIFT (Scale Invariant Feature Transform), developed by Lowe [21] in 1999, and SURF (Speeded Up Robust Features), developed by Bay et al. [3] in 2006. The main advantages of the SURF descriptor lie on the fact that it detects less keypoints and describes them with half of the dimensions that SIFT uses to describe its keypoints (*i.e.*, 64 dimensions). However, the SIFT feature descriptor is still better to represent some transformations and distortions as rotation, scaling and blurred images [17].

Although the above mentioned descriptors have demonstrated to achieve good results, their dimensionality is still large. Considering applications where millions of images are involved, or even less than millions but high resolution images, each image will be represented by hundred or thousands of descriptors each one with several dimensions. This last situation allows us to see that the amount of information generated by the descriptors needs to be carefully reduced, otherwise, it will incur on costly machine processes.

It is straightforward to recognize two aspects to be solved, which are the quantity of keypoints and the descriptor dimensionality. The issue related to the number of keypoints can be addressed by using a bag-of-feature approach [7] since it preserves the feature vector individual influences while adding generalization, as each visual word represents a set of similar features. On the other hand, to deal with the descriptors high dimensionality, it is possible to use some dimensionality reduction techniques, such as Principal Component Analysis (PCA) [16] or Linear Discriminant Analysis (LDA) [2], which compute projection matrices by using different mathematical models to explain the data.

Dimensionality reductors are broadly used in such cases, where they usually take into consideration some important information to construct the projection matrices which can maintain the distinctiveness of the original feature vectors. This is, vectors integrating the projection matrices are computed by weighting aspects such as covariance or correlation between the classes and variables, which is expected to conduct any data projected onto a projection matrix to a better reduced description. This is the main reason that motivates us to apply reduction techniques through the task of reducing feature vectors.

In the literature, the PCA technique has already been used in [18] to reduce 3042-dimensional feature vectors to 32 dimensions. Computing reduced feature vectors through linear dimensionality reduction techniques proved to be a suitable approach for reducing the consumption of computational resources while maintaining or even achieving better performance than the original feature vectors.

1.2 Objectives and Contributions

The main objective of this work is to demonstrate that the reduction of the amount of information generated by SIFT and SURF feature vectors through bag-of-features and dimensionality reduction techniques are suitable for image analysis applications.

Some contributions arising from this work include the usage of less disk memory for applications that demand reading and storing high amounts of descriptor information, improvement in terms of computational time to process lower dimensional feature vectors, as well as achievement of better performance when the dimensionality reduction techniques can remove redundancy or noise from the data.

1.3 Text Organization

This dissertation is organized as follows.

Chapter 2 briefly describes some concepts related to this work, such as SIFT and SURF image descriptors, linear dimensionality reduction techniques and bag-of-feature techniques.

Chapter 3 compares the performance of both SIFT and SURF descriptors against their reduced versions. The reduced descriptors are projected onto a matrix of eigenvectors trained with the PCA technique [41]. This comparison is evaluated by means of two applications, image matching, and image retrieval.

Chapter 4 extends the work described in Chapter 3 by applying other dimensionality reduction techniques and conducting experiments over large datasets for image matching and retrieval purposes [42].

Chapter 5 presents an image retrieval application addressed by SIFT and SURF descriptors reduced through LDA, where the reduced number of keypoints are modeled by a single bag-of-feature representation [40].

Finally, Chapter 6 concludes the dissertation with final remarks and directions for future work.

Chapter 2

Background

This chapter briefly describes some concepts related to the topic investigated in this work, such as local feature descriptors, linear dimensionality reduction techniques, and bag-of-features, for image analysis and computer vision areas.

One important issue is to measure features or characteristics of objects present in the images. These characteristics can be based on attributes such as intensity, color or texture. An example of local characteristic is a pattern of a certain image region that differs from its immediate neighborhood.

Local characteristics are broadly used in object recognition applications, such as in scene classification, texture analysis, image retrieval and video mining, which evidence the importance of obtaining distinctive local characteristics. Such local characteristics can be classified according to the description of the image that they provide [39]:

- repeatability: relates two images from the same object viewed from different viewpoints;
- distinctiveness / informativeness: presents a significant variation in contrast with the intensity patterns underlying of the detected characteristic;
- locality: describes part of the image to reduce the likelihood of occlusion;
- quantity: extracts enough characteristics to properly describe an image, even if the image is small;
- accuracy: finds features with precise positions, even if there are variations and deformations;
- efficiency: extract features in a manner that can be used in time-critical applications.

Ideally, a local feature must be robust to noise presence or lighting changes and invariant with respect to affine transformations. In several situations, local characteristics, individually, are not enough to describe an image, such that the combination of features becomes necessary. Therefore, a set of local characteristics must be selected properly so they can efficiently represent an image, taking into consideration the application to be performed. However, when several characteristics are joined, it occurs a trade-off between accuracy and time consumption.

2.1 Feature Descriptors

Several feature descriptors for image analysis have been proposed in the literature. In the following sections, two important descriptors for detection of keypoints in images are described.

2.1.1 Scale Invariant Feature Transformation

Scale Invariant Feature Transformation (SIFT), as explained in [22], is a technique for object description that uses characteristics invariant to scaling, translation, and rotation, and partially invariant to changes of lighting. This method allows a robust object recognition, even in partially occluded images. The SIFT method consists of four main steps:

1. detection of scale-space extrema: this stage detects candidates, to be taken as keypoints, at several octaves and scales (Figure 2.1), by performing a Difference-of-Gaussian (DoG) function (Figure 2.2). One octave groups several copies of the image within the same size, then each octave represents a different resolution for the image. Scale is varied in each octave, from image to image, by smoothing each image with a different Gaussian kernel.

2. localization of keypoints: a detected candidate is selected according to a measure of desired stability (Figure 2.3).

3. assignment of orientation: one or more gradient orientations are assigned according to the keypoint neighborhood (Figure 2.4). Furthermore, this feature processing is invariant to scaling, translation, and orientation, since such characteristic is the basis of the feature construction.

4. description of keypoints: for each keypoint neighborhood, the obtained gradients are transformed into an 8-bin histogram representation, which will allow the invariance to high levels of changes or distortions (Figure 2.5).

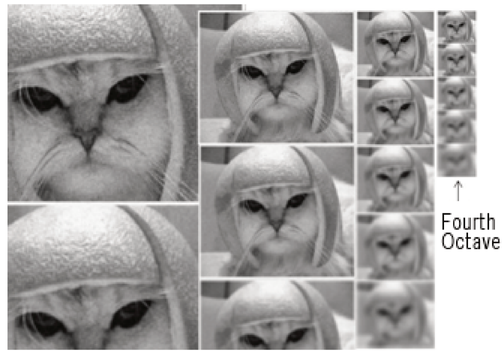


Figure 2.1: Multi-scaled image (figure extracted from [35]).

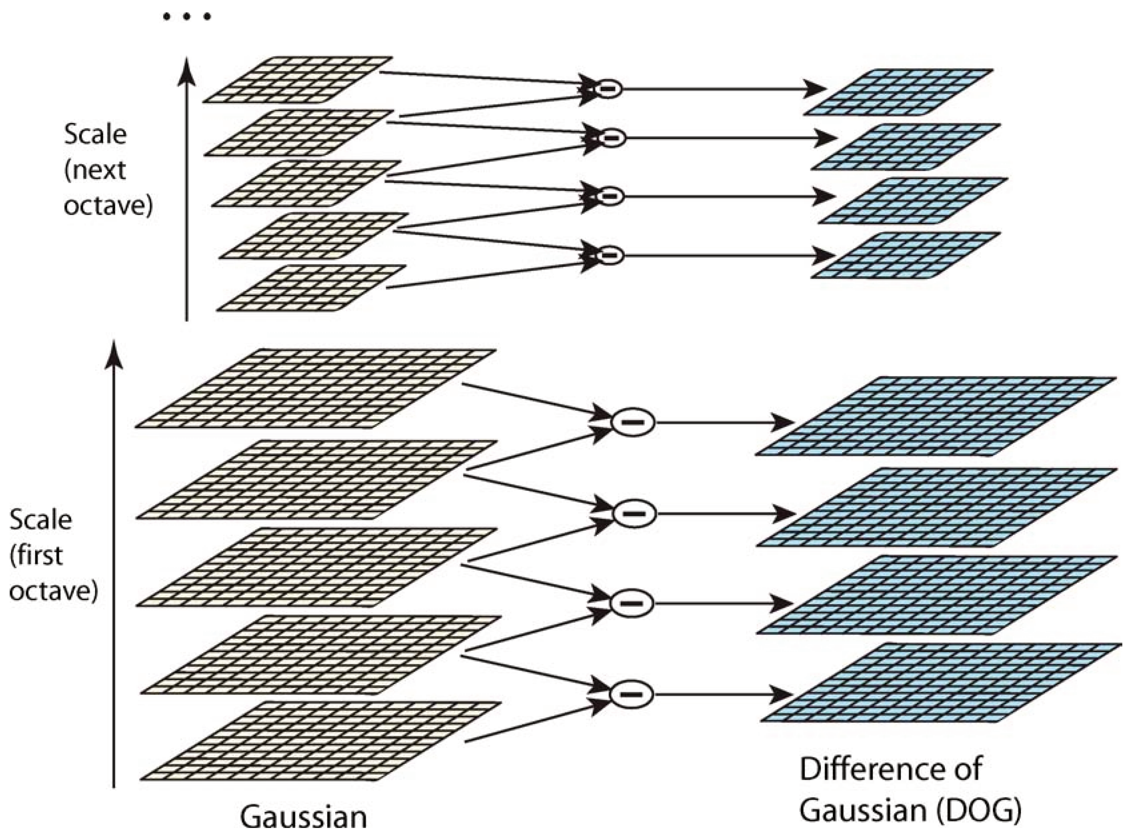


Figure 2.2: Difference-of-Gaussian (figure extracted from [22]).

Therefore, the keypoints represent a set of relevant points of an image that describe it and will make possible for a computer to recognize that image. A local region of the image, that is, the neighborhood of one keypoint, is described by creating a high-dimensional vector that represents the gradients corresponding to each point in the region. This description becomes the representation of the keypoint which will be used to differentiate

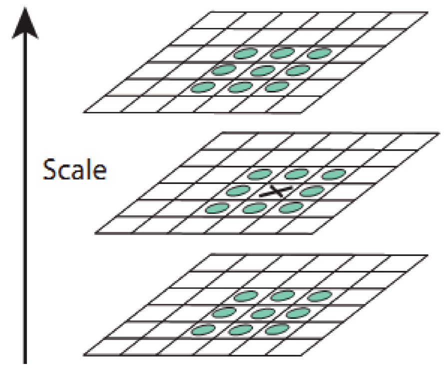


Figure 2.3: Detection of keypoints (figure extracted from [22]).

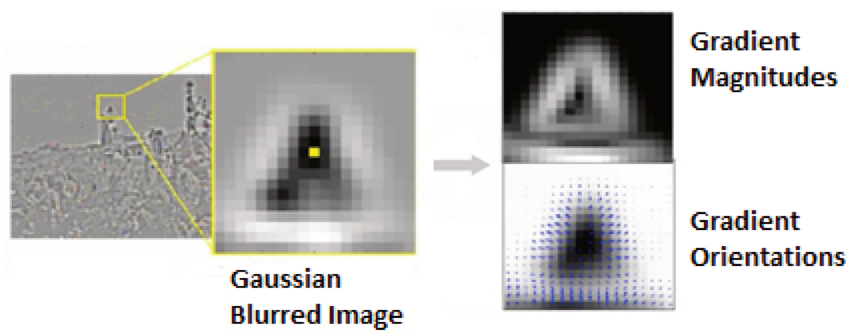


Figure 2.4: Example of a region gradient orientation (figure extracted from [34]).

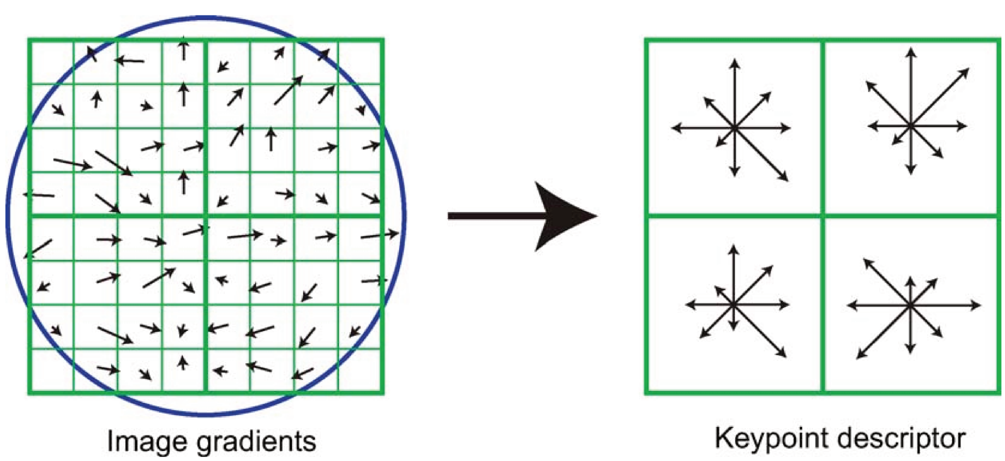


Figure 2.5: Representation of a SIFT descriptor (figure extracted from [22]).

one keypoint from others.

Typically, a large number of points of interest can be extracted from images, which can lead to robustness in the description of small objects. The points of interest are detected in a wide range of scales. Points of interest on small scales can be used to detect small or partially occluded objects, whereas points of interest on larger scales allow good performance for images subject to noise. The technique is efficient, such that thousands of points of interest can be extracted from an image with relatively low computational cost [22].

2.1.2 Speeded Up Robust Features

The Speeded Up Robust Feature (SURF) algorithm also detects and describes keypoints in an image. Inspired partially over SIFT, it aims at achieving lower computational time while maintaining a similar accuracy [3]. SURF performs similar or even better than the SIFT algorithm with respect to repeatability, distinctiveness and robustness, and can also be calculated more efficiently. In part, this is achieved by the use of integral images that allows the application of efficient convolutions, as well as by taking advantage of the use of a Hessian matrix on the detection stage, and a distribution-based descriptor. The main procedures involved in SURF are:

1. creation of integral images [43]: the input of an integral image $I_\Sigma(p)$ at one point $p = (x, y)$ represents the sum of all the pixels in the input image I of a rectangular region formed by the point p and the origin, $I_\Sigma(p) = \sum_{i=0}^{x} \sum_{j=0}^{y} I(i, j)$. The computed $I_\Sigma(p)$ allows to calculate the sum of the intensities over any rectangular area, independently of its size, by performing only four additions (Figure 2.6).

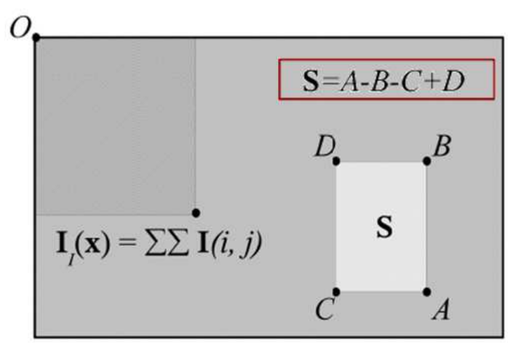


Figure 2.6: Integral image (figure extracted from [45]).

2. representation of scale-space: a box filter is used to obtain an approximation of the second order derivative of Gaussian (Figure 2.7). The initial box filter is 9×9 pixels with $\sigma = 1.2$. The convolution between the image and the box filter will generate

an initial scale layer, called as the $s = 1.2$ (directly related to the previous value: $\sigma = 1,2$). The following layers are obtained by filtering the image with gradually larger masks, considering the discrete nature of the integral images and the specific structure of box filters.

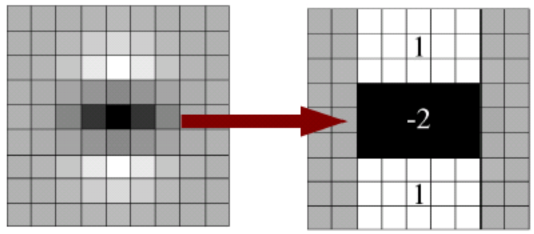


Figure 2.7: Approximation of the second order Gaussian by a 9×9 box filter (figure extracted from [3]).

3. identification of keypoints using Fast-Hessian [25]: the keypoints are located by applying non-maximum suppression over a neighborhood of $3 \times 3 \times 3$ pixels (Figure 2.8). Afterwards, the maximum values for the Hessian matrix determinant are interpolated through the method proposed by Brown et al. [5].

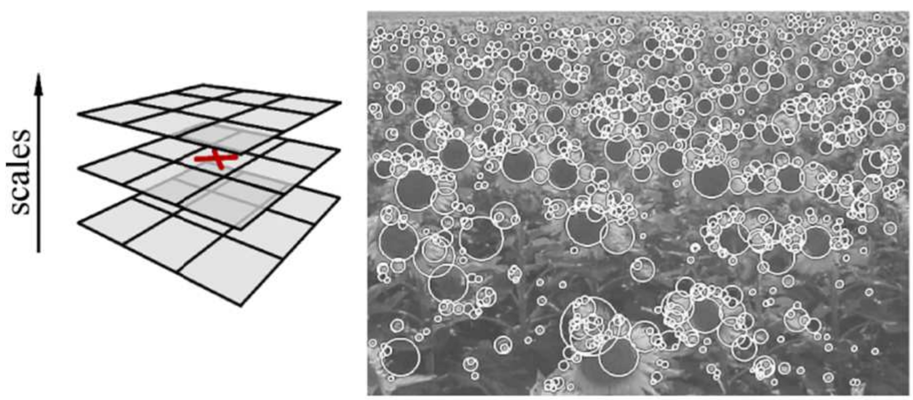


Figure 2.8: Non-maximum suppression to detect keypoints (figure extracted from [3]). Left figure shows the $3 \times 3 \times 3$ neighborhood of an interest point. Right figure shows by several circles the detected interest points, each circle size denote the scale where the interest point was detected.

4. construction of each keypoint descriptor:

   (a) keypoint description: the application of the Haar wavelets [24] will generate some x and y values for each keypoint. The wavelet responses are computed over x and y direction, with a Gaussian centered on the keypoint, in its detected scale. This responses are then represented as vectors in a space where the horizontal and vertical response strengths correspond to the abscissa and ordinate axes.

