UNIVERSITY OF CALIFORNIA
SANTA CRUZ

ROBUST VISUAL RECOGNITION WITH LOCALLY ADAPTIVE REGRESSION KERNELS

A dissertation submitted in partial satisfaction of the
requirements for the degree of

DOCTOR OF PHILOSOPHY

in

ELECTRICAL ENGINEERING

by

Hae Jong Seo

June 2011

The Dissertation of Hae Jong Seo
is approved:

Professor Peyman Milanfar, Chair

Professor Benjamin Friedlander

Professor Silvio Savarese

Tyrus Miller
Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures

List of Tables

Abstract

Robust Visual Recognition with Locally Adaptive Regression Kernels

by

Hae Jong Seo

Visual recognition concerns identifying objects in an image or actions in a video. Recent progress in network, storage, and computational power makes visual recognition algorithms practical in such applications as surveillance, medical image analysis, visual image search, and more. Although current learning-based frameworks achieve state of the art performance on the existing benchmark databases, they are often slow in training phase and require a large number of training examples. However, a single image can be the only example available in such applications as automatic passport control at airports and image retrieval from the Web. As such, developing a sophisticated descriptor is a key to visual recognition from a single (or a few) examples.

In this work, we propose to use a novel descriptor, *locally adaptive regression kernels* (LARK). LARKs have several advantageous properties: (i) LARK is robust to illumination variations, local deformation, and presence of data uncertainty, (ii) LARKs capture local geometry exceedingly well by taking advantage of geodesic distance over the Euclidean distance, and (iii) LARKs can be computed from multi-dimensional data. Thus, they are applicable to a wide variety of problems, such as generic object detection, action recognition, saliency detection, and more. We also develop a real-time detection framework by efficiently computing LARKs. The comprehensive experimental results presented in each chapter will show the superiority of the LARKs over other descriptors.

Dedicated to my pregnant wife and the baby boy in her.

Acknowledgments

This thesis is not possible without the support from many people around me. I want to especially thank my advisor, Prof. Peyman Milanfar who has taken a great effort to guide me, from theory to professional presentations. He always encouraged and supported my research ideas. His words of encouragement and constructive criticism have helped me through many obstacles during my Ph.D. I have learned from him not only how to conduct high-quality research, but also how to become a man of wisdom.

My deep gratitude also goes to my other thesis committee members, Prof. Benjamin Friedlander and Prof. Silvio Savarese, and my mentor, Dr. Gary Bradski for guiding me during summer internship at Willow Garage. Also, I would like to thank Prof. Michael Elad for providing the valuable feedback to my journal papers.

I thank the former members of MDSP Lab, Assistant Prof. Farsiu, Dr. Robinson, Dr. Shaharam, Dr. Poonawala, Dr. Takeda, Dr. Atsunori, and Dr. Sroubek, and the current members, Priyam, Xiang, Chelwhon, Hossein, and Erik. I am proud to have been a part of MDSP Lab due to a great collaborative research environment and awesome friendship. I also would like to thank my best friend, Assistant Prof. Ji-Hun Seo and brother-in-law, Dr. Taeyoung Ha so much for their seamless inspiration.

Lastly, but most importantly, this thesis is impossible without love, belief, and sacrifice of my parents and parents-in-law. My gratitude to them is beyond what I can express in words. I would like to thank my family and my wife's family for their affection. Needless to say, I owe my lovely wife, Julee Woo, everything.

Santa Cruz, California
June 9th, 2010
Hae Jong Seo

Chapter 1

Introduction

Abstract – We address the *visual recognition problem* which concerns identifying objects in image and actions in video. In order to overcome disadvantages of the popular learning-based recognition paradigm, we introduce a sophisticated descriptor, *locally adaptive regression kernel* (LARK), which measures a pixel level self-similarity based on geodesic distance. LARKs in 2-D (3-D) capture local (space-time) geometric structures exceedingly well, thus they are useful for the visual recognition problem.

1.1 Visual Recognition Problem

Today, a huge number of images and videos are available online and the number is rapidly growing. Thus, visual recognition is a very important component in many computer vision systems. Areas where such systems are deployed are diverse and include such practical applications as surveillance, security, video conference, video forensics, medical image analysis, human-robot interaction, computational photography, mobile vision, etc. as shown in Fig. **1.1**. These applications are get-

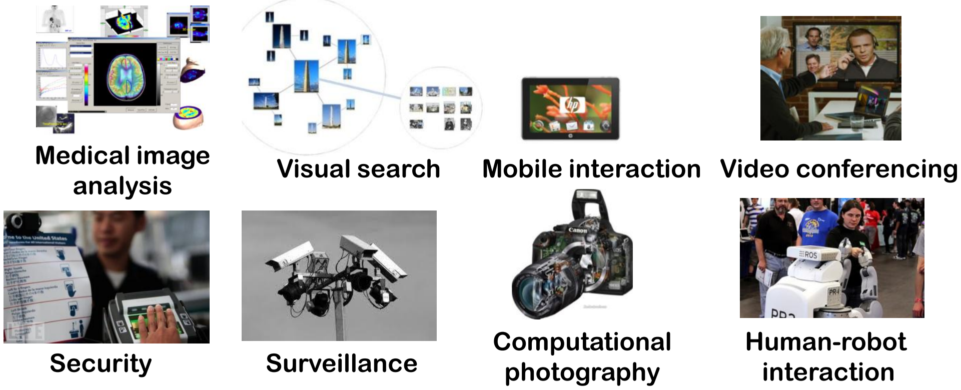


Figure 1.1: There are many practical applications of visual recognition.

