IMAGE FEATURE BASED OBJECT DESCRIPTION, DETECTION AND REAL TIME TRACKING

(İMGE KARAKTERİSTİK TABANLI NESNE TASVİRİ, SAPTAMA VE GERÇEK ZAMANLI TAKİBİ)

by

Ramazan YILDIZ, B.S.

Thesis
Submitted in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE

In

MASTER OF SCIENCE

Date of Submission : May 24, 2013
Date of Defense Examination : June 13, 2013

Supervisor : Assoc. Prof. Dr. Tankut ACARMAN

Committee Members : Assoc. Prof. Dr. Temel ÖNCAN

Asst. Prof. Dr. Murat AKIN

Acknowledgements

First and foremost I offer my sincerest gratitude to my supervisor, Assoc. Prof. Dr. Tankut ACARMAN, who has supported me throughout my thesis with his patience and knowledge whilst allowing me the room to work in my own way. I attribute the level of my Master's degree to his encouragement and effort and without him this thesis, too, would not have been completed or written. One simply could not wish for a better or friendlier supervisor.

Besides my advisor, I would like to thank my managers at Netaş, Mahmut BİBİNOĞLU, Taner ARIKAN, İbrahim KARADOĞAN, Sait ŞENER, Orhan Başar EVREN and Öner TEKİN for their encouragement and the tolerance given to me to accomplish my studies. Special thanks to Netaş, for supporting young professionals to continue higher education.

Finally, I thank my parents, my wife, my brothers, my mother and father for supporting me throughout all my studies at University.

May 22, 2013
Ramazan YILDIZ

Table of Contents

Acknowledgements………………………….……………….…….……ii
Table of Contents ………………………….………………………..….iii
List of Figures…...…………………………..……………………….....vi
Abstract…………...…………………………………………………....ix
Resumé……..……………...……………………………………….....x
Özet……………..…………...………………………………….........xi
1 Introduction 1
2 State Of Art... 3
2.1 Noise Removal with Gaussian Blurring.. 3
2.2 Canny Edge Detection 7
2.2.1 Step 1: Noise Removal 10
2.2.2 Step 2: Gradient Image Representation 11
2.2.3 Step 3: Edge Direction Detection 12
2.2.4 Step 4: Edge Orientation Detection, Classification of Edge Directions ... 12
2.2.5 Step 5: Non-maximum Suppression 13
2.2.6 Step 6: Hysteresis.. 14
2.3 SIFT: Scale Invariant Feature Transform 14
2.3.1 Local Extrama Detection 15
2.3.2 Local Extremum Localization 16
2.3.3 Orientation Affectation.. 18
2.3.4 Key Point Descriptor 19
2.4 SURF: Speeded Up Robust Features 20

2.4.1 Integral Images 21
2.4.2 Hessien Detector of SURF 21
2.4.3 SURF Orientation Affectation 24
2.4.4 SURF Descriptor 25
2.5 Conclusion of SIFT and SURF 27
2.6 Object Category Recognition HMAX Model 27
2.6.1 About HMAX Implementation 27
3 Our Researches 34
3.1 Local Features Approach 34
3.1.1 Generic Points with Face Dataset 34
3.1.1.1 Selection of proper SIFT implementation 34
3.1.1.2 Generic Points 36
3.1.1.3 About Lowe’s Matching 41
3.1.1.4 Our Matching 43
3.1.1.5 Generic Points Aided Matching Result 46
3.1.1.6 A Dangerous Problem False Match Elimination 48
3.1.1.7 A Triangular Relative Positioning Model 50
3.1.1.8 Triangular Similarity 53
3.1.1.9 Triangular Parallelism 53
3.1.1.10 Generic Points Aided Elimination Results 54
3.1.1.11 Some Test Results for Object Detection 57
3.1.1.11.1 First Frame 57
3.1.1.11.2 Second Frame 57
3.1.1.11.3 Third Frame 58
3.1.1.11.4 Forth Frame 59
3.1.1.12 Assessments and Perspectives for Generic Points with Face Dataset 60

3.1.2 Generic Points with Car Dataset 61
3.1.2.1 Image Local Feature Based Video Object Description and Tracking 61
3.1.2.2 Methodology 62
3.1.2.3 Car Data Set Matching Methodology 65
3.1.2.4 Car Data Set Generic Points Aided Matching Versus Lowe’s Matching..65
3.1.2.5 Car Data Set False Match Elimination 67
3.1.2.6 Tracking with Generic Points 71
3.1.2.7 Local Features Approach Assessments 73
3.2 Global Features Approach 73
3.2.1 Real Time Object Detection Using Image Global Features and Detection Based Object Tracking 74
3.2.1.1 Vehicle Detection Using Haar Features 74
3.2.1.2 Vehicle Detection Verification 76
3.2.1.2.1 X-Symmetry 77
3.2.1.2.2 Horizontal Edges 79
3.2.1.2.3 Detection History 82
3.3 Experimental Results 84
3.3.1 Assessment of Local Features Approach 84
3.3.2 Assessment of Global Features Approach 86
4 Conclusion 89
5 References 91
Biographical Sketch……………..……………………..………………93

List of Figures

Figure 2.1 - Smoothing effect on edge detection (A) 5
Figure 2.2 - Smoothing effect on edge detection (B) 5
Figure 2.3 - Smoothing effect on edge detection (C) 6
Figure 2.4 - Smoothing effect on edge detection (D) 6
Figure 2.5 - Smoothing effect on edge detection (E).. 7
Figure 2.6 - Canny Edge Detection Algorithm... 8
Figure 2.7 - Discrete approximation to Gaussian function with σ=1.4 10
Figure 2.8 - Sobel operators.. 11
Figure 2.9 - Edge orientation ranges... 13
Figure 2.10 - Gaussian Scale Space.. 15
Figure 2.11 - Gaussian sampling 16
Figure 2.12 - Matching problem... 18
Figure 2.13 - SiFT Descriptor... 19
Figure 2.14 - Gaussian function and box filters.. 22
Figure 2.15 - SURF Orientation 25
Figure 2.16 - SURF Descriptor Extraction.. 26
Figure 2.17 - First part of HMAX model... 29
Figure 2.18 - C1 extraction.. 30
Figure 2.19 - C1 patch extraction 31
Figure 2.20 - C2 response extraction.. 32
Figure 2.21 - C2 similitude response extraction... 33
Figure 3.1 - Frontal view and 45 Degrees rotated view comparisons 36
Figure 3.2 - Planar Perspective Transformations.. 38
Figure 3.3 - Planar Perspective Transformations.. 39
Figure 3.4 - Transformed image matching to the query image 41
Figure 3.5 - SIFT Matching... 42

Figure 3.6 - Generic Points Aided Sift Matching 44
Figure 3.7 - A comparison of our model 47
Figure 3.8 - Different objects are being mismatched because of the elevated distance ratio 0.77 of Generic Points Matching. 49
Figure 3.9 - Perfectly distributed Generic Point false matching. 50
Figure 3.10 - Randomly distributed false matchings which is general case. 52
Figure 3.11 - Generic Points Aided matching versus classical SiFT matching. 55
Figure 3.12 - Generic Points Aided matching versus classical SiFT matching. 56
Figure 3.13 - Vedaldi vs Generic Points 1 57
Figure 3.14 - Vedaldi vs Generic Points 2 58
Figure 3.15 - Vedaldi vs Generic Points 3 59
Figure 3.16 - Vedaldi vs Generic Points 4 59
Figure 3.17 - Images A, B, C, D, E, F are perspective planar transformations in different directions of image G. Transformation ratios are [0.9, 0.9] and [1.0, 0.8]. 64
Figure 3.18 - Lowe SIFT [A] and Vedaldi SIFT [B] matching versus Generic Points aided matching [C-D-E-F]. 66
Figure 3.19 - Generic points matching (blue column) versus Vedaldi matching (red column) 69
Figure 3.20 - Correct matching comparison 70
Figure 3.21 - False matching comparison 71
Figure 3.22 - Tracking with generic points 72
Figure 3.23 - (a) Example of Haar-like features. (b) Cascade of boosted classifiers. 74
Figure 3.24 - Symmetry search 78
Figure 3.25 - Edge image horizontal line search 80
Figure 3.26 - Discontinuity on lines 81
Figure 3.27 - Detection and tracking 82
Figure 3.28 - Detection and tracking 83
Figure 3.29 - Tracking with Local features, correct matchings' per car rear 85
Figure 3.30 - Tracking with Local features, false matchings' per car rear. 85
Figure 3.31 - Istanbul TEM Highway video dataset metrics 86
Figure 3.32 - LISA-Q Front FOV video dataset metrics 1 87
Figure 3.33 - LISA-Q Front FOV video dataset metrics 2 87

Figure 3.34 - A capture from our recognition and tracking schema on LISA-Q Front FOV 1 - Rush hour.. 88

Figure 3.35 - A capture from our recognition and tracking schema on Istanbul TEM Highway.. 88

Abstract

Our thesis is based on image feature extraction, description of objects using extracted features, real time detection and comparison of State of art technics. Using global and local image features, two separate proceedings carried out. First proceeding is based on use of local image features and second one is based on global image features. Local image feature extraction technics such as SIFT and SURF are comparatively inspected and using SIFT a new object description and video object tracking project is carried out. For description, we introduced Generic Points notion extracted with use of different perspective planar transformations. These points are robust against object geometrical deformations than the State of art SIFT. Based on Generic Points, a feature tracking algorithm is designed running over video images. Generic features are integrated with RAMOSAC tracker. To provide real-time efficiency, CUDA GPU implementation of SIFT is used. Global image feature extraction technics such as Hmax and Haar Like Simple Features are inspected. The implementation of early Hmax is inspected and a referential feature extraction algorithm is described. Well known Haar Like Simple Features technic is adopted and used for describing object models. This technic is used with Adaboost classifier for preliminary object detection over on road vehicle video records. Using image analyze technics, a suit of detection algorithm is designed to validate preliminary detections. Using color channels for texture analyze, object symmetric features searched. Using edge level images, horizontal lines are detected inside detected region of interests. Symmetric feature search and prominent horizontal line frequency detection are used for validation. Temporal detection story is used for tracking and validation as well. A new proceeding is carried out which enables active safety for on road vehicle navigation.

Resumé

Notre thèse est basée sur l'extraction de particularités d'image, la description et détection d'objet en temps réel et la comparaison des techniques de l'état de l'art. Nous avons développé deux différents projets d'études. Le premier est basé sur les techniques SIFT et SURF, ceux deux sont analysées d'une manière comparative et ils sont utilisés pour l'extraction des particularités locales d'image. Une nouvelle technique pour la description et le suivi d'objet est proposée. La notion des Points Génériques, basée sur les transformations perspectives planaires, est introduit. Ces points sont plus robustes contre les déformations géométriques d'image. En utilisant les points génériques, un algorithme de suivi de particularités est développé et utilisé avec les vidéos de trafic. Les points génériques sont utilises avec l'algorithme RANSAC pour le suivi. Pour garantir le temps réel, l'implémentassions avec CUDA GPU de SIFT est utilisé. D'autre part, les algorithmes Hmax et Haar Like Simle Features sont analysés. Ceux deux servent pour l'extraction des particularités globales. La technique Haar Like Features est adoptée et d'en proposée une nouvelle projet de détection et le suivi d'objets. La description d'objet est fait avec cette technique, et avec le classificateur Adaboost la détection préliminaire d'objets a étés réalisées sur les vidéos de trafic. Une suite d'algorithme de détection est développée en utilisant les techniques d'analyse d'image pour valider les détections préliminaires. Les canaux de couleurs sont utilisés pour l'analyse de texture d'image, une symétrie est recherchée sur la région d'intérêt. En utilisant l'images aux niveaux des contours, la fréquence des lignes horizontaux est détectée. Ces critères sont utilisées pour la validation. L'histoire de détection temporale est utilisée pour le suivi et la validation d'objet. Un nouveau travaille est mis en place qui puisse permettre la sécurité active pour la navigation des véhicules routiers.

Özet

Bu çalışma imge karakteristik temelli nesne tasviri, gerçek zamanlı nesne saptama, hareketli nesne takip etme konularını ele almaktadır. Global ve lokal imge karakteristiklerini ayrı ayrı konu edinen, iki ayrı çalışma ortaya konmuştur. İlk çalışma lokal karakteristik kullanımı temellidir. Bu kapsamda literatür çalışmaları olarak bilinen SIFT ve SURF teknikleri mukayeseli olarak incelendi. SIFT kullanılarak, bir nesne betimleme yöntemi geliştirildi. Bu betimleme video nesne takibi için kullanıldı. Betimleme aşaması için, Çesitli perspektif düzlemsel dönüşümler kullanılarak elde ettiğimiz ve Generic Points adını verdiğimiz karakteristik kümesi kullanıldı. Nesne geometri bozulmalarına karşı daha az kırılgan olan bu karakteristik kümesi ile, tasvir edilen nesneleri, hareket halinde iken takip eden bir algoritma geliştirdik. Bu algoritma RAMOSAC takip algoritmasi ile entegre edildi. Gerçek zamanlılığı sağlamak için SIFT CUDA GPU uygulaması kullanıldı. İkinci çalışmamız, global imge karakteristik çıkarım tekniklerini ele almakta. Hmax ve Haar Like Simple Features algoritmaları ele alındı. Hmax açık kaynak uygulama kodu, aşama aşama incelendi. İmge Global karakteristik çıkarım algoritması dokümante edildi. Haar Like Features tekniği benimsendi ve gerçek zamanlı bir saptama ve takip projesi geliştirildi. Araç tamponlarını içeren bir imge kümesi bu teknik ile betimlendi. Haar Like Simple Features karakteristik çıkarımı ile Adaboost sınıflandırıcı algoritması kullanılarak öncül nesne saptaması yapıldı. Öncül saptama sonuçları, imge işleme teknikleri kullanılarak validasyona tabi tutuldu. Renk kanalları kullanılarak imge doku analizi yapıldı ve simetrik karakteristik araması yapıldı. İmge kenar analizi yapılarak, yatay ve dominant çizgiler algilamasi yapıldı. Simetrik karakteristikler ve yatay dominant çizgi frekans indikatörlerini kullanarak öncül saptamanın validasyonu yapıldı. Zamansal saptama özgeçmişi tutuldu, nesne saptama validasyonu ve nesne takibi için kullanıldı. Yol üzerinde araç seyri için aktif güvenlik sağlayabilecek bir proje önerildi.

1 Introduction

This document presents state of art technics used in computer vision domain, discusses and compares them. Beside low level image processing technics, sophisticated technics are presented such as Canny Edge Detection algorithm, Scale Invariant Feature Transform, Speeded up Robust Features Extraction, Biological Visual Cortex Inspired Features-HMAX Model, Haar Like Simple Features.

Canny Edge Detection algorithm is a low level image processing technic incorporates other low level image processing notions.

Scale Invariant Feature Transform and Speeded up Robust Features Extraction are used for local image analyses. These two approaches extract distinctive local image features. These are local feature extraction technics.

HMAX and Haar Like Simple features are global image features; qualifies thickness and orientation of contours, junctions, intensity changes, geometry of shapes and objects etc. We inspected in detail the HMAX model implementation of Jim Mutch and extracted a referential algorithm of this implementation which clarifies in detail how biologically inspired features are extracted. Haar Like Simple features is much more efficient in simple and useful feature extraction than HMAX model, as it uses box filters inspired from Haar Wavelet Transform with integral images.

Based on these global and local feature extraction technics, we carried out proceedings that will be presented in this document. Our proceedings are separately based on these two different approaches.

First approach is the use of local features for image object description and tracking. A system is developed which provides a more robust description than the state of art technic SIFT and adapted it to be useful for tracking of an already detected object. Object detection is not incorporated within this study. Our tests are done over two data sets. Description and matching stages are firstly tested over a face image set containing different peoples; frontal and rotated pictures captured from different perspectives, including difficult scene conditions such as noise, shadow etc...

Our system is then tested with a car image data set, afterwards used for video object tracking. On road traffic videos are used to track vehicle rears, under real scene conditions. Both data set results are discussed, useful metrics are reported.

Second approach is the use of global features for image object description, detection and tracking. A video object detection and object tracking algorithm is developed and discussed. The algorithm runs in real time. Haar Like Simple Features are used for a preliminary detection, region of interests are extracted. Afterwards validation algorithms are developed based on low level image analyses which makes robust and safe our detection process. Finally, tracking is handled with a clever algorithm that we developed, which is based on temporal information of detection history. Different real world videos are tested. Video records captured from Istanbul TEM highway are tested.

2 State Of Art

Sate of Art approaches and important image analyze techniques are subjected in this section.

2.1 Noise Removal with Gaussian Blurring

Gaussian blur is also known as Gaussian smoothing. It is obtained by convolving an image with a Gaussian function. It is a well-known technic in image/video processing and also it's used in other signal processing disciplines. It's typically useful for reducing noise in signals like image. Reducing noise corresponds to reducing details in image. The visual effect of smoothing an image with Gaussian blur is similar to view the considered image via a translucent screen. This effect is different from the well-known Bokeh effect. Bokeh effect is produced with the shadow of an object under usual illumination or with an out of focus lens.

Gaussian blurring is generally used as a pre-processing step in computer vision domain to represent image global features at different scales. For example scale space representation in SIFT (Scale Invariant Feature Transform) technic.

Gaussian blur is obtained in general by convolving image using a kernel of Gaussian values. Blurring operation can be done in one single pass by convolution with a two dimensional Gaussian kernel.

However dividing this process into two passes is possible thanks to separable property of Gaussian blur operation and this second approach requires fewer operations in total.

Two one dimensional kernels are used per horizontal and vertical directions. Resulting effect is similar to the one obtained with a two dimensional kernel, however it's more efficient in computation time.

At discrete points, filter defined by a Gaussian kernel is subjected to sampling to obtain a discrete representation. The mentioned discretization points are in general corresponds to each pixel’s midpoint. Normally this decreases computation time however with small filter kernels, Gaussian function point sampling with fewer samples leads to error. To prevent this and maintain accuracy with a slight computation time, the Gaussian function integration over each pixel's area is used.

Noise in image is in general an aspect of electronic noise; it's a variation of color and brightness in image. it's not present in real image scene. Image noise is undesirable spurious or extraneous information in image, a faulty information added at capture time by image capturing product. In image processing noise removals is important and in most cases it’s removed with Gaussian function. As an example, Gaussian smoothing is in general used for edge detection as a preliminary step. Edge detection algorithms assume image as a perfect 2d signal without having any signal information. Yet in reality images have lots of noise. Edge detection algorithms are in general sensitive to noise. E.g. Laplacien filter.

Gaussian blurring before detection of image edges reduces noise and improves the edge detection algorithm performance. This is commonly referred to LoG filtering, Lapcacian of Gassian filtering

Fallowing captures show the effect of Gaussian blurring on edge detection. More smoothing leads to detection of fewer edges. See Figure 2.1, Figure 2.2, Figure 2.3, Figure 2.4 and Figure 2.5:

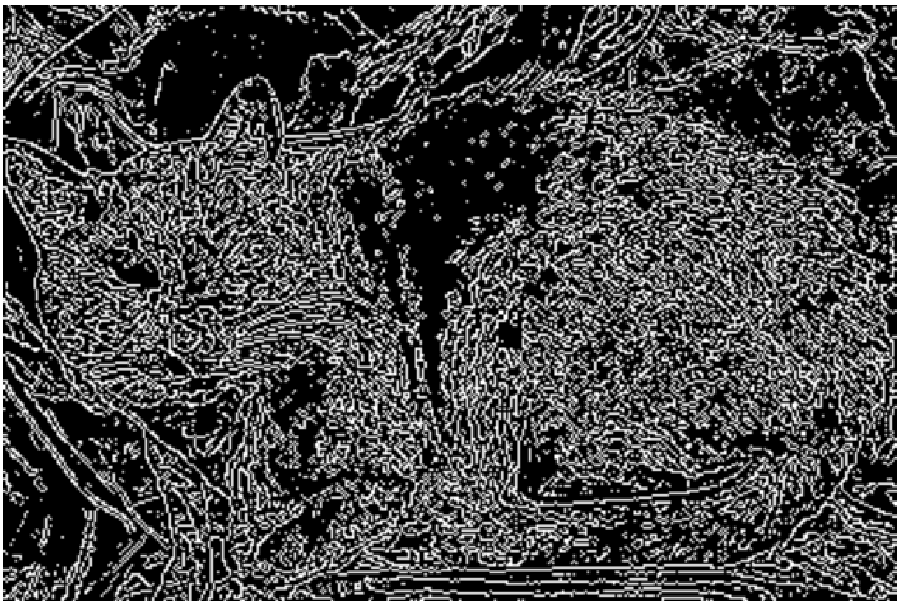


Figure 2.1 - Smoothing effect on edge detection (A)

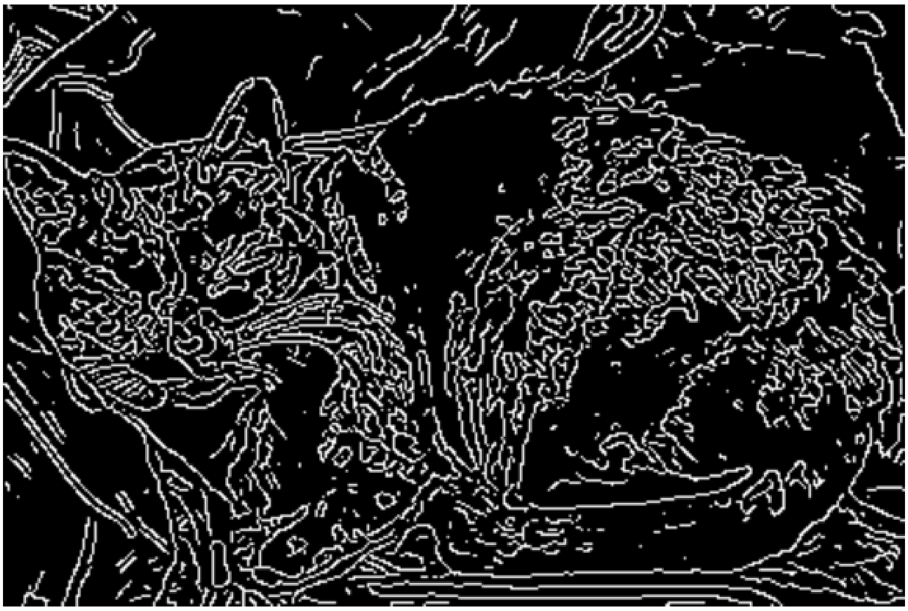


Figure 2.2- Smoothing effect on edge detection (B)

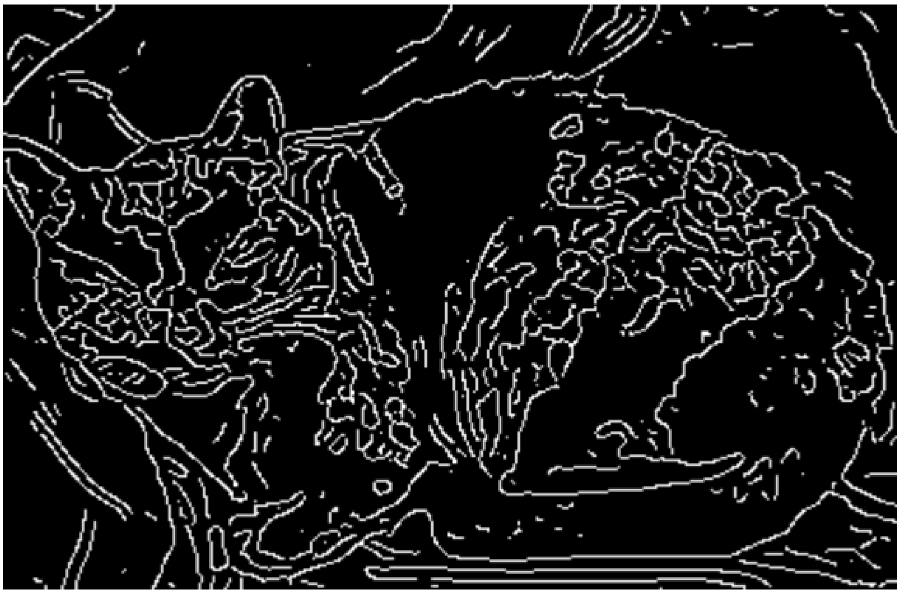


Figure 2.3 - Smoothing effect on edge detection (C)

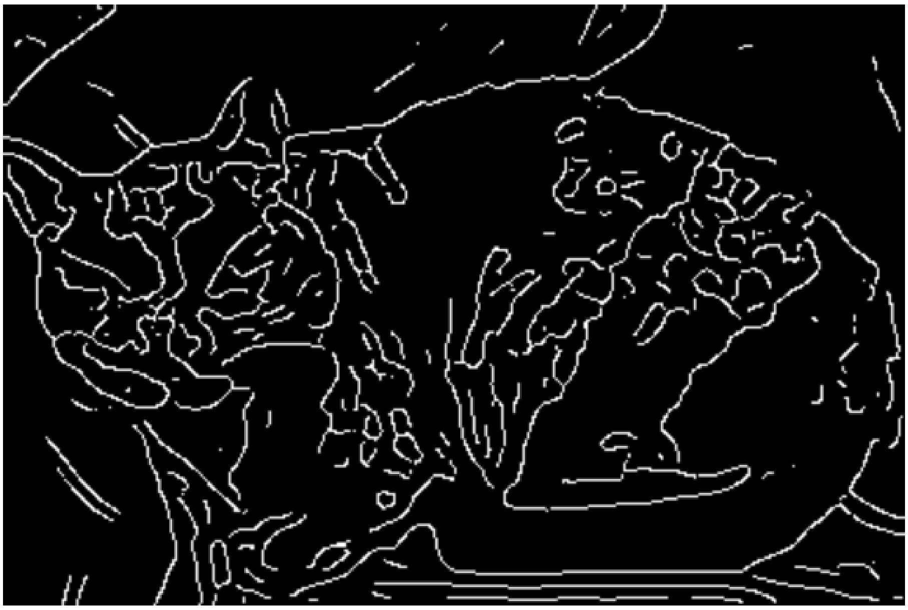


Figure 2.4 - Smoothing effect on edge detection (D)

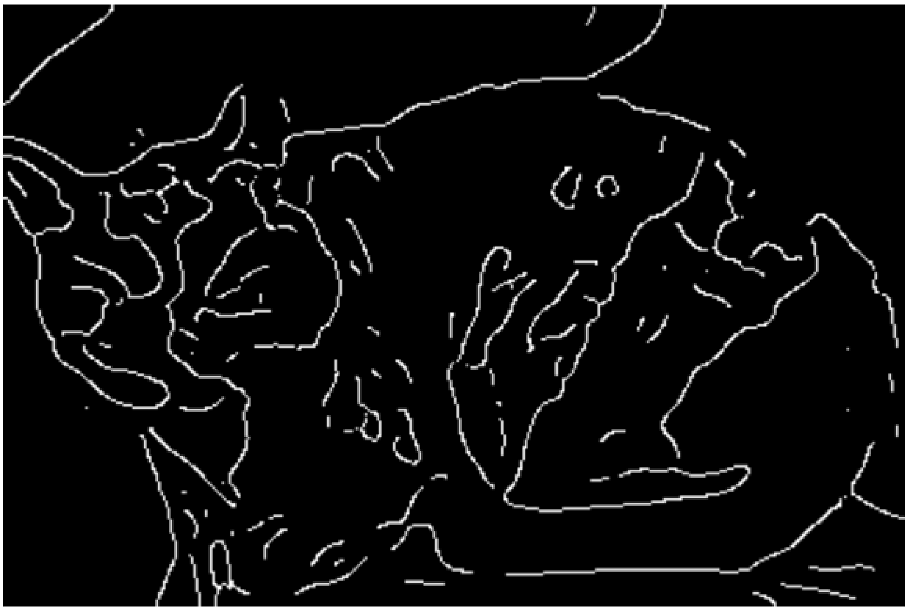


Figure 2.5 - Smoothing effect on edge detection (E)

Before edge detection, noise removal and scale space representation of treated image is achieved with Gaussian blurring.

2.2 Canny Edge Detection

There are lots of edge detection methods with different detection algorithms. The Canny operator is designed to be an optimal detector; its detection process considers other approaches as well. It treats image at gray scale level and transforms the treated image to an image showing the positions of intensity discontinuities.

The Canny algorithm operates in multiple steps. Firstly, using Gaussian blurring, the image is smoothed. Secondly a filter consisting of a simple 2D first derivative operator is applied over smooth image. This derivative filtering is the high first spatial derivative. It refines and concretizes image regions. In the gradient magnitude image of treated image, edges become highlighted in the form of rising ridges. The Canny algorithm then depicts pixels corresponding to the top of these rising ridges. Pixels which are not on the ridge top are marked with zero value. This achieved with a process known as non-maximum suppression. This operations lead to the extraction of thin lines in the output image. Edge extraction process is achieved with a well-known method, namely hysteresis. Hysteresis is controlled with two thresholds, lower and upper thresholds: T1 and T2 with T1 greater than T2. Extraction process starts on a pixel point which is on a ridge and with a value higher than T1. Process than continuous in both directions out from the pixel point up to that the ridge falls below T2. Hysteresis ensures that edges are not broken into multiple fragments due to image noise.