(b) assignment of orientation: The sum of the wavelet responses in x and y, within a sliding window ($\frac{\pi}{3}$), will represent a new vector, which lends its orientation to the interest point (Figure 2.9).

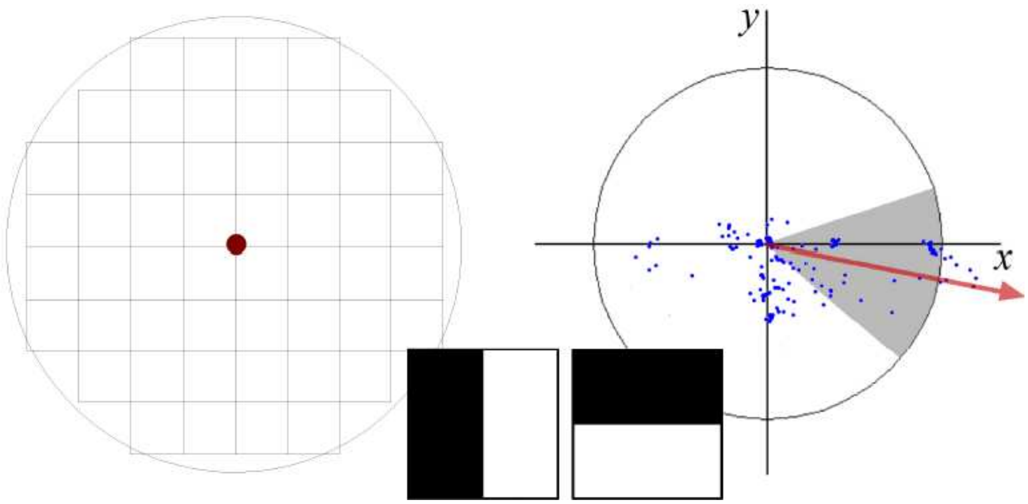


Figure 2.9: SURF descriptor orientation assignment (figure extracted from [45]).

(c) descriptor extraction: a square region is constructed around the keypoint, and oriented in the direction previously assigned. The Haar wavelet responses denote the horizontal direction by dx and the vertical direction by dy. Therefore, the response absolute value sum is also calculated in order to know the polarity and intensity changes. Thus, it creates a vector v of four dimensions to describe its underlying structure, $v = (\Sigma dx, \Sigma dy, \Sigma|dx|, \Sigma|dy|)$ (Figure 2.10).

2.2 Linear Dimensionality Reduction Techniques

Reduction of dimensionality is one of the most important tasks in multivariate analysis and is especially critical for multivariate regressions [23]. Several of the independent variables used in a regression show a high correlation, since those variables can measure the same characteristics.

Some dimensionality reduction techniques that perform data transformation from high dimensional spaces to lower dimensional spaces are presented in the following sections.

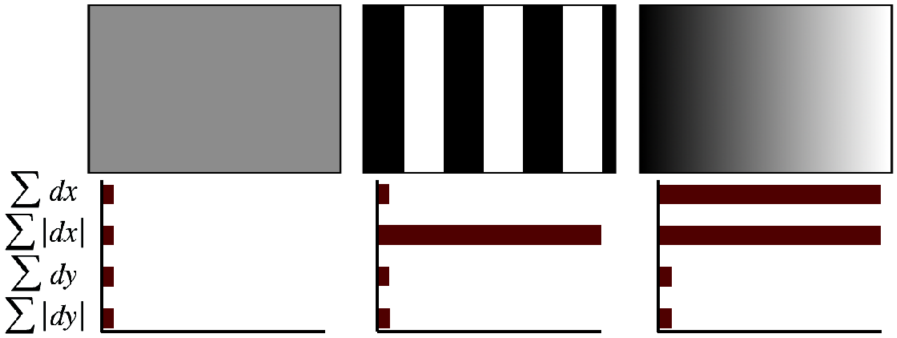


Figure 2.10: SURF descriptor (figure extracted from [3]).

2.2.1 Random Projections

In linear dimensionality reduction the Random Projection (RP) technique [38] has proven to be an efficient way to reduce data dimensionality [4]. This technique is based on the Johnson and Lindenstrauss lemma [8, 15], which shows that the distances between points in some space is nearly preserved if they are projected onto a randomly selected space. It has also being applied to image and text data [4]. This states that any data matrix, X, with n points in a $\Re^d$ space can be projected over a matrix of random vectors, R, in an $\Re^p$ space, where $d \gg p$, as shown in Equation 2.1.

$$A^{n\times p} = R^{n\times d} \times X^{d\times p} \tag{2.1}$$

2.2.2 Independent Component Analysis

The Independent Component Analysis (ICA) [13] is a blind source separation technique. This means that it estimates original source signals from several mixed signals, without having important details such as the transmission channel characteristics. Therefore, this technique is applied to find independent signals.

As it is explained in [28], we can think of some independent signals from several sources, which could include noise. More precisely, there are N non-observable independent signals, $s_i(t)$, $i = 1, ..., N$, such that each signal, s_i, is spammed by some fixed probability distribution at some time t. Also, we have N signals, $x_i(t)$, corresponding to N sensors, obtained by a mixture of the original independent signals. The sensors should be spatially distributed in order to record different mixture of the sources, which means that each sensor one unique signal will be stronger that the others. Therefore, Equation 2.2 is modeled as

$$x(t) = As(t) \tag{2.2}$$

where A is a mixing matrix, from which there is no known information. Therefore, the objective is to model the Equation 2.3

$$s(t) = Wx(t) \tag{2.3}$$

where W is the un-mixing matrix (A^{-1}). Estimating the un-mixing matrix will lead to better approximation of the original signals.

In order to compute the un-mixing matrix, W, there are several methods available in the literature. This includes second-order methods, such as PCA and Factor analysis, and higher-order methods, such as Projection Pursuit, Redundancy Reduction, and Blind Deconvolution [13]. In this work, we use the higher-order method based on the deflation approach [9].

2.2.3 Principal Component Analysis

The Principal Component Analysis (PCA) [31] is a technique for dimensionality reduction recommended when there is a large number of numeric variables and it is needed to find a set with fewer artificial variables, that are called the principal components, that have the largest variance in the observed variables. The principal components can be used as predictors or criterion variables in subsequent analyses.

Usually, when obtaining a dataset corresponding to a series of variables (that is possibly large), there can be some redundancy in the variables, so that some variables are correlated, which means that they are measuring the same information. Due to such redundancy, it is possible to reduce the number of observed variables to some small number of principal components (the artificially created variables). In PCA, such components correspond to those that are responsible for the largest variance in the dataset.

The objective of PCA is to convert a set of highly correlated variables into a set of independent variables. This is obtained using linear transformations, and possibly reducing the number of variables. The main steps in PCA are described as follows [11].

Consider a data sample vector x as

$$x = (x_1, ..., x_n)^T \tag{2.4}$$

and such that the average sample vector is given by the expectation

$$\mu_x = E\{x\} \tag{2.5}$$

and the corresponding covariance matrix is

$$C_x = E\{(x - \mu_x)(x - \mu_x)^T\} \tag{2.6}$$

By definition, the covariance matrix is symmetric and allows obtaining an orthogonal basis by finding its eigenvalues, denoted by e_i, and the corresponding eigenvectors, denoted by λ_i. That is, we have

$$C_x e_i = \lambda_i e_i, \qquad i = 1, ..., n \tag{2.7}$$

Assuming that the eigenvectors are distinct, they can be obtained by the characteristic equation

$$|C_x - \lambda I| = 0 \tag{2.8}$$

where I is the identity matrix of the same order as C_x, and $|.|$ denotes the matrix determinant.

After obtaining the eigenvalues and eigenvectors, it is obtained the matrix A that contains all the eigenvectors, in decreasing order of eigenvalues. Applying the equation

$$y = A(x - \mu_x) \tag{2.9}$$

it is possible to obtain a new point y, in an orthogonal coordinate system defined by the eigenvectors. The components of vector y are the coordinates in such an orthogonal basis. Also, to recover the original data sample x from y, it is possible to use the transformation

$$x = A^T y + \mu_x \tag{2.10}$$

This is possible since the orthogonality of A implies that $A^{-1} = A^T$. Thus, the original vector x was projected on the axis of the new basis and recovered by using a linear combination of this coordinate system.

It is possible to represent the original dataset in terms of just a few eigenvectors (they correspond to some coordinates in the orthogonal basis). It is denoted by A_K, the orthogonal matrix obtained from the first K eigenvectors from the matrix A, then leads to the following transformations

$$y = A_K(x - \mu_x) \tag{2.11}$$

and

$$x = A_K^T y + \mu_x \tag{2.12}$$

These transformations allow us to reduce the dimensionality of the original large representation and to recover the original sample with a minor information loss.

2.2.4 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a supervised technique that aims at reducing the dimensionality of a data set considering different types of indicated classes. According to [2], the criteria used to reduce the data representation in LDA is the maximization of the ratio of the between-class variance and the within-class variance to guarantee the maximum data separability. To obtain this objective, the LDA algorithm calculates two matrices that measure the between and within variances.

The first matrix is called the between-class scatter matrix (S_b) and is defined by

$$S_b = \sum_{i=1}^{c} p_i \times ((\mu_i - \mu_T)(\mu_i - \mu_T)^T) \tag{2.13}$$

where p_i and μ_i are the respectively a priori probability and the average corresponding to class i, The a priori probabilities can be given by the number of elements in each class divided by the total number of elements. The total average, denoted by μ_T, is defined by

$$\mu_T = \sum_{i=1}^{c} p_i \times \mu_i. \tag{2.14}$$

The second matrix is the within-class scatter matrix (S_w) and is defined by

$$S_w = \sum_{i=1}^{c} p_i \times ((x_i - \mu_i)(x_i - \mu_i)^T) \tag{2.15}$$

where x_i is the data matrix of i.

In Equation 2.15, the covariance matrix of each class is multiplied by its corresponding a priori probability, that is given in advance, and the accumulated sum gives the between-class scatter matrix.

Recall the separability criterion of LDA is

$$\max\left(\frac{S_b}{S_w}\right) \tag{2.16}$$

Thus, for a given transformation matrix W, the LDA objective function is defined as

$$J(W) = \frac{|W^t S_b W|}{|W^t S_w W|} \tag{2.17}$$

where the within-class scatter matrix, S_w, and the transformation matrix, W, must obey the restrictions $|S_w| \neq 0$, so that S_w has an inverse matrix, and $W^T W = I$, that validates that W is orthogonal.

The objective function maximum, W_{max}, satisfies

$$\frac{d}{dW}J(W_{max}) = 0 \tag{2.18}$$

from where we get

$$(W_{max}^T S_w W_{max})2S_b W_{max} - (W_{max}^T S_b W_{max})2S_w W_{max} = 0. \tag{2.19}$$

Dividing the last expression by $W_{max}^T S^w W_{max}$

$$2S_b W_{max} - \frac{W_{max}^T S^b W_{max}}{W_{max}^T S^w W_{max}} 2S_w W_{max} = 0 \tag{2.20}$$

and making $\lambda = \frac{W_{max}^T S^b W_{max}}{W_{max}^T S^w W_{max}}$, it follows that

$$\frac{S_b}{S_w} W_{max} = \lambda W_{max} \tag{2.21}$$

This means that the matrix W_{max} that maximizes the ratio of the scatter matrices represents the set of eigenvectors of matrix S_b/S_w.

2.2.5 Partial Least Squares

Partial Least Squares (PLS), developed by Wold et al. [44], is a technique that generalizes and combines the features of PCA and multiple regression [32]. The goal is to predict a variable Y from another variable X, and describe their common structure.

When Y is a vector and X is a complete ordering, it is possible to predict Y through commonly used multiple regression. However, when the number of predictors is large compared to the number of observed variables, X becomes susceptible to singularities, and the regression approach is no longer viable.

The technique consists in finding components of X that are relevant to Y. Specifically, the regression performed by PLS searches for a set of components that forms a simultaneous decomposition of X and Y with the constraint that such components "explain" as best as possible the covariance between X and Y. This is summarized as

$$\begin{aligned} X &= TP^T + E \\ Y &= UQ^T + F \end{aligned} \tag{2.22}$$

Equations (2.22) contain blocks of variables X and Y, where X is an $n \times N$ matrix, Y is an $n \times M$ matrix. Also, T and U are matrices of sizes $N \times p$ and $M \times p$, respectively, that contain p components; P and Q are matrices of sizes $N \times p$ and $M \times p$, respectively, that represent eigenvectors; E and F are residual matrices.

The maximum covariance between vectors w e c is computed as Equations (2.23) and (2.24) to estimate the projection vectors that maximize the covariance of X and Y. Vectors t and u are the obtained components with reduced dimensionality.

$$[\mathrm{cov}(t,u)]^2 = [\mathrm{cov}(Xw, Yc)]^2 = \max_{|r|=|s|=1} [\mathrm{cov}(Xr, Ys)]^2 \tag{2.23}$$

$$t = Xw \qquad u = Yc \tag{2.24}$$

Finally, in each iteration of PLS, the matrices X and Y are recovered according to an obtained projection. In the next iteration, to obtain a new component, it is performed the so called deflation of X and Y, that is, subtracting the components from each data matrix the components that were already explained by previous obtained projections. The deflation process is described by Equation (2.25). The method stops when the norm of some component t shrinks beyond a given threshold, so that the matrix T cannot further explain the data set X. The deflation is more formally stated as

$$X = X - tp^T \quad \text{and} \quad Y = Y - uq^T \tag{2.25}$$

where the matrix T is the concatenation of the obtained vectors t.

The classical form of the PLS algorithm is based on the NIPALS (Nonlinear Interval Partial Least Squares) algorithm [29], that is briefly described as follows:

Algorithm 1 – NIPALS Algorithm

Input: Data matrix (X); label matrix (Y); score vector (u); score vector (t); threshold (thr)

Output: Projection matrix (T), projection matrix (U)

While X and Y are not completely deflated
- *While $|t| > thr$*
  - $w = X^T u / (u^T u)$
  - $||w|| \to 1$
  - $t = Xw$
  - $||c|| \to 1$
  - $u = Yc$
  - $c = Y^T t / (t^T t)$
- *Deflat X and Y matrices*
- *Add $t \to T$*
- *Add $u \to U$*

- **Input**: In order to perform the NIPALS algorithm, there must exist the X data matrix, with its respective Y label matrix; also, the u and t score vectors must be randomly predefined; and a threshold to stop searching for better correlated data must be set.

1. **Line 1**: Perform lines 2 to 9 while matrix X and Y are not completely deflated.

2. **Line 2**: Continue searching for more correlated data while the magnitude of the t score vector is greater than the selected threshold.

3. **Lines 3-5**: Project X onto u to find the corresponding weight vector w. Normalize w to the unit length. Project X onto w to find the new t score vector.

4. **Lines 6-8**: Project Y onto t to find the corresponding weight vector c. Normalize c to the unit length. Project Y onto c to find the new u score vector.

5. **Lines 9**: Deflat X and Y matrices.

6. **Lines 10-11**: Add the recent obtained score vectors, t and u, to the projection matrices T and U.

- **Output**: The projection matrix T will be used to reduce dimensionality from new X matrices.

2.3 SIFT Compressions

In this section, two techniques are presented for optimizing the SIFT description process by reducing the feature vector dimensionality.

2.3.1 PCA-SIFT

PCA-SIFT is a technique, proposed by Ke and Sukthankar [18], that combines the SIFT algorithm and the PCA linear dimensionality reduction technique. This method is based on the construction and evaluation of local feature representations by using the SIFT detected keypoints, however, using a 3042-dimensional feature vector that is then reduced using a projection matrix built with PCA. The algorithm works as follows:

1. a 41×41 window, centered at each keypoint, is extracted at a given scale. This window must satisfy the following properties:

    - it is centered at a local maximum;

  - it is rotated so the gradient dominant orientation will be aligned with the vertical axis;
  - it only contains information of the appropriate scale to its keypoint.

2. the input vector, of size $2 \times 39 \times 39 = 3042$, is normalized to the unit magnitude, in order to minimize the effects of the illumination change;

3. during an off-line stage, the PCA technique is applied to the input vector to reduce its dimensionality.

Their results showed that PCA-SIFT was faster than the original SIFT descriptor due to its lower dimensional size.

2.3.2 KPB-SIFT

Gangqiang et al. [47] proposed a variation of SIFT using kernel projection techniques. The kernel function used is the Walsh-Hadamard, which was chosen due to its good performance at the feature discrimination and computational efficiency.

In order to obtain KPB-SIFT keypoints and descriptions we must perform the following steps:

1. the interest points are detected by the scale invariant feature detection, which consist on three sub-steps:
   (a) select the points that represent a peak (more contrast) at a determined scale-space;
   (b) localize the selected points;
   (c) assign the respective orientation to each point.

2. the orientation gradient patches are built by:
   (a) obtaining the corresponding patches to each detected keypoint;
   (b) rotating each local patch to be vertically aligned to its dominant gradient orientation;
   (c) resizing the patches to 32×32 pixels;
   (d) computing the gradient norms, per each patch, for 4 directions $(0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4})$.

3. the final descriptor is the result of joining the reduced versions of the 4 gradient norms directions onto the Walsh-Hadamard kernel, obtaining:

(a) two groups of 12 dimensions for the vertical and horizontal gradient norms, and

(b) two groups of 6 dimensions for the diagonal gradient norms.

The KPB-SIFT algorithm generates 36-dimensional descriptors that improves the performance in image processing applications in terms of speed and scalability. Furthermore, KPB-SIFT does not require an off-line phase as PCA-SIFT. However, the results have showed that this method is advantageous only for applications where features can be represented by small descriptors and do not require high computational load.

2.4 Bag-of-Features

The Bag-of-Feature (BoF) model is inspired on the Bag-of-Word (BoW) concept [36, 37]. This model enables the creation of an image global feature build over its own local features. It characterizes the distribution of feature vectors present in the image using a histogram, that is, a sparse vector of occurrence counts of local image features. The BoF representation preserves local descriptors individual influence and adds generalization by grouping features from the same nature.