Canny operator can be tuned with tree parameters: Smoothing level Gaussian kernel width and upper and lower thresholds used by hysteresis. Increasing Gaussian kernel width reduces sensitivity of Canny against noise, leads to loose of finer details in the image. Edge localization errors will also be increased when Gaussian width is increased. On the other hand, the optimal way for hysteresis parameters is to set the higher threshold to a high value and the lower threshold to a low value. Setting the lower threshold to a high value leads to noisy edges to be broken. Setting the upper threshold to a low value increases spurious and undesirable edge fragments.

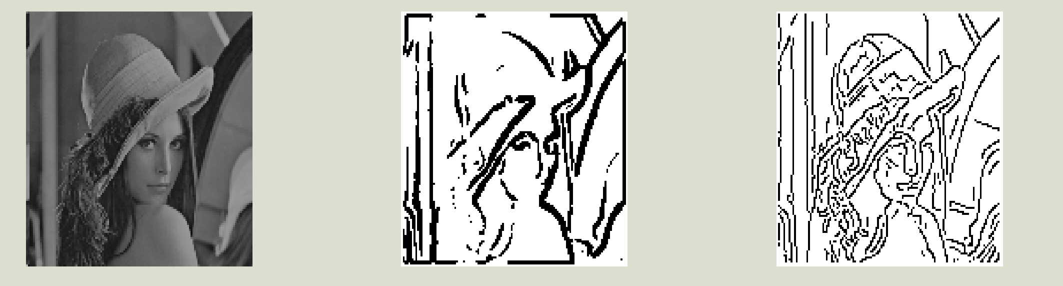


Figure 2.6 – Canny Edge Detection Algorithm

Edges are important global features and are fundamental features in computer vision. Edges consist of strong intensity contrasts, change in intensity from one pixel to the one adjacent. And edge detection significantly removes useless data, while keeping useful structural global image properties.

The Canny edge detector is adopted as the optimal way by lots of significant studies. While Canny edge detector was developed, many of the existing edge detectors were considered. A part from these existing edge detection algorithms, a list of criteria is fallowed and achieved in Canny edge detector.

As a criterion, achieving low error rate is the most important one for Canny. Extraction of real edges and discarding non edges from the image with a feasible error rate is the most important criterion for Canny.

The second important criterion is the localization of edge points. The distance between edge pixels in the detector output image should be consistent with corresponding ones in input image.

The third criterion is having only one response per single edge. The first two criterions are not perfect for achieving this problem, so a special algorithm is developed and integrated with detector to ensure this criterion.

Satisfying these criterions, the Canny detector first applies Gaussian smoothing to remove image noise.

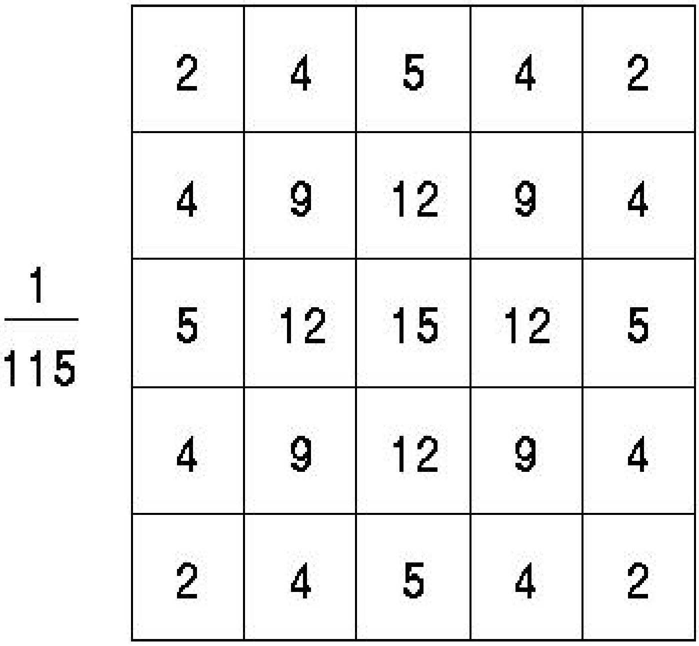
Secondly applying a spatial derivative filter, finds significant ridges in gradient image. Then the Canny algorithm then tracks these ridges and removes pixels that are not at the maximum of ridges. This is known as non-maximum suppression.

After then, gradient image is subjected to hysteresis operation. Hysteresis tracks along the remaining pixels located on ridges and have not yet been removed. Lower and Upper thresholds are used with Hysteresis operation. If the magnitude of the considered pixel is below the first threshold (higher threshold), then it's discarded; otherwise if it's greater, then it's marked as an edge making pixel. If the magnitude of the considered pixel is between the higher and lower thresholds then it's discarded unless there is a path from considered pixel to a pixel with a gradient value above the high threshold. The fallowing example with steps better clarifies the Canny algorithm ant its effects.

2.2.1 Step 1: Noise Removal

Canny edge detection algorithm imposes some consecutive steps. The first step is Gaussian blurring. Before edge detection and localization, image noise is removed with Gaussian filter. The Gaussian function can be modeled and filtering can be computed using a simple mask. This simple mask is usually much smaller than the image size. The simple mask slides over the image, computes contributions of pixels which are corresponding to the simple mask area at a time. If the width of Gaussian mask is large, then detectors sensitivity to noise lowers. Edge localization errors slightly increase, while Gaussian width is increasing.

A Gaussian mask may be determined as shown below:



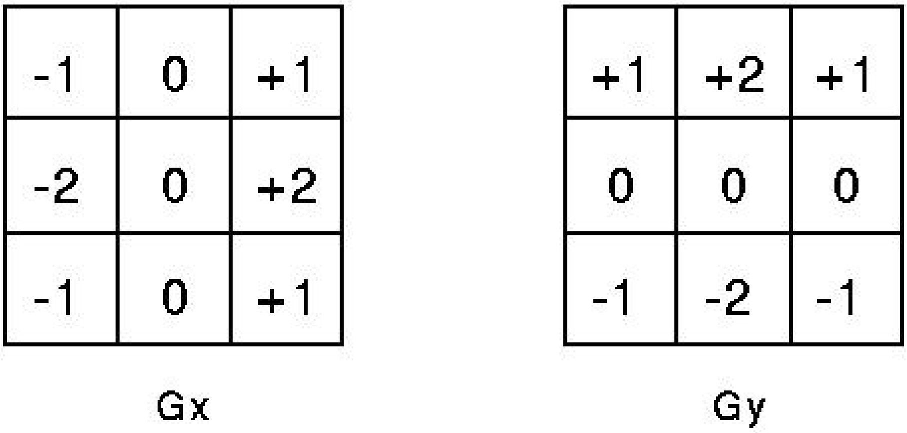
$$\frac{1}{115}\begin{array}{|c|c|c|c|c|}\hline 2 & 4 & 5 & 4 & 2 \\ \hline 4 & 9 & 12 & 9 & 4 \\ \hline 5 & 12 & 15 & 12 & 5 \\ \hline 4 & 9 & 12 & 9 & 4 \\ \hline 2 & 4 & 5 & 4 & 2 \\ \hline \end{array}$$

Figure 2.7 - Discrete approximation to Gaussian function with σ=1.4

2.2.2 Step 2: Gradient Image Representation

After noise removal with Gaussian smoothing, the next step is taking the gradient of the image to find edge strengths. The 2D spatial gradient measurement on the image is achieved with the Sobel operator. Then, the edge strength (absolute gradient magnitude) at each pixel is approximately found. Sobel operator consists of two 3x3 convolution masks. First one measures the gradient response in y-direction and the other one measures gradient response in x-direction.

These convolution masks can be defined as fallowing:



-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

Figure 2.8 - Sobel operators

The magnitude (edge strength) of the gradient is then approximated using the formula:

$$|G| = |Gx| + |Gy|$$

2.2.3 Step 3: Edge Direction Detection

Using the gradient values in the x and y directions, the edge direction can be computed easily. Below formula is used for finding edge direction:

theta = invtan (Gy / Gx)

2.2.4 Step 4: Edge Orientation Detection, Classification of Edge Directions

Once en edge direction is known, then we relate it to a direction which can be plotted on an image.

To illustrate this, we can consider a 5x5 image aligned as follows:

• • • • •

• • • • •

• • **p** • •

• • • • •

• • • • •

Then, this making relation idea can be seen by looking at pixel "p", there are only eight possible directions around the pixel "p": 0 and 180 degrees (in the horizontal direction), 45 and 225 degrees (along the positive and negative diagonal), 90 and 270 degrees (in the vertical directions), 135 and 315 degrees (along the negative diagonal). Opposite direction pairs makes a single direction. So there are 4 directions 0, 45, 90 and 135 degrees. The edge orientation will be related to one of these four orientations depending on which direction it is close to. We can imagine this as a semicircle divided into 5 regions, like it's seen in Figure below.

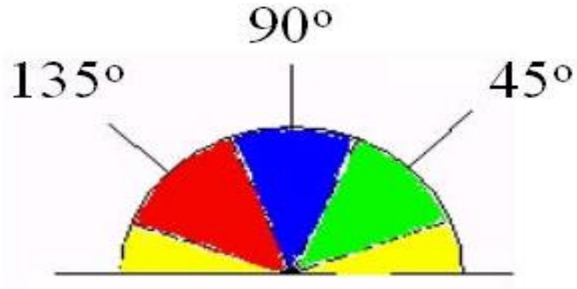


Figure 2.9 – Edge orientation ranges

Thus, considering Figure 2.9, edge orientation of an edge with a direction corresponding to yellow area (0-22.5 degrees and 157.5-180 degrees) is assumed as 0 degrees. Edge orientation of an edge with a direction corresponding to green area (22.5 67.5 degrees) is assumed as 45 degrees orientation. Edge orientation of an edge with a direction corresponding to blue area (67.5-112.5 degrees) is assumed as 90 degrees orientation. Edge orientation of an edge with a direction corresponding to red area (112.5-157.5 degrees) is assumed as 135 degrees orientation.

2.2.5 Step 5: Non-maximum Suppression

Edge orientation extraction step is followed by non-maximum suppression (NMS) step. NMS suppress any pixel value that is not considered to be an edge that is not located on any ridge. NMS is used to trace a thin line along the edge in the edge direction.

2.2.6 Step 6: Hysteresis

Hysteresis is the final step, used as a means of discarding streaking. Streaking notion is the split up of an edge segment due to operator output fluctuating below and above the used threshold. Hysteresis uses two thresholds; upper and lower thresholds (T1 and T2 respectively).

Any pixel with gradient value higher than T1 is assumed as an edge making pixel. Then any pixel connected to this edge pixel with a gradient value higher than T2 and lower than T1 selected as edge making pixel as well. Others are discarded. Hysteresis merges contours to be meaningful edges and discards meaningless contours as well.

2.3 SIFT: Scale Invariant Feature Transform

SIFT is a method to detect and extract good local particularities, key points which are invariant to image affine transformations, changes in rotation and image scale. These local particularities are robust to the changes in illumination, noise and view point of 30 degrees. These are the points which are distinctive and safe to be re-identified in different scenes. SIFT points are used in many computer vision domains. Especially, it is useful for motion tracking, 3D reconstruction, image auto stitching, object recognition, object identification etc…

SIFT is presented for the first time in 1999 by David Lowe, since then this method is a good reference.

This method can be described in 4 steps:

- Local Extrama Detection
- Local Extremum Localisation
- Orientation Affectation

- Key Point Descriptor

2.3.1 Local Extrama Detection

In this step, candidate interest points (candidate key points) are extracted. In order to do that, initially, image is filtered with Gaussian Filter.

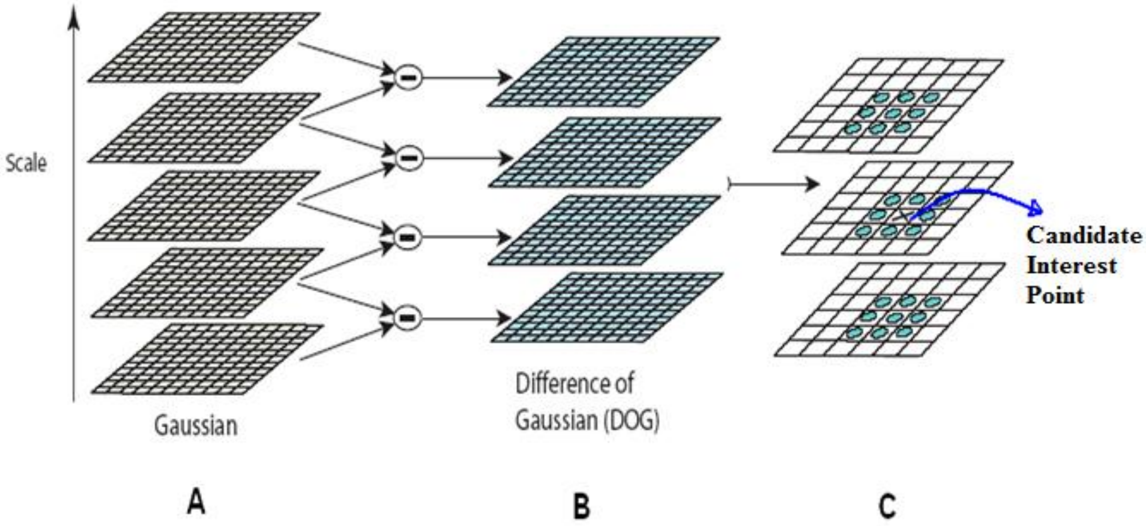


Figure 2.10 – Gaussian Scale Space

A spatial scale is obtained, Figure 2.10: A, and then with the well-known function of Difference of Gaussian, DoG, the difference images are obtained, Figure 2.10: B. Then, candidate interest points are extracted by maxima-minima detection over the scale space: Each point is compared to its 26 neighborhood points and checked if it is the maximum or the minimum. If the point is a maxima-minima, it is kept as a candidate interest point.

2.3.2 Local Extremum Localization

Local Extrema Detection produces a lot of points and some of them are not stable.

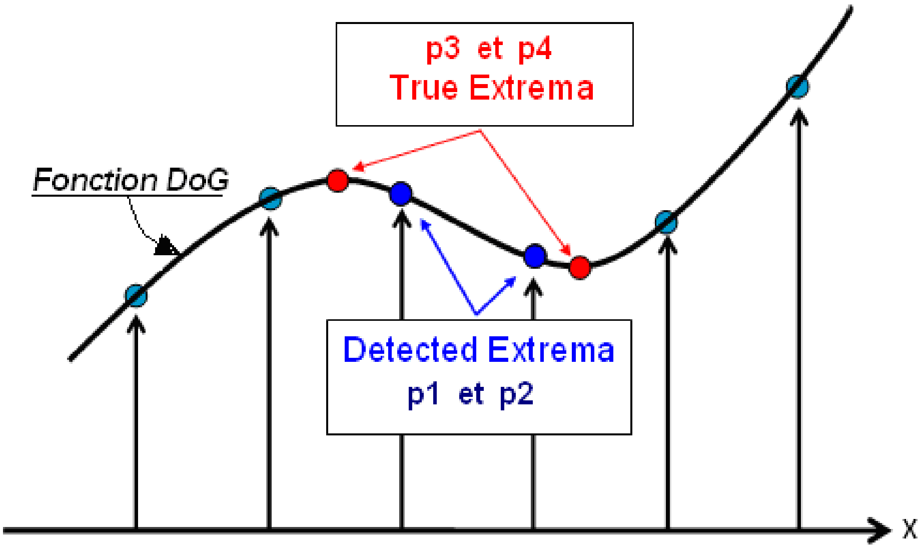


Figure 2.11 - Gaussian sampling

In reality, the function, DoG, finds us a sampling of spatial scale. This function is continuous and its derivate also. As it is illustrated in Figure 2.11, p1 and p2 are found as the local maxima-minimas, but if DoG is considered in real space rather than discrete space, the true extremas can be localized as p3 and p4. Lowe is used the studies of Brown to do this.

In the neighborhood of a candidate point, for example p1, the Taylor series of DoG function is defined. It is a second order polynomial:

$$D(\vec{x}) = D + \frac{\partial D^T}{\partial \vec{x}} \vec{x} + \frac{1}{2} \vec{x}^T \frac{\partial^2 D^T}{\partial \vec{x}^2} \vec{x} \text{, in which}$$

$\frac{\partial D}{\partial \vec{x}}$ is the Gradient, giving $\dot{\vec{x}}$

$\frac{\partial^2 D}{\partial \vec{x}^2}$ is an Hessian matrix, giving $\ddot{\vec{x}}$

$\vec{x} = (x, y, \sigma)$

By minimizing this polynomial (deriving and then finding the annulation point), we can find the estimated location $\hat{x}$ of $\vec{x}$:

$$\hat{x} = -\frac{\partial^2 D}{\partial \vec{x}^2}^{-1} \frac{\partial D}{\partial \vec{x}}$$

By using this polynomial and this estimated point, $\hat{x}$, we can decide whether a point is proper. If $\hat{x} > 0.5$, this extremum is near to another extremum and it is not proper, not safe to be selected as a distinctive point in matching time. In this case this point (p1) is eliminated and the interpolation continues with another candidate point. The value of the polynomial on a proper candidate point is useful to eliminate the points which are sensitive to noise. 0.03 is the contrast threshold used by Lowe to eliminate noise sensitive candidate points: if $D(\hat{x})$ <0.03 then this point is eliminated also.

A problem of DoG function is that it produces many candidate points on rectilinear low level image lines. As it is illustrated in Figure 2.12, a point is difficult to match with rectilinear points.

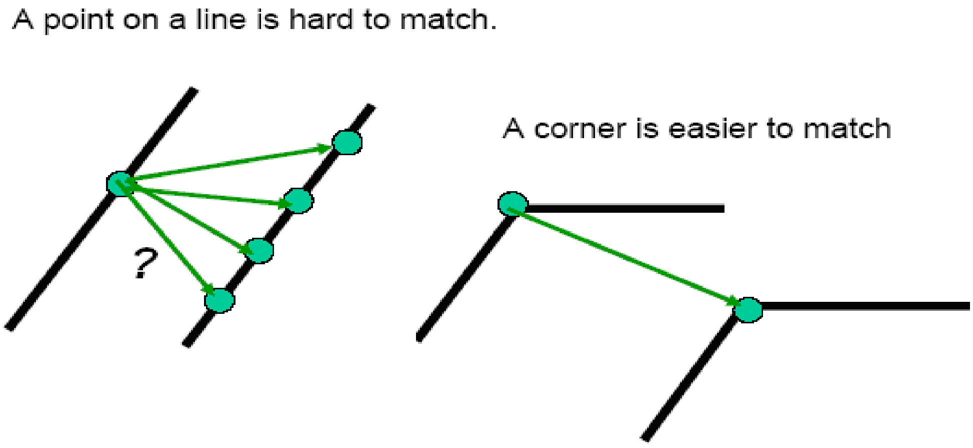

A point on a line is hard to match.
?
A corner is easier to match

Figure 2.12 - Matching problem

To handle this problem, the method known as Principal Curvature Analyze is used. According to this analyze, the eigen values of the Hessien Matrix of the $D(\hat{x})$ polynomial are used to establish a relation between two principal curvatures. If the ratio between two principle curvatures is bigger than 10, the candidate point is not proper, so it is eliminated.

By applying this criterion, the best points among multitudinous points extracted by DoG are selected and these are the interest points from now on.

2.3.3 Orientation Affectation

Based on low level local image properties, we can assign an orientation to the interest point and represent its descriptor relative to this orientation.

In order to do that, the gradient magnitudes in the neighborhood of the interest point are calculated and then with the Arctan function, the angle values which give the orientations are obtained. A Gaussian circular weighting is also applied. Then, the most significant orientation is considered as the orientation of the descriptor. Some other significant orientations are also kept with the principal one also.

By assigning a relative orientation, descriptors which are invariant to the image rotations will be obtained.

2.3.4 Key Point Descriptor

In the Figure 2.13, key point descriptor calculation is illustrated. Interest points are characterized by their surrounding low level image information. So far, every interest point is represented by 4 factors, (x-y coordinates, scale factor and orientation info). In the concerned scale image, over a 16x16 window centered at the interest point, the gradient values in 8 directions are being calculated by applying a Gaussian circular weighting to make the gradients near the key point more contributive.

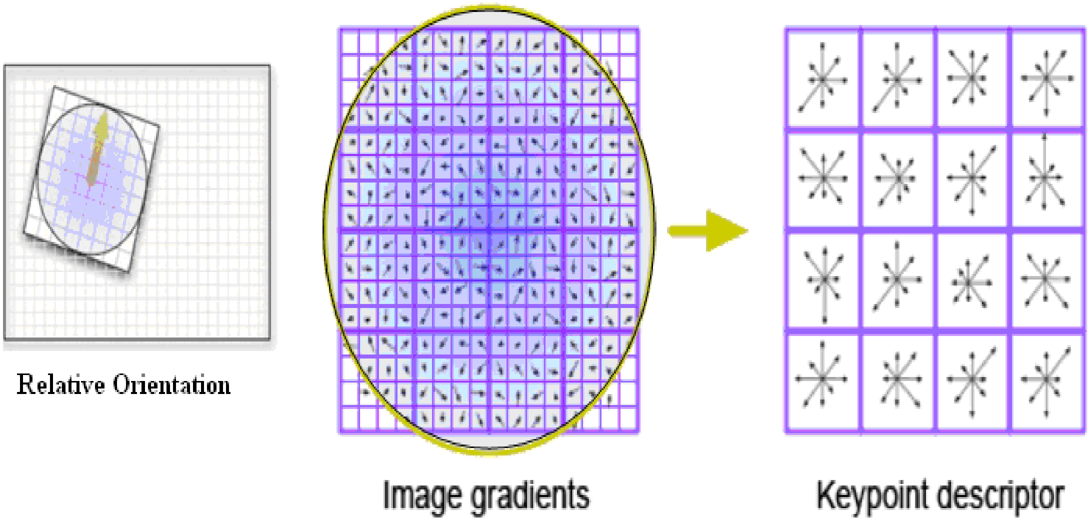


Figure 2.13 - SiFT Descriptor

In 4x4 sub windows, the total contributions in 8 directions of the weighted gradient values are represented. This produces a keypoint descriptor of 4x4x8 = 128 elements which is used as a vector of 128 elements in matching operations.

2.4 SURF: Speeded Up Robust Features

SURF is a recent method published in 2006 by Bay et al. This is a SiFT-like method which is much faster than SiFT and gives approximately the same results with SiFT.

SURF can be presented in two steps:

Detection: Interest points are selected on distinctive locations like corner, blob, T-Junction.
Its descriptor is stable and SURF can detect the same interest points under different visual conditions.

Description: The neighborhood of each interest point is represented by a characteristic vector. The obtained descriptor is distinctive and robust against noise, detection errors, geometric and photometric deformations.

SiFT is one of the best methods but its applications are not fast enough.: the description of a 1000x700 pix. image causes the extraction of 4000 points approximately. This task takes 6 seconds. As SiFT generates elevated dimensional descriptors (128), matching level takes more time. SiFT is an ideal method except that it is not suitable for some applications necessitating rapidity. About this fact, Lowe claims in Lowe04 that it is necessary to develop new methods like SiFT, appropriated to the nature of the desired application types. SiFT, in fact, shows us the most possible invariant, robust and distinctive description. Using SiFT, it is possible to develop varied applications after the adaptation to the desired application. SURF is a method derived from SiFT and some popular SiFT-like methods. SURF is fast enough at matching time, which is a problem for SiFT at application level, and original enough as a new method.

The SURF detector is based on a Hessien matrix with an approximation by using some box filters. The integral images are used to diminish the calculation complexity. In the neighborhood of the extremum point, the Haar Wawelet distributions are described.

2.4.1 Integral Images

The integral image defined at the point (x, y) is the sum of values of all pixels which are above and left of the point (x, y). It is easy to calculate the integral images effectively and their use provides some facilities. For example, the sum of pixel values over a sub region of image can be achieved in 4 simple linear operations rather than accumulation of all values.

2.4.2 Hessien Detector of SURF

The detector of SURF is based on a Hessien matrix. Given the point X=(x,y) of the image I. The matrix H(X, σ) at the point X and at the scale σ is defined as fallow :

$$H(x,\sigma) = \begin{pmatrix} L_{xx}(x,\sigma) & L_{xy}(x,\sigma) \\ L_{xy}(x,\sigma) & L_{yy}(x,\sigma) \end{pmatrix}$$

$L_{xx}(x,\sigma)$ is the convolution of image with $\frac{\partial^2}{\partial x^2} g(\sigma)$, the second order Gaussian derivation. $L_{xy}(x,\sigma)$, $L_{yy}(x,\sigma)$ are similar also. And SURF introduces approximations of these operators.

The use of Gaussian function is optimal for spatial scaling analysis. In reality, by using Gaussian function, the images are needed to be discretized. In sampling step, as it is illustrated in Figure 2.11, the risk of aliasing is encountered. Gaussian filter of SiFT is used as an approximation of “Scale Normalized Laplacien of Gaussian function”.

Rather than using an approximated Gaussian filter, in SURF06, this phenomena is modeled by using box filters which are the approximations of second order derivatives of Gaussian function. In this way, spatial scaling and discretization of image with Gaussian filter are not necessary.

Thanks to the facilities of integral images, this approximation is independent from the size of used images.

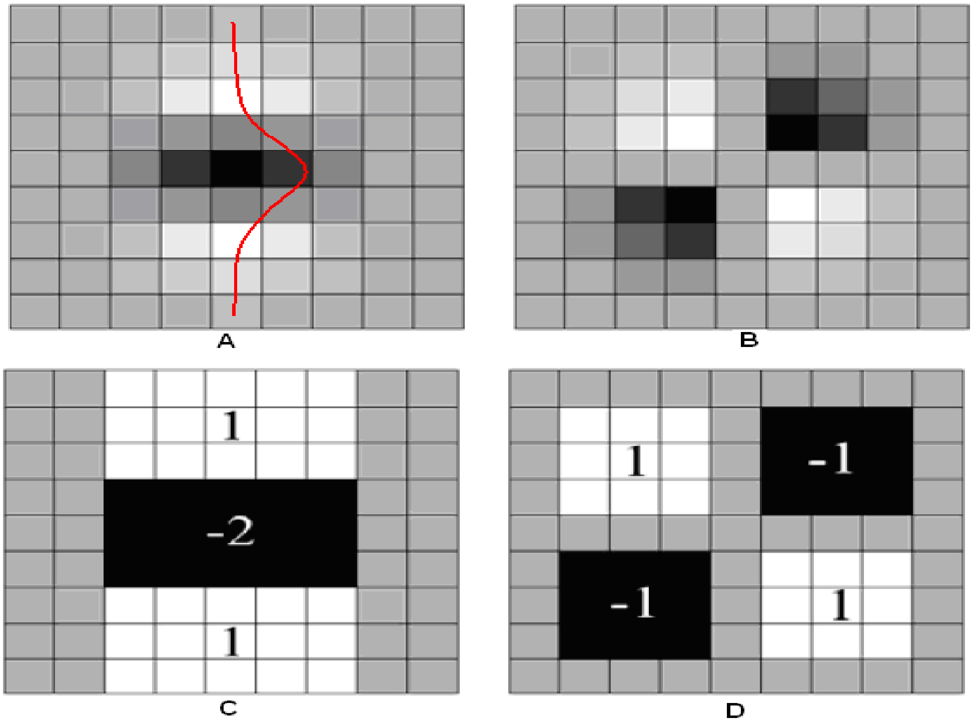


Figure 2.14 - Gaussian function and box filters

The Figure 2.14 represents these approximations. Figure 2.14(A) illustrates the convolution of the image with second order derivative of the Gaussian function $L_{yy}(x,\sigma)$. Figure 2.14(B) is the one of $L_{xy}(x,\sigma)$. The obtained results in Figure 2.14(A) and Figure 2.14(B) are approximately modeled by box filters shown in Figure 2.14(C) and Figure 2.14(D) : Gray regions have 0 value, white regions have 1 and black regions have -1 or -2 values. When $\sigma = 1.2$, the relation between Gaussian derivatives and these 9x9 box filter approximations, D_{xx}, D_{xy} and D_{yy} are defined as fallowing:

$$\frac{|L_{xy}(1.2)|_F}{|L_{xx}(1.2)|_F} \cong 0.9 \frac{|D_{xy}(9)|_F}{|D_{xx}(9)|_F}, \qquad |X|_F \text{ is the Frobenius Norm}$$

And the determinant of Hessian matrix is being calculated approximately as fallowing:

$$\det(H_{approx}) = D_{xx} D_{xy} - (0.9 D_{xy})^2$$

As in Lowe04, Spatial scaling is generated as a pyramid containing different level of filtered images. Different scales of the same octave are obtained via convolution of preceding image and octaves are obtained by sampling of image. In SURF06, the use of box filters with integral images contributes to avoid doing all of these scaling and sampling operations. It is possible to apply a lot of box filters directly with different sizes. Moreover, all of the box filters can be applied in parallel at the same time. Whereas in SiFT, it is a must to wait the generation of each spatial scale to be able to calculate the next one, so SiFT Gaussian extraction is a serial treatment. In SURF06, spatial scaling is done by increasing the sizes of box filters, rather than changing the size of images as in SiFT.

SURF06 uses box filters with sizes 9x9, 15x15, 21x21 and 27x27. Different size of filters are the approximations of second order Gaussian derivatives with different σ values: for the filter of size 27x27 new scaling factor determined as $(27/9)\sigma$ = 3σ =3x1.2 = 3.6. As the Frobenius norm remains constant for box filters with different sizes, the property of the invariance to scale the changes in SiFT is provided in SURF.

To detect and localize the interest points among spatial scales, non extremum elimination over 26 neighbor points (9 elements in the upper scale, 9 in the lower scale and 8 surrounding the keypoint in the same scale) is being used as in SiFT. For the localization of extremums, the determinant of the approximated Hessien matrix is interpolated at the candidate extremum point as in the SiFT.

2.4.3 SURF Orientation Affectation

To provide rotation invariance, a reproducible orientation is assigned to the extremum point: First, over an extremum point centered circular region of 6s radius, Figure 2.15, wavelet responses in x and y directions are calculated. In order to do this, box filters of size 4s are used. As integral images are used, this task is done easily: the differences between the sum of the elements in the black region and the sum of the elements in white region of the box, S2-S1 (Figure 2.15 Haar wavelet response in y direction). The regions S1 and S2 are represented with 6 points (A, B, C, D, E, F). Their differences can be calculated in 6 simple operations, since integral images are used.

$$S2-S1 = (A + C - B - D) - (D + F - C - E) = A + F + 2(C-D) - B - E.$$

After calculating the wavelet responses over the circular region of 6s, the results are weighted with a gaussian function to make the ones near the extremum point more effective. This increases the stability of the descriptor. The weighted responses in x and y directions are represented as vectors which are oriented in x and y axes, dx and dy. Over the region of 6s, a slice of 60 degree is selected, Fig 2.2.2, and this slice is turned 360 degree slowly, e.g. 10 degree per 10 degree. Meanwhile, the vectors remaining inside the slice are summed as $\sum \vec{d}_x + \sum \vec{d}_y$, and the resulting vectors orientation is kept for that position of the slice. The biggest vectors orientation is admitted as the orientation of the extremum point.

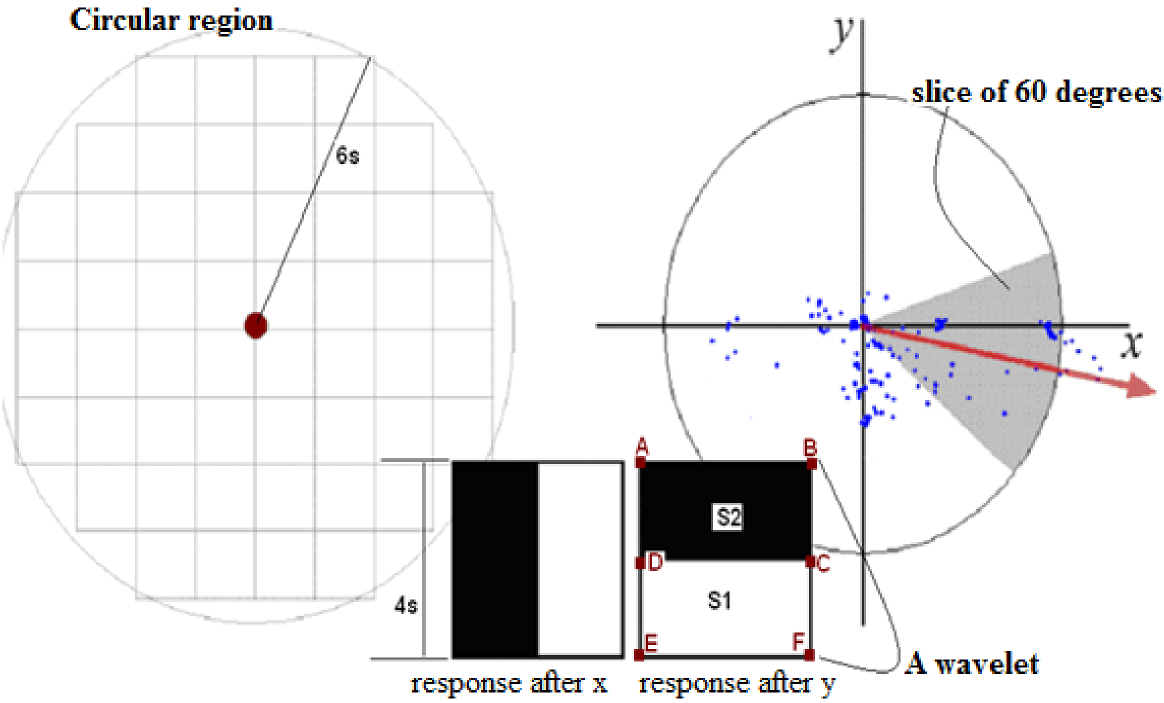


Figure 2.15 - SURF Orientation

2.4.4 SURF Descriptor

Firstly, a square region of 20s, centered at interest point, is determined and rotated to the orientation of the interest point Figure 2.16. This square region is divided into 16 sub square regions of 5s x 5s.

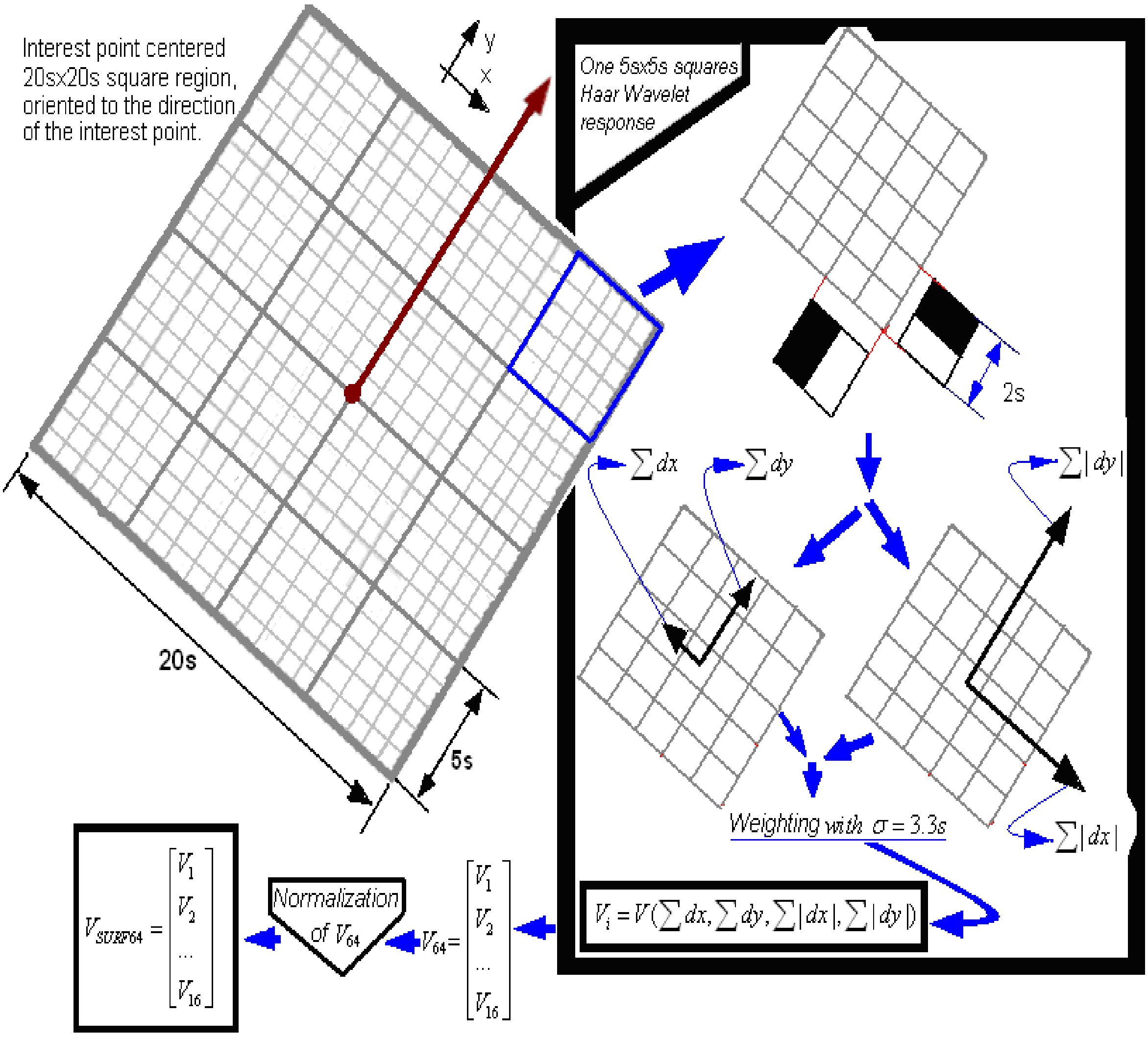

Interest point centered 20sx20s square region, oriented to the direction of the interest point.
y
x
One 5sx5s squares Haar Wavelet response
2s
20s
5s
$\sum dx$
$\sum dy$
$\sum |dy|$
$\sum |dx|$
Weighting with $\sigma = 3.3s$
$V_i = V(\sum dx, \sum dy, \sum |dx|, \sum |dy|)$
$V_{64} = \begin{bmatrix} V_1 \\ V_2 \\ \ldots \\ V_{16} \end{bmatrix}$
Normalization of V_{64}
$V_{SURF64} = \begin{bmatrix} V_1 \\ V_2 \\ \ldots \\ V_{16} \end{bmatrix}$

Figure 2.16 - SURF Descriptor Extraction

For each sub region, Haar Wavelet responses (intensity changes) are calculated with box filters of size 2s. The sum of responses in x directions forms the resulting vector Σdx and in y direction Σdy. The absolute sum of the responses forming $\Sigma |dx|$ and $\Sigma |dy|$ vectors. For all of 16 regions these vectors are obtained and over all vectors a Gaussian weighting is applied to make the descriptor robust against geometrical deformations and localization errors.

By putting together the weighted vectors of each sub regions, a new vector of 4 elements are obtained: $V(\Sigma dx, \Sigma dy, \Sigma \mid dx \mid, \Sigma \mid dy \mid)$. As there are 16 sub regions, we have a resulting vector of 16x4 = 64 elements and after normalizing it, SURF descriptor is obtained.

2.5 Conclusion of SIFT and SURF

Haar responses in SURF have an aim to generate image information which is invariant to illumination changes. But it is not as successful as in SIFT. SURF is invariant against noise, image rotation changes, however it is weak against view point changes, SURF is not as good as SIFT but it provides 3 times faster description of images.

2.6 Object Category Recognition HMAX Model

The biologically inspired HMAX model was firstly proposed by Riesenhuber and Poggio, and lately revised by Serre et al., who introduced a learning step based on the extraction of lots of random patches. T. Sere et al. compared this model with their own SiFT based model and proved that SiFT is less adaptable to object category recognition. In the context of HMAX, we extracted a detailed referential algorithm of this method by analyzing the recent implementation of Jim Mutch. The algorithm is explained in detail with explicitly designed schemas.

2.6.1 About HMAX Implementation

In the schemas below that we generated, it is easy to understand how HMAX C2 features are extracted to the use of a classifier like NN or SVM. C2 features are generally used for object class recognition. In this context SiFT features have less success than HMAX C2 features.

In PART 1 of the algorithm from training images, a random image is selected. From this random image, RANDOM PATCH EXRACTION process extracts the PATCH_i. This extraction task is repeated for randomly selected 250 images from training images.

In PART 2 of the algorithm, for only one image, RANDOM PATCH EXTRACTION process extracts the structure PATCH_i which includes 4 patches of different sizes. For each size, there exist 4 patches of different orientations. RANDOM PATCH EXTRACTION process is applied 250 times to extract 250 PATCH_i structures.

In PART 3 of the algorithm, we have 4 classes of images. Training images, Background Training images, Test images and Background Test images. First, each class is handled individually: for all images of the same class, C2 SIMILUTUDE RESPONSE EXTRACTION process is applied. The results of all 4 classes are gathered in the same vector which is called C2 Features. Then these C2 features are used with a classifier.

PART 1

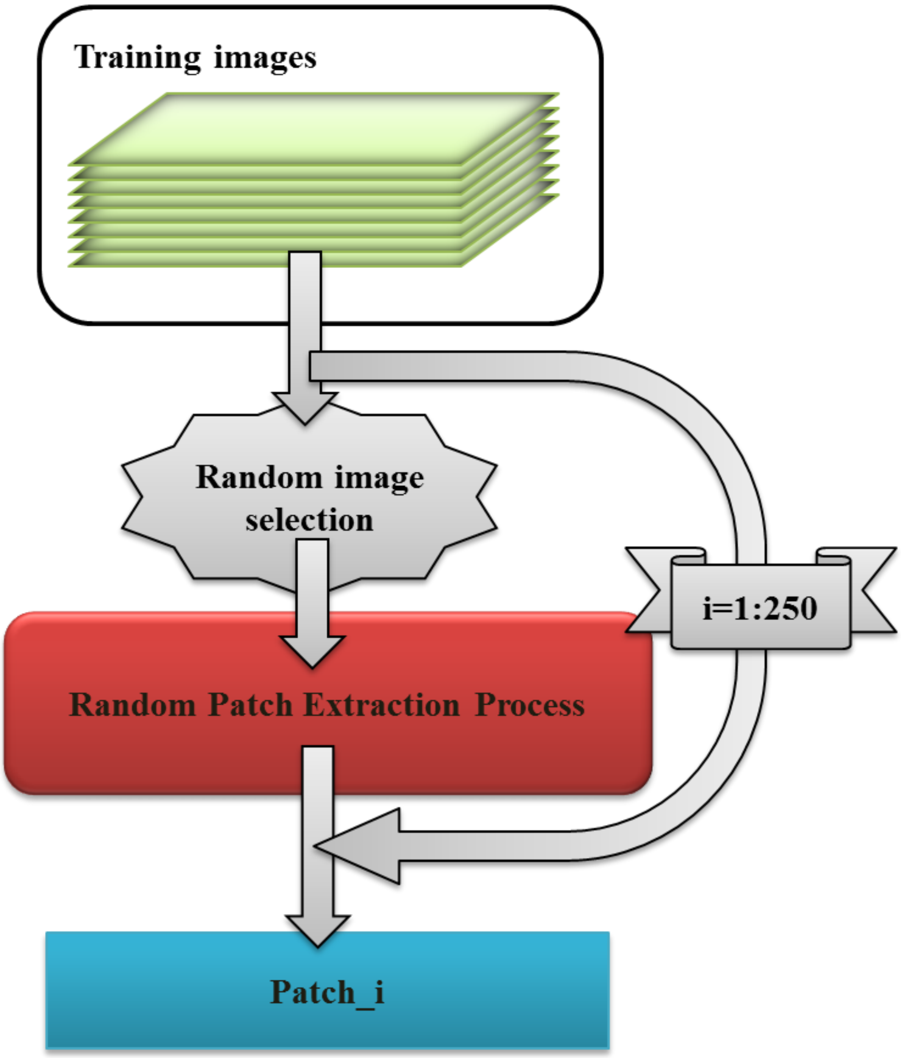


Figure 2.17 - First part of HMAX model

PART 2

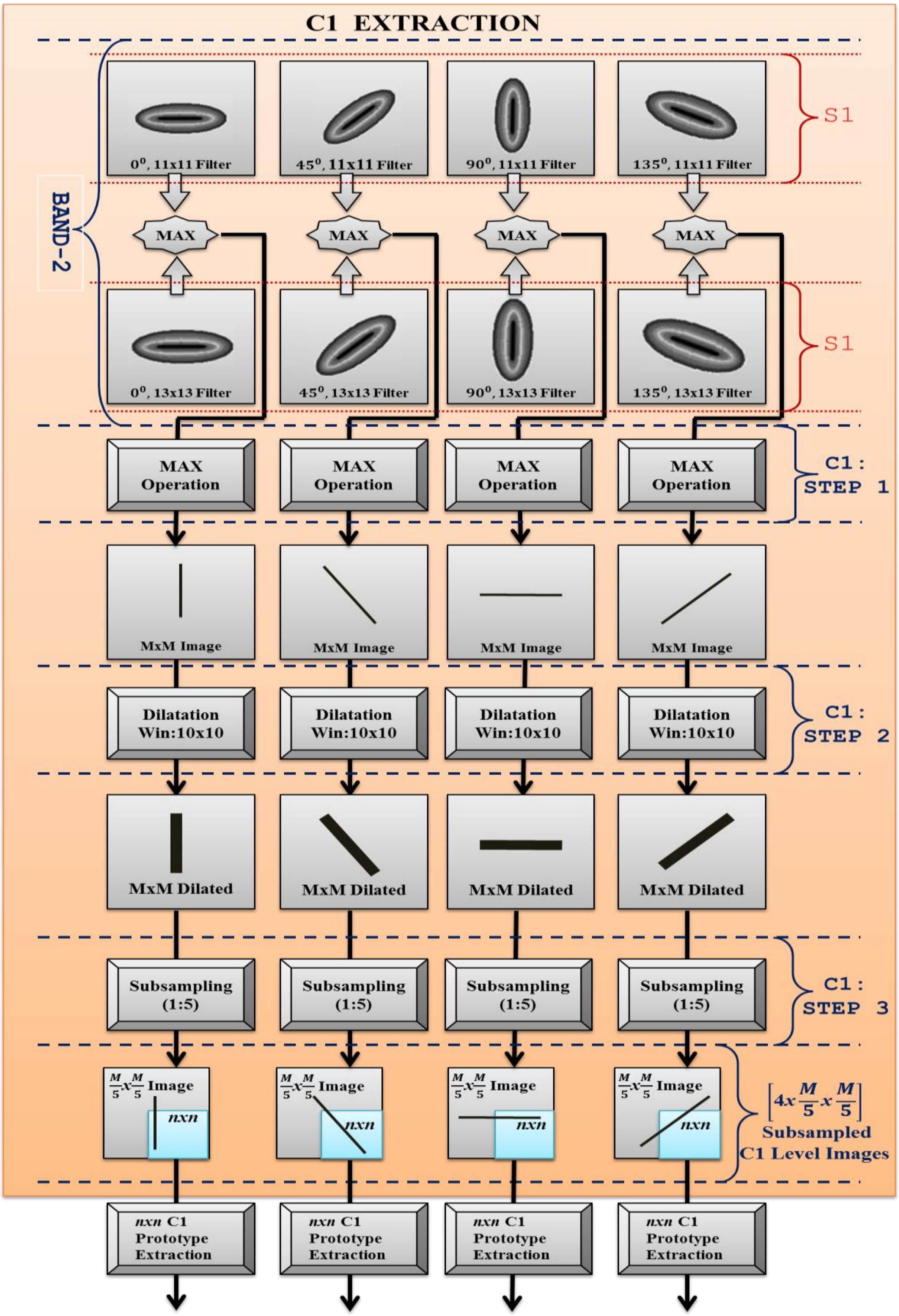


Figure 2.18 - C1 extraction

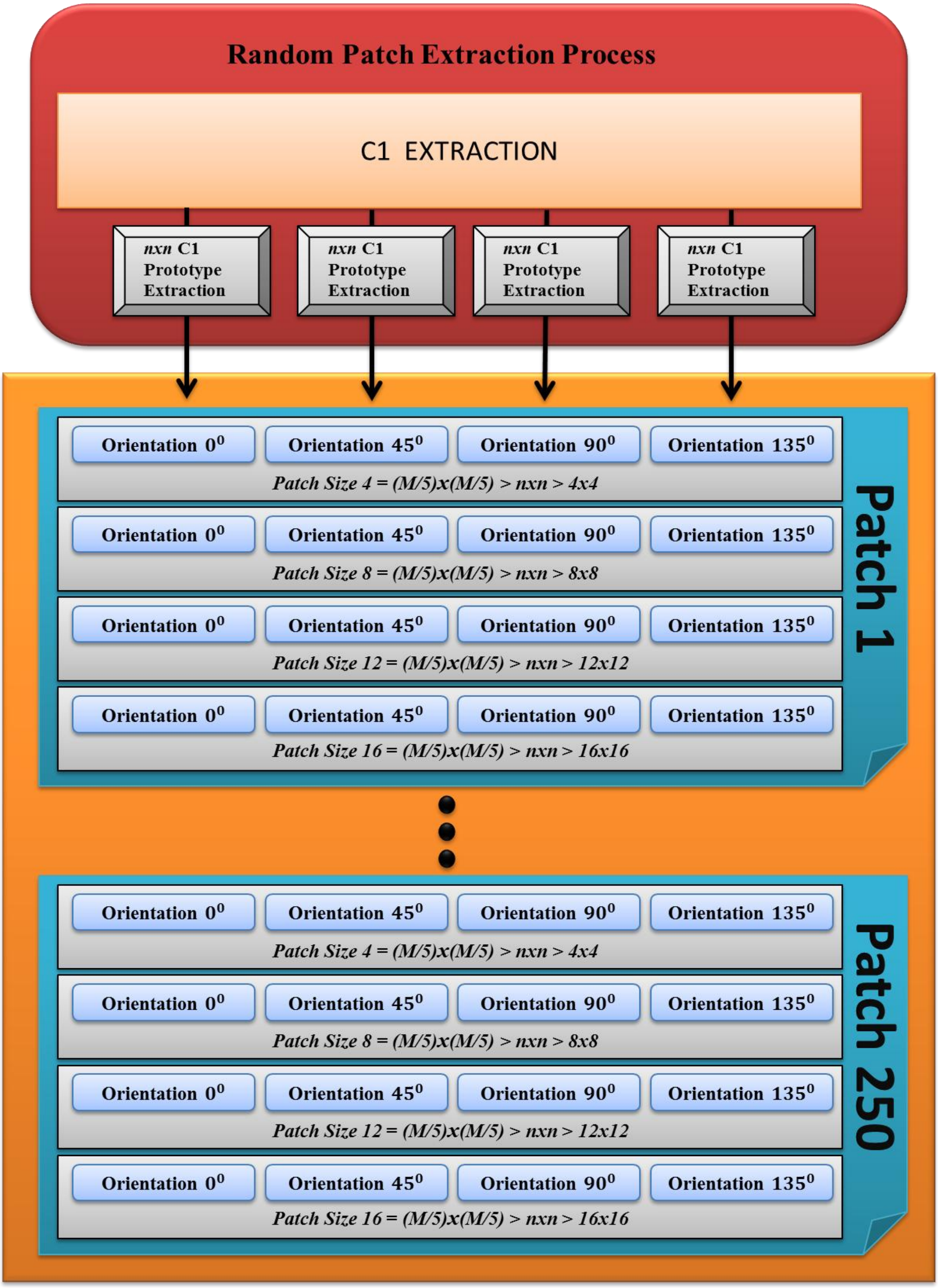


Figure 2.19 - C1 patch extraction

PART 3

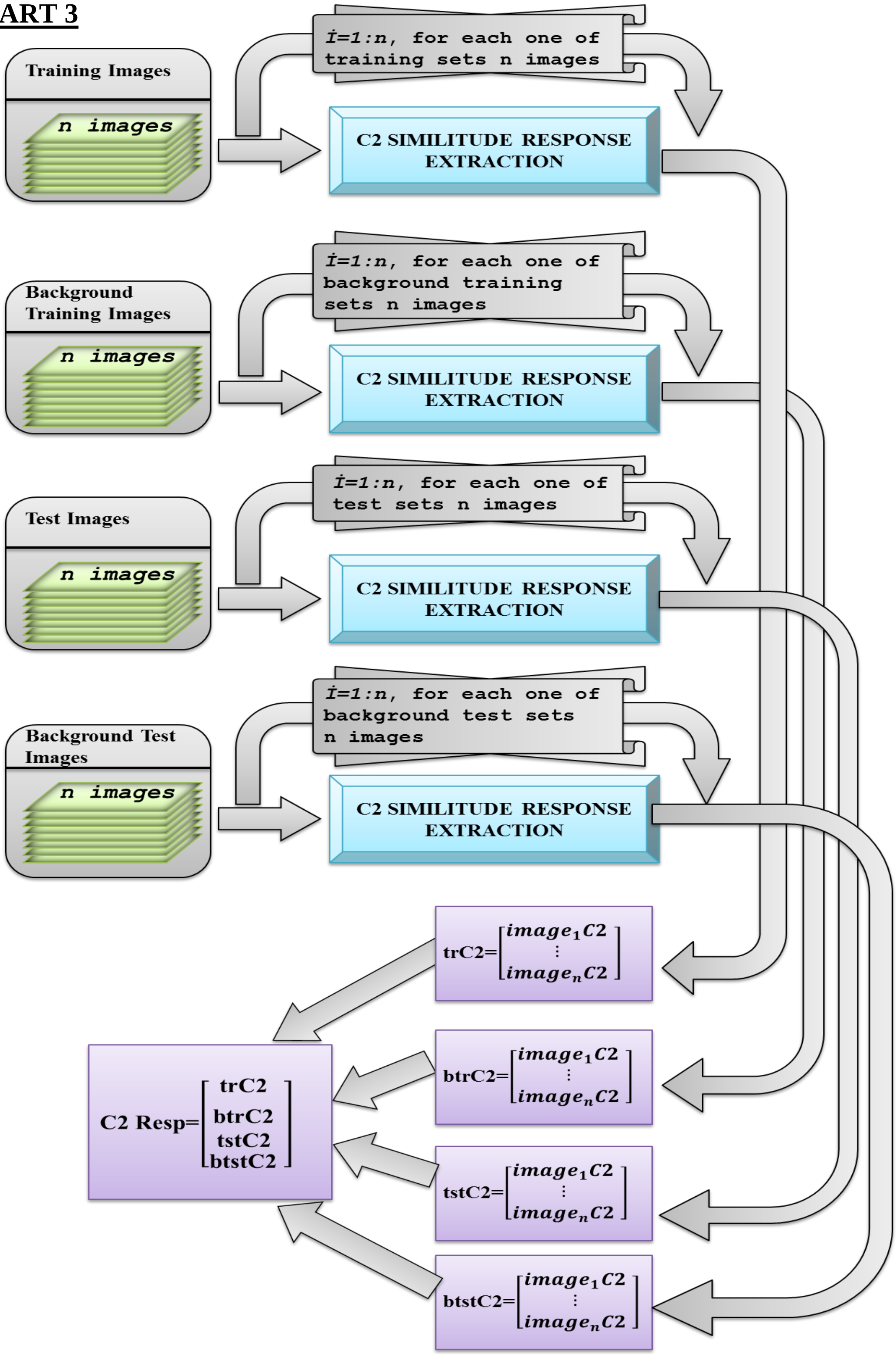
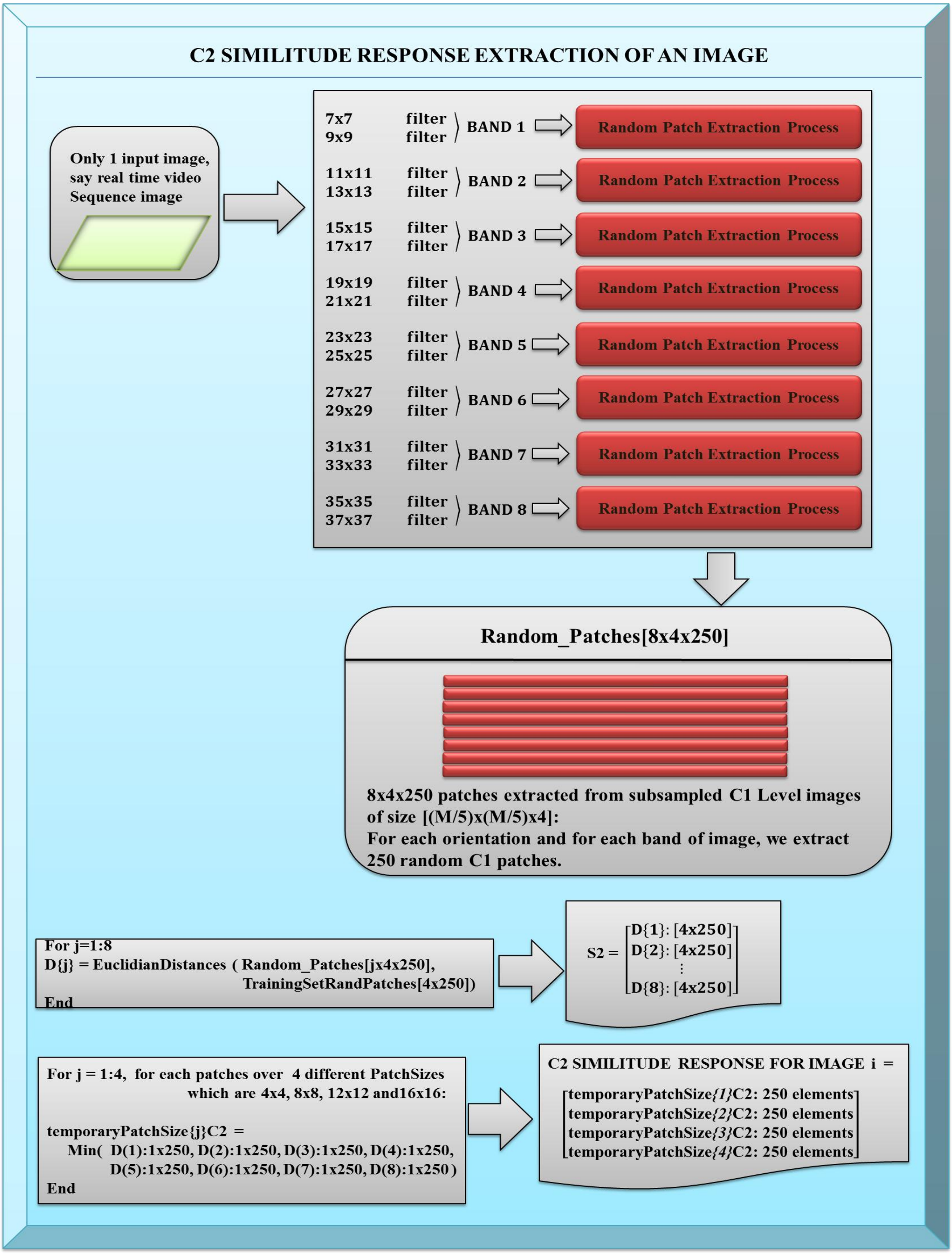