To represent an image through bag-of-features(Figure 2.11), a set of keypoints must initially be detected. After feature detection, various local patches are used to represent the image as numerical vectors. Examples of descriptors that could be used are SIFT [21] and SURF [3], as described above. Another step to represent images using bag-of-features is the codebook (dictionary) generation, where a codeword is considered as a representative of similar patches. A common technique for generating the codebook is to apply the k-means clustering over all the vectors, such that the codewords are defined as the centers of the clusters. The codebook size corresponds to the number of clusters. Then, each patch in the image is mapped to a specific codeword and the image can be represented by the histogram of the codewords.

In order to apply the BoF model, we must perform two separated stages. The first one builds the vocabulary, whereas the second stage builds a BoF representation for a given image. There are several ways to perform these stages, nevertheless the following steps describe the process as it is performed in Chapter 5:

1. Building the vocabulary:

   (a) Local features extraction: a dense sampling scheme is used to obtain points distributed in a dense grid over each image in our training dataset. This dense sampling approach allows to describe equally distributed points in an image, which have the advantage of considering both heterogeneous and homogeneous regions.

   (b) Feature space quantization: since several d-dimensional points where described, the k-means is applied over them in order to obtain n d-dimensional visual words. The visual word quantizers group the different detected points.

2. Obtaining the BoF representation:

   (a) Local feature extraction: for a given image, each point in the dense grid is described.

   (b) Coding: the hard assignment scheme is applied, that is, a point is labeled with id of its nearest visual word in the vocabulary.

   (c) Pooling: builds a histogram with n buckets, representing the frequency, of each of the n visual words appearing in the image.

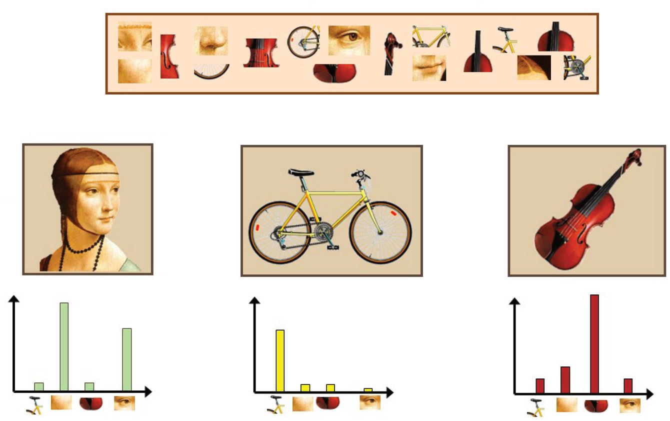


Figure 2.11: Bag-of-feature representation (figure adapted from [10]). The top image represents a vocabulary of visual words. The three following images are associated each one with a histogram. Each histogram has higher frequencies on the visual words related to the nature of the respective image.

Chapter 3

Dimensionality Reduction Through PCA over SIFT and SURF Descriptors

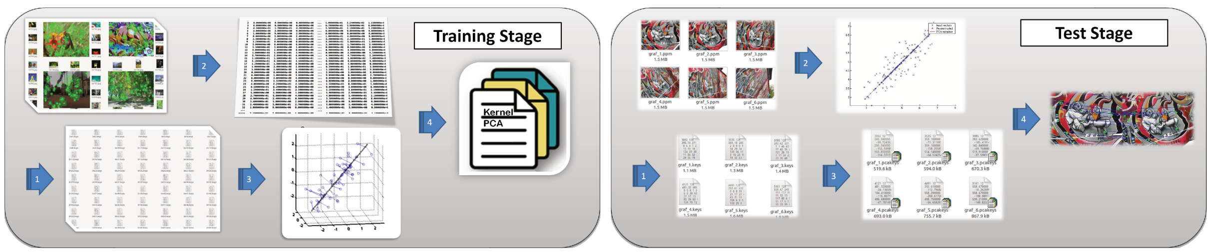
Preamble

One of the constant challenges in image analysis is to improve the process for obtaining distinctive object characteristics. Feature descriptors usually demand high dimensionality to adequately represent the objects of interest. The higher the dimensionality, the greater the consumption of resources such as memory space and computational time. Scale-Invariant Feature Transform (SIFT) and Speeded Up Robust Features (SURF) present algorithms that, besides of detecting interest points accurately, extract well suited feature descriptors. The problem with these feature descriptors is their high dimensionality. There have been several works attempting to confront the curse of dimensionality over some of the developed descriptors. In this work, we apply Principal Component Analysis (PCA) to reduce SIFT and SURF feature vectors in order to perform the task of having an accurate low-dimensional feature vector. We evaluate such low-dimensional feature vectors in a matching application, as well as their distinctiveness in image retrieval. Finally, the required resources in computational time and memory space to process the original descriptors are compared to those resources consumed by the new low-dimensional descriptors.

3.1 Methodology

Our work aims at finding suitable projection matrices to reduce the dimensionality of SIFT and SURF feature vectors by applying the PCA method. Since PCA-SIFT descriptor is used to perform a similar task, its results are also employed in the comparison.

The main steps of the proposed methodology are illustrated in Figure 3.1. Each stage is described in the following sections.



(a) Training Stage (b) Test Stage

Figure 3.1: Training and test stages. (a) in the training stage, interest points are detected and described over images that will not be used in the test stage, then 40000 interest point feature vectors are joined to form the training matrix, and PCA is applied over them in order to get the kernel PCA; (b) in the test stage, interest points are detected and described over the test images, then it projects them over the kernel PCA to obtain the reduced feature vectors, and, finally, it evaluates them.

3.1.1 Training of the Descriptor Eigenspace

A training stage is needed as PCA requires the computation of a covariance matrix and its eigenvectors, what would result in a high computational cost if performed online.

We trained different projection matrices (kernels) for each of the mentioned descriptors, with 40000 feature vectors extracted from sample images that are only used during the training stage.

3.1.2 Generation of Ground-Truth

In order to evaluate matching experiments, we used the Inria Graffiti Dataset [14], which contains a group of images which suffered from different geometric and photometric transformations as rotation and scaling, blurring, warping, illumination variance, and JPEG compression. The first three sets of transformed images have two inner subsets, one of them contains images with distinctive edge boundaries, the other one contains repeated textures of different forms.

Even more important, the Inria Graffiti Dataset contains, for every group of images, different projective transformations, expressed in 3×3 matrices. These matrices allow us to generate the ground-truth by mapping any point from the first image in a group into any other image in the same group.

To validate a match, we have two relevant interest points: p in the first image and q in the second image. We used the transformation matrix provided in the dataset to map p in the second image, obtaining p'. Then, p and q are considered a correct match if p' and q are sufficiently close in space and scale. As mentioned in [18], two points are close in space if the distance between them is less than σ pixels, where σ is the standard deviation to generate the used scale. Two points are close in scale if their scales are within $\sqrt{2}$ of each other.

3.1.3 Descriptor Matching

The descriptor matching process is detailed as follows: given two sets of feature vectors, A and B, with their respective interest point locations, for each feature vector in A, we compute the Euclidean distance, denoted as D_E, to each feature vector in B. Then, for each pair of feature vectors in A and B, if their D_E is smaller than an estimated threshold, we consider to have a match between the respective interest points.

There are different strategies to consider a corresponding interest point. SIFT works better with the nearest neighbor distance ratio strategy (refereed to as NNDR) and PCA-SIFT works better with the nearest neighbor strategy (refereed to as NN). The NN strategy selects the corresponding interest points which present the smallest Euclidean distance under the threshold value. On the other hand, the NNDR strategy considers to have a match when the distance ratio between the two smallest Euclidean distances is under a given threshold. If the mentioned statement is true, then it selects the corresponding interest point with smaller Euclidean distance. Both strategies are being used on the matching process.

3.1.4 Evaluation Metrics

To evaluate the matching performance, we use *recall vs. 1-precision*, as recommended in [1]. Recall (Equation 3.1) measures the ratio between the number of correct matches retrieved over the total of commit matches. As we can achieve a 100% of recall by returning a set with all possible matches, we notice that the recall measure is not enough; therefore, it is also calculated the imprecision (1-precision). The precision (Equation 3.2) measures the ratio between the quantity of correct retrieved matches over the number of retrieved matches, and the imprecision (Equation 3.3) measures the ratio between the number of false retrieved matches over the total number of retrieved matches. So, if we retrieved every possible matches, it would result in a high imprecision.

$$\text{Recall} = \frac{\text{Correct matches retrieved}}{\text{Total of correct matches}} \tag{3.1}$$

$$\text{Precision} = \frac{\text{Correct matches retrieved}}{\text{Total of matches retrieved}} \tag{3.2}$$

$$\text{1-Precision} = \frac{\text{Incorrect matches retrieved}}{\text{Total of matches retrieved}} \tag{3.3}$$

In order to obtain the recall vs. 1-precision curve, the metrics are computed for various thresholds. Starting with a small value where the retrieved matches will be zero or almost zero, and incrementing it the value to a larger threshold where every interest point in the image base (first image of the group) is matched to one in the corresponding image.

3.2 Results

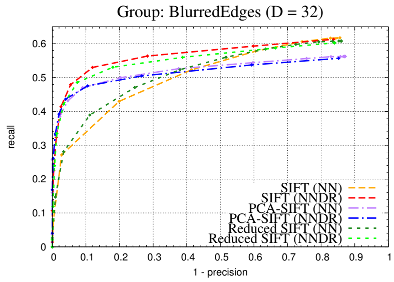
We execute SIFT and SURF algorithms over every group of images contained in the Inria Dataset and evaluate their matching performance. To obtain reduced descriptors, we project the descriptors of the SIFT and SURF interest points onto the trained kernel PCA.

3.2.1 Comparing SIFT, SURF and PCA-SIFT Reduced Dimensionality Descriptors

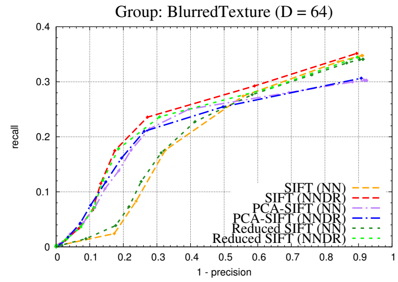
This first experiment compares SIFT, PCA-SIFT (note that the PCA-SIFT descriptor is the one of 3042 dimensions), and the Reduced-SIFT descriptor. We evaluate the mentioned descriptors when their dimensionality is reduced to 12, 20, 32, 36, 46 and 64 dimensions. In the same manner, we evaluate the SURF descriptor and the Reduced-SURF descriptor, this latter also reduced to 12, 20, 32, 36, 46 and 64 dimensions.

The recall vs 1-precision curves of the reduced descriptors, which achieved a performance similar to the original descriptors, are shown in Figures 3.2 and 3.3. Each curve shows one of the transformations contained in the Inria Dataset and indicates the number of dimensions reduced with PCA. The NN and NNDR abbreviations next to each method name identify the strategy used to generate the presented curve.

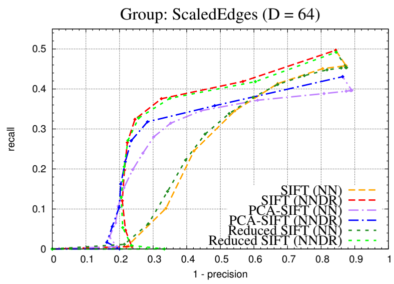
Figures 3.2(a) and (h) show that Reduced-SIFT and the PCA-SIFT descriptors achieving similar responses to the original SIFT descriptor by using only 32 dimensions. Figures 3.2(b-g) show that Reduced-SIFT achieves a similar response to the original SIFT descriptor by using only 36 and 64 dimensions, outperforming again the PCA-SIFT de-



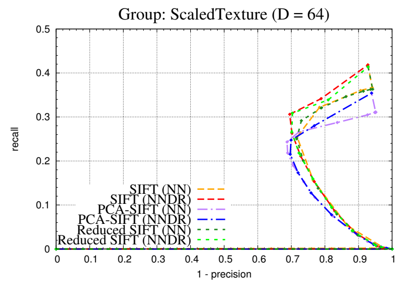
(a) Blurred Edges



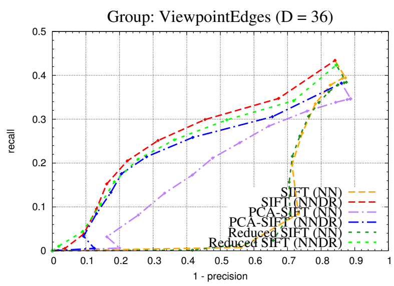
(b) Blurred Texture



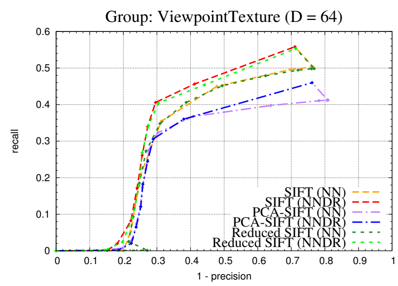
(c) Scaled Edges



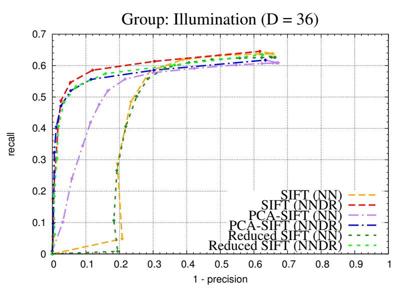
(d) Scaled Texture



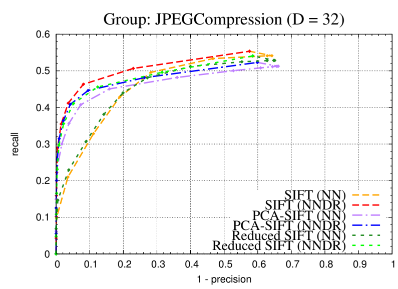
(e) Viewpoint Edges



(f) Viewpoint Texture

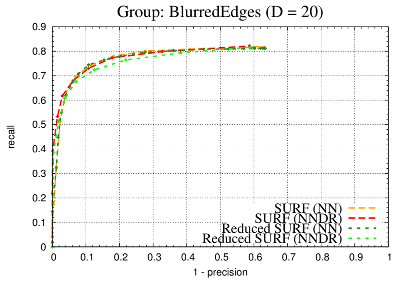


(g) Illumination

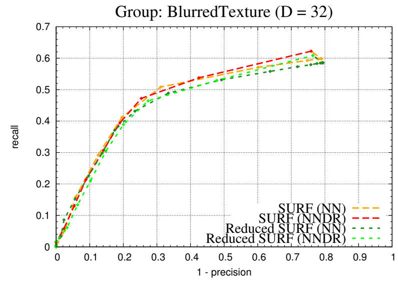


(h) JPEG Compression

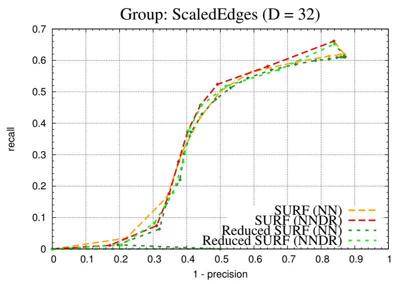
Figure 3.2: SIFT and Reduced-SIFT descriptors matching performance using the Inria Dataset transformations. Each pair of figures (from (a) to (h)) presents the descriptor performance over different groups of images presenting distortions/transformations over distinctive edges or repeated textures.



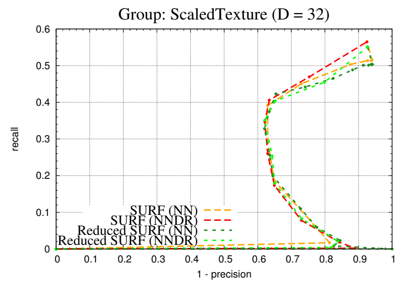
(a) Blurred Edges



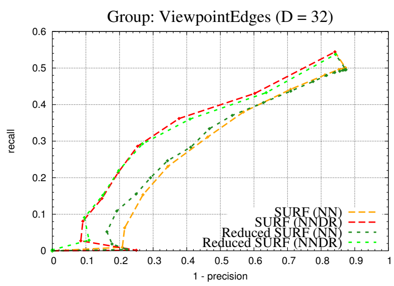
(b) Blurred Texture



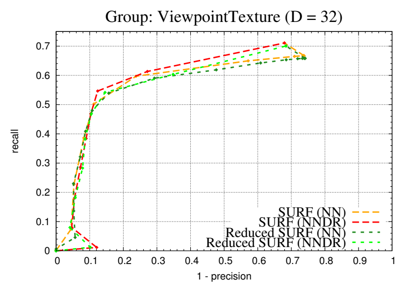
(c) Scaled Edges



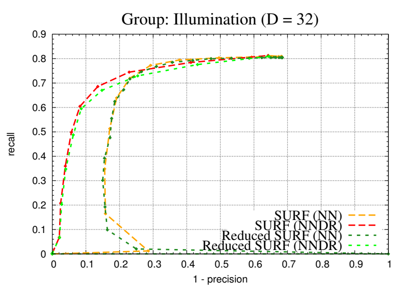
(d) Scaled Texture



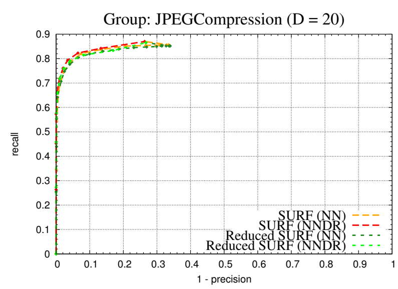
(e) Viewpoint Edges



(f) Viewpoint Texture



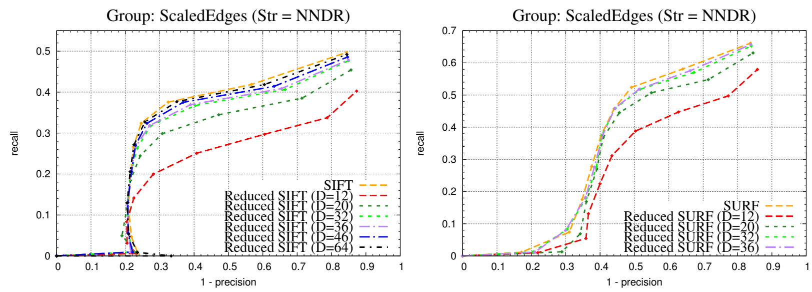
(g) Illumination



(h) JPEG Compression

Figure 3.3: SURF and Reduced-SURF descriptors matching performance using the Inria Dataset transformations. Each pair of figures (from (a) to (h)) presents the descriptor performance over different groups of images presenting distortions/transformations over distinctive edges or repeated textures.

scriptor recall. Meanwhile, Figures 3.3(b-g) show that Reduced-SURF descriptor achieves almost the same result as the SURF descriptor by using 32 dimensions.



(a) SIFT - Dif. dimensions (b) SURF - Dif. dimensions

Figure 3.4: Performance achieved by Reduced-SIFT and Reduced-SURF at different dimensions.

Figure 3.4 show the matching performance over different dimensions. On the right of the figure is the Reduced-SIFT descriptor, on the left is the Reduced-SURF descriptor. This reduced descriptors achieve good performances using 32 dimensions. Then, it is recommended to use 32 dimensions if we desire to maintain almost the same performance as the original descriptors.

3.2.2 Image Retrieval Application

We evaluated an image retrieval application using the dataset provided by Ke and Sukthankar [18]. This dataset consists of thirty images divided into 10 groups of three images each, so each image has two corresponding images. For each image, we perform the matching with all the others and obtain a ranking of the three images with best correspondence.

Each ranking is scored according to the number of corresponding images it contains: two points if both corresponding images appear in the ranking, one point if only one corresponding image appears; and zero points, otherwise. This means that we will have a maximum of 60 points if, for every image, the two corresponding images are returned in the ranking.

We evaluated the percentage of image retrieval for the SIFT, PCA-SIFT and Reduced-SIFT descriptors using different number of dimensions. We applied both matching strategies and used the threshold that obtained the best results.

Results are shown in Tables 3.1 to 3.4 reporting the descriptor, number of dimensions, threshold used to obtain the image retrieval value, percentage of the threshold (where the maximum threshold is represented by 100%, which also means a high imprecision), and the percentage of image retrieval calculated as the sum of all obtained scores divided by 60 (the maximum score to obtain).

Tables 3.1 and 3.2 compare SIFT, PCA-SIFT and Reduced-SIFT descriptors. The first table uses the nearest neighbor (NN) strategy and shows that PCA-SIFT descriptor outperforms the SIFT descriptor by using 32 dimensions, while the Reduced-SIFT descriptor provides a result equal or better than the SIFT descriptor by using 20, 32 and 36 dimensions. The second table uses the nearest neighbor distance ratio (NNDR) strategy and shows that PCA-SIFT descriptor achieves a better performance than the Reduced-SIFT descriptor. It is important to notice that the PCA-SIFT and Reduced-SIFT descriptors achieve a better result than the SIFT descriptor with a lower threshold percentage in the majority of the cases. In some cases, as shown in table 3.1 for reduced SIFT to 32 and 36 dimensions, reducing dimensions can even achieve better results, we assume this occurs because of some noise that was suppressed by using less dimensions.

Table 3.1: Image retrieval performed with SIFT, PCA-SIFT and Reduced-SIFT (Nearest Neighbor Strategy)

Descriptor	**Dimensions**	**Threshold**	**Percentage**	**Retrieval**
SIFT	128	250	45%	68%
PCA-SIFT	12	1500	13%	48%
PCA-SIFT	20	2500	17%	58%
PCA-SIFT	32	3500	24%	70%
PCA-SIFT	36	4000	25%	67%
Reduced-SIFT	12	75	23%	57%
Reduced-SIFT	20	125	33%	68%
Reduced-SIFT	32	150	33%	70%
Reduced-SIFT	36	150	33%	70%

Table 3.2: Image retrieval performed with SIFT, PCA-SIFT and Reduced-SIFT (Nearest Neighbor Distance Ratio Strategy)

Descriptor	**Dimensions**	**Threshold**	**Percentage**	**Retrieval**
SIFT	128	0.80	80%	65%
PCA-SIFT	12	0.60	60%	57%
PCA-SIFT	20	0.80	80%	69%
PCA-SIFT	32	0.80	80%	75%
PCA-SIFT	36	0.80	80%	77%
Reduced-SIFT	12	0.80	80%	57%
Reduced-SIFT	20	0.90	90%	67%
Reduced-SIFT	32	0.70	70%	67%
Reduced-SIFT	36	0.80	80%	70%

Tables 3.3 and 3.4 show a comparison between SURF and Reduced-SURF descriptors. The first one uses the NN strategy and shows Reduced-SURF descriptor (20-dimensional

feature vector) with a threshold percentage of 40% achieving similar response to the SURF descriptor with 70% image retrieval for a threshold percentage of 41%. The second one uses the NNDR strategy, where the Reduced-SURF descriptor (36-dimensional feature vector) achieves similar response to the SURF descriptor with 77%. Both responses were achieved for a threshold percentage of 80%. As all the descriptions are in different spaces, it is not possible to compare the threshold values. For this reason, we show a percentage coming from a normalized value using the 0 as the minimum threshold and x as the maximum, where x represents the smallest threshold capable to retrieve every point.

Table 3.3: Image retrieval performed with SURF and Reduced-SURF (Nearest Neighbor Strategy)

Descriptor	**Dimensions**	**Threshold**	**Percentage**	**Retrieval**
SURF	64	0.35	41%	70%
Reduced-SURF	12	0.20	31%	65%
Reduced-SURF	20	0.30	40%	68%
Reduced-SURF	32	0.30	38%	65%
Reduced-SURF	36	0.40	50%	67%

Table 3.4: Image retrieval performed with SURF and Reduced-SURF (Nearest Neighbor Distance Ratio Strategy)

Descriptor	**Dimensions**	**Threshold**	**Percentage**	**Retrieval**
SURF	64	0.80	80%	78%
Reduced-SURF	12	0.70	70%	58%
Reduced-SURF	20	0.80	80%	63%
Reduced-SURF	32	0.80	80%	73%
Reduced-SURF	36	0.80	80%	77%

3.2.3 Comparing Computational Time and Memory Space

In order to perform this experiment, we used a computer with an Intel Core i7-2670QM CPU with 2.20GHz and 8 Gbytes of RAM.

Table 3.5 was obtained from running the description process over all possible images in an interval of time of about ten minutes. As SURF performed faster, it achieved the higher quantity of described keypoints, which is represented by the value of 100%. The Reduced-SURF could not perform better since it executes the SURF algorithm and then projects the kernel PCA; even though, Reduced-SURF achieved among 90% of the keypoints described by SURF. As PCA-SIFT executes all the SIFT algorithm (detection and description) in our experiments, it was slower than it could be.

Table 3.5: Interest point description process for about ten minutes

Descriptor	Dimensions	Files	Keypoints
SURF	64	3560	1168200
SIFT	128	972	760668
PCA-SIFT	36	463	348724
Reduced-SURF	12	3285	1077466
Reduced-SURF	20	3285	1077479
Reduced-SURF	32	3230	1061001
Reduced-SURF	36	3238	1063347

Table 3.6 was also obtained from running the description and the matching process over all possible images in an interval of time of about ten minutes. At this time, the Reduced-SURF performed better because of its low-dimensional vectors. Results for SIFT and PCA-SIFT descriptors are not reported due to their slow execution time.

Table 3.6: Interest point description and matching processed for about ten minutes

Descriptor	Dimensions	Files	Keypoints
Reduced-SURF	12	3075.6	1012460.5
Reduced-SURF	20	3022.5	994357.6
Reduced-SURF	32	2958.5	974094.2
Reduced-SURF	36	2898.2	954022.2
SURF	64	2870.1	944186.3

Table 3.7 evaluates the matching time spent for each feature vector size. It is also interesting to note the gain of space used by the reduced descriptor. Experiments were executed over 10000 images with approximately 3 million interest points.

Table 3.7: Running time for interest point matching

Descriptor	Dimensions	Time	Space
Reduced-SURF	12	134.0s	532 MB
Reduced-SURF	20	158.7s	778 MB
Reduced-SURF	32	185.8s	1.1 GB
Reduced-SURF	36	201.3s	1.3 GB
SURF	64	247.6s	2.2 GB

3.3 Discussion

This work demonstrated that the Reduced-SIFT and Reduced-SURF can be applied as their low dimensional feature vectors still present similar behavior as the original descriptors.

As reducing the feature vectors involves an extra computational time, it is needed to maintain a performance similar to the original descriptor. The reduction should be moderated so that at least 20 or 32-dimensional feature vectors could be used.

The gain of the reduced descriptors is better manifested when the detected interest points are not discarded fast, for instance, in video tracking application. However, when the descriptors are used repeatedly, as in image retrieval applications, the matching would be the dominant process, which represents an advantage to our reduced descriptor.

Chapter 4

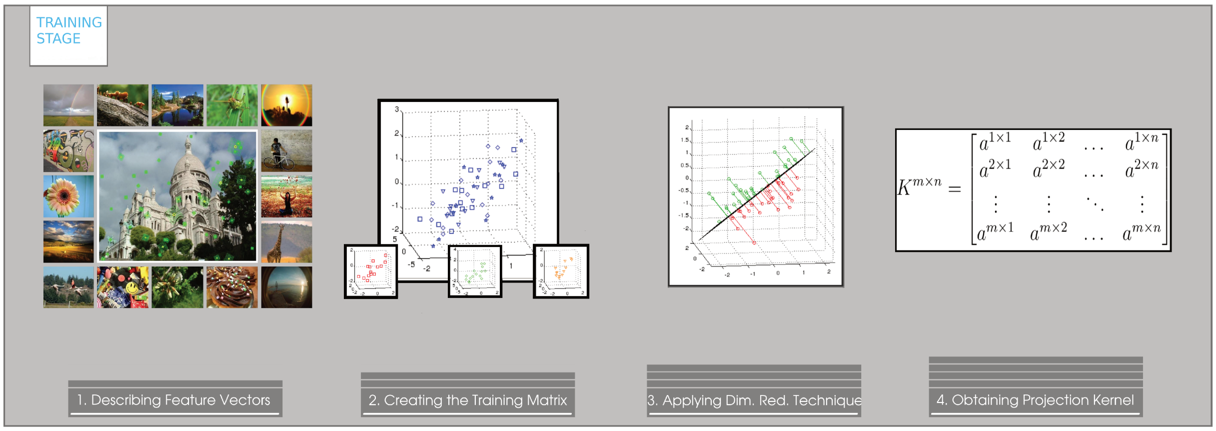
Linear Dimensionality Reduction Applied to SIFT and SURF Feature Descriptors

Preamble

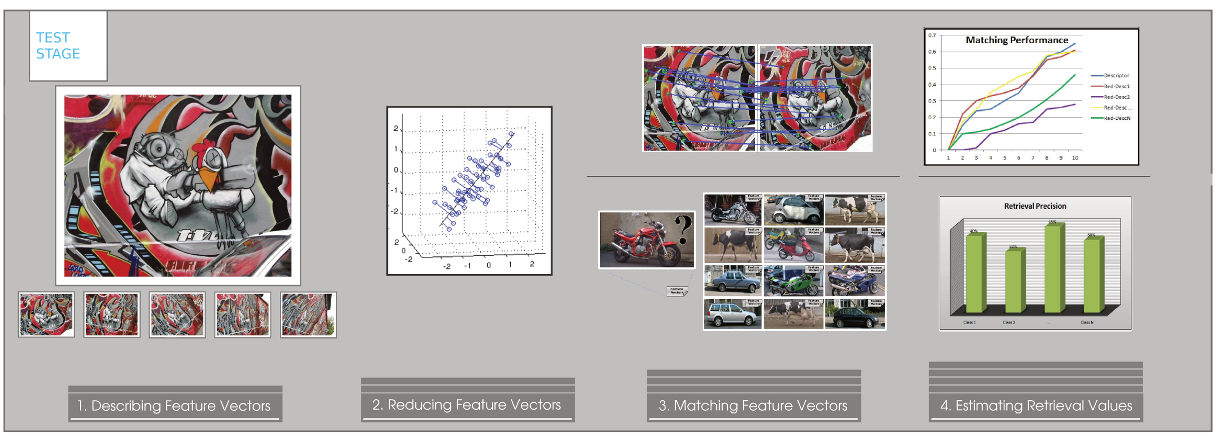
Robust local descriptors usually consist of high dimensional feature vectors to describe distinctive characteristics of images. The high dimensionality of a feature vector incurs into considerable costs in terms of computational time and storage. It also results in the curse of dimensionality, which affects the performance of several tasks that use such feature vectors, such as matching, retrieval and classification of images. To address these problems, it is possible to employ some dimensionality reduction techniques, leading frequently to information loss and, consequently, accuracy reduction. This work aims at applying linear dimensionality reduction to the SIFT and SURF descriptors. The objective is to demonstrate that even risking to decrease the accuracy of the feature vectors, it results in a satisfactory trade-off between computational time and storage requirements. We perform linear dimensionality reduction through Random Projections (RP), Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA) and Partial Least Squares (PLS) in order to create lower dimensional feature vectors. These new reduced descriptors lead us to less computational time and memory storage requirements, even improving accuracy in some cases. We evaluate such reduced feature vectors in a matching application, as well as their distinctiveness in image retrieval. Finally, we assess the computational time and storage requirements by comparing the original and the reduced feature vectors.

4.1 Evaluation Methodology

This work aims at reducing the dimensionality of SIFT and SURF feature vectors by applying the dimensionality reduction techniques mentioned in the previous section. To accomplish that, the main steps of the proposed methodology are illustrated in Figure 4.1. Each stage is described in the following sections.



(a) Training Stage



(b) Test Stage

Figure 4.1: Training and test stages. (a) during the training stage, feature vectors are described over images that will not be used in the test stage; the obtained feature vectors are joined into a single training matrix, then a dimensionality reduction technique is applied over this training matrix in order to obtain the projection matrix; (b) the test stage detects and describes interest points over the test images, then it projects them over a projection matrix to obtain the reduced feature vectors, subsequently the matching task is computed between one image and a set of selected images and, finally, a comparison between the matching task and a ground truth is evaluated and shown.

4.1.1 Learning the Projection Matrices

An off-line computation is performed in this stage. This is due to the expensive computational costs demanded by the linear reduction techniques to calculate a projection matrix. Furthermore, there is no need to re-compute the projection matrices as the considered applications area performed.

In order to learn a projection matrix, the dimensionality reduction technique receives a set of feature vectors grouped into a matrix, referred to as training matrix. This training matrix is denoted as $X^{n \times d}$, where n represents the number of feature vectors and d represents the dimension of each feature vector. After applying the dimensionality reduction, a projection matrix is obtained, denoted as $K^{d \times p}$ (with $p < d$). Then, it is possible to reduce a m-dimensional feature vector to at most n dimensions. To reduce feature vectors to lower dimensions, it is unnecessary to recompute the projection matrix, only performing a projection of its descriptors to the projection matrix as

$$T^{l \times p} = M^{l \times d} \times K^{d \times p} \tag{4.1}$$

resulting in the low dimension matrix T.

Projection Matrix in the Matching Task

A set of randomly selected images, collected from the Mirflickr-1M Dataset [27], was used to extract $40,000$ feature vectors to compose the training matrix for PCA.

The training matrix for LDA and PLS techniques differs from the PCA matrix in the fact that every feature vector has one more dimension, assuming a value of 1 if it corresponds to an interest point that was detected within the default threshold or 0, otherwise. These arbitrarily selected values allow PLS and LDA techniques to distinguish the difference between both types of interest point descriptor by giving different weights to their related projections.

Projection Matrix in the Image Retrieval Task

The feature vectors of the first seventy-five images from each class were computed to compound the training matrix for PCA. For the LDA and PLS techniques, it is needed to add the identifier of the corresponding class to each feature vector in the training matrix.

4.1.2 Feature Vector Matching

In order to perform the matching process, two sets of feature vectors, A and B, are necessary with their respective interest point locations. The Euclidean distance, denoted as D_E, is computed from each feature vector in A to each feature vector in B. Then, for each pair of feature vectors in A and B, if their D_E is smaller than an estimated threshold, we consider to have a match between the respective interest points.

In this work, we employ two different approaches to evaluating a corresponding interest point: the nearest neighbor (NN) strategy and the the nearest neighbor distance ratio (NNDR) strategy. The NN strategy selects the corresponding interest points which present the smallest Euclidean distance under the set threshold. On the other hand, the NNDR strategy considers to have a match when the distance ratio between the two smallest Euclidean distances is under the set threshold. If the mentioned statement is true, then it selects the corresponding interest point with smaller Euclidean distance.

The feature vector matching stage is applied in both matching and image retrieval tasks.

4.2 Experiments and Results

Experiments conducted on two tasks, matching and image retrieval, were proposed to validate the following statement (the data set considered and the evaluation metrics are described in Sections 4.2.1 and 4.2.2). SIFT and SURF feature vectors, when reduced to lower dimensions, can maintain or even improve a similar accuracy as they would achieve in the original space (results shown in Sections 4.2.3 and 4.2.4). An additional experiment to measure the computational time and storage usage was performed to compare the consumption of these resources when using the original and the reduced feature vectors (results shown in Section 4.2.5).

4.2.1 Datasets and Ground Truth

Two datasets were selected to perform matching and image retrieval experiments.

Matching Task For matching experiments, we used the Inria Graffiti Dataset [14]. This dataset contains 8 groups of images (6 images per group). Each group of images is subject to different geometric and photometric transformations such as rotation, scaling, blurring, warping, illumination variance, and JPEG compression. The first three sets of transformed images have two inner subsets, one of them contains images with distinctive edge boundaries, the other one contains repeated textures of different forms.

Every group of images, in the Inria Graffiti Dataset, contains five 3×3 homography matrices. Each of these homography matrices represents a projective transformation from the first image to one of the other five images, which allows us to map any point from the first image to any other image belonging to its group. Therefore, to validate a match, it is necessary to have two interest points: p in the first image and q in any other image, denoted by i, belonging to the same group. The homography matrix related to the image 1 (base image) to the image i allows to map p into i, obtaining p'. Then, p and q are considered a correct match if p' and q are sufficiently close in space and scale. As mentioned in [18], two points are close in space if the distance between them is less than σ pixels, where σ is the standard deviation to generate the used scale. Two points are close in scale if their scales are within $\sqrt{2}$ of each other.

Image Retrieval Task For image retrieval experiments, we employed the TU Darmstadt Dataset [20]. This dataset contains 300 images divided into three categories: cars, motorbikes and cows, where each one contains 100 images.

4.2.2 Evaluation Metrics

This subsection presents the metrics employed to measure results, matching and image retrieval experiments, to better understand the results obtained through the experiments.