Training Images
n images
i=1:n, for each one of training sets n images
C2 SIMILITUDE RESPONSE EXTRACTION
Background Training Images
n images
i=1:n, for each one of background training sets n images
C2 SIMILITUDE RESPONSE EXTRACTION
Test Images
n images
i=1:n, for each one of test sets n images
C2 SIMILITUDE RESPONSE EXTRACTION
Background Test Images
n images
i=1:n, for each one of background test sets n images
C2 SIMILITUDE RESPONSE EXTRACTION
trC2=[image1C2 ⋮ imagenC2]
btrC2=[image1C2 ⋮ imagenC2]
tstC2=[image1C2 ⋮ imagenC2]
btstC2=[image1C2 ⋮ imagenC2]
C2 Resp=[trC2 btrC2 tstC2 btstC2]

Figure 2.20 - C2 response extraction


C2 SIMILITUDE RESPONSE EXTRACTION OF AN IMAGE
Only 1 input image, say real time video Sequence image
7x7 filter
9x9 filter
BAND 1
11x11 filter
13x13 filter
BAND 2
15x15 filter
17x17 filter
BAND 3
19x19 filter
21x21 filter
BAND 4
23x23 filter
25x25 filter
BAND 5
27x27 filter
29x29 filter
BAND 6
31x31 filter
33x33 filter
BAND 7
35x35 filter
37x37 filter
BAND 8
Random Patch Extraction Process
Random_Patches[8x4x250]
S2 =
D{1}: [4x250]
D{2}: [4x250]
⋮
D{8}: [4x250]
C2 SIMILITUDE RESPONSE FOR IMAGE i =
temporaryPatchSize{1}C2: 250 elements
temporaryPatchSize{2}C2: 250 elements
temporaryPatchSize{3}C2: 250 elements
temporaryPatchSize{4}C2: 250 elements

8x4x250 patches extracted from subsampled C1 Level images of size [(M/5)x(M/5)x4]:
For each orientation and for each band of image, we extract 250 random C1 patches.

```
For j=1:8
D{j} = EuclidianDistances ( Random_Patches[jx4x250],
                            TrainingSetRandPatches[4x250])
End
```

```
For j = 1:4,  for each patches over  4 different PatchSizes
                   which are 4x4, 8x8, 12x12 and16x16:

temporaryPatchSize{j}C2 =
   Min(  D(1):1x250, D(2):1x250, D(3):1x250, D(4):1x250,
         D(5):1x250, D(6):1x250, D(7):1x250, D(8):1x250 )
End
```

Figure 2.21 - C2 similitude response extraction

3 Our Researches

We have two proceedings based on use of image local features and image global features.

3.1 Local Features Approach

SIFT is very successful for image scene description tasks, however its performance is not very good when object description is in question. Our purpose was to understand SIFT and try to adapt it to the object description. We proposed a method using local features approach. The method is developed using face data set, tracking adaptation part is done with car dataset and tested on video records.

3.1.1 Generic Points with Face Dataset

Generic points are first evaluated with face dataset and then assessment is done with car dataset.

3.1.1.1 Selection of proper SIFT implementation

To our examinations, Vadaldi’s SIFT implementation is the most resembling implementation to the original SIFT implementation of Lowe. The results below are obtained by using Vedaldi SIFT implementation with the distance ratio 0.6, optimal value suggested by Lowe.

We obtain at about 200 key points, for the images below, see Figure 3.1, which are 256x256 at gray level. We used human face as object for pre-examinations. Images, including mainly the frontal and 45 degrees turned views of the same face, are compared.

When first image is compared to the second one as in Figure 3.1(A) we obtain 6 couples of SIFT point matched, whereas the result is only 3 points when second image is compared to the first image in Figure 3.1(B) That is because of the use of distance ratio to eliminate suspicious matchings' by comparing the most resembling point and the second most resembling point.

Using the results of both cases, union of the sets of matched points has 7 points. We can observe that we are able to obtain new matches, by gathering the results of comparison of the first image key points to seconds and second image key points to the ones of first image. Another more safe-matching point may help us in critical matching situations. *As Lowe claims it, to be able to decide on whether 2 objects are the same, we have to find minimum 3 couple of points correctly matching between both of the objects. So it will be possible to extract some useful information about 3D scene of the image.*

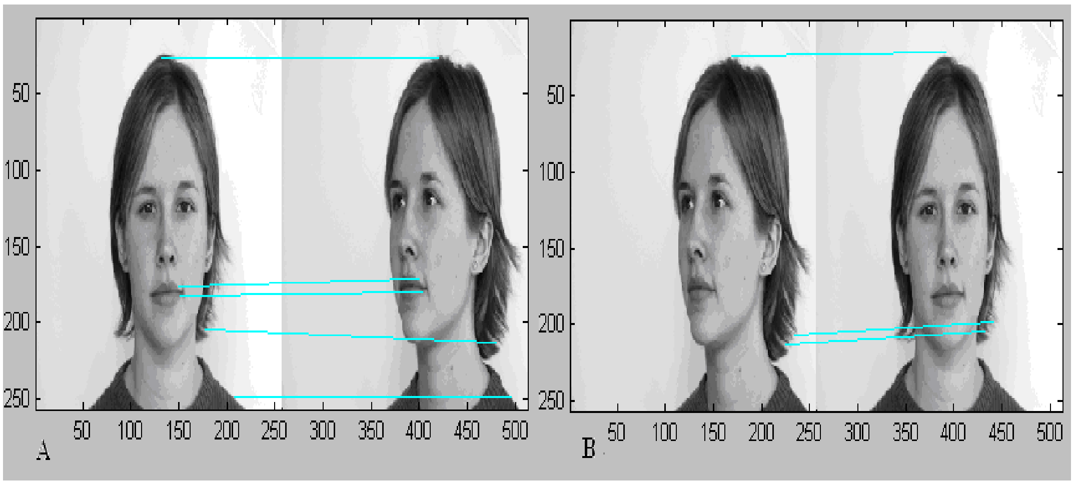


Figure 3.1 - Frontal view and 45 Degrees rotated view comparisons

3.1.1.2 Generic Points

When face is rotated 45 degrees the shape of the object also changes, the orientation of contours, the length and thickness of lines are modified. Especially during low level analyses, spatial intensity changes occur when object is rotated, and regarding this case, SIFT is less invariant, as SIFT descriptors are calculated over a neighborhood centered at a special point by the weighted contributions of gradients. Even the position of a key point is unchanged; but the 128 values of descriptor vector are changed, so this causes the generation of new descriptors that we are unable to match. At this point, we can think about modeling the transformation that the key point vectors are passing during the object rotations. As we are in 2D, it is difficult to model such a change.

On the other hand, to achieve performance and safe matching among the images in the database, it would be fine and judicious to select the SIFT points which are the most robust to such object rotations. An image, in our case, is compared to another with 200 key points in 0.03 second. Before initiating a match, if we can select a minimum set, which is including only the necessary points instead of a matching operation over

200x200 points, we will be able to diminish this task to a matching operation over more or less 50x50 points.

For the purpose of modeling partial flexibility to low level image particularity changes occurring during object rotation transformations, in 2D we tried to apply a perspective planar transformation and then by matching the transformed image points to its original untransformed image, we expected to obtain the key points which are robust to object shape deformations.

The idea is based on an inner step of C1 global image particularity extraction, which is still a better model to describe objects for object category detection tasks: In C1, oriented filters are used to extract the responses of contours in 4 directions and 2 different sizes of filters are used, 11x11 and 13x13, with a MAX operation the significant contour responses are then extracted. Afterwards, the dilatation operation is applied to the extracted contour images. Dilated contour images become roughly described contour images. In fact this produces a general representation of the global image particularities, such as contours, junctions etc. By using this representative model, we are able to compare other objects' global features for the purpose of detecting the class of the object. By using the same approach, we can try to represent a SIFT descriptor with some logical deformations.

For our situation, since our materials are SIFT key points, we can represent a SIFT point with some logical deformations; and among extracted whole set of SIFT points, we can select the ones which will be the most flexible and safest ones against partial object form changes. Moreover, we can try to generate new SIFT points, by a partial modification of original ones, which will be more suitable to match rotated objects points.

We applied some planar perspective transformations to the considered image, the transformation ratio is chosen in order to keep visual context of the image; the face in the transformed image should be visually identifiable. We can then match the transformed image to the same untransformed image and obtain pertinent points whose quantity is, on average, 40-50 for an image of 200-250 descriptors. These points can be used to describe this image with a selected subset of original descriptors or with the whole set of the original descriptors. These points should be the most flexible ones, and should be adapted to our SIFT object description.

To figure out this:

For this illustration, 2 images are used, frontal and 45 degrees rotated views of the same person. Left image, including 45 degrees rotated object, is submitted to a planar perspective transformation. The transformation is from an image plane of **m**x**m** (256x256) to **m**x**0.6m** image plane:

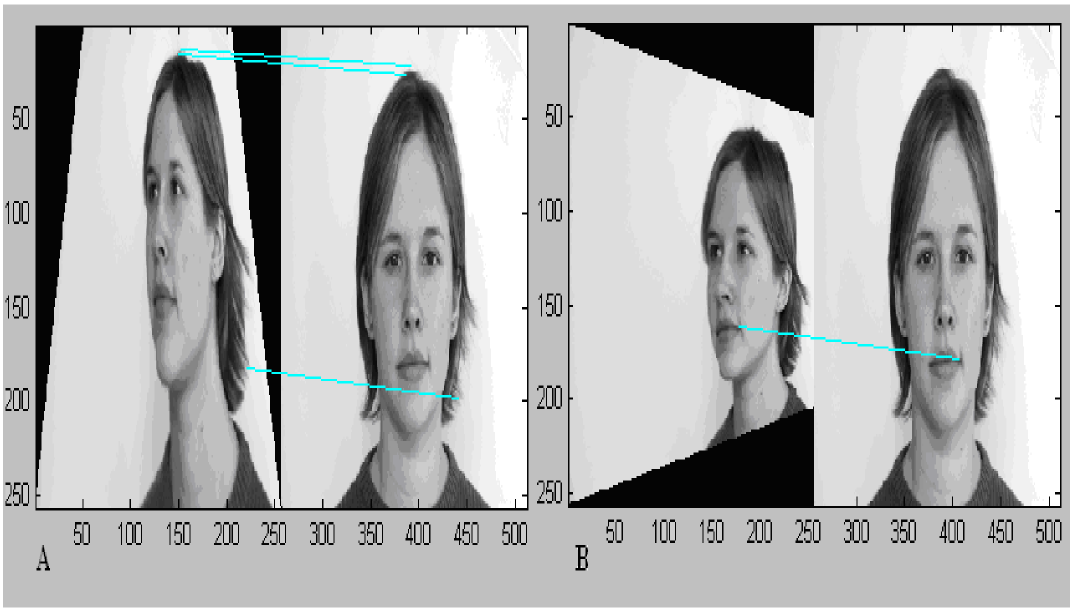


Figure 3.2 - Planar Perspective Transformations

In Fig 3.2.1(A), right image is transformed from **[(0,0), (m,0), (m,m), (0,m)]** into **[(0.2*m, 0) ,(0.8*m, 0), (m, m), (0,m)] a**nd compared to the frontal view which is at right, one more different match is obtained when we compare the result with the ones in Figure 3.2(A) and Figure 3.2(B).

In Figure 3.2 (B), same image is transformed into **[(0,0), (m,0.2m), (m,0.8m), (0,m)]**. We obtain a new match which is different from the 2 points in Figure 3.2 (A) which are on the lips; this one is detected at the right-end of lips.

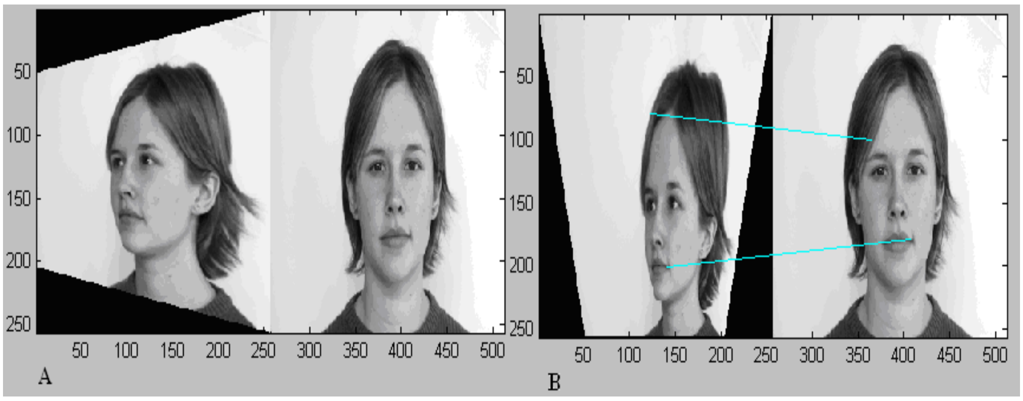


Figure 3.3 - Planar Perspective Transformations

In Figure 3.3(A), same image is transformed into **[(0,0.2m), (m,0), (m,m), (0,0.8m)]**. We obtain no matches.

In Figure 3.3(B), same image is transformed into **[(0, 0), (m,0), (0.8m,m), (0.2m,m)]**. One new match is obtained which is between the hairs of forehead.

In addition to 7 different matchings (union of points in Figure 3.4 (A) and Figure 3.4 (B)), we have 3 new points obtained using transformed images.

We observe that transformed images' key points can help us in the same way. Let us that we know a restricted set of original keypoints which will include the 7 points(in

Figure 3.4) and some distorted keypoints resembling to this restricted set. Then, it seems we can obtain a useful description of the object.

We can apply the same perspective transformations to the left image (right image is frontal view, left image is 45 degrees rotated view) and compare the transformed images with its untransformed version: For each comparison, Figure 3.4 (A), (B), (C) and (D) we obtain a set of key points of 50 points approximately while untransformed image has 200-250 key points.

Union of the mentioned point sets will be used as principal points. And the union of these matching points belonging to the transformed images, union of all of the points shown in Figure 3.4 at left side images, will be called Generic Points from now on. And these Generic Points will be used to describe their matching points situated on the untransformed image, in Figure 3.4 union of all of the points at right side images.

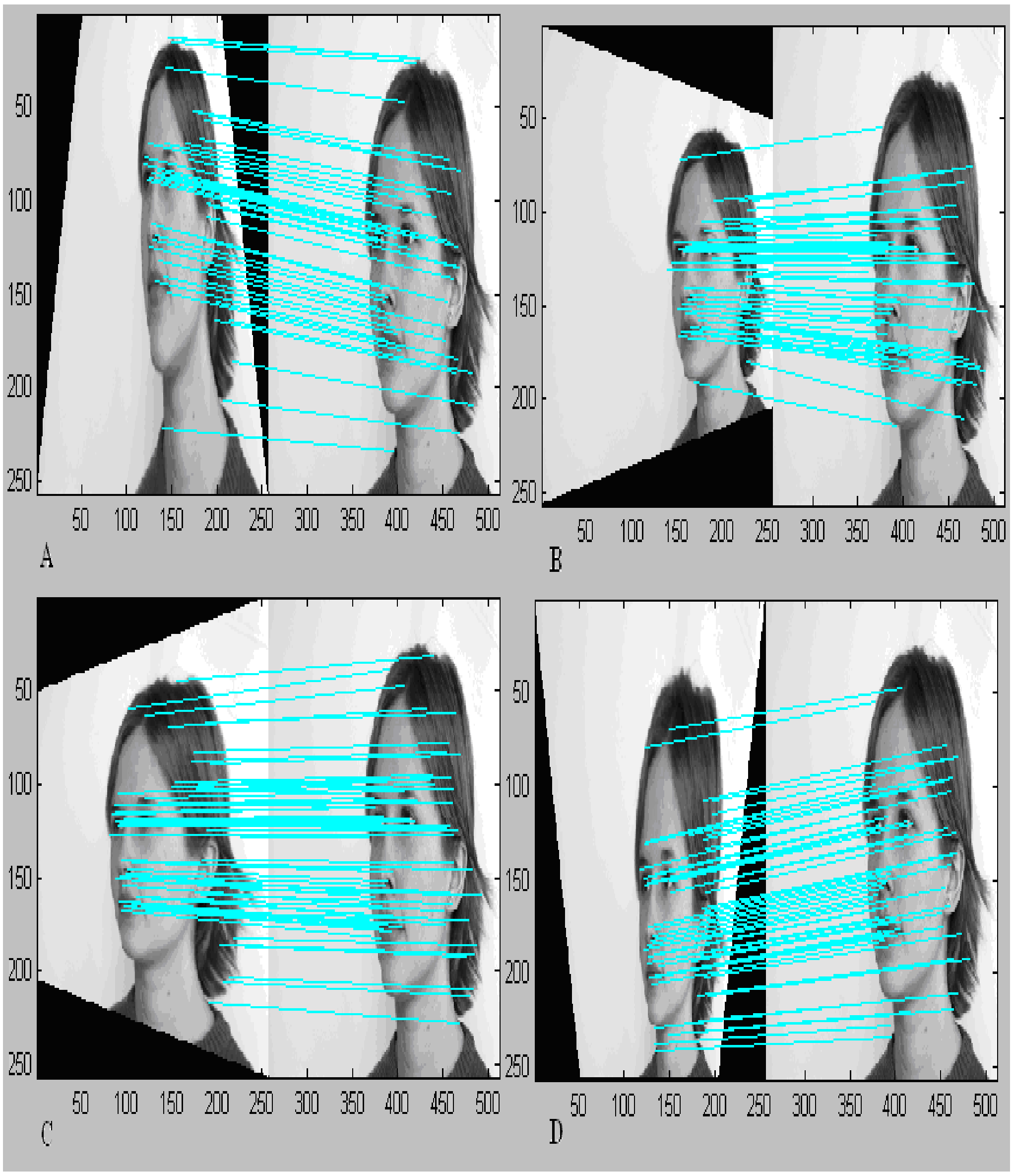


Figure 3.4 - Transformed image matching to the query image

3.1.1.3 About Lowe's Matching

To find the resemblance between 2 vectors (descriptors), we can find Euclidian distances as it is done in HMAX models C2 features similitude response calculation. Rather than doing so, Lowe is proposing to compute the angle ratios of 2 vectors approximately. If the computed angle ratio is a small value, then the resemblance that we find is very close to the resemblance found via Euclidian distance calculation. It is cheap to calculate the dot products of 2 vectors; arccosines of the dot products of two unit vectors is the angles’ ratio we are looking for.

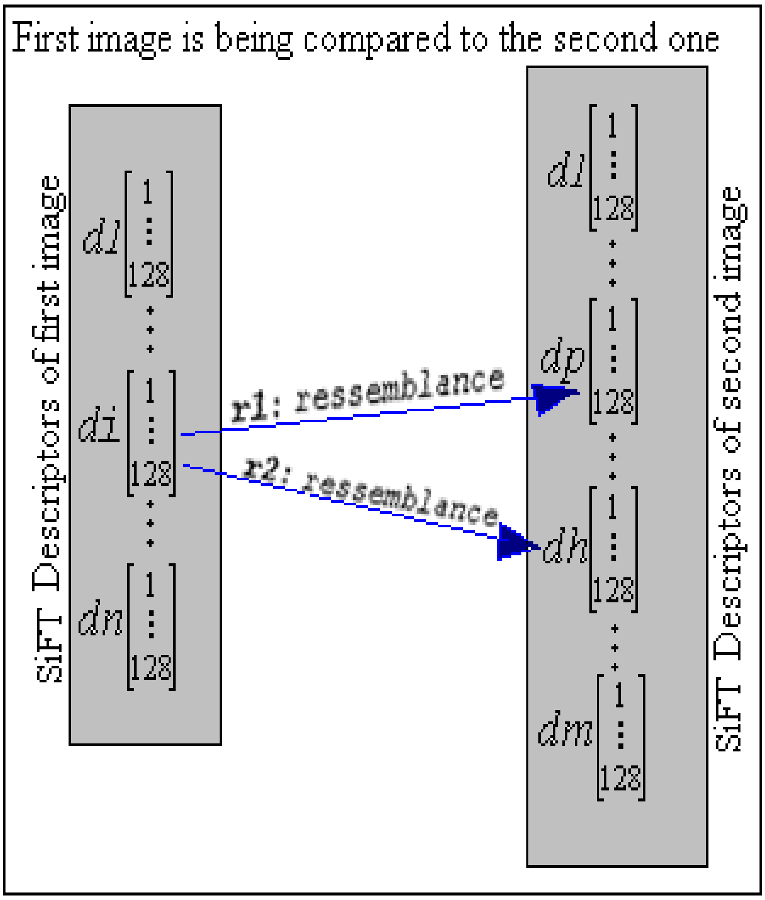

First image is being compared to the second one
SiFT Descriptors of first image
SiFT Descriptors of second image
d1
di
dn
dp
dh
dm
1
128
r1: ressemblance
r2: ressemblance

Figure 3.5 - SIFT Matching

In the image above each descriptor of the first image is compared to the second images’ all m descriptors. The found m resemblances are sorted and 2 highest resemblances are taken into consideration. If one of the 2 highest resemblances is not smaller than 0.6 times the other one, both of them are rejected even if the resemblances are very strong. Because, as it is seen in the Picture for *di*, we have *dp* and *dh* as matches, so we are unable to decide on. That’s why they are eliminated.

Analyzing this method, we can say that descriptors of the second image are competing within each other on being the safest match. So we can say that, if we decrease the quantity of the second image descriptors, we will have a more unsafe matching; a description with less number of keypoints looks dangerous. But as the elements of the first image are compared to the second image's whole set of points one by one, we can try to select the best keypoints of the first image, which are most robust to the object 3D rotations. We think of these points as more favorable to the description of an object.

3.1.1.4 Our Matching

In the image below, the first image, which is going to be matched to a SiFT descriptor image data base, is pretreated and a subset of its descriptors is selected to be compared with the descriptors of the second images descriptors. Firstly, this subset of the first image is compared to the second image with Lowe's matching method, and then, found matches are noted. Secondly, generic descriptors of this subset are compared to the second image. This generic key points are used as representatives of the subset elements. Our matching is simply Lowe's matching method, but with a distance ratio 0.77.

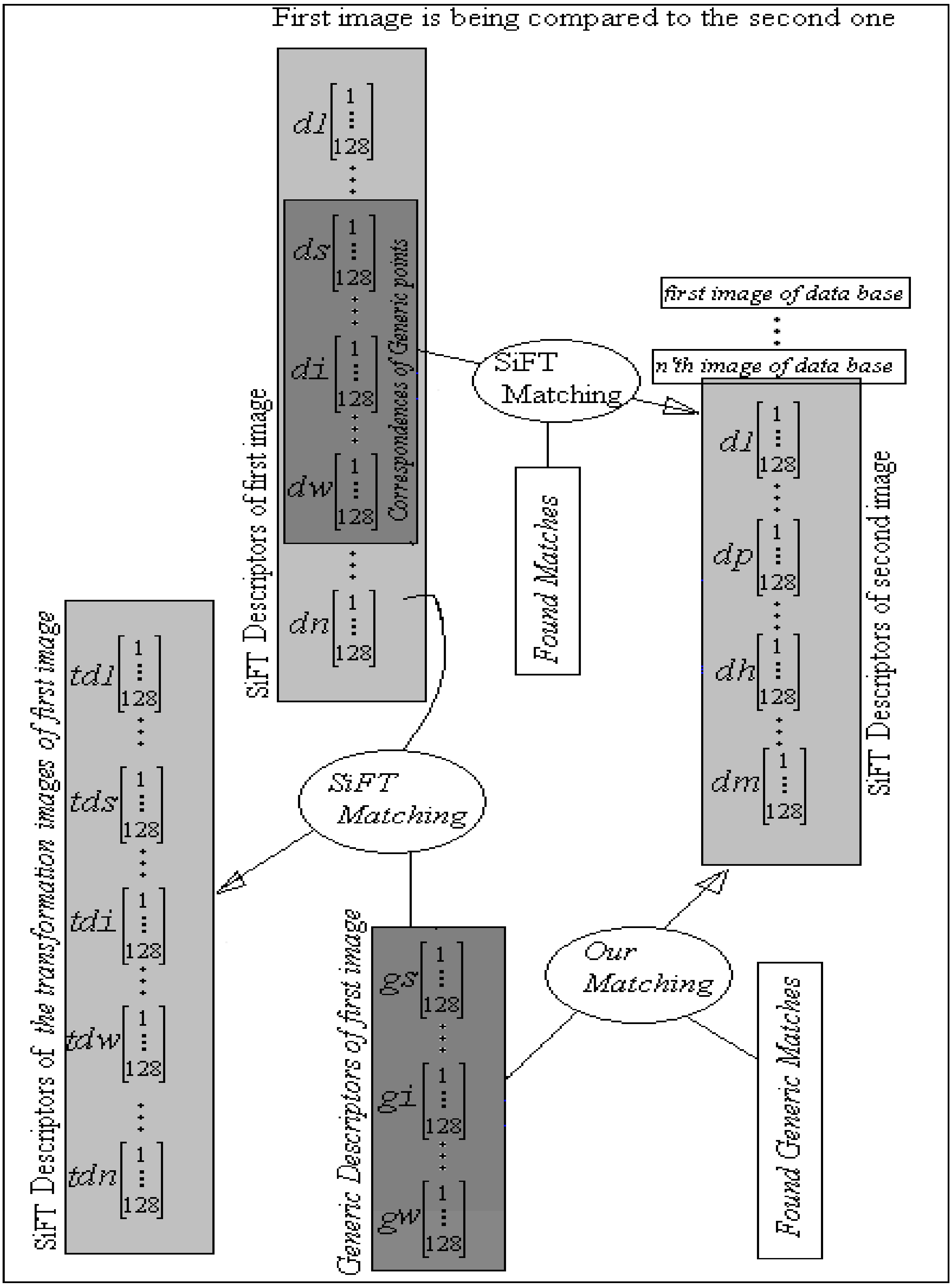

First image is being compared to the second one
SiFT Descriptors of first image
d1
ds
di
dw
dn
1
128
Correspondences of Generic points
SiFT Matching
Found Matches
first image of data base
n'th image of data base
SiFT Descriptors of second image
d1
dp
dh
dm
SiFT Descriptors of the transformation images of first image
td1
tds
tdi
tdw
tdn
SiFT Matching
Generic Descriptors of first image
gs
gi
gw
Our Matching
Found Generic Matches

Figure 3.6 - Generic Points Aided Sift Matching

Surprisingly, for these generic keys, such an elevated distance ratio is not causing much false matches. The reason is that, during first matching operation, some keys of the subset are matched. If a principal object into the first image appears in the second image, then:

- SIFT matching will find some matches for the subset elements of the first image, say M to this matched ones set,
- SIFT matching won't be able to decide certainly on some points, say U to this set, because of strong alternative candidate matches,
- SIFT matching will be able to decide on some wrong points, say E to this set, because of elevated dissemblance.
- Then, generic points will help us to match unmatched ones, too. The set U will find their matches easily. Even if the ratio is elevated, we will be able to find the correct matches among the elements of U easily with some minority quantity of false matching.