Matching Task To evaluate the matching performance, we use *recall vs. 1-precision* curves, as recommended in [1]. Recall (Equation 4.2) represents the measure of the ratio between the number of correct matches retrieved over the total of matches that are expected to be retrieved. It is important to notice that a recall of 100% would be achieved if a set with all possible matches is returned, so if the recall measure is presented alone, it loses its relevance. Therefore, since the precision measure means the ratio between the number of correct matches retrieved over the total of matches retrieved, the 1-precision value is also considered to know when the recall represents some good result. These two measures are defined as

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{4.2}$$

$$\text{1-Precision} = \frac{\text{False Positives}}{\text{True Positives} + \text{False Positives}} \tag{4.3}$$

Image Retrieval Task As there is still a remaining of 25 images per class from the TU dataset and saved to be used in the testing stage, we mixed them into the set and compute for each image a rank list of the top corresponding 24 images. The ranking list can also be performed for different threshold values. Finally, from all these lists we found the best threshold to perform our experiments.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{4.4}$$

4.2.3 Image Matching Task

In this experiment, SIFT and SURF feature vectors are compared to their respective reduced feature vectors by using the recall vs 1-precision curves. The original feature vectors were projected onto the projection matrices built with RP, PCA, LDA and PLS methods. It is important to notice that the sequence composed of dimension reduction with PCA and feature extraction with SIFT, tested in this work (referred to as PCA+SIFT), is not the same as the approach called PCA-SIFT proposed in [18].

The recall vs 1-precision comparison curves are shown in Figures 4.2, 4.3, 4.4 and 4.5. Figures 4.2 and 4.4 show plots for every transformation, represented in the Inria Dataset, using the NN strategy, whereas Figures 4.3 and 4.5 show plots for every transformation using the NNDR strategy. In addition, each plot presents the lower dimensional feature vector that achieved an accuracy similar to its respective original feature vector. SIFT feature vectors were reduced to 12, 20, 32, 36, 46 and 64 dimensions, whereas SURF feature vectors were reduced to 12, 20, 32 and 36 dimensions.

Figures 4.2 and 4.3 show that PCA+SIFT achieved similar results to the original SIFT descriptor using only 32, 36 and 64 dimensions. This means that the PCA+SIFT descriptor can achieve a high accuracy even when the feature vector is reduced to 25 to 50% of its original size.

Figures 4.4 and 4.5 also show that PCA+SURF achieved similar responses to the original SURF descriptor using 20 and 32 dimensions. This means that the PCA+SURF descriptor can achieve a high accuracy even when the feature vector is reduced to 31.25 to 50% of its original size.

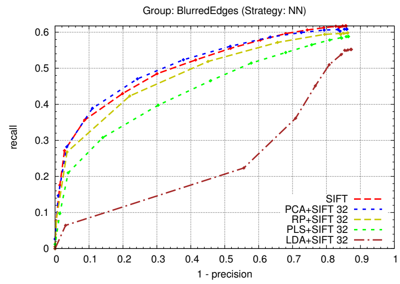
For both descriptors, the PCA method performed better. Figures 4.6 and 4.7 compare SIFT and SURF performance to their corresponding PCA+SIFT and PCA+SURF using different dimensions. It is shown that PCA+SIFT achieves a similar accuracy to SIFT at 32 dimensions, as well as PCA+SURF achieves a similar accuracy at 20 dimensions.

4.2.4 Image Retrieval Task

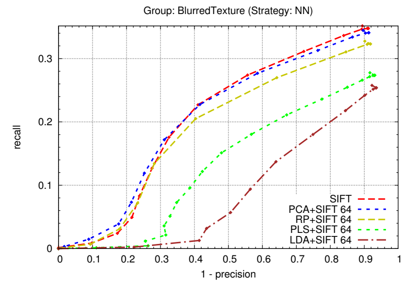
Image retrieval experiments were conduced to demonstrate that the reduced feature vectors can perform as well or even better than the original feature vectors.

This experiment differs from the one presented in [41] because of the nature of the data set used. The data set used in [41] considered 30 images separated into 10 groups, where each group contained the same object viewed from a different angle. On the other hand, the TU Darmstadt data set contains 300 images separated into 3 groups, and normally each group presents different objects. Such difference leads to a lower retrieval precision.

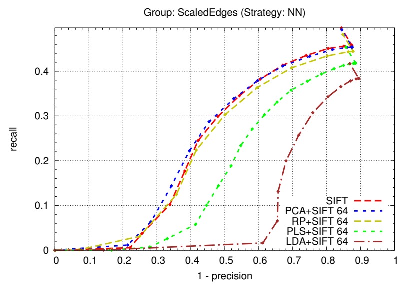
The SIFT and SURF feature vectors were reduced to a set of 2, 4, 8, 12, 20, 32, 36, 46, 64 dimensions and 2, 4, 8, 12, 20, 32, 36 dimensions, respectively.



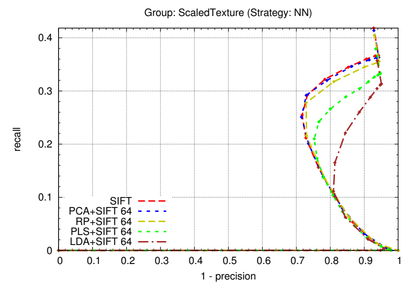
(a) Blurred Edges - NN



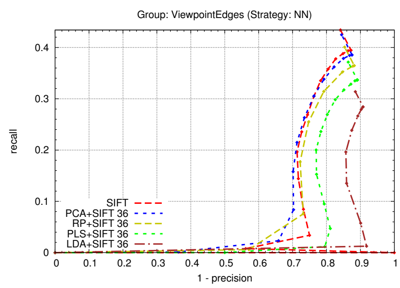
(b) Blurred Texture - NN



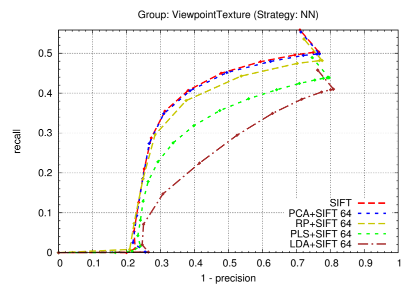
(c) Scaled Edges - NN



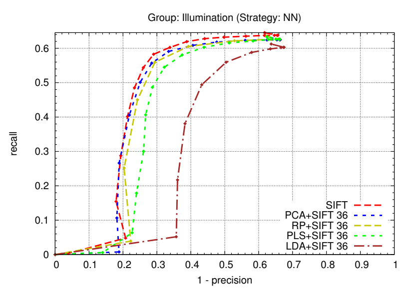
(d) Scaled Texture - NN



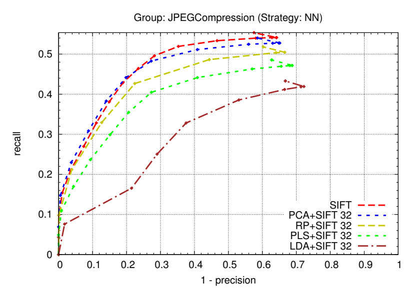
(e) Viewpoint Edges - NN



(f) Viewpoint Texture - NN



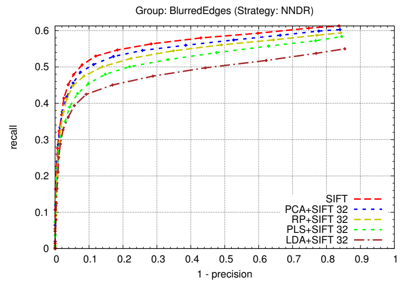
(g) Illumination - NN



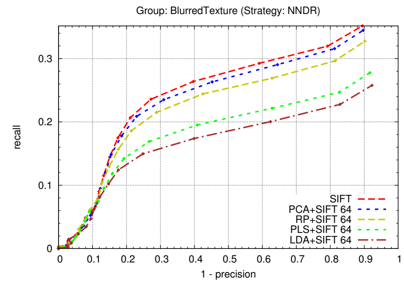
(h) JPEG Compression - NN

Figure 4.2: SIFT Feature Vectors and corresponding Reduced Feature Vector comparison using the NN strategy

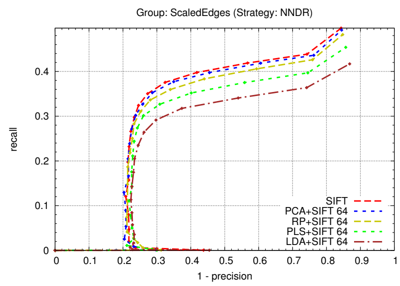
Each table presented in this subsection contains the following fields: descriptor, which refers to the original descriptor or a reduced descriptor; dimensions indicating the number



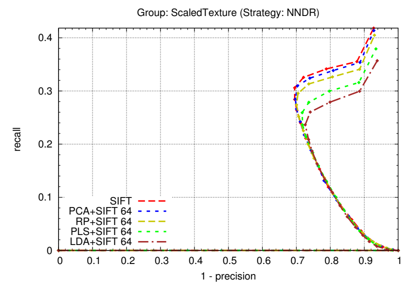
(a) Blurred Edges - NNDR



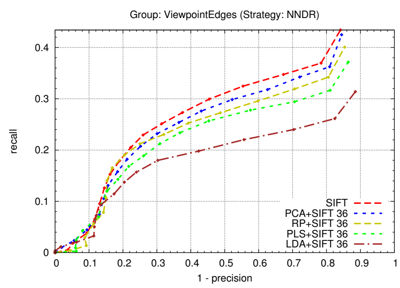
(b) Blurred Texture - NNDR



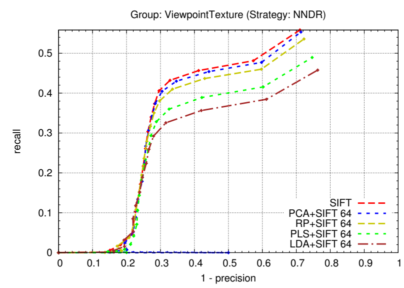
(c) Scaled Edges - NNDR



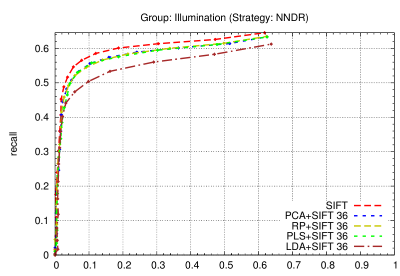
(d) Scaled Texture - NNDR



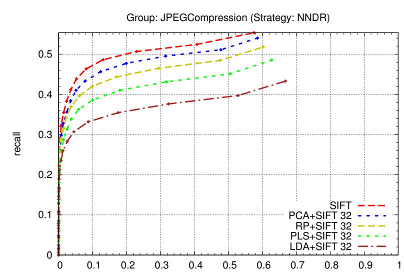
(e) Viewpoint Edges - NNDR



(f) Viewpoint Texture - NNDR



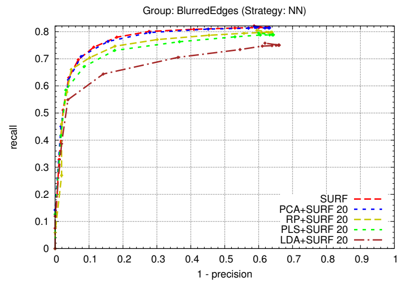
(g) Illumination - NNDR



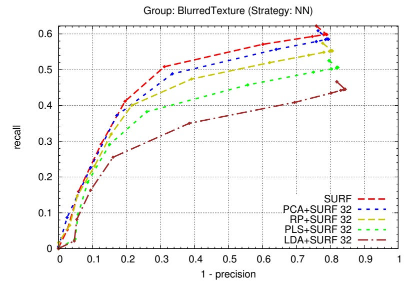
(h) JPEG Compression - NNDR

Figure 4.3: SIFT Feature Vectors and corresponding Reduced Feature Vector comparison using the NNDR strategy

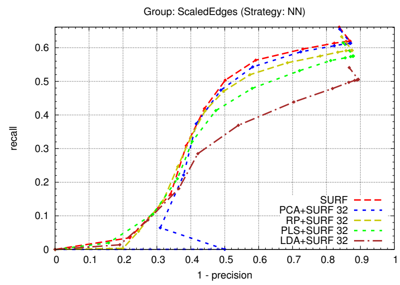
of dimensions used by each descriptor; threshold, which indicates the value where the best retrieval was obtained; percentage, which indicates the position in the interval where the



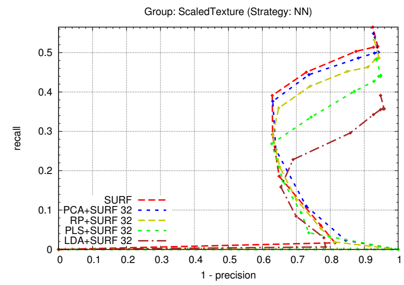
(a) Blurred Edges - NN



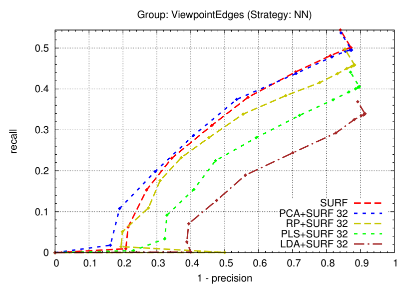
(b) Blurred Texture - NN



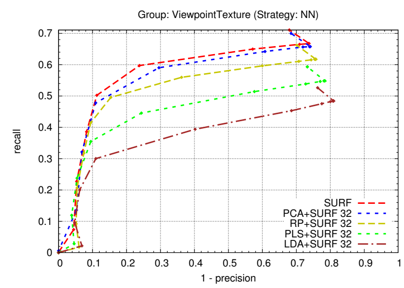
(c) Scaled Edges - NN



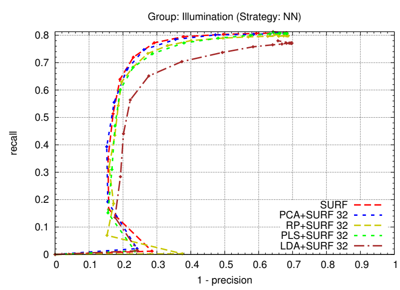
(d) Scaled Texture - NN



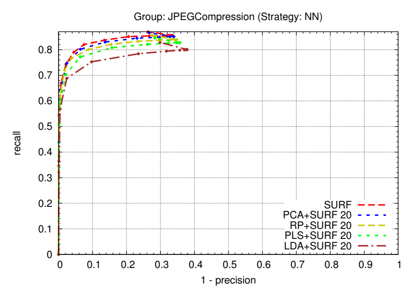
(e) Viewpoint Edges - NN



(f) Viewpoint Texture - NN



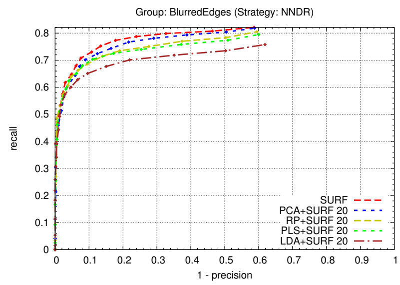
(g) Illumination - NN



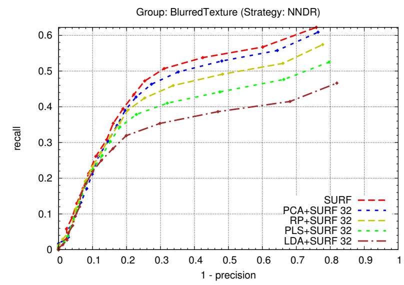
(h) JPEG Compression - NN

Figure 4.4: SURF Feature Vectors and corresponding Reduced Feature Vector comparison using the NN strategy

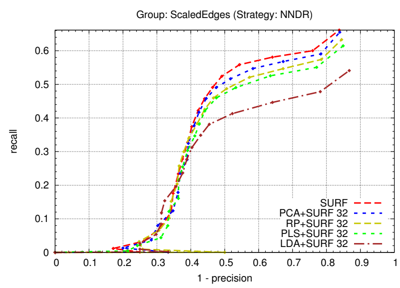
current threshold is suitable; the retrieval indicating the percentage of correct images retrieved over the total of images retrieved.



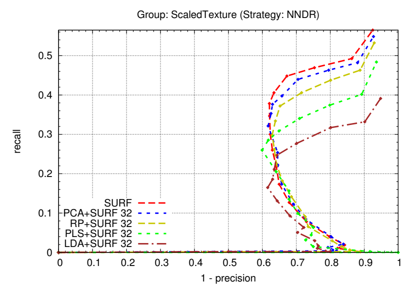
(a) Blurred Edges - NNDR



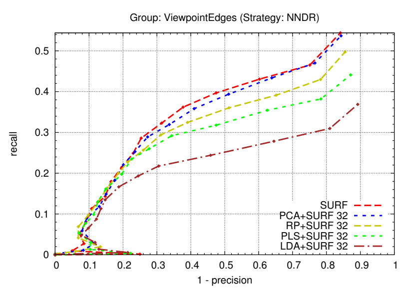
(b) Blurred Texture - NNDR



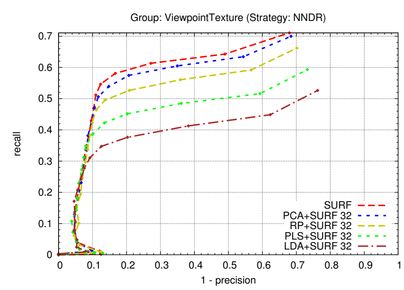
(c) Scaled Edges - NNDR



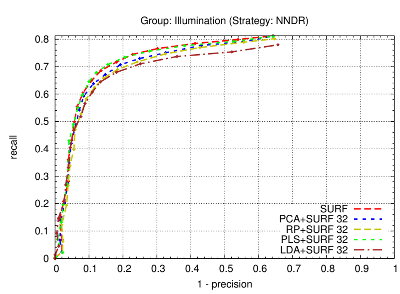
(d) Scaled Texture - NNDR



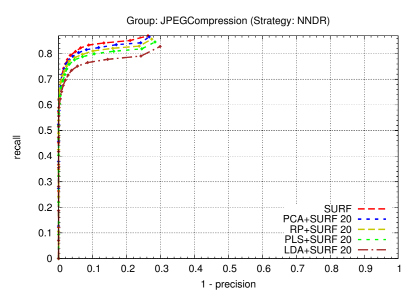
(e) Viewpoint Edges - NNDR



(f) Viewpoint Texture - NNDR



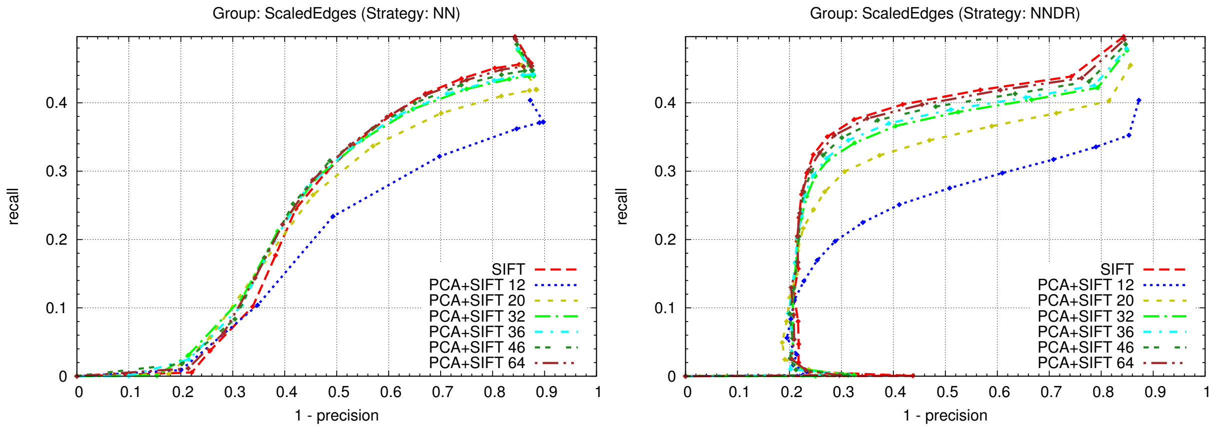
(g) Illumination - NNDR



(h) JPEG Compression - NNDR

Figure 4.5: SURF Feature Vectors and corresponding Reduced Feature Vector comparison using the NNDR strategy

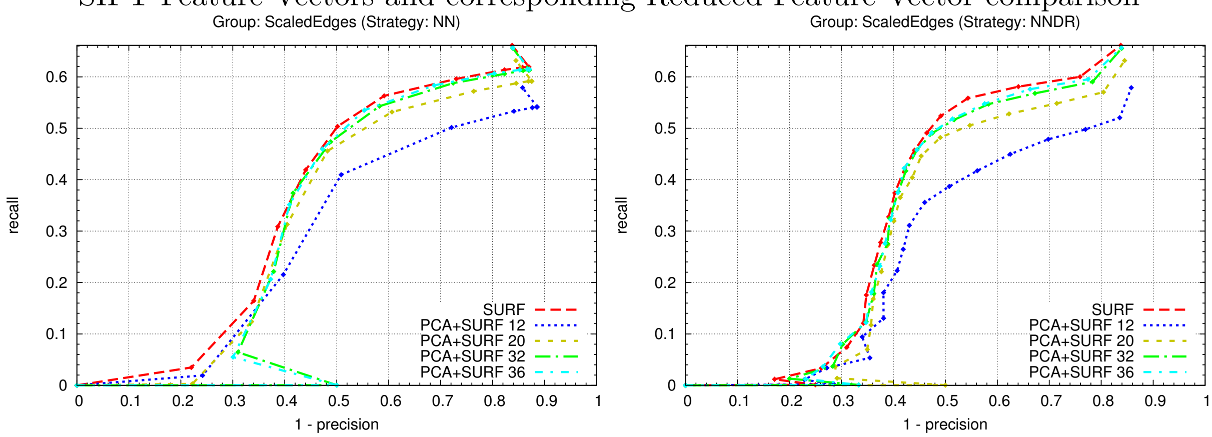
It is important to observe that the percentage reveals the tolerance of the current threshold. A high percentage means a high threshold tolerance, which leads to a higher



(a) Different dimensions - NN (b) Different dimensions - NNDR

Figure 4.6: Performance achieved by PCA+SIFT with both strategies at different dimensions

SIFT Feature Vectors and corresponding Reduced Feature Vector comparison



(a) Different dimensions - NN (b) Different dimensions - NNDR

Figure 4.7: Performance achieved by PCA+SURF with both strategies at different dimensions

imprecision.

Figure 4.8 presents a comparison between the retrieval results computed for SIFT feature vectors and every reduced feature vector computed. It can be seen that the features reduced by using the PCA projection matrix outperformed all others. Table 4.1 focuses on the 32-dimensional feature vector results since it achieved a retrieval value close to the achieved by the original descriptor. Furthermore, as PCA technique demonstrated to perform better, Table 4.2 presents in detail the retrieval values achieved by the PCA+SIFT descriptor with different feature dimensions. It can be seen that, between 12 and 36 di-

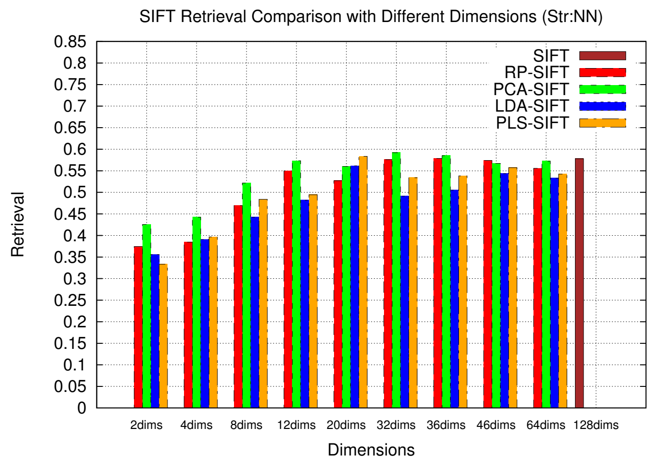


Figure 4.8: SIFT - Retrieval Comparisons (Str: NN).

Table 4.1: Reduced-SIFT - Retrieval Comparisons (Str: NN).

Descriptor	**Dims.**	**Thr.**	**Perc.**	**Retrieval**
SIFT	128	100.00	16.67%	57.81%
RP+SIFT	32	50.00	16.67%	57.60%
PCA+SIFT	32	75.00	15.00%	59.20%
LDA+SIFT	32	4.00	33.33%	49.12%
PLS+SIFT	32	0.01	33.33%	53.39%

Table 4.2: Retrieval Comparison between Dimensions (Str: NN).

Descriptor	**Dims.**	**Thr.**	**Perc.**	**Retrieval**
SIFT	128	100.0	16.67%	57.18%
PCA+SIFT	2	50.0	25.00%	42.56%
PCA+SIFT	4	75.0	25.00%	44.21%
PCA+SIFT	8	25.0	7.14%	52.16%
PCA+SIFT	12	50.0	13.33%	57.33%
PCA+SIFT	20	50.0	11.11%	56.00%
PCA+SIFT	32	75.0	15.00%	59.20%
PCA+SIFT	36	75.0	15.00%	58.51%
PCA+SIFT	46	75.0	14.29%	56.69%
PCA+SIFT	64	100.0	18.18%	57.23%

mensions, the retrieval response is usually higher and more precise than considering the original descriptor.

Figure 4.9 shows a comparison between the retrieval results achieved with the NNDR strategy for SIFT feature vectors and every reduced feature vector. In this case, the features projected onto the LDA projection matrix and onto PLS projection matrix outperformed the other reduced features. We believe this is due to the classification ability.

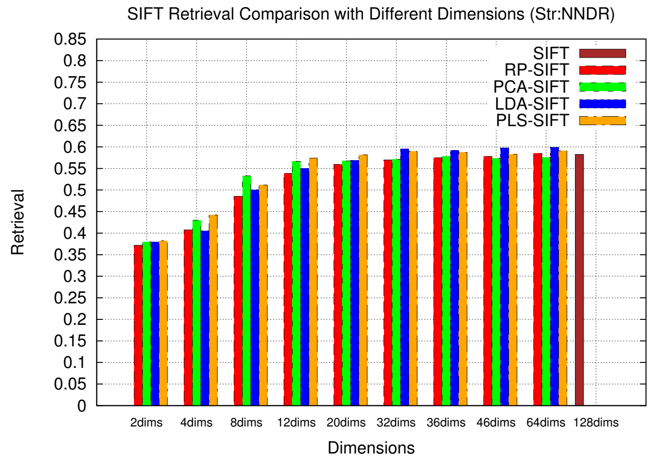


Figure 4.9: SIFT - Retrieval Comparisons (Str: NNDR).

Table 4.3: Reduced-SIFT - Retrieval Comparisons (Str: NNDR).

Descriptor	**Dims.**	**Thr.**	**Perc.**	**Retrieval**
SIFT	128	0.525	52.50%	58.24%
RP+SIFT	36	0.525	52.50%	57.49%
PCA+SIFT	36	0.475	47.50%	57.76%
LDA+SIFT	36	0.600	60.00%	59.20%
PLS+SIFT	36	0.525	52.50%	58.72%

Table 4.4: Retrieval Comparison between Dimensions (Str: NNDR).

Descriptor	**Dims.**	**Thr.**	**Perc.**	**Retrieval**
SIFT	128	0.525	52.50%	58.24%
LDA+SIFT	2	0.850	85.00%	37.92%
LDA+SIFT	4	0.700	70.00%	40.53%
LDA+SIFT	8	0.325	32.50%	50.08%
LDA+SIFT	12	0.425	42.50%	54.99%
LDA+SIFT	20	0.550	55.00%	56.85%
LDA+SIFT	32	0.600	60.00%	59.52%
LDA+SIFT	36	0.600	60.00%	59.20%
LDA+SIFT	46	0.600	60.00%	59.73%
LDA+SIFT	64	0.650	65.00%	59.89%

Table 4.3 focuses on the 36 dimensional feature vector achieving better retrieval than the original feature vector. Table 4.4 presents the retrieval values achieved by the LDA+SIFT descriptor for different dimensions. It can be seen that, between 32 and 64 dimensions, the retrieval response is higher than the original descriptor, however, the precision is lower.

Figure 4.10 shows the NN strategy obtaining better results than the NNDR strategy. In this experiment, the PCA reduced feature vectors achieved a better performance, being reduced to 18.75% of the original size. Table 4.5 shows that the other reduced descriptors achieved close results to the original feature vector for 12 dimensions. Table 4.6 presents the retrieval values achieved by the PCA+SURF descriptor for different dimensions. It can be seen that, between 12 and 36 dimensions, the retrieval and precision responses are higher than considering the original descriptor.

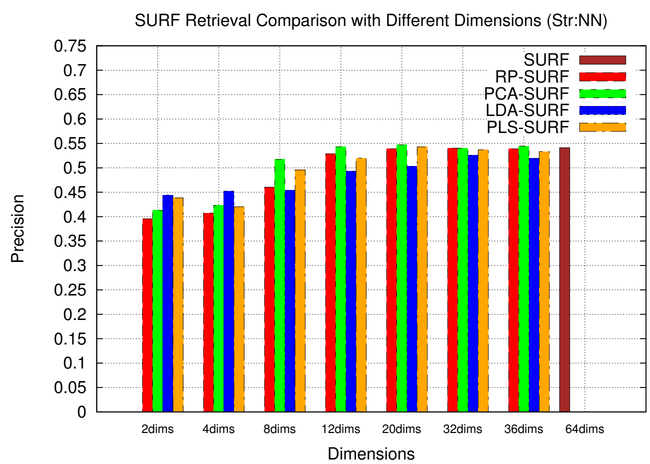


Figure 4.10: SURF - Retrieval Comparisons (Str: NN).

Table 4.5: Reduced-SURF - Retrieval Comparisons (Str: NN).

Descriptor	**Dims.**	**Thr.**	**Perc.**	**Retrieval**
SURF	64	0.425	38.64%	54.13%
RP+SURF	12	0.075	14.29%	52.85%
PCA+SURF	12	0.200	22.22%	54.29%
LDA+SURF	12	1.000	11.76%	49.33%
PLS+SURF	12	0.005	11.76%	52.00%

Finally, Figure 4.11 shows a comparison between the retrieval results computed with the NNDR strategy for SURF feature vectors and each reduced feature vector. Once again, the PCA reduced features obtained superior results. Table 4.7 focuses on the 32

Table 4.6: Retrieval Comparison between Dimensions (Str: NN).

Descriptor	Dims.	Thr.	Perc.	Retrieval
SURF	64	0.425	38.64%	54.13%
PCA+SURF	2	0.100	18.18%	41.28%
PCA+SURF	4	0.050	71.43%	42.29%
PCA+SURF	8	0.150	18.75%	51.73%
PCA+SURF	12	0.200	22.22%	54.29%
PCA+SURF	20	0.300	31.58%	54.72%
PCA+SURF	32	0.350	33.33%	54.03%
PCA+SURF	36	0.400	38.10%	54.45%

Table 4.7: Reduced-SURF - Retrieval Comparisons (Str: NNDR).

Descriptor	Dims.	Thr.	Perc.	Retrieval
SURF	64	0.650	65.00%	50.56%
RP+SURF	32	0.575	57.50%	50.45%
PCA+SURF	32	0.625	62.50%	50.99%
LDA+SURF	32	0.575	57.50%	48.32%
PLS+SURF	32	0.625	62.50%	50.35%

dimensional feature vectors reduced with the PCA projection matrix, achieving better results than the original feature vector. The PLS reduced feature vector retrieval is close to the original feature vector retrieval. Table 4.8 presents the retrieval values achieved by the PCA+SURF descriptor for different dimensions, where features reduced to 32 and 36 dimensions obtained higher accuracy while having a lower imprecision than the original feature.

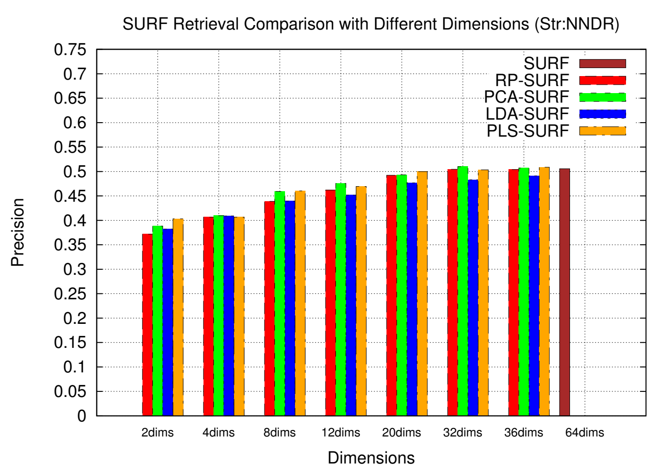


Figure 4.11: SURF - Retrieval Comparisons (Str: NNDR).

Table 4.8: Retrieval Comparison between Dimensions (Str: NNDR).

Descriptor	Dims.	Thr.	Perc.	Retrieval
SURF	64	0.650	65.00%	50.56%
PCA+SURF	2	0.225	22.50%	38.83%
PCA+SURF	4	0.875	87.50%	40.96%
PCA+SURF	8	0.700	70.00%	45.92%
PCA+SURF	12	0.500	50.00%	47.57%
PCA+SURF	20	0.575	57.50%	49.33%
PCA+SURF	32	0.625	62.50%	50.99%
PCA+SURF	36	0.625	62.50%	50.72%

4.2.5 Memory Storage and Computational Time

To perform this experiment, we randomly selected 10,000 images from the Mirflickr-1M Dataset. Computational time and memory storage required to compute these processes can be seen in the following figures and tables presented in this subsection. We used an Intel Core i7-2670QM CPU computer with 2.20 GHz and 8 Gbytes of RAM.

Tables 4.9 and 4.10 show the space required to store 10,000 image descriptors using an average of 804 and 325 keypoints per SIFT and SURF description per image, respectively. On the other hand, Figures 4.12 and 4.13 present a comparison to better observe the benefits, in terms of space required, of using reduced feature vectors. As it can be observed, the descriptor storage is proportional to their dimensions. The most distinctive PCA+SIFT with 36-dimensions and PCA+SURF with 32 dimensions use approximately one third of the memory required by their respective original SIFT and SURF descriptors.

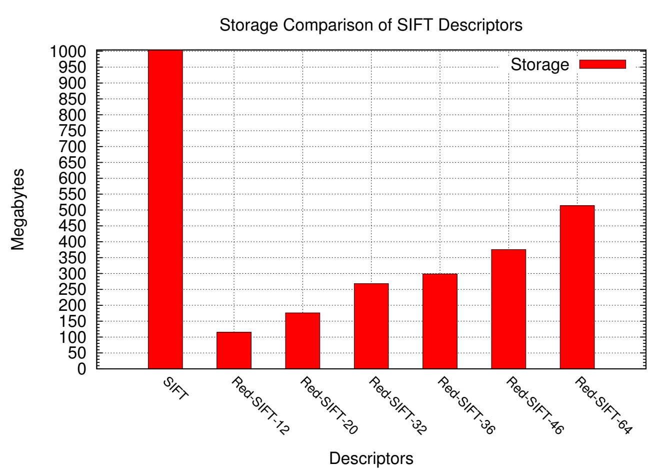


Figure 4.12: SIFT Descriptors - Storage Comparison.

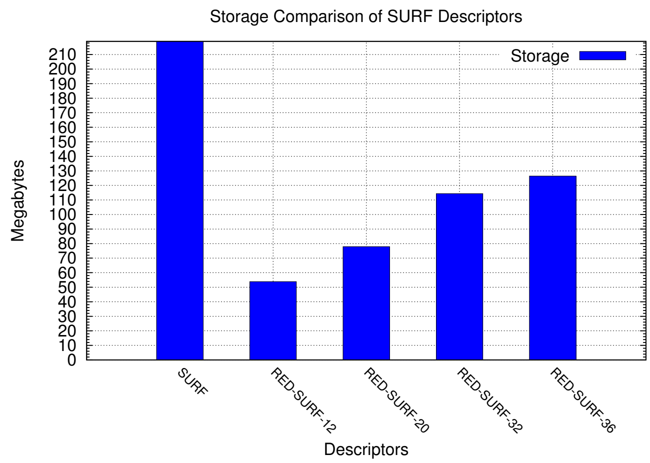


Figure 4.13: SURF Descriptors - Storage Comparison.

Tables 4.11 and 4.12 show the computational time, in minutes, consumed to finally perform the matching process. First, the description stage is performed. Then, SIFT and SURF original feature vectors are ready to start the matching process, while the reduction stage is still needed in order to obtain the reduced feature vectors. Once the reduction stage is performed, every reduced descriptor is ready to be matched.