If no principal object appeared in the second image, then we would obtain some false matchings' because of the elevated ratio of generic points. But for this case, the decision whether we are facing the correct object will be easy. Because in the presence of the correct object we have correct matches more than normal case, and surprisingly, we are obtaining too fewer mismatches for generic ones; and in the presence of different objects we will have less points with wrong, abnormal relative positionings. Then we would eliminate all of the matches.

These generic points are functioning as voting elements and as a powerful descriptor of the real descriptors that they represent.

3.1.1.5 Generic Points Aided Matching Result

Since the public version of Lowe’s SIFT implementation does not exist, we have no chance to modify his implementation while needed. With the intention of using a public implementation of SIFT, we have compared the existing SIFT implementations to find out the best one, like JiFT(Jann’s Invariant Feature Transform), Robert HESS implementation, Vedaldi etc. In order to use the distance ratio 0.6 with Lowe’s matching method, the best one seems Vedaldi’s implementation.

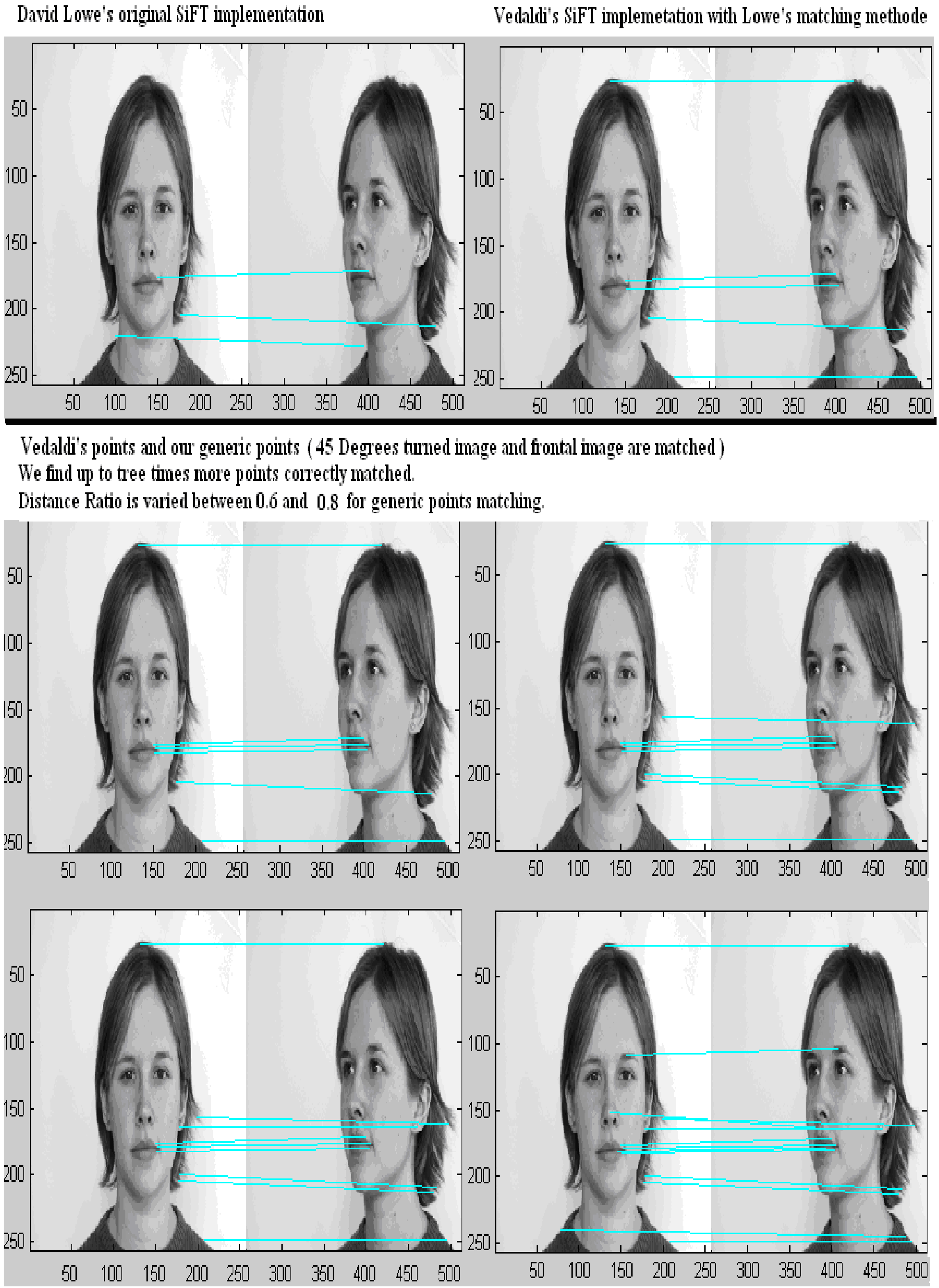
As a first result of our implementation, we obtained the comparison in Figure 3.7. The frontal view of an object is compared with its 45 degrees rotated view. The original implementation of Lowe finds 3 matches, one is on the upper lip, the other one on the extremity of hair and the third one belongs to the scene, not useful for describing the object.

Vedaldi’s implementation with Lowe’s matching method gives us a better result. 5 points, all belonging to the object, are correct matches. The matching distance ratio used is 0.6, the optimal value of Lowe. If we increase this value, false matchings’ occurs. If this value is decreased, fewer matching points are obtained.

According to this first result, our implementation seems to be giving us new matching points. For generic descriptors:

When we use distance ratio as 0.6, we obtain 6 matches (upper left image): one more matching on the lips.

With distance ratio 0.7 we obtain 8 matching points: 2 new matches on the extremities of hairs and the one on the lips.



David Lowe's original SiFT implementation

Vedaldi's SiFT implemetation with Lowe's matching methode

Vedaldi's points and our generic points (45 Degrees turned image and frontal image are matched)
We find up to tree times more points correctly matched.
Distance Ratio is varied between 0.6 and 0.8 for generic points matching.

Figure 3.7 - A comparison of our model

With distance ratio 0.75 we obtain 9 points, all of them are matching correctly (image lower left): A new matching point appears on the ears.

With distance ratio 0.8 we obtain 13 matching points of which 11 are correct and two are false matchings': One overhead, one on eyebrow, 4 points on the lips, one on the ear, 3 on the extremity of hairs, one correct matching on the shoulder and one on the nose which is mismatched to a place between nose and upper lip, and one false matching on the shoulder.

According to our researches, 0.77 is seemed to be a good distance ratio for matching of generic points. With this ratio it seems we will have minimum 2 times more matching points.

However one can ask why we do not have false matchings' with the use of an increased distance ratio, while the original SIFT fails with such higher ratios.

3.1.1.6 A Dangerous Problem False Match Elimination

We observed that if two images include the same object then we obtain fewer false matching and a lot of correct matches which are more than the SiFT matching of Lowe. However, if the compared images mainly include different objects, then we have the risk of having in general randomly distributed mismatches as in Figure 3.8.

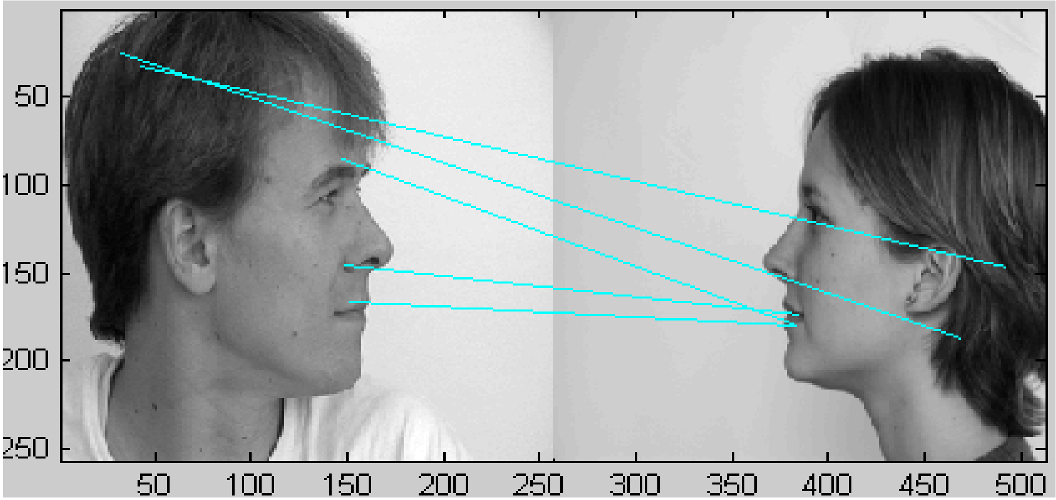


Figure 3.8 - Different objects are being mismatched because of the elevated distance ratio 0.77 of Generic Points Matching.

To solve this problem, we tried to develop an algorithm which will eliminate randomly distributed matches. As Lowe claims, some existing works done earlier to accomplish resolving this problem were not satisfying. However, in our case some additional generic points are used.

The effects of these generic points can be used as a voting medium: thanks to these points, more correct matches are found than normal case while we compare two same objects; and while we compare two different objects, in general abnormally distributed false matchings' are obtained. Therefore, with a strong probability, we will be able to decide whether it is the object we are looking for.

The case of obtaining perfectly distributed false matchings, as can be seen on the left side of Figure 3.9, is observed rarely, however this case is found 2 times while comparing an image to 206 images.

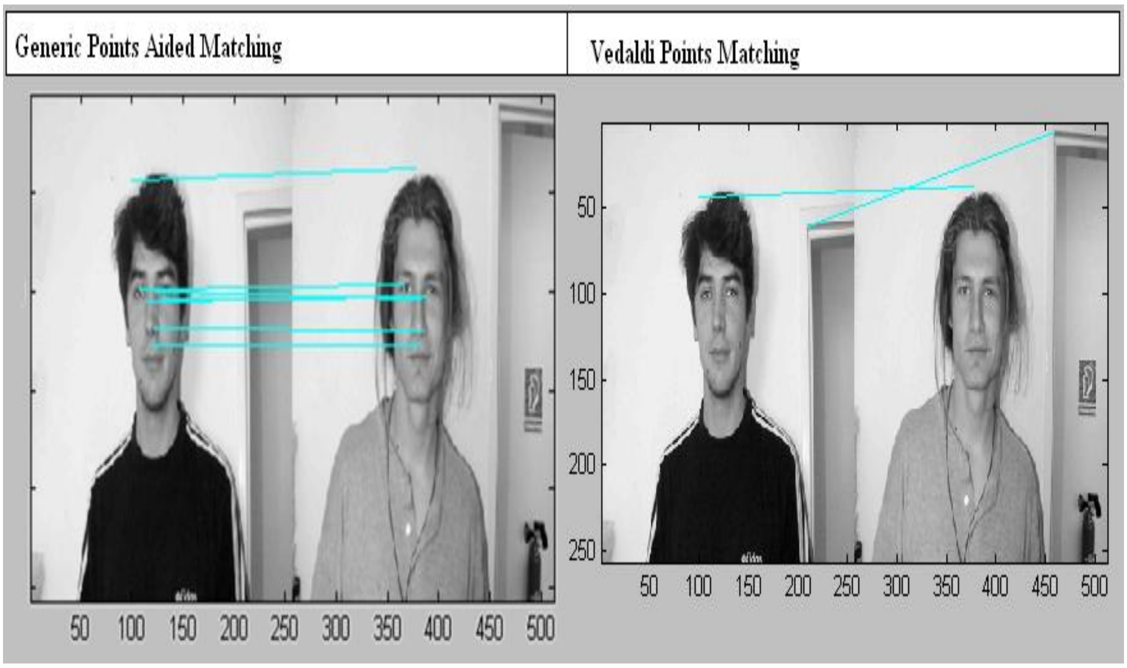


Figure 3.9 - Perfectly distributed Generic Point false matching.

3.1.1.7 A Triangular Relative Positioning Model

Generic points provide lots of false matchings, when the objects are not the same. However, the distributions of these points are highly varied. The relative positions of the points on an object (relativity of each one to others) are remained partially same in another image of the same object. As SiFT points have a limited object rotation invariance of 30-45 degrees, a relative similarity of point distributions between 2 images should be obtained.

In Figure 3.10, in image 2, the triangle ABC is not similar to the triangle A’B’C’: B is in the midst of A and C, because the longest edge of this triangle is |AC|. However, the situation is not kept on the triangle constituted by the pairs of ABC in new image. As the longest edge of A’B’C’ is |A’B’|, C’ is in the midst of A’ and B’. We can say that there is an anomaly. In some cases, we can see that the midst points are not kept, but matching pairs are correct. However, generally this similarity is kept.

Another remark is that we are unable to track the similarities of the points belonging to the independent parts of an object. For example, the arm of a human can be translated in different images and even if the viewed side of a hand is kept same in its new location, its relative position to the face of the human will change. In this case, our triangular relative positioning thought will become worthless.

In another case, the head of a person and its shoulders positions will be changed, in this case, the points matched on shoulders and the points matched on the heads will not constitute similar triangles. Relative positions of head and shoulders of a person can be varied by rotation and movement transformations at most, on average, 50-60 degrees. The SiFT points are used in our approach; these points are invariant to rotations at most 45 degree. So it would be possible for us to adapt our triangulation method to handle the relative position changes between partially-moving and partially-rotating parts of an object like the case of a head and the shoulders.

This adaptation can be realized with some control parameters, but the position change of a hand seems difficult to overcome. Our conception is, a hand is an object itself, it is attached to the head indirectly and the arm of a person is also the same; shoulders of a person, or main body of a person is directly attached to the head and partially moving or rotating.

So, we can adopt these related parts as an integrated object. This conception is assumed valid for all object classes. Eventually, he elimination by this similarity model will provide us more points on the principal parts of objects. The matches belonging to the moving or translating parts of the main object, even if they are correct matches, will be eliminated in our approach. We try to obtain object-center concentrated points with this model.

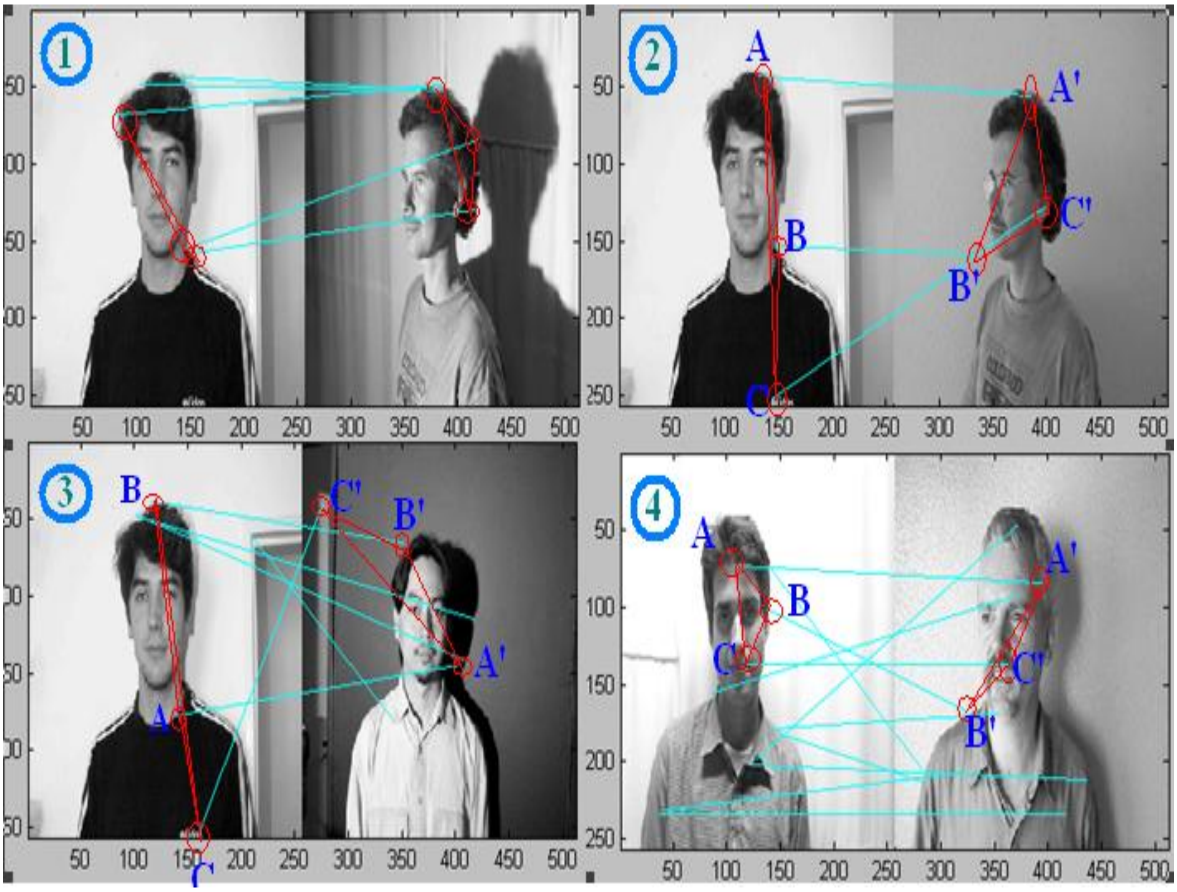


Figure 3.10 - Randomly distributed false matchings which is general case.

Randomly distributed matching anomalies are observed, more than once in each of 4 images above Figure 3.10. We used these anomalies as a medium of decision.

In 3rth image for example, left image has 6 points and a triple combination of these points generates us 20 different triangles, by the same way the 20 triangles of the right side image are extracted and the midst points of these triangles are compared. If the error ratio is bigger than an optimum value, 0.5 for our case, then we are confirmed that this is not our object according to our model.

3.1.1.8 Triangular Similarity

According to the test of midst point comparison, both images are voted according to their regularity of point distribution. If the result of voting is positive, then we apply another criterion to estimate false matching ones, similarity of both of the triangles. In the third image in Figure 3.10, the triangle ABC compared to the triangle A’B’C’. Say the ratio of |AB|/|A’B’| is rAB and by the same way rBC and rAC. According to this example the ratio rBC is abnormally big or small. According to this, C or B is probably a false matching. We consider rAC and rAB to decide on. If rAC>p*rAB then we say that C is abnormal, in other case rAB<p*AC we decide this time that B is abnormal. The p factor is a flexibility parameter.

Of course this is a cruel elimination, we will have probably some correct matches which are eliminated also, but the remnant ones should be more perfect matches.

3.1.1.9 Triangular Parallelism

Another technique that we tried is the parallelism of matching pairs: It is supposed that the sizes of both images being compared are known, thus by using the size information of images and the points positions inside the images we can try to define a parallelism criterion. In the same figure, Figure 3.10, in image 2, |AA’| and |BB’| are almost parallel lines but the inclination of |CC’| is roughly different. Of course our objects can be rotated in 2D, but even in the 2D rotation cases this criterion is helping to eliminate anomalies.

We noticed that Triangular Similarity is a better eliminating method, and it eliminates matchings belonging to the scene elements also, it is keeping the matches belonging to the principal objects, because the density of matched points corresponds on the principal object. We can say that this eliminating method has a tendency of being object centered.

3.1.1.10 Generic Points Aided Elimination Results

For our tests we used an image collection of 256x256 gray level .pgm images. This collection includes 206 pictures of 13 people: Each person has 12-20 images in the collection. This collection includes difficult images having strong shadows, rotation up to 90 degrees, different principal objects, use of cluttered scenes, and smiling, laughing faces with closed eyes… All of these are serious difficulties for SiFT matching.

For the first person, we captured the matching results in Figure 3.11. Left side column includes our result and right side column includes Vedaldi SiFT points with Lowe Matching Method. During our examinations we observed that the SiFT points of Vedaldi are more successful than Lowe's SiFT points. So we adopted it for this test.

The red frame contains our correct identifications. These points are correctly matched and there are more than 3 points being part of the object. For image 1 both methods are successful, but in 2, 3, 4 our method is succeeding perfectly.
For the images 2, 3 and 4 Vedaldi Matching is failing.

And the interesting remark is that our matching points are generally located over the head of the person, which is the significant part of a human for identification, or which might be the main part of another object for identification. On the body or on the scene there are very few points.

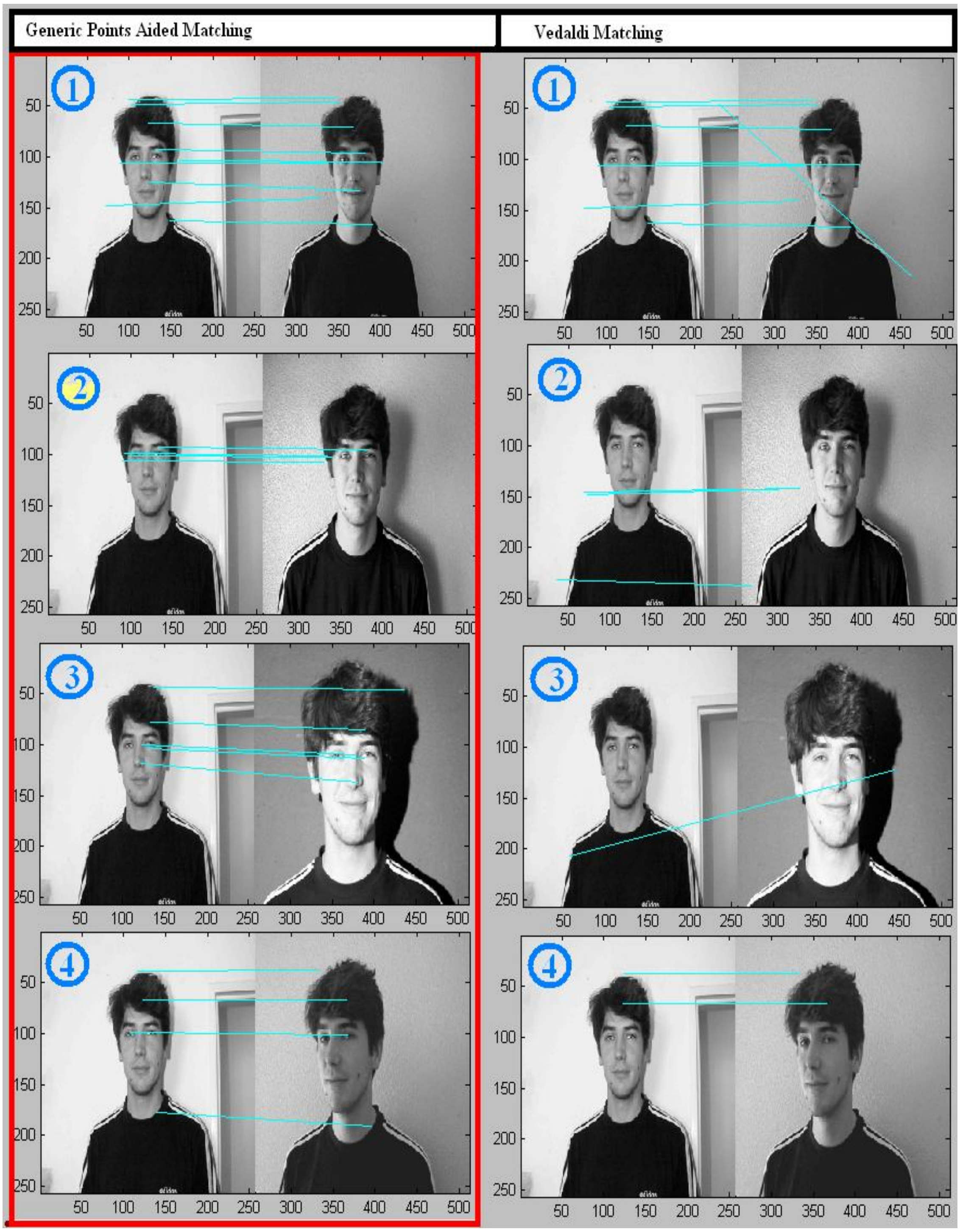


Figure 3.11 - Generic Points Aided matching versus classical SiFT matching.

In the blue frame in Figure 3.11, we have also correct matches. But we have fewer than 3 points corresponding onto the object; other points are the parts of the scene. So as a scene recognition result we have 3 more correct identifications. On the other hand Vedaldi's implementation is succeeding for only the fifth image.

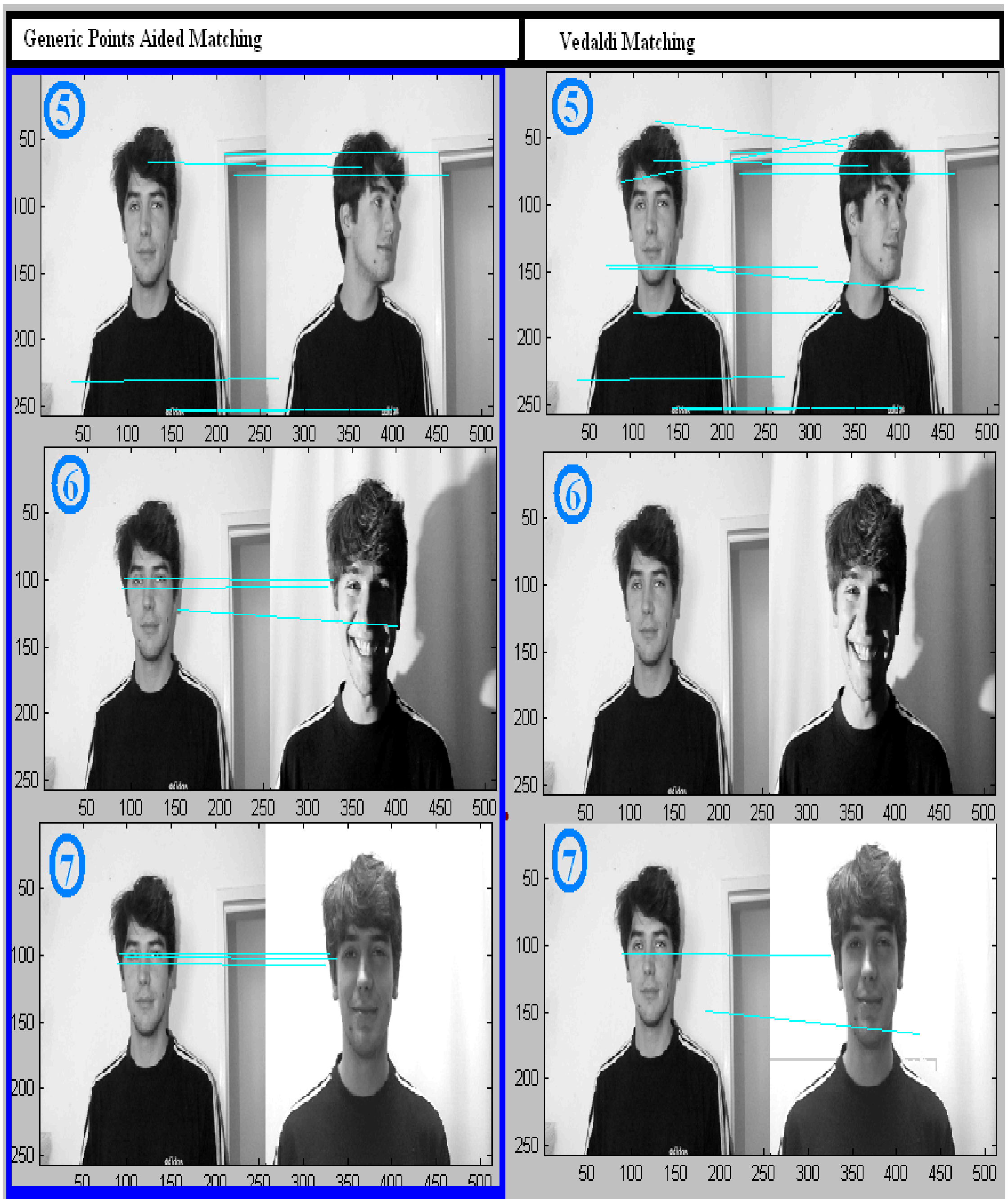


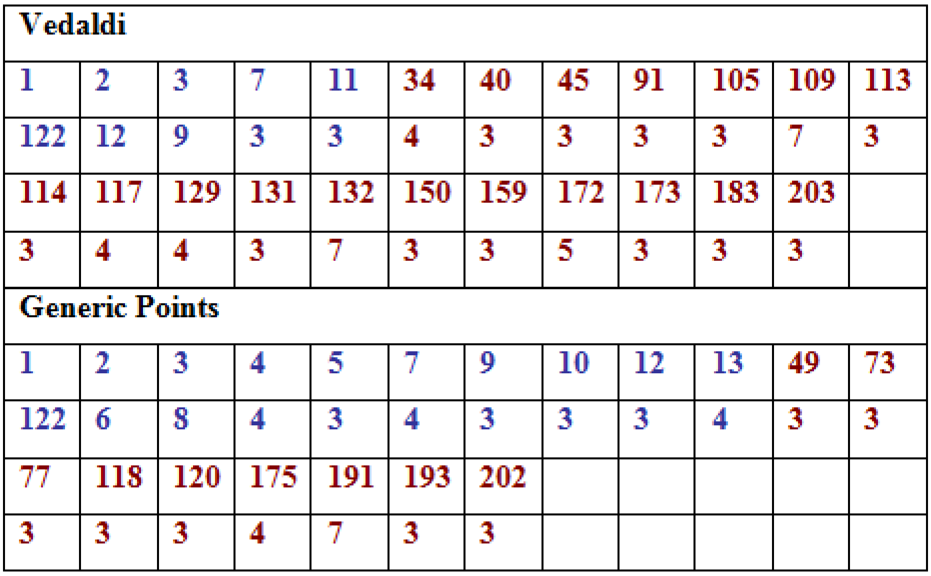
Figure 3.12 - Generic Points Aided matching versus classical SiFT matching.

3.1.1.11 Some Test Results for Object Detection

In the tables below upper lines is the image indices and lower lines are the match quantities that are found. Blue ones are the correctly identified images. Red ones are the erroneously identified images over 206 images.

3.1.1.11.1 First Frame

For image 1 Vedaldi is finding 4 correct detections and 18 false detections. Our Generic points are giving 9 true detections and 9 false detections. So, greater true detection and fewer errors.

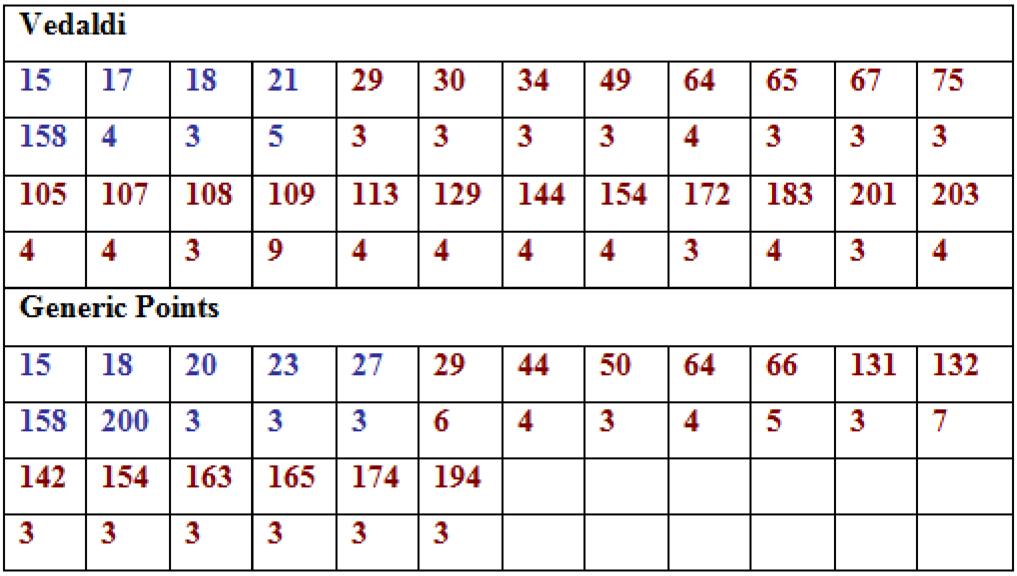


Vedaldi											
1	2	3	7	11	34	40	45	91	105	109	113
122	12	9	3	3	4	3	3	3	3	7	3
114	117	129	131	132	150	159	172	173	183	203	
3	4	4	3	7	3	3	5	3	3	3	
Generic Points											
1	2	3	4	5	7	9	10	12	13	49	73
122	6	8	4	3	4	3	3	3	4	3	3
77	118	120	175	191	193	202					
3	3	3	4	7	3	3					

Figure 3.13- Vedaldi vs Generic Points 1

3.1.1.11.2 Second Frame

For 15th image, Vedadi gives 3 correct detections and 21 false matches. Our approach gives 4 correct detections and 13 false detections. To decrease false matching quantity, a hard elimination method is applied. Among eliminated matches there may exist some correct ones too, so our results are giving fewer false detections.

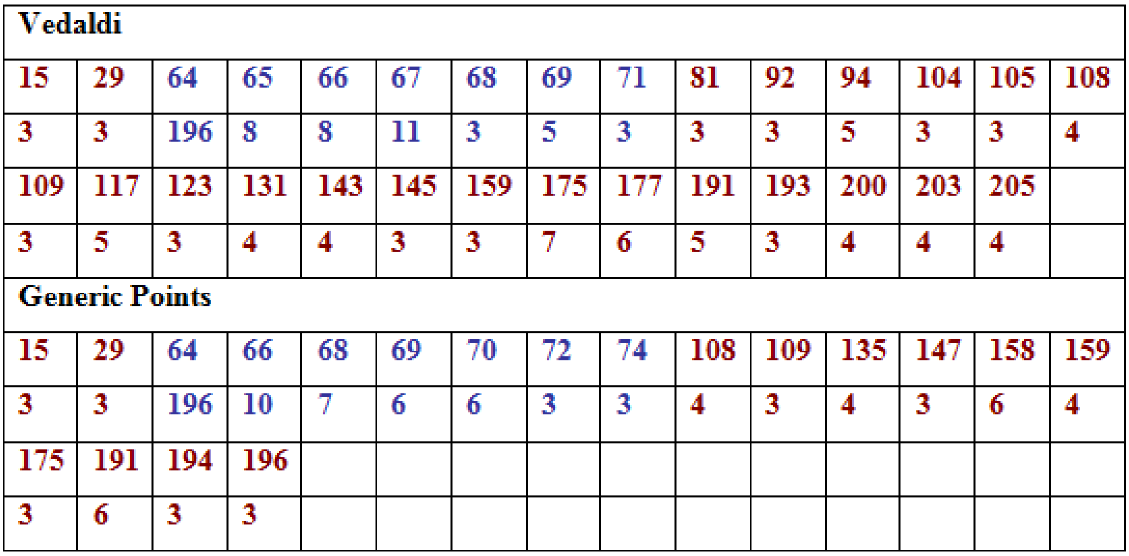


Vedaldi											
15	17	18	21	29	30	34	49	64	65	67	75
158	4	3	5	3	3	3	3	4	3	3	3
105	107	108	109	113	129	144	154	172	183	201	203
4	4	3	9	4	4	4	4	3	4	3	4
Generic Points											
15	18	20	23	27	29	44	50	64	66	131	132
158	200	3	3	3	6	4	3	4	5	3	7
142	154	163	165	174	194						
3	3	3	3	3	3						

Figure 3.14 – Vedaldi vs Generic Points 2

3.1.1.11.3 Third Frame

For the person in the 64th image, Vedaldi has 6 detections and 23 false detections. Our approach is giving 6 correct detections and 12 false detections. We obtained visibly less error ratio in this frame.

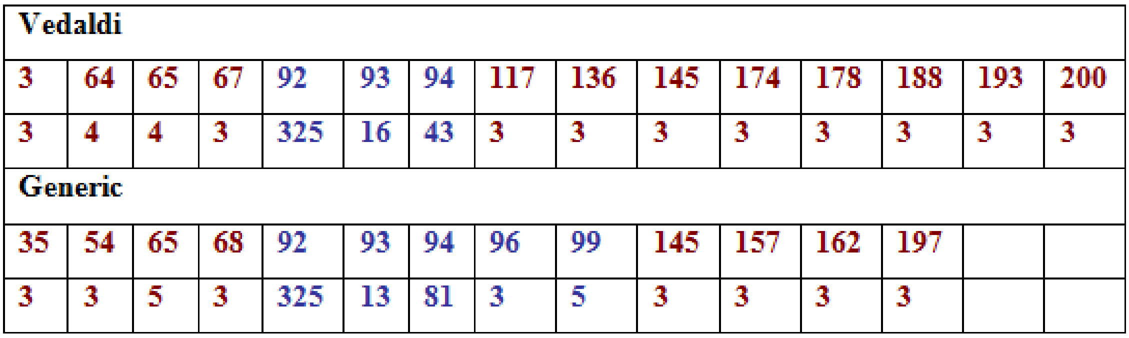


Vedaldi														
15	29	64	65	66	67	68	69	71	81	92	94	104	105	108
3	3	196	8	8	11	3	5	3	3	3	5	3	3	4
109	117	123	131	143	145	159	175	177	191	193	200	203	205	
3	5	3	4	4	3	3	7	6	5	3	4	4	4	
Generic Points														
15	29	64	66	68	69	70	72	74	108	109	135	147	158	159
3	3	196	10	7	6	6	3	3	4	3	4	3	6	4
175	191	194	196											
3	6	3	3											

Figure 3.15 - Vedaldi vs Generic Points 3

3.1.1.11.4 Forth Frame

For the person in the 92th image, Vedaldi gives 2 true detections and 11 false detections and our result gives 4 true detections and 8 false detections.



Vedaldi														
3	64	65	67	92	93	94	117	136	145	174	178	188	193	200
3	4	4	3	325	16	43	3	3	3	3	3	3	3	3
Generic														
35	54	65	68	92	93	94	96	99	145	157	162	197		
3	3	5	3	325	13	81	3	5	3	3	3	3		

Figure 3.16 – Vedaldi vs Generic Points 4

3.1.1.12 Assessments and Perspectives for Generic Points with Face Dataset

We might have chosen not to use matching points of transformed images as the representatives of the original ones, with a distance ratio 0.77. Instead of this, we would realize SiFT matching normally and then rematch all of the unmatched ones with an elevated distance ratio 0.77. So we would obtain some more correct matches and some false matches (that were to be eliminated).

But the second comparison costs expensive for us, say we have 500 SiFT points for both images and after first matching with 0.6 distance ratio we have 490 unmatched points, for the second matching with 0.77 distance ratio we will have to realize 490*500 matching operations in addition to first matching of 500*500.

In our case we use 100-200 generic points to do this task and as these are the points robust to object form deformations surely we obtain more matches for 3D partially rotated images with a rotation limit of 45 degrees.

In this model firstly, the query image is transformed in 4 directions and the query image is compared to these transformed images. Obtained SiFT matching points are used as generic points. Somehow it seems that without transforming query image, we might obtain nearly almost of the same generic points by transforming the query images SiFT points with the same transformation matrix. By comparing the original SiFT points with the transformed ones we obtain most probably the same generic points.

A SiFT point can be represented with its transformed matches for the use of object description, so for future studies it might be judicious to try modeling the changes happening on SiFT points while the object is returning in 3D degree by degree. There should be a way of approximating 2D image global features, like contour, junction etc. to the rotated object images global features.

Our hypothesis is the fact that while transforming 2D images with planar perspective transformation, we are keeping the visual context of images. After transformation also the images are visually identifiable and it is surprising for us that with transformed images we find new matches that the original image couldn’t catch and these new matching points can have pairs on the original image.

Set of Generic points is functioning as a medium of bridge between a point of query image and a point of the compared database image.

3.1.2 Generic Points with Car Dataset

In this section, Generic Points Aided Robust Description is subjected to assessment with car data set.

3.1.2.1 Image Local Feature Based Video Object Description and Tracking

In this section we tested our proceeding on a car image set and used it for video object tracking. Similar to face data set test, towards extraction of image features, each image is preprocessed by applying different perspective planar transformations, and a set of points, which are robust with respect to geometrical deformations, is obtained. These transformations are chosen in a manner to preserve the perceptional identities of the principal objects existing in the transformed images. The main contribution of this study consists of comparing the trained images with the transformed images and gathering a set of the most stable points which are representing the principal objects of the trained images. These stable points derived by the set of the trained images, are then used as a robust description and tracking of the objects in motion. In order to improve reliability of the presented method, an algorithm is proposed to correct the mismatches which occur at point matching stage. This correction algorithm is same as used for face data set test. The results of the studied method are compared with classical SIFT matching.

Better results illustrate the effectiveness and the robustness of the SIFT based object description that we proposed.

3.1.2.2 Methodology

The set of multi-view car profiles (Ozuysal, 2009), is used to perform SIFT algorithm. When the view of an image object is changed, the orientation of contours, the length and thickness of lines is modified. Especially during low level analysis, spatial intensity changes occur when the object is rotated, and regarding this case, SIFT is less invariant, as SIFT descriptors are calculated over a neighborhood centered at a special point by the weighted contributions of gradients. With such transformations, even the position of a key point is remained unchanged but the 128 values of descriptor vector are changed, the generated descriptors are unmatched. To avoid the possible unmatched descriptors' generation, transformations are modeled in which the key point vectors are existing during the object rotations. But in a 2D modeled scene, it is difficult to model such a change.

On the other hand, to achieve performance and robustness of matching, it would be fine and judicious to select the SIFT points which are the most robust to such object rotations for the purpose of modeling a partial flexibility to low level image particularity changes occurring during the object view change transformations. Using a scene in a plane, applying a suit of perspective planar transformations and then by matching the transformed images points to its original untransformed image, the key points can be obtained which are robust to object shape deformations.

The idea is based on an inner step of C1 global image particularity extraction method (HMAX), which is still a better model to describe objects for object category recognition tasks: In C1, oriented Gabor filters are used to extract the responses of contours in 4 directions and 2 different sizes of filters, 11x11 and 13x13, are used with a MAX operation; the significant contour responses are then extracted. Afterwards, the dilatation operation is applied to the extracted contour images. Dilated contour images become roughly described contour images. In fact this produces a general representation of the global image particularities, such as contours, junctions etc.

By using this representative model, we are able to compare other objects' global features for the purpose of detecting the class of the object. By using a similar approach, we can represent a SIFT point with some logical deformations; and among extracted whole set of SIFT points, we can select the ones which will be the most flexible and safest ones against partial object form changes. Moreover, we can generate new SIFT points, by a partial modification of original ones, which will be more suitable to match rotated objects points.

We applied some planar perspective transformations to the considered image, the transformation ratio is chosen in order to keep visual context of the image; *e.g.* a human face in a transformed image should remain visually identifiable or a car rear should remain recognizable. According to our trial and errors, the perspective planar transformation ratios (0.90, 0.90) and (1.00, 0.80), fit best to our scheme. The transformed images are matched to the same untransformed image and obtain pertinent points whose quantity is, on average, 40-50 for an image of 200-250 descriptors.

These points can be used to describe considered car image object with a selected subset of original descriptors and with their corresponding new matches. These points should be the most flexible ones, and should be adapted to our SIFT object description.

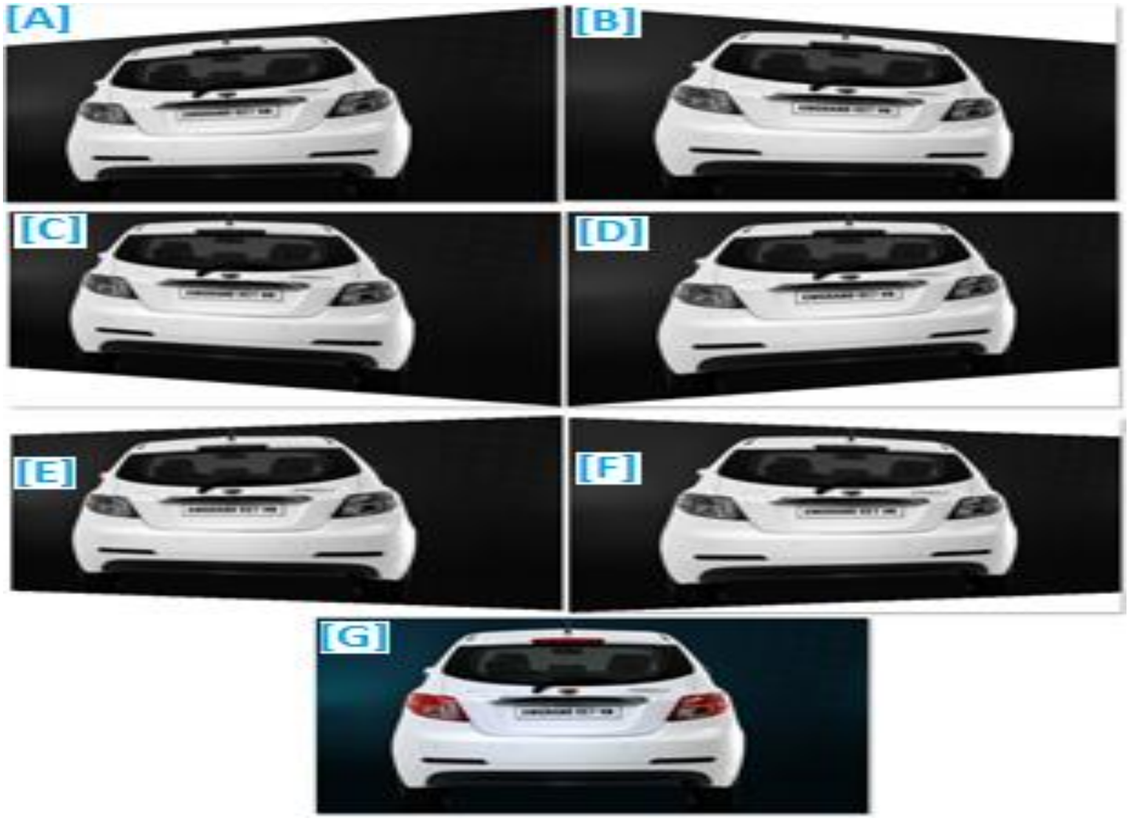


Figure 3.17 - Images A, B, C, D, E, F are perspective planar transformations in different directions of image G. Transformation ratios are [0.9, 0.9] and [1.0, 0.8].

Perspective transformations given in Figure 3.17(A), (B), (C), (D), (E), (F) of the same car image (given in Figure 3.17(G)) are compared. For each comparison, (in Figure 17, A-G, B-G, C-G, D-G, E-G, F-G), a set of key points of approximately 50 points is obtained per transformation while untransformed image has approximately 200-250 key points.

Union of the matching point sets over different transformations is used as principal points set. These are the “Generic Points” used for car image set: the union of these matching points belonging to the transformed images. Generic Points will be used to describe their matching points (principal points) situated on the untransformed image (G).

These transformations are over 6 directions as seen in Figure 3.17(A), (B), (C), (D), (E) and (F). In total, 6 transformed images exist. In a real traffic video, over consecutive frames, global image features (contours, junctions etc.) are less distorted in vertical axis; however, sudden view changes in the horizontal-axis frequently may happen (for example in the overtaking event); as a result, horizontal distortions arise on global and local image features to which SIFT has less tolerance. Therefore, horizontal transformations (towards horizontal axis) are preferred.

3.1.2.3 Car Data Set Matching Methodology

The matching methodology used for face data set is applied. And similar results are obtained. Generic points aided description is compared to Veldaldi matching. Better results are obtained. Then false match correction algorithm is applied and comparison is performed. Below results are obtained.

3.1.2.4 Car Data Set Generic Points Aided Matching Versus Lowe’s Matching

Without applying correction algorithm, we obtained the compared results given in Figure 3.18. The view of the car’s rear is compared with its 45 degrees rotated view image. The original implementation of Lowe generates 9 matches (Figure 3.18 (A)), most of them are located on the plate of the car since it’s more textured. Vedaldi’s implementation with Lowe’s matching method generates a better result, (Figure 3.18 (B)), 10 correctly matching points are obtained.

The matching distance ratio used is 0.6 which is the optimal value of Lowe’s algorithm. If the ratio is increased, false matches occur. If the ratio is decreased, fewer matching points are obtained.

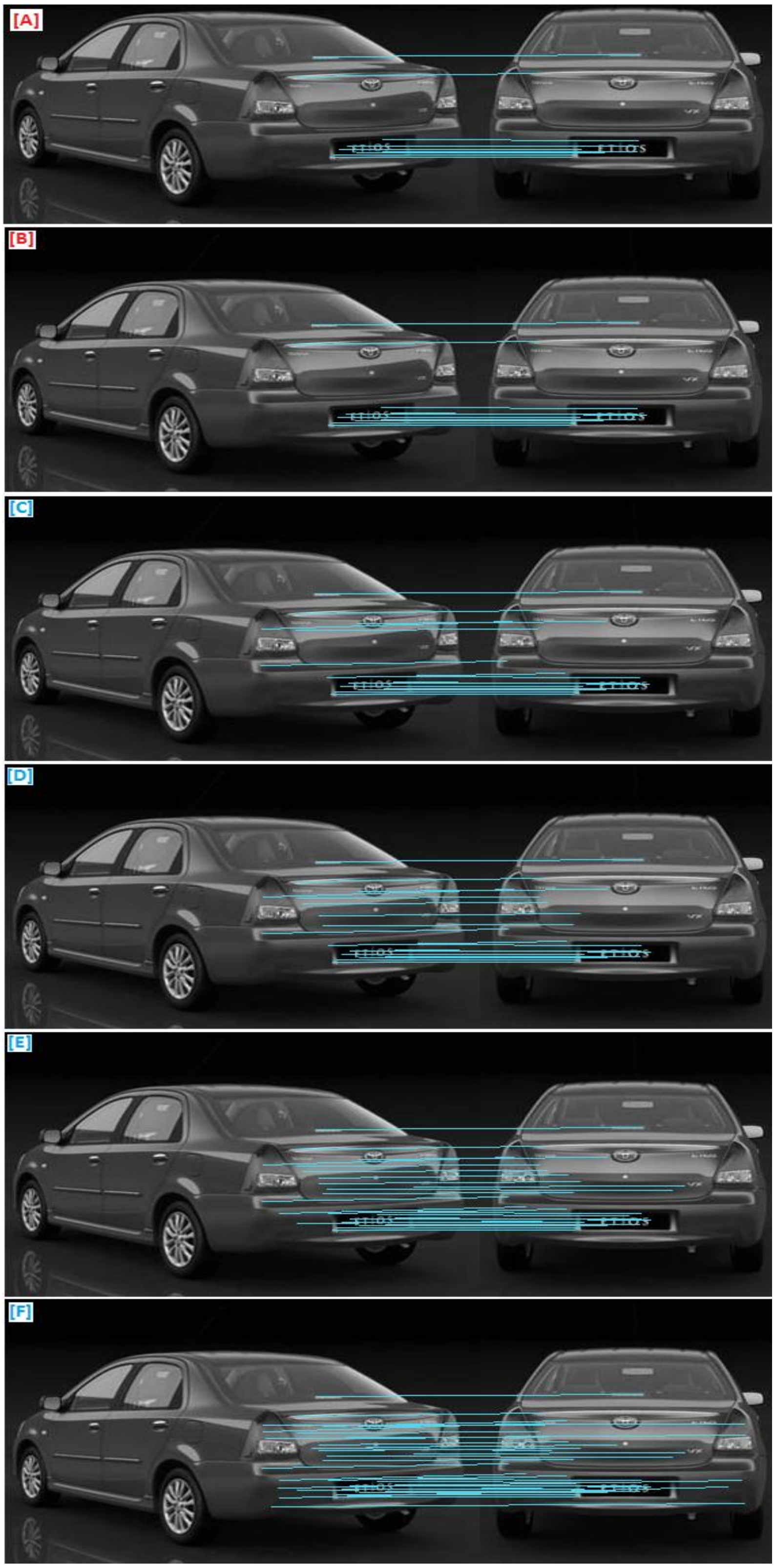


Figure 3.18 - Lowe SIFT [A] and Vedaldi SIFT [B] matching versus Generic Points aided matching [C-D-E-F].

According to this first result, the presented implementation is detecting new matching points, for generic descriptors: when the distance ratio as 0.6 is used, (Figure 3.18(C)), 15 matches are obtained. The distance ratio of 0.7, as the algorithm parameter, leads to 19 matching points, (Figure 3.18 (D)). With distance ratio of 0.75, 24 points are generated, (Figure 3.18 (E)), all of them are matching correctly. With distance ratio of 0.8, 30 matching points with 27 correct and 3 are false matches are obtained (see Figure 3.18 (F)).

The distance ratio of 0.75 is tuned for matching of generic points. Using this ratio, more than double matching points are obtained versus the other distance ratio values.

3.1.2.5 Car Data Set False Match Elimination

If two images include the same object, then fewer false matching and more correct matches than the SIFT matching of Lowe is obtained. However, if the compared images mainly include different objects, then the risk of having in general randomly distributed mismatches increases. To solve this problem, algorithm is enhanced to eliminate randomly distributed matches. In our case study, a subset of robust SIFT points are selected and their corresponding generic points are used for matching. The effects of these generic points can be used as a voting medium: thanks to these points, more correct matches are found than normal case while we compare two same objects; and while we compare two different objects, in general abnormally distributed false matches are obtained. Therefore, with a high probability, we will be able to decide whether it is the object we are looking for or eliminate false matches if much more fewer than correct matches which are coherently distributed relative to each other.

The relative positions of the points on the object (relativity of each one to others) are remained partially same in another image of the same object. As SIFT points have a limited object rotation invariance of 30-45 degrees, a relative similarity of point distributions between 2 matched images should be maintained and useful for false match elimination. Outliers are cleaned based on this idea and decision is made upon relative distributions of matching pairs.

For the tests, the multi-view image set, given in (Ozuysal, 2009), containing 20 sequences of car images subject to rotation of 360 degrees, is used. This data set includes images subject to shadows, rotation, different principal objects in the background, and to the cluttered scenes. Car surfaces are less textured and poor in including local distinctive information. All of these effects arise serious challenges for SIFT matching. For the first car rear view frame, the matching results in Figure 3.19 are obtained. Left side column includes the developed algorithm's results and right side column presents Vedaldi SIFT points with Lowe Matching Method. The blue frame in Figure 3.19 contains the generated correct identifications. These points are correctly matched and there are more than 3 points being part of the object. If fewer than 3 points remain after discarding outliers, then the match is rejected.

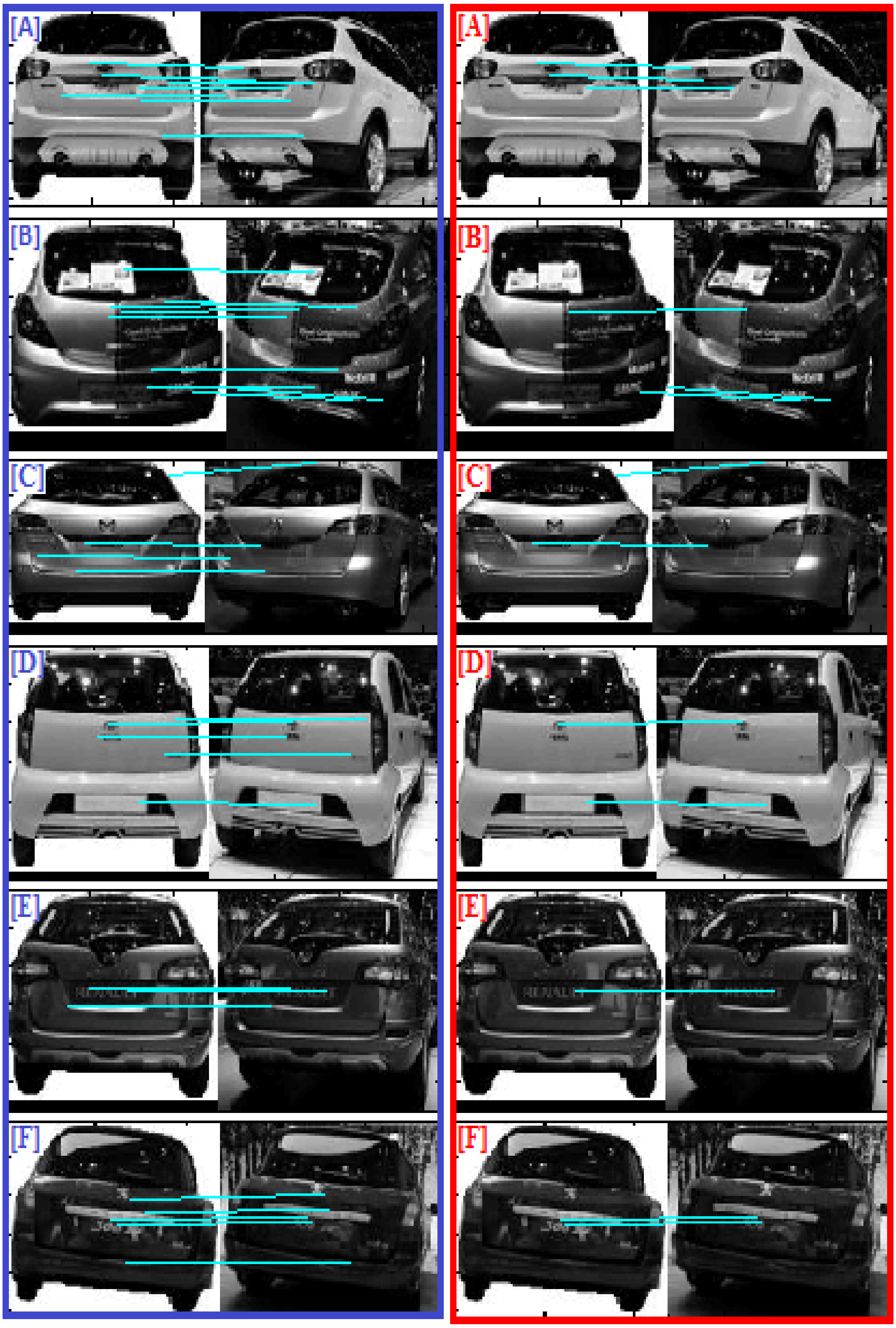


Figure 3.19 - Generic points matching (blue column) versus Vedaldi matching (red column)

For image pairs [A] and [B] in Figure 3.19, both of the methods are successful: In [A], Generic Points are constituted by 7 correct matching, whereas Vedaldi's algorithm generates 3 correct matchings. In [B], Generic Points has 9 correct matchings versus Vedaldi with 3 correct matches. For remaining image pairs, [C], [D], [E] and [F], Generic Points succeeds perfectly whereas Vedaldi SIFT matching algorithm is failed.