Table 4.9: SIFT descriptor storage.

Descriptor	**Dims.**	**Storage**
SIFT	128	1004.09 MB
Reduced-SIFT	12	115.01 MB
Reduced-SIFT	20	173.33 MB
Reduced-SIFT	32	268.30 MB
Reduced-SIFT	36	298.96 MB
Reduced-SIFT	46	375.60 MB
Reduced-SIFT	64	513.56 MB

Table 4.10: SURF descriptor storage.

Descriptor	**Dims.**	**Storage**
SURF	64	219.07 MB
Reduced-SURF	12	53.85 MB
Reduced-SURF	20	77.95 MB
Reduced-SURF	32	114.35 MB
Reduced-SURF	36	126.43 MB

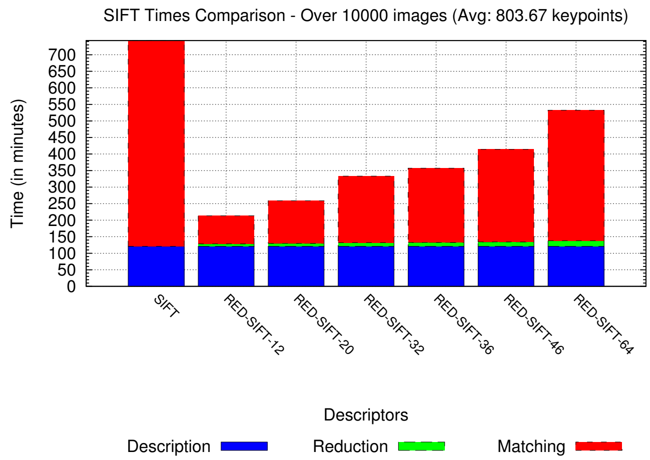


Figure 4.14: SIFT - Time for matching comparison.

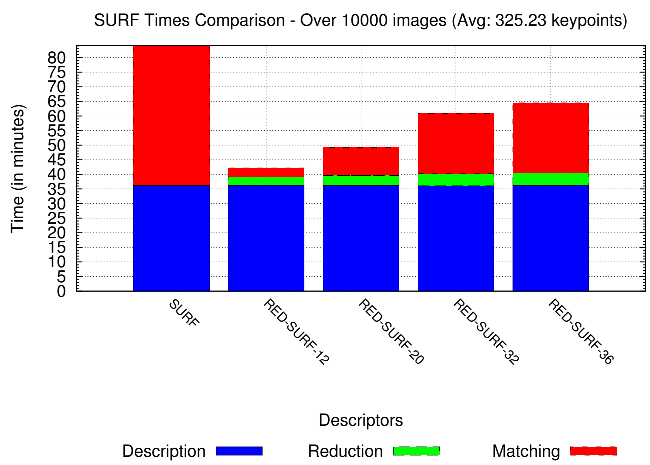


Figure 4.15: SURF - Time for matching comparison.

Table 4.11: SIFT - Time to perform Matching.

		Time in minutes		
Descriptor	**Dims.**	**Desc.**	**Reduc.**	**Matching**
SIFT	128	120.99	0.00	621.90
Reduced-SIFT	12	120.99	7.49	84.80
Reduced-SIFT	20	120.99	8.77	129.10
Reduced-SIFT	32	120.99	10.89	201.30
Reduced-SIFT	36	120.99	11.65	224.50
Reduced-SIFT	46	120.99	13.40	279.60
Reduced-SIFT	64	120.99	16.62	394.90

Table 4.12: SURF - Time to perform Matching.

Descriptor	Dims.	Time in minutes: Desc.	Reduc.	Matching
SURF	64	36.26	0.00	47.93
Reduced-SURF	12	36.26	2.84	3.11
Reduced-SURF	20	36.26	3.36	9.57
Reduced-SURF	32	36.26	4.08	20.54
Reduced-SURF	36	36.26	4.22	24.00

Description and reduction stages are computed once, which is not supposed to occur with the matching stage. In these experiments, the first two stages were computed over the selected $10,000$ images, but only 10 images (with the average number of keypoints) were selected to be matched against all the others. Note that performing an all-vs-all matching leads to a dominating time on matching, where description and reduction times would be hardly perceived, and the gap between matching times for feature vectors and reduced feature vectors would increase significantly.

Finally, it is important to note that memory storage and computational time requirements for the SIFT descriptor are not proportional to the SURF requirements, since both descriptors can detect different number of interest points to describe a same image.

4.3 Discussion

This work shows experimental evidences regarding the viability of reducing feature vectors, up to 90% of their original sizes, while maintaining or even improving the accuracy and precision achieved by their original feature vectors.

Feature vectors reduced by the projection matrix PCA performed better in the majority of the cases. This can be due to the fact that it takes into consideration the relation between features. Therefore, the PCA technique is suitable for applications where there is no need for classification.

The Random Projection technique can construct a projection matrix faster than the other techniques since it does not take any data into consideration. However, this feature is not interesting for this work since it pre-computes the projection matrices due to its fixed training set.

We believe that LDA and PLS techniques can perform even better than PCA for applications where it is important to identify several classes. As it was demonstrated through the image retrieval experiments, with SIFT descriptors and the NNDR strategy, LDA and PLS yielded superior results. Additional tests with more classes are intended to be performed as future work to prove our premise.

The storage required for a reduced feature vector is far lower than the required for an original feature vector. This represents a significant advantage in using the reduced set of features.

The extra computational time spent in reducing the feature vectors is well paid off when the application demands exhaustive work with the described reduced feature vectors, such as in image retrieval and classification tasks.

Chapter 5

Dimensionality Reduction Through LDA and Bag-of-Features Applied to Image Retrieval

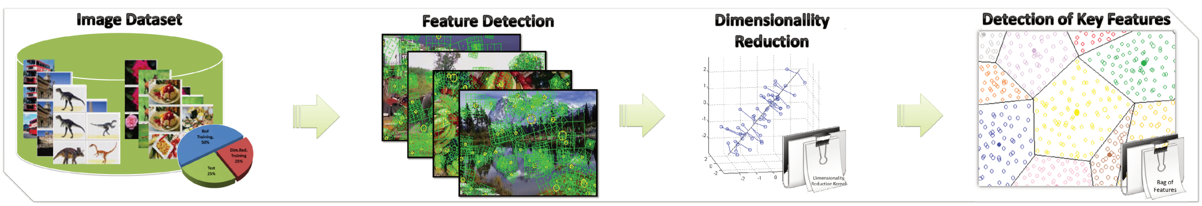
Preamble

Content-based image retrieval applications represent a relevant task in computer vision field. Many researches have been conducted to improve the retrieval process by means of discriminative descriptors, such as Scale-Invariant Feature Transform (SIFT) and Speeded Up Robust Features (SURF), which describe high dimensional feature vectors. Problems involving high dimensional feature vectors usually require high computational time to process the data, high storage requirements to keep relevant information, and may suffer with low accuracy depending on the noise contained in the data. Most of the solutions addressing these problems propose a trade-off between accuracy, time and storage or even impose restrictions on the action of the application by compromising the distinctiveness of the feature descriptors. In this paper, we propose to apply linear dimensionality reduction kernels to reduce the dimensions of SIFT and SURF feature vectors and employ bag-of-features to create global features to maintain or even enhance the accuracy of the retrieval, while spending less storage and computational time requirements. The experiments compare the results achieved by the reduced feature vectors to the results obtained by the original features, demonstrating to gain in accuracy while reducing computational time and storage.

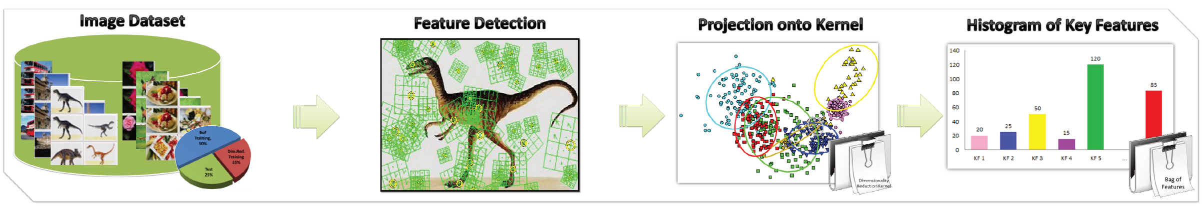
5.1 Methodology

The main purpose of this work is to maintain an approximate accuracy of the original descriptors while reducing the computational time and storage requirements by using dimensionality reduction techniques and bag-of-features.

The main steps of the proposed methodology are illustrated in Figure 5.1. Each stage is described in the following sections.



(a) Training Stage



(b) Testing Stage

Figure 5.1: Training and testing stages. (a) in the training stage, 25% images are separated from each group to create their feature vectors and then obtain a projection matrix (kernel), after that other 50% images from each group are used to create new reduced feature vectors to compute the bag-of-features; (b) in the testing stage, feature vectors are created and described over the last 25% images, then projected over the kernel to obtain the reduced feature vectors and, finally, the histogram of key features is computed to be compared to others.

5.1.1 Training Stage

In order to reduce a feature vector, a projection kernel is usually created. This process may demand a high computational time, however, since every feature vector needs to be projected on the same kernel, this process must be executed on a training stage. To perform this phase, 75 images from each group in the Corel Dataset [6] were selected. Feature vectors are densely detected and described by using SIFT or SURF algorithms over the 750 images.

Then, a dimensionality reduction technique (such as PCA, ICA, RP, PLS or LDA) is applied to the 250 images that are used to learn the projection kernel. If the projection kernel is learned with PLS or LDA, then each feature vector is associated with the image class identifier. Once the projection kernel is already computed, the feature vectors corresponding to the remaining 500 images are projected onto the kernel to obtain the reduced feature vectors. The reduced feature vectors of the 500 images are used to create the bag-of-features to obtain the dictionary through the k-means algorithm.

5.1.2 Testing Stage

Two elements are needed to perform the testing stage: the projection kernel and the bag-of-features.

The feature vectors for the 250 images belonging to this phase are densely detected and computed. Then, these features are projected onto the projection kernel to produce the 250 files of reduced feature vectors. After that, a histogram of n bins is created to have a global descriptor for each image, where n refers to the number of key features in the bag-of-features. Each bin is initialized with 0. To fill this histogram, it is calculated the Euclidean distance between each reduced feature vector and the key features in the bag-of-features.

A reduced feature vector belongs to some key feature when its Euclidean distance is the smallest one, then the bin corresponding to this key feature is increased by one. Subsequently, the histogram is normalized to one. This histogram will represent a unique feature vector describing one image. Finally, the number of similar images can be efficiently computed between these global descriptors.

5.1.3 Evaluation Metrics

During the testing stage, the histograms indicating the number of reduced feature vectors belonging to each key feature are computed for every image in the test group (25 images from each of the ten Corel Dataset groups). Then, each histogram is compared to the other 249 histograms, and a ranking of the first 24 images with histograms with maximum similarity measure is maintained, given by Equation (5.1), where H_x and H_y represent two histograms, n denotes the number of key features, $k_i^{H_x}$ and $k_i^{H_y}$ represent the respective number contained by the i-th bin of their histograms.

$$\mathrm{sm}(H_x, H_y) = \sum_{i=1}^{n} \min(k_i^{H_x}, k_i^{H_y}) \tag{5.1}$$

The retrieved precision value is given by Equation (5.2). An image belonging to the class A will achieve 100% of retrieved precision if its top 24 matched images are composed of only images belonging to its class

$$\text{Retrieved Precision} = \frac{\text{Correct Retrieved Images}}{\text{Total Images to Retrieve}} \tag{5.2}$$

5.2 Experimental Results

To validate our methodology, experiments with an image retrieval application over the Corel Dataset [6] are conducted. This data set is composed of 10 groups of images, each one containing one hundred images. Dense SIFT and Dense SURF algorithms are executed to describe each image in the data set. After that, dimensionality reduction is applied to the descriptors.

The following sections compare the original and reduced feature descriptor in terms of accuracy, computational time and storage requirements.

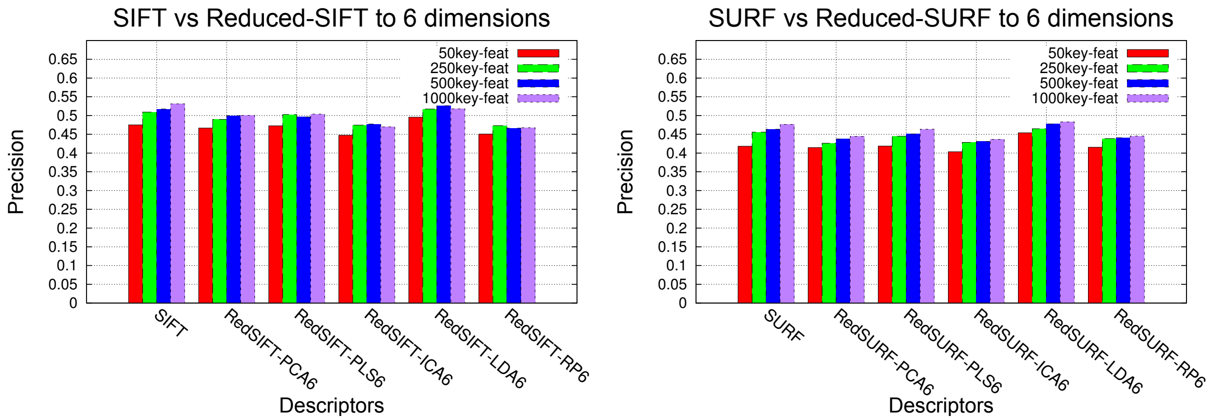
5.2.1 Precision Evaluation

This first experiment compares the retrieved precision achieved by SIFT and SURF descriptors to the correspondent precision achieved by the reduced descriptors. PCA, PLS, ICA, LDA and RP techniques are used to reduce the feature vectors to 6, 12, 20 and 32 dimensions. Furthermore, the generation of several bag-of-features, consisting of 50, 250, 500 and 1000 key features, is performed.

In Figure 5.2, it is shown that the performance obtained by executing the experiments with SIFT (128 dimensions) and SURF (64 dimensions) and their respective 6 dimensional reduced feature vectors. It is noticed that even performing with these few dimensions, most of the reduced feature vectors achieved a similar precision in relation to the original feature vector (with a difference of at most 4% and 6% from the original), and it is worth mentioning that the feature vectors reduced with the LDA dimensionality reduction kernel achieved even superior results than the original feature vectors.

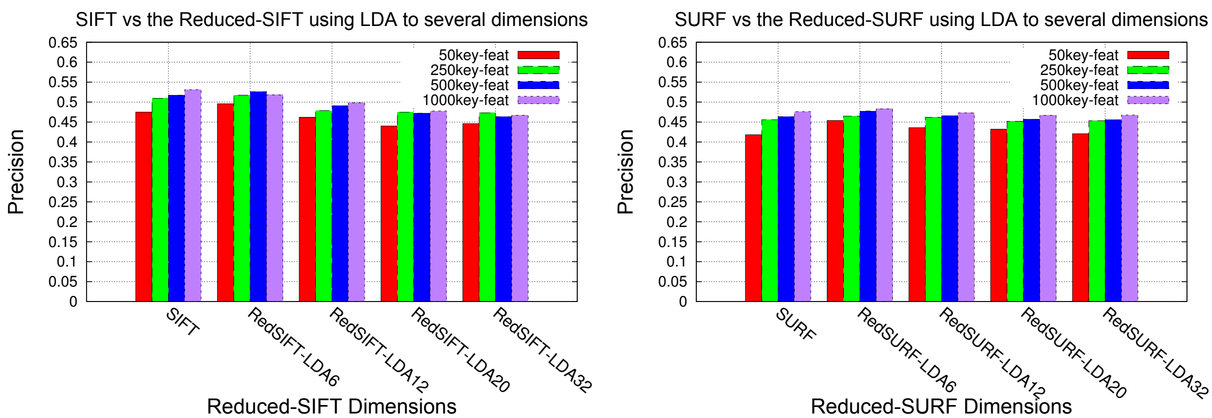
We assume that these favorable results are achieved by every dimensionality reduction kernel due to the fact that not only a small amount of information is removed from the original vectors but also possible noise. Furthermore, PLS and LDA dimensionality reduction techniques report better performance, which can be explained by their ability of marking a feature vector as belonging to a specific class, which aids to better identify the feature vectors of different classes while projecting them onto the kernel.

Figure 5.3 compares the precision retrieved by SIFT and SURF to their respective LDA reduced feature vectors. It is important to notice that, in both cases, the 6 dimensional reduced feature vectors produce superior results than any other reduced feature vector.



(a) (b)

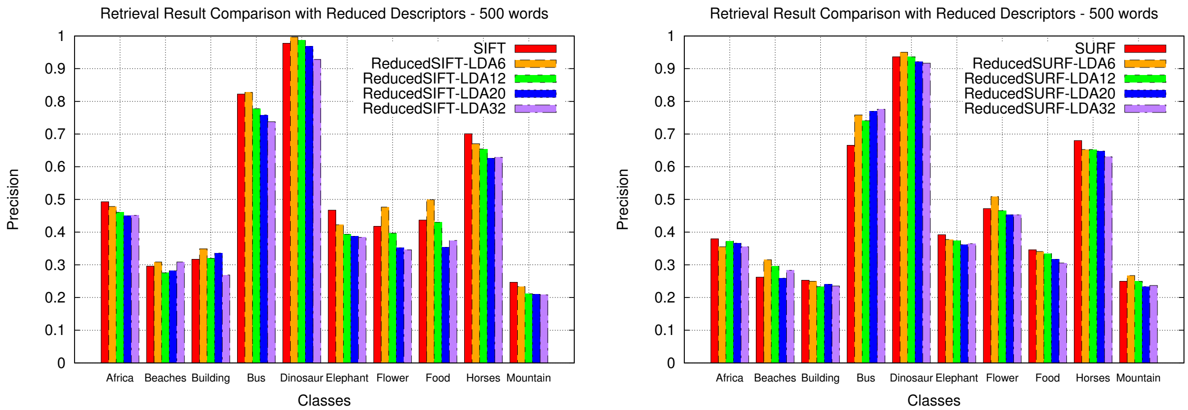
Figure 5.2: Comparison between the original feature vectors and the 6 dimensional reduced feature vectors using bag-of-features of 50, 250, 500 and 1000 key features (a) SIFT compared to RedSIFT-PCA, RedSIFT-PLS, RedSIFT-ICA, RedSIFT-LDA, RedSIFT-RP; (b) SURF compared to RedSURF-PCA, RedSURF-PLS, RedSURF-ICA, RedSURF-LDA, RedSURF-RP.