Results for 20 cars over 600 images are reflected in the overlaying plots Figure 3.20 and Figure 3.21. For each car presentation, 30 rear captures are selected from multi view data set. Each rear capture of the same car has a different viewpoint varies approximately by 3 degrees beginning from -45 degrees up to 45 degrees.

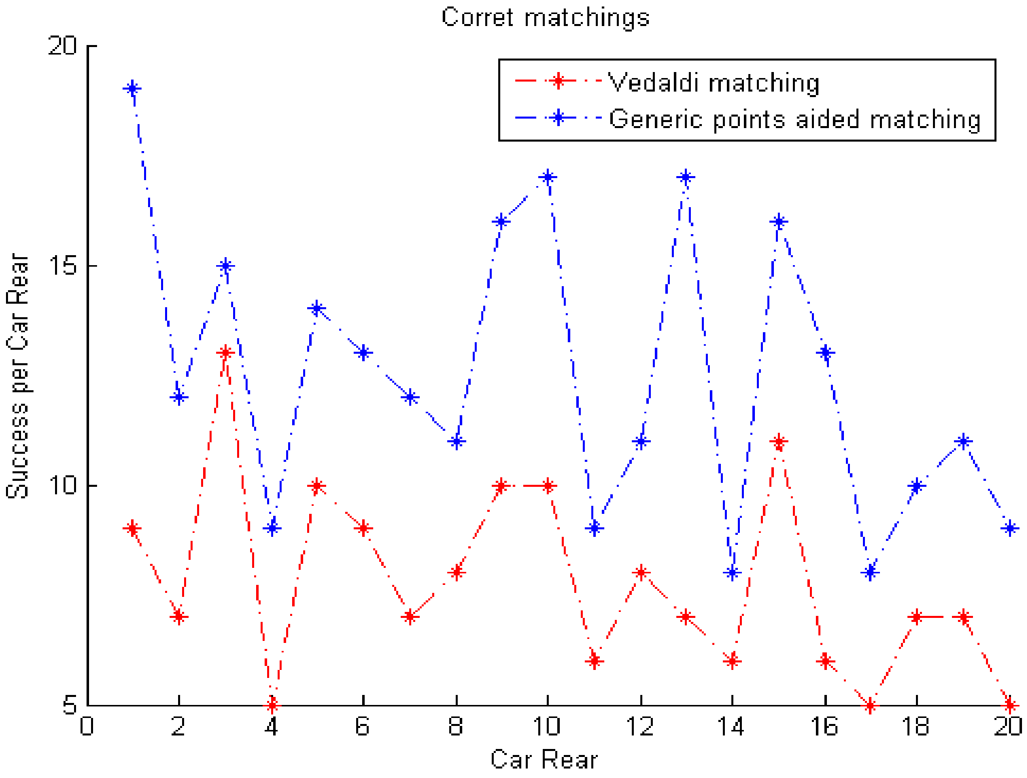


Figure 3.20 - Correct matching comparison

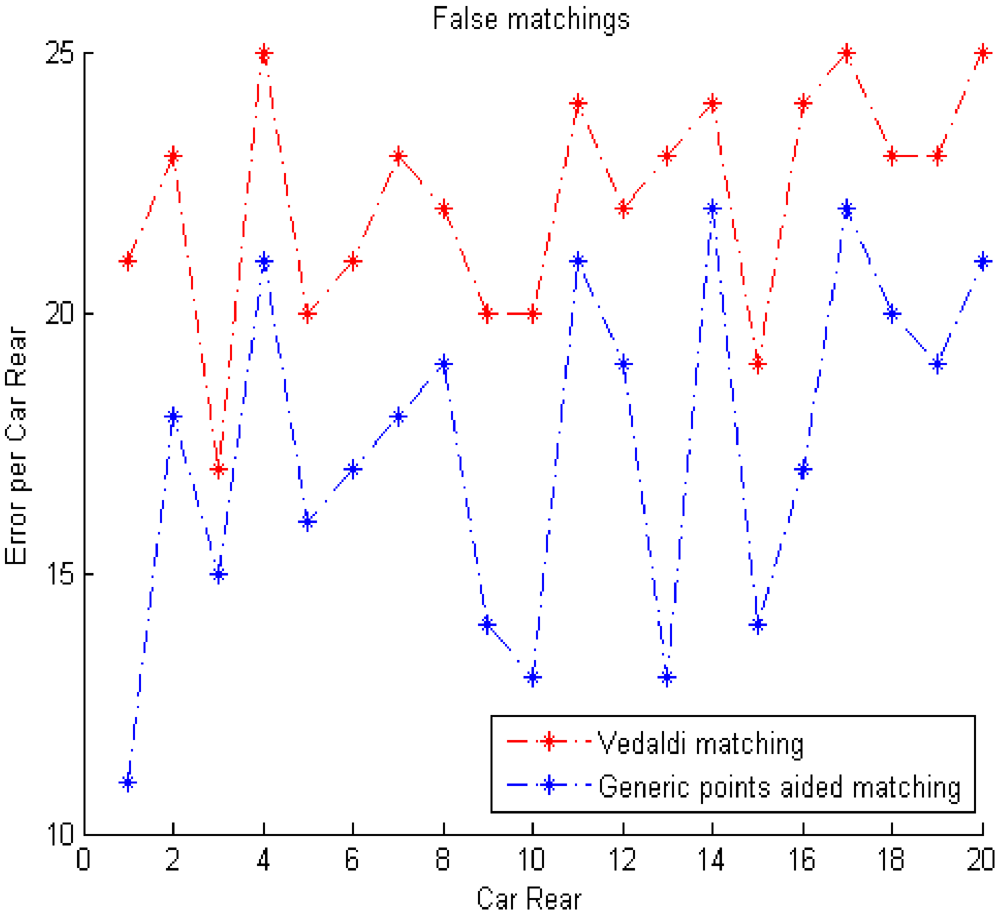


Figure 3.21 - False matching comparison

Correct matches are compared in Figure 3.20. The presented matching algorithm performs better than Vedaldi SIFT implementation with Lowe's matching in terms of correct matches. False detections are compared in Figure 3.21. Generic points aided matching is lesser error-prone.

3.1.2.6 Tracking with Generic Points

Generic points are more reliable in matching over difficult image pairs. It is deployed for car tracking video. To achieve real-time efficiency, the GPU CUDA implementation of Andrea Vedaldi's SIFT is used. Generic points are obtained with the same implementation as well. For feature tracking, a statistical tracking method called RAMOSAC which is based on the well-known RANSAC algorithm is used. Our Generic Points aided description is combined with RAMOSAC which is a feature tracker working with SIFT/SURF features. This method is evaluated in an attempt to lower the detection error rate and to achieve more robustness to car rear view changes.

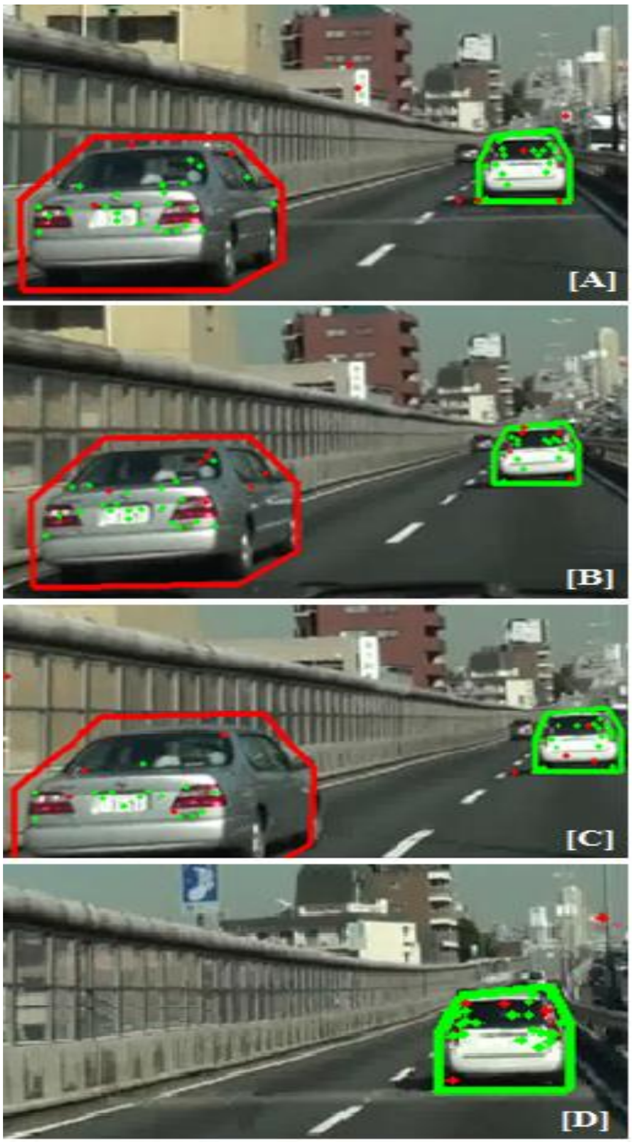


Figure 3.22 – Tracking with generic points

Tracking with Vedaldi SIFT description and tracking with Generic Points Aided SIFT description are compared. The proposed method assures more robust tracking. SIFT itself is sensitive for object view changes and heavy shadow cases are not handled successfully. Generic points aided description is better subject to such view changes and heavy shadow conditions. Under cluttered scenes it may assure more reliable tracking thanks to generic points being more powerful at matching phase. A capture from tracking is given in Figure 3.22. A set of features in the region of interest per car rear object is tracked over successive frames. Matching points are green dots and unmatched points are with red color dots. The detection rate of the proposed method illustrates its robustness and real-time performance.

3.1.2.7 Local Features Approach Assessments

For Generic Points extraction, the presented hypothesis assures the visual context of images even though images are subject to planar perspective transformations. After transformation, the images are visually identifiable and new matches are generated. But the original image descriptors cannot detect these new matches generated by the presented study and these new matching points can have pairs on the original image. Generic points are functioning as a medium of bridge between a point of query image and a point of the compared image. A false matching elimination algorithm is used to increase the robustness. The presented scheme is compared with Vedaldi SIFT implementation and Lowe's matching method and enhanced robustness versus cluttered image scenes is obtained. Feature tracking is realized in video images. Generic features are integrated with RAMOSAC tracker. To provide real-time efficiency, CUDA GPU implementation of Vedaldi SIFT is used.

3.2 Global Features Approach

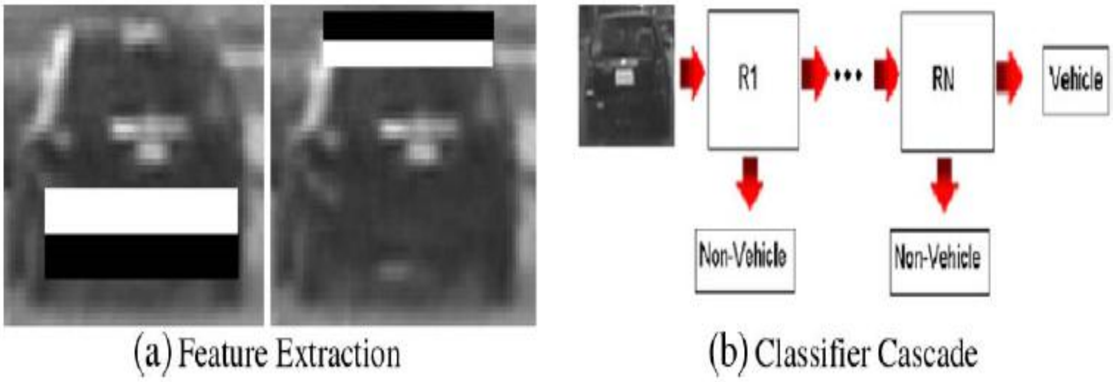
Global feature extraction schemas HMAX and Haar-Like Simple features are analyzed and Haar-Like Simple Features approach is adopted and used for developing our second proceeding.

3.2.1 Real Time Object Detection Using Image Global Features and Detection Based Object Tracking

Data sets consisting of on-road video records are used and the developed proceeding is subjected to assessment.

3.2.1.1 Vehicle Detection Using Haar Features

To detect candidate vehicles, we used boosted cascade of simple Haar like rectangular features, as was introduced by Viola and Jones in the context of face detection. Various studies have been used this approach for vehicle detection. Haar-like rectangular features are well suited to object shape detection. These features are sensitive to image global features like edges, bars, vertical and horizontal details, and symmetric structures. Examples of Haar-like rectangular features used in vehicle detection are seen in Figure 3.23 (a). The original algorithm used by Viola and Jones allows for rapid object detection that can be suitable for a real time system, use of integral images provides fast and efficient feature extraction. Extracted resulting values are effective weak learners, which are then classified by Adaboost.



(a) Feature Extraction (b) Classifier Cascade

Figure 3.23 - (a) Example of Haar-like features. (b) Cascade of boosted classifiers.

Adaboost performs classification based on a weighted majority vote of weak learners. It is a discriminative learning algorithm. A cascade of classifier stages is constructed with Adaboost learning. With very little effort, preliminary stages in the cascade eliminate most of the false negative regions. Scores that are computed from feature extraction makes the decision for rejection at each stage. Candidate vehicles are eliminated stage by stage within the cascade, and remaining candidates after final stage, considered as positive detections, see Figure 3.23 (b).

One of the most relevant contributions of Viola and Jones is the introduction of integral image use. Integral images are defined like lookup tables in the form of a matrix same size of the original image.

Each element of the matrix contains the sum of all pixels located on the up-left region of the original image. This provides effective processing; using only 4 lookups, sum of rectangular areas, and so difference between two rectangular areas in the image at any position and scale is known.

Haar-like features are extracted with box filters which tend to have behavior like Haar wavelets of degree 1.

A Haar-like feature feature is extracted by summing up the pixel intensities over two adjacent rectangles and then subtracting the two sums. Basically, it is the difference between pixel intensity sums over two rectangles, side to side regions total pixel intensity changes raises a global feature usefull for a weak learner like adaboost.

This difference is then used to categorize subsections of an image. For example, let us say we have an image database with human faces. It is a common observation that among all faces the region of the eyes is darker than the region of the cheeks. Therefore a common haar feature for face detection is a set of two adjacent rectangles that lie above the eye and the cheek region. The position of these rectangles is defined relative to a detection window that acts like a bounding box to the target object (the face in this case).

In the detection phase of the Viola–Jones object detection framework, a window of the target size is moved over the input image, and for each subsection of the image the Haar-like feature is calculated. This difference is then compared to a learned threshold that separates non-objects from objects. Because such a Haar-like feature is only a weak learner or classifier (its detection quality is slightly better than random guessing) a large number of Haar-like features are necessary to describe an object with sufficient accuracy. In the Viola–Jones object detection framework, the Haar-like features are therefore organized in something called a classifier cascade to form a strong learner or classifier.

3.2.1.2 Vehicle Detection Verification

Step 1:
İf candidate vehicle bounding box has not an x symmetry axis, then detection is rejected.

Step 2:
Vehicles tend to have lots of horizontal lines. At least 4 horizontal lines, each one longer than 10 pixels should rely inside the box.

Step 3:
Detection history may be a healthy indicator for tracking and as well for detection. Candidate vehicle bounding box center must have 28 close matches among last 30 frames in which detection occurred. Distance between the center of candidate vehicle bounding box and the center of detection from the recorded history must be twice smaller than the radius of bounding box. This prevents arising of false positives.

If either step 2 or Step 3 is not verified, then detection is rejected.

3.2.1.2.1 X-Symmetry

Colored image data relying inside bounding box of detected candidate vehicle is analyzed and symmetry is searched around vertical axis, based on image data. To do this a sliding window is used, which is of half size of the bounding box.

In Figure 3.24, bounding box of size (2w, 2h) is convolved with the sliding window W. W slides pixel per pixel, each time image data corresponding W is scanned from bottom to up and the sum of colored pixels is noted as $\sum 1$.

Similarly, each time the data corresponding to the left side window L is scanned from bottom to up and pixel values are summed, then noted as $\sum 2$. At each slide operation difference of $\sum 2$ and $\sum 1$ is noted as Δi. At the end of all iterations x-axis value corresponding to Δi with minimum value over all iterations, is considered as candidate symmetry axis. If the value hold by corresponding Δi is not sufficiently small then symmetry detection is rejected. Windows L and W have variable width, varying between [0, w], w is half width of bounding box.

Colored data is considered to detect symmetry axis. We used 3 channels colored and subsampled smooth images to achieve this. Only 1 channel value is considered at each iteration. Considering 3 channels together and 1 channel together provided similar results.

Alternatively, to detect symmetry axis we used edge images, which are 1 channel images, containing only white or black colored contours of the original image. Edge images are obtained with Canny edge detection algorithm. Before applying Canny algorithm, colored images are blurred and subsampled, to reduce noise and increase performance.

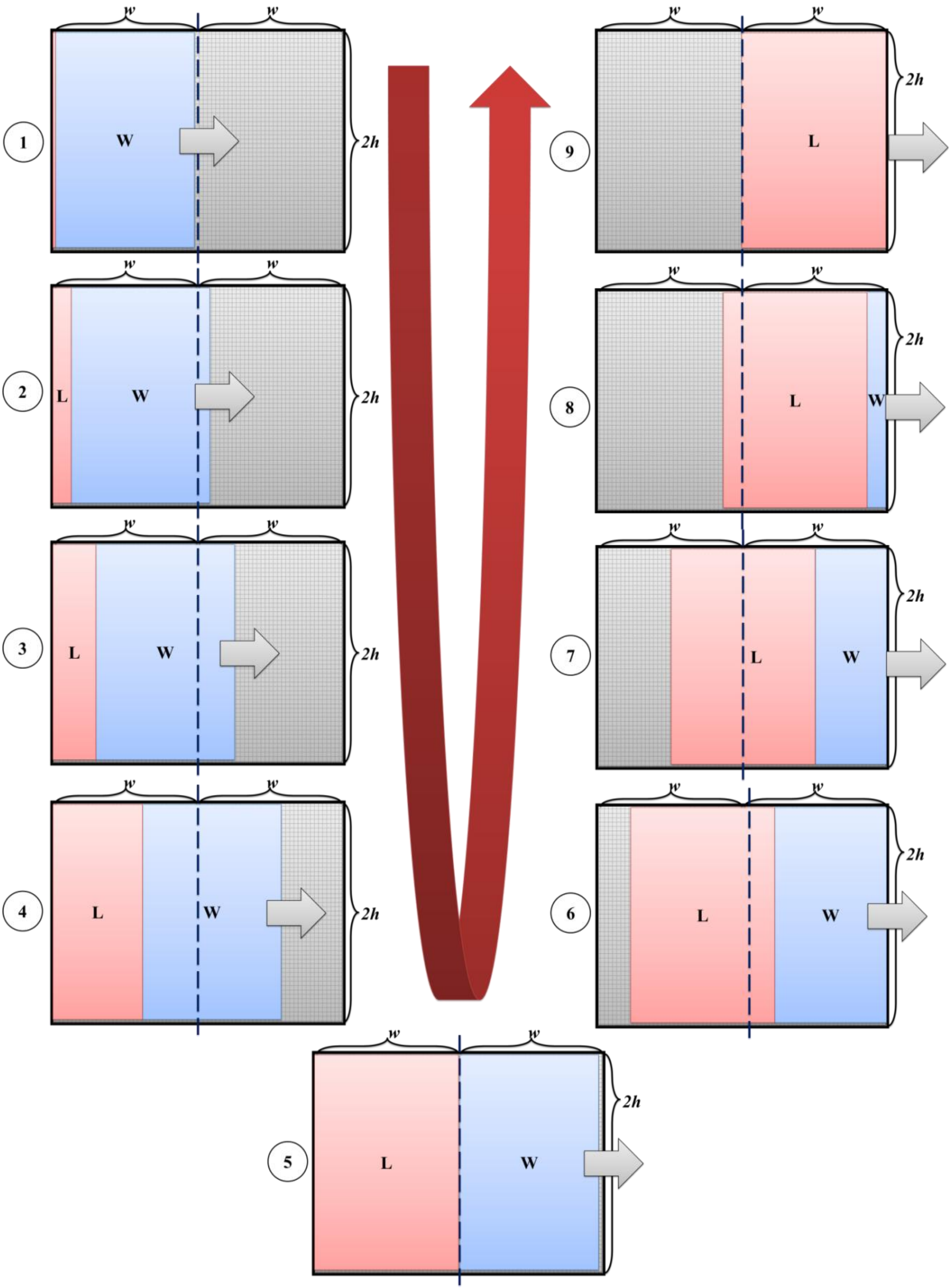


Figure 3.24 - Symmetry search

Similar sliding window approach is applied and at each iteration the ratio between white pixels count to whole pixel count corresponding to sliding window is calculated. Similarly left side window L is scanned and same ratio is extracted. The two ratios belong to L and W are compared at each iteration and the ratio pair providing minimal delta is considered to detect symmetry axis. If the two ratios are not sufficiently close to each other, then symmetry detection is rejected.

The two approaches are compared and we adopted the first approach which is considering image data comparison over 1 color channel.

3.2.1.2.2 Horizontal Edges

Analyzing low level vehicle image global features, one can consider that shape of a vehicle produces horizontal edges significantly more than on-road background textures produce and these horizontal edges are significantly long and condensed parallel to each other inside the vehicle bounding box.

Quantification of prominent horizontal edges is a good indicator when validating a candidate vehicle given with a bounding box. To achieve this we used edge images. İmages are first blurred and then subsampled by 2, irrelevant noise data is eliminated.

Afterwards, Canny edge detection algorithm is used to extract edge images consisting of only white and black colored data, containing only relevant edges marked with white color over black background, see Figure 3.26.

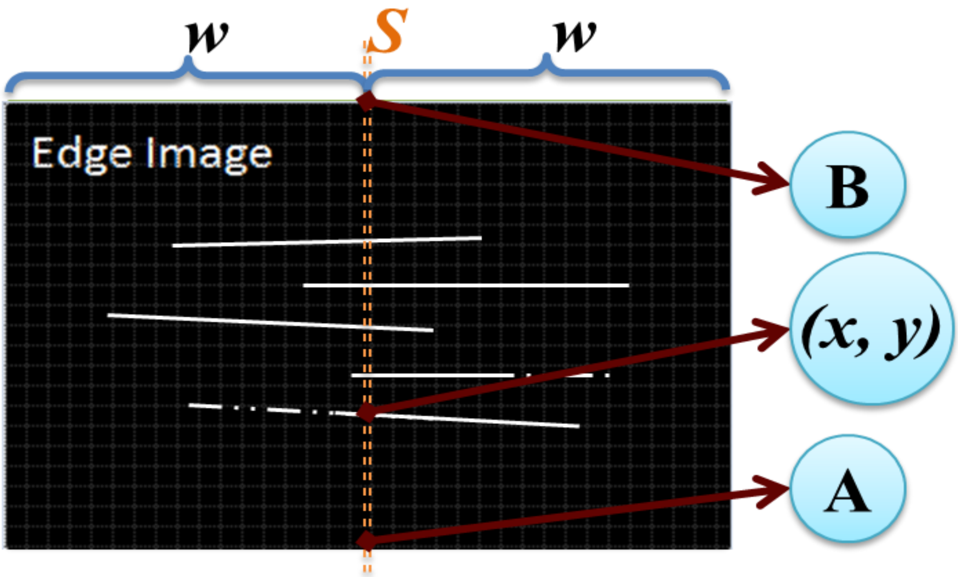


Figure 3.25 - Edge image horizontal line search

Edge Image (candidate vehicle bounding box) is scanned beginning from bottom to top and horizontal edges are searched with a suitable algorithm. Scan is started from the point A (see Figure 3.25) and iteratively goes along with bounding box symmetry axis S, up to the point B (see Figure 3.25). Scan step is 1 px and at each iteration, say for a given central point (x, y) located on symmetry axis S, continuous line search is done with horizontal scans towards left and right directions. Left side horizontal scan starts from the point (x, y), ends up at the point (x-w, y) and respectively right side horizontal scan starts from (x, y), ends up at (x+w, y).

Scan step used for horizontal scans is 1 px. At each iteration of horizontal scan, value of the pixel (x, y) on which scan step corresponds, is checked. If value of the pixel (x,y) is white and (x,y-1) and (x,y+1) is black then next iteration is continued with (x+1,y), otherwise values of vertical neighborhood pixels (x,y+1) and (x,y-1) are checked; if (x,y+1) is white then next step is continued with (x, y+1); respectively if (x, y-1) is white then next step is continued with (x, y-1), if x is not incremented more than 1 iteration than scan is ended up. This provides tolerance for detection of horizontal edges which are not compactly parallel to y axis but considerable as globally horizontal relative to bounding box base.

We use another tolerance providing factor, which is a maximum gap parameter set to 2 pixels in our detection system. According to the walking algorithm mentioned above, if (x, y) is black then we check (x,y-1) and (x,y+1). If those upper and lower neighbors are also black then we give a chance and re-iterate algorithm with (x +1, y). If still no white pixel is detected, one more chance is given. So gaps on edges are tolerated up to 2 pixels. Gaps over edges may arise like it's seen in Figure 3.26, marked with B.

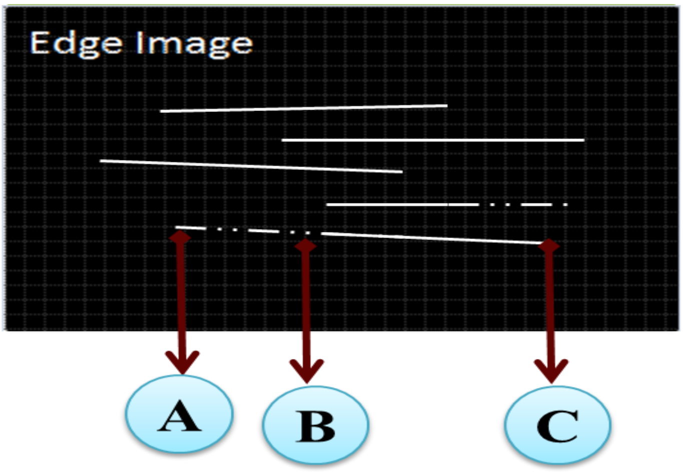


Figure 3.26 - Discontinuity on lines

The points A and C seen in Figure 3.26 above, are the horizontal scan end points, these are detected line en up points.

At the end of 1 horizontal scan, using line end up points, we check the obliquity of line. If the angle between line and horizontal axis y is more than 10 degrees then detected horizontal edge is rejected.

Another constraint that we set for a horizontal line is the strength of line; if detected horizontal lines length is less than 10 pixels then it is rejected. We consider only prominent lines for safety of validation.

3.2.1.2.3 Detection History

Temporal information is useful for tracking issues. It's incorporated in a variety of studies. We use temporal information not only for tracking but for detection stage also. It's used as validation constraint besides horizontal line frequency analyzes and x-symmetry search.

At each frame processing, history is updated, only the vehicles detected within last 30 frames are conserved, and others are discarded from history.

When detection is achieved, detected car is searched within the history, if it is already detected within last 30 frames, then its position is updated and it's marked as active.

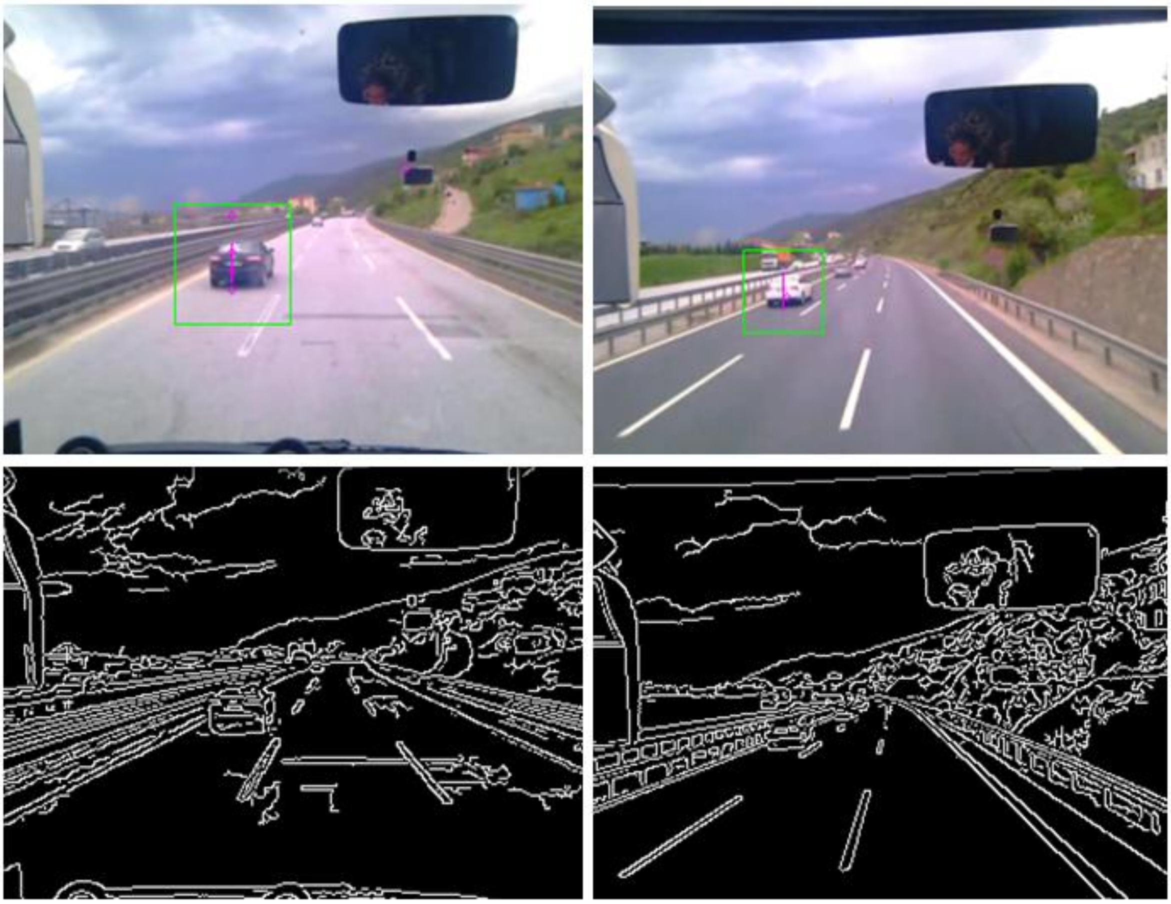


Figure 3.27 - Detection and tracking

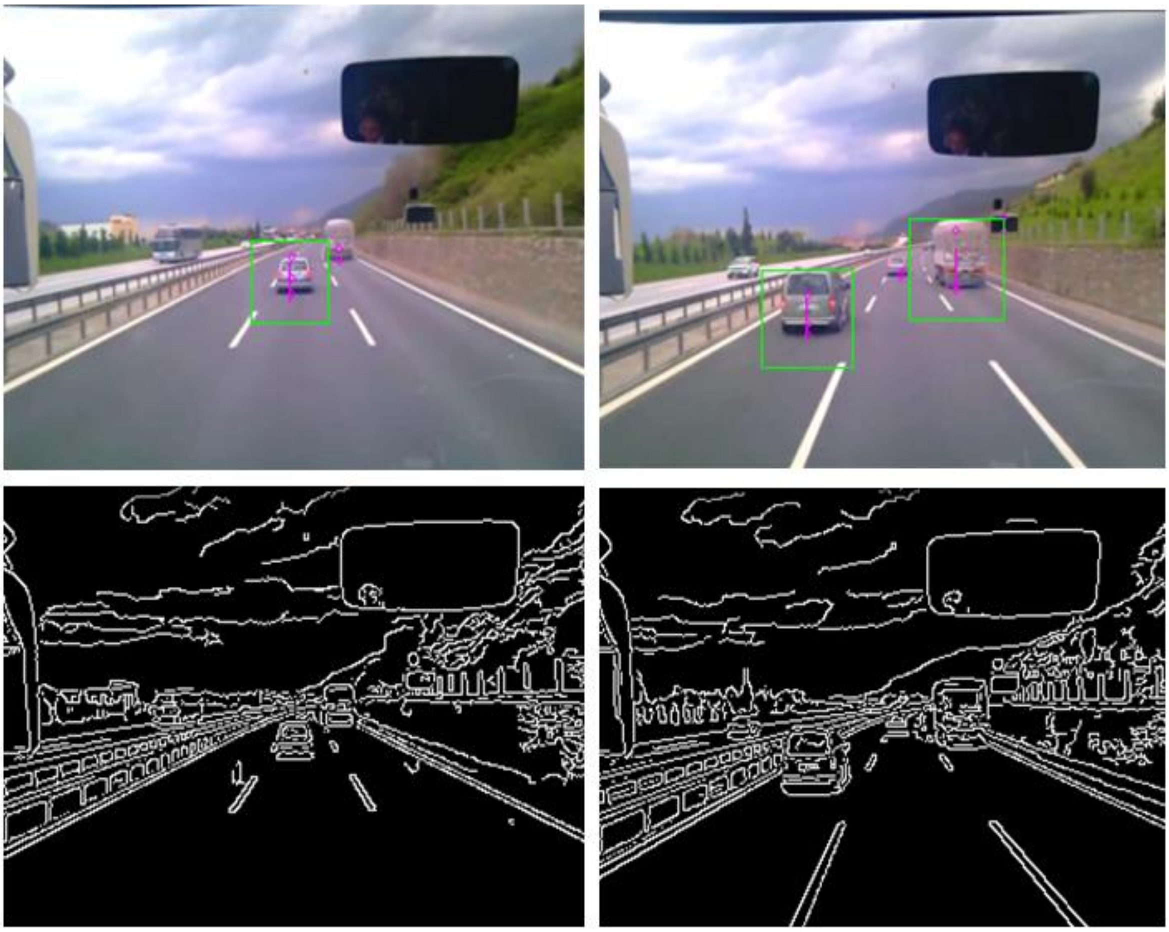


Figure 3.28 - Detection and tracking

The states of vehicles recorded in history are updated per frame process, if a vehicle is not re-detected within last 15 frames, then it’s marked as invalid in history. But it’s not discarded from the history, unless it’s redetected within last 30 frames.

While drawing detected vehicles on the screen per frame process, we do not draw only the vehicles detected and validated at current frame, all of the vehicles from the history with active states are drown per frame. Unless a detected vehicle is lost over 15 successive frames, we do not discard it.

A suite of capture from our real time implementation is seen in Figure 3.27 and Figure 3.28.

3.3 Experimental Results

Local and global features based proceedings are subjected to assessment in this section. They are compared with State of art techniques results, reflected weak and strong aspects per approach.

3.3.1 Assessment of Local Features Approach

To assess local feature based description methods, multi-view car rear dataset is used, (Ozuysal, 2009). This data set includes images subject to shadows, rotation, different principal objects in the background and cluttered scenes.

Car surfaces are less textured and poor in including local distinctive information. SURF, SIFT and SURF GPRD performances are given in Figure 3.29 and Figure 3.30 : Tewenty sequences of cars used, captured as they rotate by 3 degrees, beginning from -45 to +45 degrees. 30 snapshots per car are used with 45 degrees of view changes.

Frontal car rear is compared with remaining captures of the same car. At least 3 correct matches are set for a car to match. If fewer than 3 correct matches then matching is rejected.

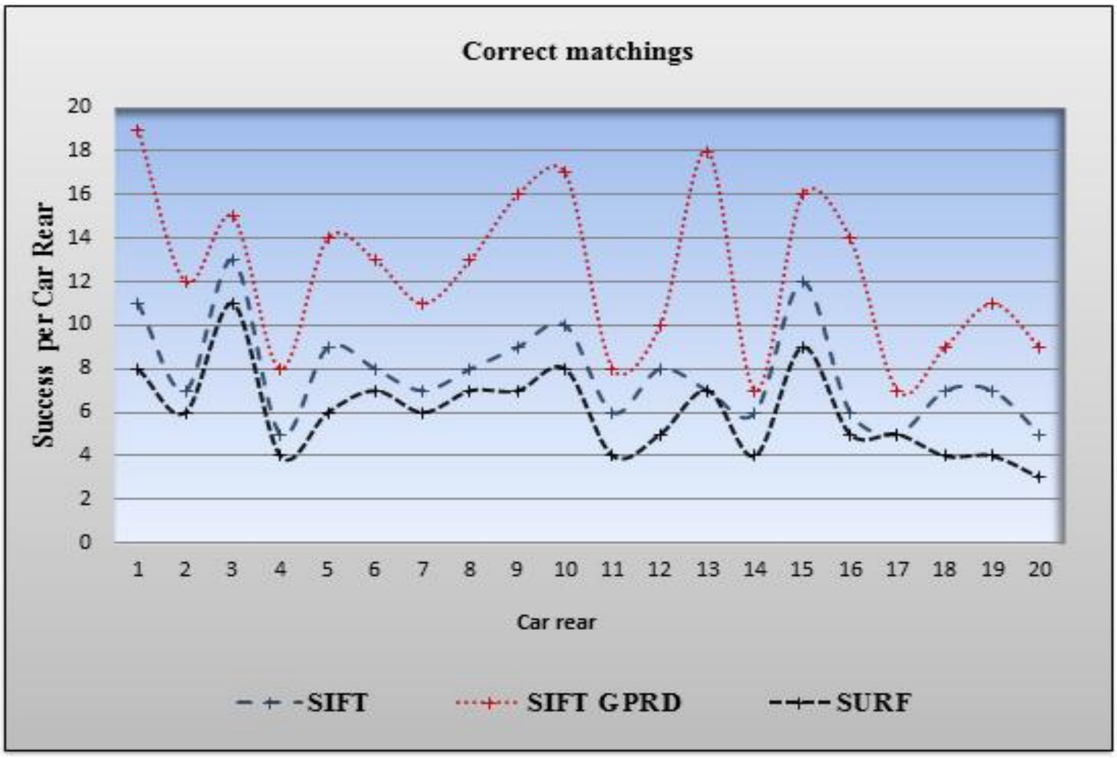


Figure 3.29 - Tracking with Local features, correct matchings' per car rear

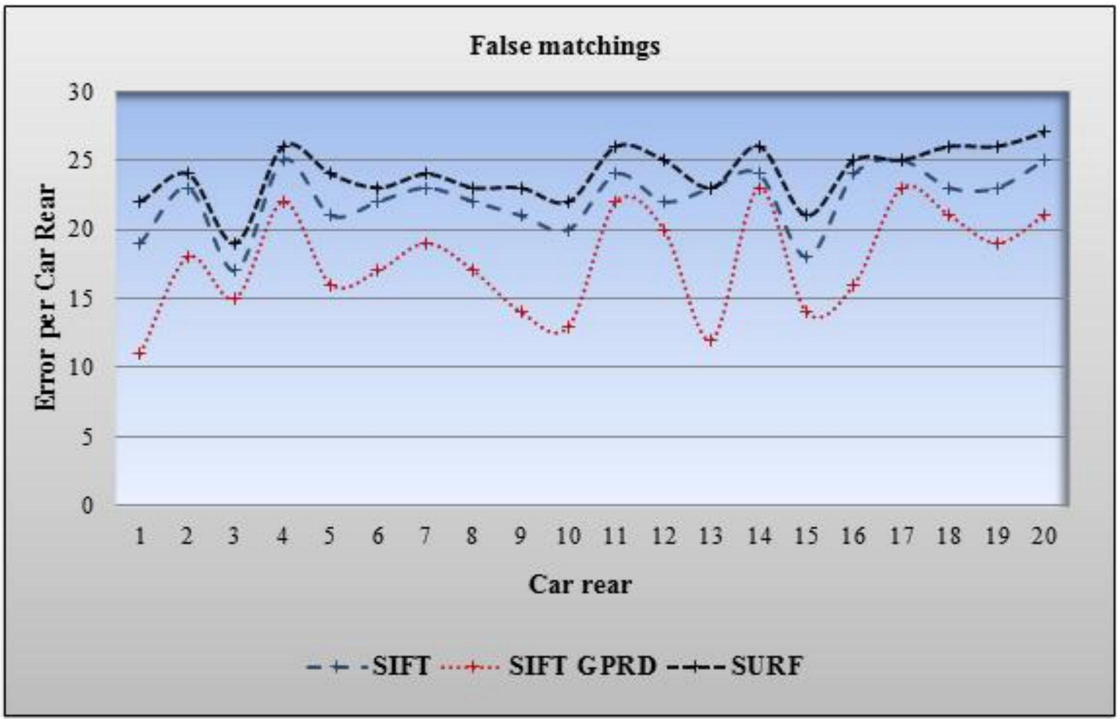


Figure 3.30 - Tracking with Local features, false matchings' per car rear.

3.3.2 Assessment of Global Features Approach

Metric definitions below are used for assessment of recognition and tracking with global features, results are reported in Table 1 and Table 2.

TPR (True Positive Rate) is a measure of recall and localization; it is defined by:

TPR = detected vehicles / total number of vehicles

FDR (False Detection Rate) is a measure of precision and localization; it is defined by:

FDR = false positives / (detected vehicles + false positives)

FP/Frame (False Positives per Frame) is the measure of robustness, localization and scalability; it is defined by:

FP/Frame = false positives / total number of frames

TP/Frame (True Positives per Frame) is the measure of robustness; it is defined by:

TP/Frame = true positives / total number of frames

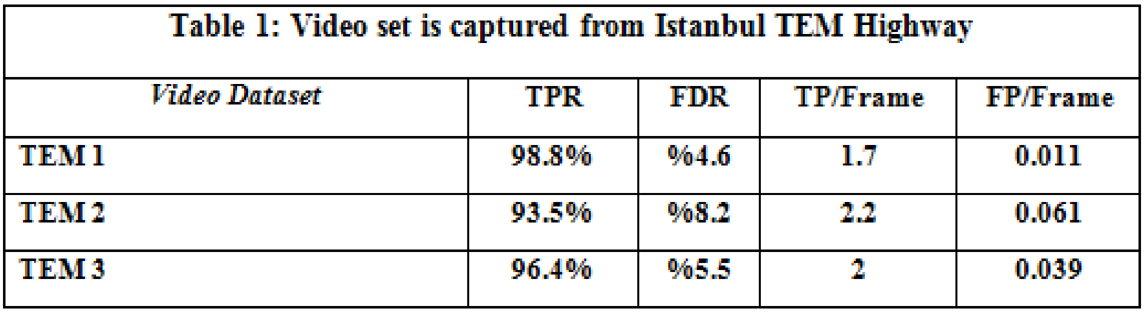


Table 1: Video set is captured from Istanbul TEM Highway

Video Dataset	TPR	FDR	TP/Frame	FP/Frame
TEM 1	**98.8%**	**%4.6**	**1.7**	**0.011**
TEM 2	**93.5%**	**%8.2**	**2.2**	**0.061**
TEM 3	**96.4%**	**%5.5**	**2**	**0.039**

Figure 3.31 - Istanbul TEM Highway video dataset metrics.

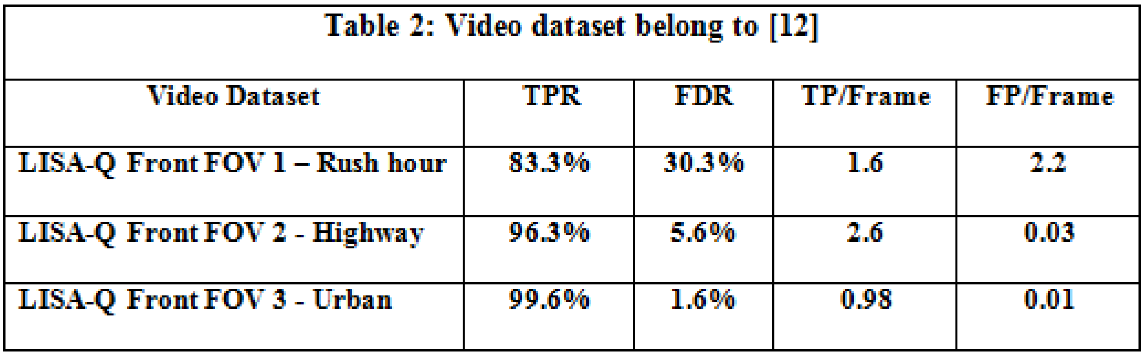


Table 2: Video dataset belong to [12]

Video Dataset	TPR	FDR	TP/Frame	FP/Frame
LISA-Q Front FOV 1 – Rush hour	83.3%	30.3%	1.6	2.2
LISA-Q Front FOV 2 - Highway	96.3%	5.6%	2.6	0.03
LISA-Q Front FOV 3 - Urban	99.6%	1.6%	0.98	0.01

Figure 3.32 - LISA-Q Front FOV video dataset metrics 1

Thanks to validation steps that we applied, our recognition and tracking system produces a less error prone result close to given by ALVeRT (Sivaraman & Trivedi , 2010). We have similar or better results with ALVeRT in Lisa-Q Front FOV 2 and Lisa-Q Front FOV 3. However in Lisa-Q Front FOV 1(Rush hour), our system is outdated by ALVeRT. On the other hand as a passively trained model, our systems metrics are better than the result of passively trained state of art model, given in (Sivaraman & Trivedi, 2010). Combination of our validation steps with an active learning method can arguably produce better performance than ALVeRT. To report the contribution of validation steps we use, the metrics below are measured over bounding boxes extracted with preliminary detections, see Table 3.

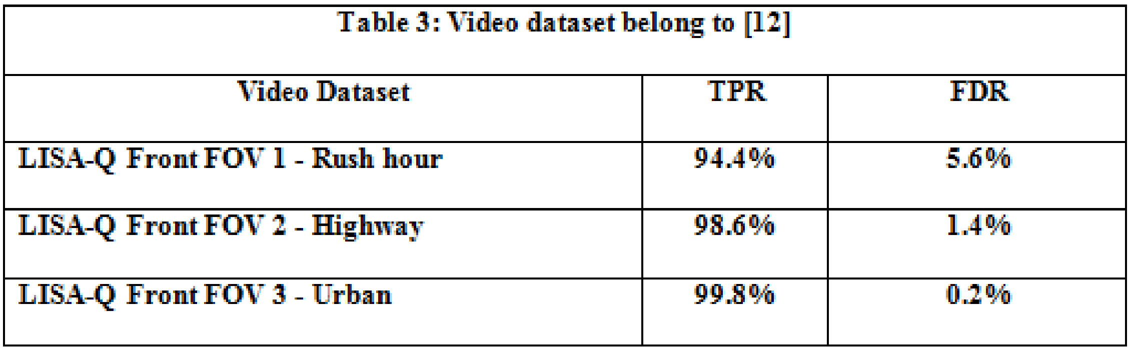


Table 3: Video dataset belong to [12]

Video Dataset	TPR	FDR
LISA-Q Front FOV 1 - Rush hour	94.4%	5.6%
LISA-Q Front FOV 2 - Highway	98.6%	1.4%
LISA-Q Front FOV 3 - Urban	99.8%	0.2%

Figure 3.33 - LISA-Q Front FOV video dataset metrics 2

Some captures from assessment of global feature based detection and tracking approach are seen in Figure 3.34 and Figure 3.35:

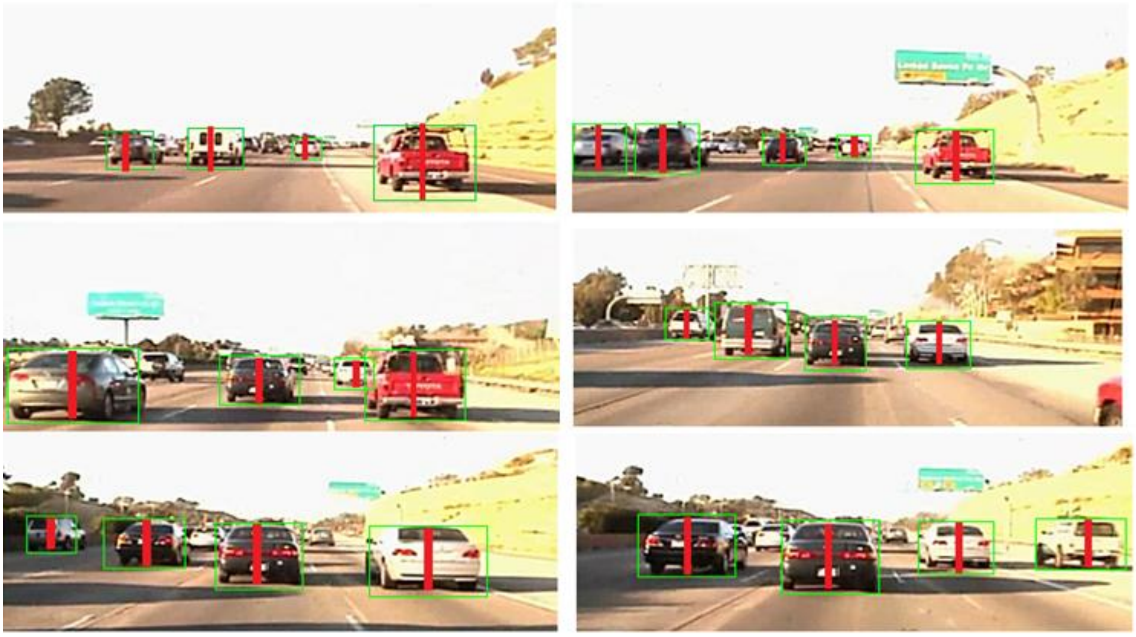


Figure 3.34 - A capture from our recognition and tracking schema on LISA-Q Front FOV 1 – Rush hour

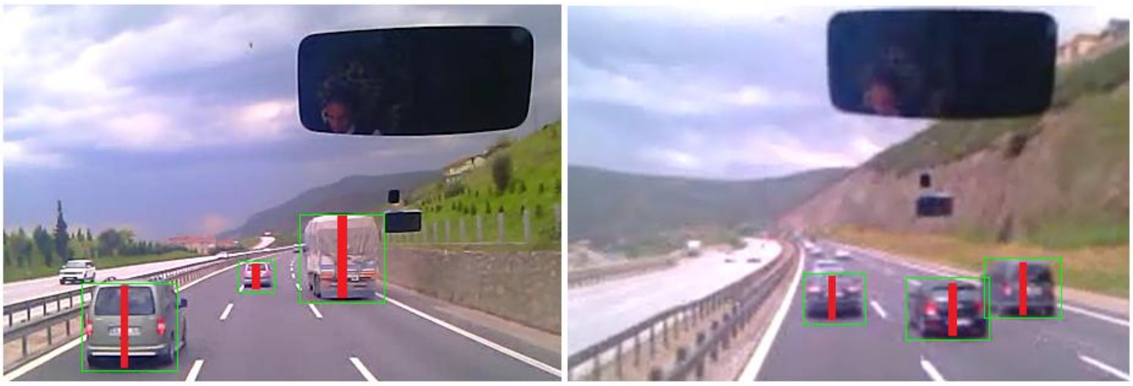


Figure 3.35 - A capture from our recognition and tracking schema on Istanbul TEM Highway

4 Conclusion

State of art image feature extraction methodologies and their implementations are analyzed. Local feature extraction methodologies SIFT and SURF are discussed and compared. Haar responses in SURF extract local features invariant to illumination changes. It is invariant against image noise, rotation changes as well. However SIFT is more robust than SURF against view point changes. So SURF is less adapted on road vehicle tracking task. However, SURF is 3 times faster than SIFT. Early systems adopted SURF as it achieves real time efficiency. SIFT performs very slow, not suitable for real time issues. However, lately SIFT is implemented on Graphical Processing Unit, which dramatically increased performance of SIFT. Use of SIFT in real time tasks became popular. On the other hand, SIFT is not much successful in object detection issues while it is the ideal way for image scene description tasks. SIFT extracts good local features which are re identifiable. But these points are not successful in describing whole object itself.

Instead local feature extraction methodologies, global feature extraction methodologies are more adapted in object description context. In this field, Haar Like Simple Features and Early Hmax Model are analyzed. The HMAX implementation of JIM Mutch is inspected; feature extraction steps are clarified and reported in this document. Haar Like Simple Features are adopted for global feature extraction and used for describing object models. Car rear data set is submitted to training, obtained indexation is used as a preliminary object detection step in our implementation with Adaboost classifier. This preliminary detection provides region of interests in which global features resembling to features of training object model.

But this is not a strong indicator. This preliminary detection itself does not achieve a feasible system, but covers most of the cases with lots of false positives. A validation algorithm is necessary which will provide robustness and keep as well the real time efficiency.

We inspected low level image characteristics of the targeted category of object, vehicle rears in our case. Vehicles have symmetric features on rears in general. Color channels are used to symmetry search; a texture based symmetry axis is searched. In addition to symmetry search, we inspected contour images extracted from detected Region of Interests. Vehicle textures generate dominant horizontal lines; horizontal dominant lines are searched. Depending on symmetric features and horizontal line frequency, validation of preliminary detection is realized. A tracking algorithm is implemented which uses temporal detection history in which previous vehicle detections are classified each time active and passive. Detection process made more robust and continuous from frame to frame. As a voting medium, the temporal detection history is used for preliminary detection validation as well. We proposed a new proceeding which enables active safety for on road vehicle navigation.

5 References

Bay H., Tuytelaars T., Gooll L. V. (2008). Speeded-Up Robust Features SURF, in Computer Vision and Image Understanding, Vol. 110, Nr. 3, p. 346 - 359.

Mutch J., URL: http://cbcl.mit.edu/jmutch/hmin/doc/, [accessed July 2013], A Minimal HMAX Implementation.

Moreno P., Marín-Jiménez M. J., Bernardino A., Santos-Victor J., Pérez de la Blanca N. (2007). A Comparative Study of Local Descriptors for Object Category Recognition: SIFT vs HMAX, in IbPRIA 1, Vol. 4477Springer, p. 515-522.

Lowe D. (September 1999). Object Recognition from Local Scale-Invariant Features, in International Conference on Computer Vision, 20--25, Kerkyra, Corfu, Greece, Proceedings, Vol. 2 , p. 1150--1157.

Lowe D. (2004). Distinctive Image Features from Scale-Invariant Keypoints, in International Journal of Computer Vision, Vol. 60 Springer Netherlands, p. 91-110.

Serre T., Wolf L., Poggio T. (2005). Object recognition with features inspired by visual cortex, in CVPR, p.994 – 1000

Ozuysal M., Lepetit V. and Fua P. (June 2006). Pose Estimation for Category Specific Multiview Object Localization, Conference on Computer Vision and Pattern Recognition, Miami, FL.

Sinha S. N., Frahm J. M., Pollefeys M. and Genc Y. (May 2006). GPU-Based Video Feature Tracking and Matching, EDGE 2006, in Workshop on Edge Computing Using New Commodity Architectures, Chapel Hill, May 2006

Sivaraman S, Trivedi M. (2010). A general active-learning framework for on-road vehicle recognition and tracking, in IEEE ITS, 11(2), 267-276

Yildiz R, Acarman T. (July 2012). Image Feature Based Video Object Description and Tracking, in IEEE International Conference on Vehicular Electronics and Safety (ICVES), Page(s): 405 – 410

Ziegler G., Tevs A., Theobalt C., Seidel H. P. (2006). GPU point list generation through histogram pyramids, in Proc. of VMV'06, p. 137--141, Aachen, Germany.

Biographical Sketch

Ramazan Yıldız was born in Doğanhisar/Turkey in 1981. He accomplished secondary school at Doğanhisar Anatolian High School in 1998 and high school education in Yozgat Science High School with full scholarship in 2001. He achieved national examination for University Entrance, in 2002, and was ranked as 2910th over 1.9 million candidate students. With this score, he studied Computer Engineering and achieved his B.S. degree in 2007 at University of Galatasaray. He continued MSc degree program in Computer Science with Computer Vision and Computer Graphics at INSA Lyon, France in 2007 and 2008. In 2010, he started MSc degree program in Computer Science at University of Galatasaray and accomplished in 2013. He started his professional career in 2008 at Nortel Networks Netaş as an international R&D software design engineer. He carried out domestic and international projects in collaboration with foreign lead engineers. He is married, speaks English and French at advanced levels and still developing technology for Netaş.