(a) (b)

Figure 5.3: Comparison between the original feature vectors and LDA reduced feature vectors using 6, 12, 20 and 32 dimensions and bag-of-features of 50, 250, 500 and 1000 key features (a) SIFT compared to RedSIFT-LDA; (b) SURF compared to RedSURF-LDA.

Finally, Figure 5.4 shows the retrieval precision obtained by each class in the Corel Dataset by SIFT and SURF feature vectors compared to their respective LDA reduced feature vectors using 6, 12, 20 and 32 dimensions and a bag-of-features composed of 500 key features, which corresponds to an appropriate balance between accuracy, computational time and storage. It is clear that LDA 6 dimensional features vectors present either better results than their original feature vectors or approximately similar.



(a) (b)

Figure 5.4: Detailed comparison between the original feature vectors and LDA reduced feature vectors using 6, 12, 20 and 32 dimensions and bag-of-features of 500 key features (a) SIFT compared to RedSIFT-LDA; (b) SURF compared to RedSURF-LDA.

5.2.2 Computational Time Evaluation

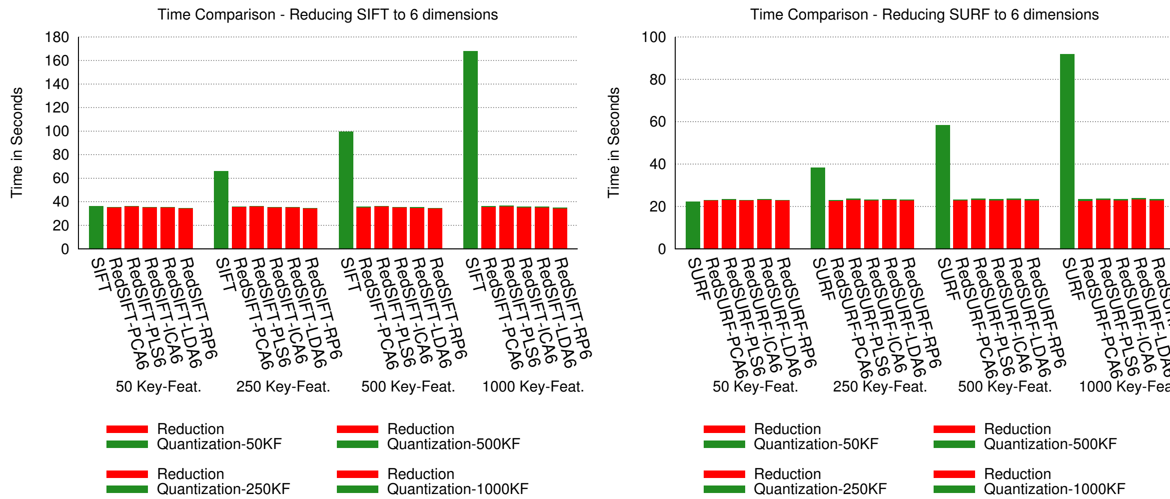
There are four phases to compute the testing stage: extracting the feature vector; reducing the original feature vector; computing the respective histogram (also called quantization phase); and calculating the retrieved precision value. The first and the last phases are the same for any feature vector or reduced feature vector, such that the reduction and quantization phases are taken into consideration to compute the difference in computational time.

The reduction phase, obviously, does not affect the SIFT or SURF descriptors; therefore, it does not represent an associated cost to them. However, as the quantization step is directly affected by the number of dimensions of the feature vectors. Figure 5.5 illustrates a comparison between the process performed with the original feature vector and its respective reduced feature vectors. clearly the time to compute the quantization phase is dominated by the number of dimensions of the feature vector, then the time spent in the reduction phase is totally remunerated by the time gained in the quantization phase.

5.2.3 Storage Evaluation

Content-based image retrieval applications usually store image descriptions to avoid computing them every time. The actual features that need to be stored will be the image histograms, however, if it is desired to change the bag-of-features to compute them, then the feature vectors must also be stored.

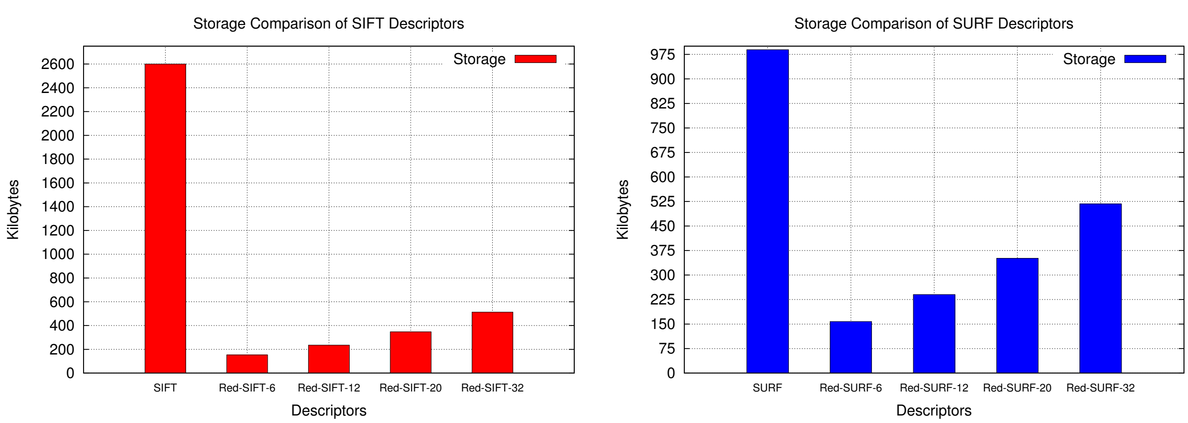
Figure 5.6 shows the storage requirements needed to store a file corresponding to dense



(a) (b)

Figure 5.5: Computational time comparison (a) SIFT compared to RedSIFT-PCA, RedSIFT-PLS, RedSIFT-ICA, RedSIFT-LDA, RedSIFT-RP; (b) SURF compared to RedSURF-PCA, RedSURF-PLS, RedSURF-ICA, RedSURF-LDA, RedSURF-RP.

feature vectors compared to their respective reduced feature vectors using 6, 12, 20 and 32 dimensions. As the 6 dimensional feature vectors perform better than the original feature vectors, it is important to notice that they represent 5.86% and 15.93% of storage size required by the original SIFT and SURF descriptors, respectively.



(a) (b)

Figure 5.6: Storage comparison (a) storage for SIFT feature vectors compared to their respective reduced feature vectors; (b) storage for SURF feature vectors compared to their respective reduced feature vectors.

5.3 Discussion

Content-based image retrieval applications usually compute image feature vectors once and employ them several times. Then, the reduction of feature vectors is desirable in terms of computational time and storage requirements. Since LDA and PLS are able to identify a feature vector as belonging to a particular class the performance they guarantee a better performance. Nevertheless, PCA, ICA and RP techniques also present an acceptable performance.

This work demonstrated the feasibility of reducing the dimensionality of descriptors, such as SIFT and SURF, while maintain a similar accuracy or even improve it when using less than 5% of the original feature vector dimensions. Computational time was not reduced only during the phase of calculating the reduced feature vectors but also when comparing the obtained histograms, as they represent an entire image through a unique vector.

Chapter 6

Conclusions and Future Work

The reduction of dimensionality, applied to SIFT and SURF descriptors, has demonstrated to maintain a similar distinctiveness while the new descriptions are even 10% of the size of the original ones. However, it is important that the selection of the appropriate linear dimensionality reduction technique occurs according to the required application.

In Chapter 3, the image matching experiments were developed with several training matrices. Some of the training matrices were compounded of descriptors taken from random images, from an image dataset that was not used in the test stage, and some other training matrices were composed of images from the same dataset of the test images. The use of these types of training matrices produced the same results, from which we deduce that, for a specific application such as matching, where there is no need to differentiate among classes, a matrix projection can be computed once and then used for several other images (*i.e.*, the training stage will be performed just once).

In Chapter 4, some new matching experiments were performed by applying other linear dimensionality reduction techniques to the SIFT and SURF descriptors. The supervised linear dimensionality reduction techniques, *i.e.*, LDA e PLS, did not show to outperform the results obtained by the PCA reduced descriptors. Therefore, we conclude that it is not a determining factor to add distinctiveness to a compact representation in order to discriminate descriptors in function of the most and least representative keypoints. However, when the linear dimensionality reduction is used through an image processing application involving different image classes, the supervised techniques can add more distinctiveness to the reduced descriptors to the point of making them more accurate even than the original descriptors.

In Chapter 5, the LDA supervised dimensionality reduction technique is employed in an image retrieval application. The results show how it maintains a similar accuracy to the original descriptors or even improve it. The bag-of-feature representation is used in order to have only one signature descriptor per image and perform faster than the process.

After the investigation carried out in this work, we conclude that just taking into consideration projection matrices which contain vectors describing the largest possible variations of the data, as done by PCA, is good enough to obtain representative reduced feature vectors. However, when it is desired to take into consideration the class to which an image belongs, as in image retrieval or classification applications, it is necessary to think about alternative projection matrices that shall be constructed by considering some relation and distinctiveness of the feature vectors from the different classes, as done by the PLS and LDA supervised techniques.

The Random Projection did not produce satisfactory results in our experiments, which can be explained because it does not take into consideration any information about the data to be reduced. In the other hand, as the ICA method searches for unique independent components over the given data, it did not produce better results compared to the other techniques since its un-mixing matrix attempts to separate independent components from sources that were not distributed, however, they are almost the same in these datasets.

In our experiments, the dimensionality reduction attains to reduce the computational time by reducing the keypoints to a bag-of-feature representation as by reducing the feature vector dimensions. As the only heavy processes, like creating a vocabulary or the matrix projection, can be done offline they do not represent any obstacle. The final stage, of reducing the dimensionality of a feature vector, is just represented by an internal product between it and the projection matrix, which is not computational expensive in terms of process and time, and leads to less disk storage usage.

As feature descriptors reduced by the PCA technique can maintain a similar accuracy than the original descriptors and feature descriptors reduced by LDA or PLS techniques add distinctiveness, a fusion of such techniques could improve the results when compared to those applied separately, which is suggested as future work.

Another promising alternative is to experiment non-linear dimensionality reduction techniques in our problem. There are several approaches investigated in the literature, some of them based on traditional methods, such as kernel PCA [12, 33] and kernel LDA [19, 30].

References

[1] S. Agarwal and D. Roth. Learning a Sparse Representation for Object Detection. In *7th European Conference on Computer Vision*, pages 113–130, Copenhagen, Denmark, May 2002.

[2] S. Balakrishnama and A. Ganapathiraju. Linear Discriminant Analysis - A Brief Tutorial. In *International Symposium on Information Processing*, Mississippi State, MS, USA, 1998.

[3] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded Up Robust Features. In *European Conference in Computer Vision*, pages 404–417, 2006.

[4] E. Bingham and H. Mannila. Random Projection in Dimensionality Reduction: Applications to Image and Text Data. In *Knowledge Discovery and Data Mining*, pages 245–250. ACM Press, 2001.

[5] M. Brown and D. Lowe. Invariant Features from Interest Point Groups. In *British Machine Vision Conference*, pages 656–665, 2002.

[6] Corel Dataset. `http://people.csail.mit.edu/torralba/research/bias/`, 2013.

[7] G. Csurka, C. R. Dance, L. F., J. Willamowski, and C. Bray. Visual Categorization with Bags of Keypoints. In *Workshop on Statistical Learning in Computer Vision*, pages 1–22, 2004.

[8] S. Dasgupta and A. Gupta. An Elementary Proof of the Johnson-Lindenstrauss Lemma. Technical Report TR-99-006, Berkeley, CA, 1999.

[9] N. Delfosse and P. Loubaton. Adaptive Separation of Independent Sources: A Deflation Approach. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume iv, pages IV/41–IV/44 vol.4, 1994.

[10] L. Fei-Fei, R. Fergus, and A. Torralba. Recognizing and Learning Object Categories. In *10th IEEE International Conference on Computer Vision)*, 2009.

[11] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 2001.

[12] H. Hoffmann. Kernel PCA for Novelty Detection. *Pattern Recognition*, 40:863–874, 2007.

[13] A. Hyvärinen and E. Oja. Independent Component Analysis: A Tutorial. *Neural Networks*, 13(4-5):411–430, 2000.

[14] INRIA. INRIA Graffiti Dataset, 2011. `http://www.inrialpes.fr/movi`.

[15] W. Johnson and J. Lindenstrauss. Extensions of Lipschitz Mappings into a Hilbert Space. In *Conference in Modern Analysis and Probability*, volume 26 of *Contemporary Mathematics*, pages 189–206. American Mathematical Society, 1984.

[16] I. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, 2002.

[17] L. Juan and O. Gwun. A Comparison of SIFT , PCA-SIFT and SURF. *International Journal of Image Processing*, 3(4):143–152, 2009.

[18] Y. Ke and R. Sukthankar. PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 506–513, 2004.

[19] S.-J. Kim, A. Magnani, and S. Boyd. Optimal Kernel Selection in Kernel Fisher Discriminant Analysis. In *23rd International Conference on Machine Learning*, pages 465–472, Pittsburgh, Pennsylvania, 2006.

[20] B. Leibe and B. Schiele. The TU Darmstadt Database (formerly the ETHZ Database), 2010. `http://pascallin.ecs.soton.ac.uk/challenges/VOC/databases.html#TUD`.

[21] D. G. Lowe. Object Recognition from Local Scale-Invariant Features. In *IEEE International Conference on Computer Vision*, pages 1150–1157. IEEE Computer Society, 1999.

[22] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, Nov. 2004.

[23] S. Maitra and J. Yan. Principle Component Analysis and Partial Least Squares: Two Dimension Reduction Techniques for Regression. In *Applying Multivariate Statistical Models*, pages 79–90, 2008.

[24] S. Mallat. *A Wavelet Tour of Signal Processing, Second Edition (Wavelet Analysis & Its Applications)*. Academic Press, 2 edition, Sept. 1999.

[25] K. Mikolajczyk and C. Schmid. Indexing Based on Scale Invariant Interest Points. *IEEE International Conference on Computer Vision*, 1:525–531, 2001.

[26] K. Mikolajczyk and C. Schmid. A Performance Evaluation of Local Descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, Oct. 2005.

[27] MIR Flickr. MIR Flickr-1M, 2010. `http://press.liacs.nl/mirflickr/`.

[28] G. R. Naik and D. K. Kumar. An Overview of Independent Component Analysis and its Applications. *Informatica (Slovenia)*, 35(1):63–81, 2011.

[29] R. Noonan and H. Wold. NIPALS Path Modelling with Latent Variables. *Scandinavian Journal of Educational Research*, 21(1):33–61, 1977.

[30] C. H. Park and H. Park. A Comparison of Generalized Linear Discriminant Analysis Algorithms. *Pattern Recognition*, 41(3):1083–1097, 2008.

[31] K. Pearson. On Lines and Planes of Closest Fit to Systems of Points in Space. *Philosophical Magazine*, 2(6):559–572, 1901.

[32] R. Rosipal and N. Krämer. Overview and Recent Advances in Partial Least Squares. In *Subspace, Latent Structure and Feature Selection Techniques*, pages 34–51. Springer Berlin / Heidelberg, 2006.

[33] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*, 10(5):1299–1319, July 1998.

[34] SIFT. Keypoint Orientation, 2011. `http://www.aishack.in/2010/05/sift-step-5-assigning-keypoint-orientation/`.

[35] SIFT. Scale Images, 2011. `http://www.aishack.in/2010/05/sift-step-1-constructing-a-scale-space`.

[36] J. Sivic and A. Zisserman. Video Google: A Text Retrieval Approach to Object Matching in Videos. In *International Conference on Computer Vision*, volume 2, pages 1470–1477, 2003.

[37] J. Sivic and A. Zisserman. Efficient Visual Search of Videos Cast as Text Retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(4):591–605, Apr. 2009.

[38] V. Sulic, J. Perš, M. Kristan, and S. Kovacic. Efficient Dimensionality Reduction Using Random Projection. In *15th Computer Vision Winter Workshop*, pages 29–36, Nové Hrady, Czech Republic, Feb. 2010.

[39] T. Tuytelaars and K. Mikolajczyk. Local Invariant Feature Detectors: A Survey. In *Foundation and Trends in Computer Graphics and Vision*, pages 177–280, 2008.

[40] R. Valenzuela, H. Pedrini, and W. Schwartz. Dimensionality Reduction Through LDA and Bag-of-Features Applied to Image Retrieval. In *IV ECCOMAS Thematic Conference on Computational Vision and Medical Image Processing*, pages 31–37, Funchal, Portugal, Oct. 2013.

[41] R. Valenzuela, W. Schwartz, and H. Pedrini. Dimensionality Reduction Through PCA over SIFT and SURF Descriptors. In *11th IEEE Conference on Cybernetic Intelligent Systems*, pages 58–63, Limerick, Ireland, Aug. 2012.

[42] R. Valenzuela, W. Schwartz, and H. Pedrini. Linear Dimensionality Reduction Applied to SIFT and SURF Feature Descriptors. *Submitted to Journal of Electronic Imaging*, 2014.

[43] P. Viola and M. Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 511–519. IEEE Computer Society, 2001.

[44] S. Wold, A. Ruhe, H. Wold, and W. J. Dunn. The Collinearity Problem in Linear Regression. The Partial Least Squares (PLS) Approach to Generalized Inverses. *SIAM Journal on Scientific and Statistical Computing*, 5(3):735–743, 1984.

[45] Wyman. Speeded Up Robust Features, 2011. `http://www.cse.cuhk.edu.hk/~lyu/seminar/06fall/Wyman.ppt`.

[46] J. Zhang, S. Lazebnik, and C. Schmid. Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study. *International Journal of Computer Vision*, 73, 2007.

[47] G. Zhao, L. Chen, G. Chen, and J. Yuan. KPB-SIFT: A Compact Local Feature Descriptor. In *ACM Multimedia*, pages 1175–1178, 2010.