ting more popular and pragmatic due to the recent advance in network, storage, and computational power. Recently, the 2-D object recognition problem (including face, pedestrian, and vehicle recognition) and the human action recognition problem have attracted much attention because of the increasing demand for developing real-world surveillance systems. However, visual recognition is a very difficult problem since objects can typically appear in completely different context and under different imaging conditions. Examples of such differences can be wide-ranging, but include differing view points, occlusion, lighting, and scale, rotation changes as shown in Fig. **1.2**. Furthermore, varying speed of actions from person to person can be a challenging factor in recognizing actions.

For the last few decades, learning-based methods for recognizing visual objects have made impressive progress. Typically, learning-based approaches involve generative or discriminative models for each category based on many training examples. In other words, these methods are mostly parametric, relying on visual object models, such as constellation [27, 28], template matching [29, 3], bags of words [30, 31],

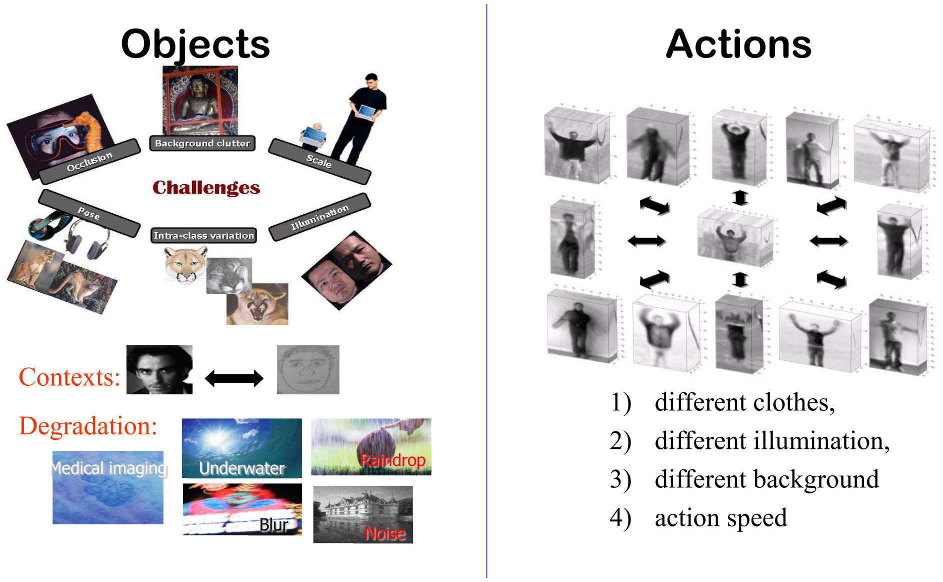


Figure 1.2: There are many challenging conditions that make visual recognition difficult in practice.

or shape models [32, 33], etc. For specific object classes, in particular faces, pedestrians and cars, detectors based on the combination of low-level descriptors combined with modern machine learning techniques have been shown effective. However, in order to achieve sufficient accuracy, these systems require a large number of manually labeled training data, typically hundreds or thousands of example images for each class to be learned. In general, the training phase is slow, and the training is necessary again when there is a new example available. Depending on the database, the system may end up with over-tuning of the parameters. Recently, Caltech 101 [32], Caltech 256 [34], Pascal VOC [35], and ImageNet [36] databases were introduced and played a pivotal role in benchmarking classification methods. While 2-D visual object recognition has recently proved capable of learning a respectably large number of categories (a couple of hundred), 3-D action recognition is still only limited to less than a dozen categories at best

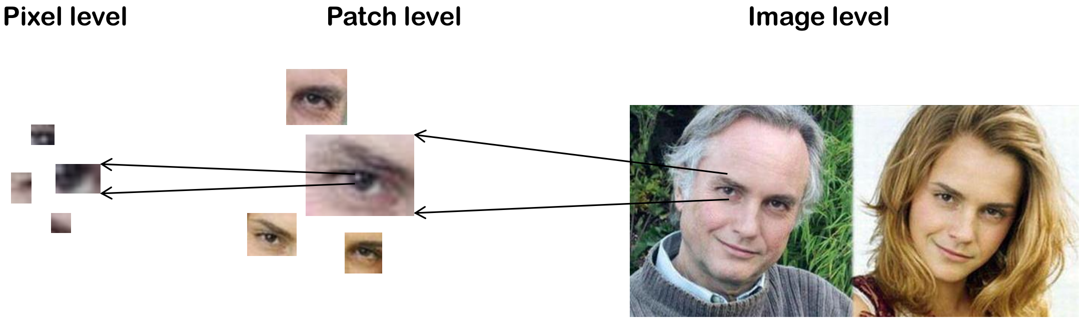


Figure 1.3: LARK descriptors measure a pixel-level self-similarity. We extend this concept to patch-level similarity for saliency detection in Chapter 2 and image-level similarity for object/action detection in Chapter 3.

(6 for KTH [7], 10 for Weizmann dataset [6], and 12 for Hollywood [37]). Furthermore, 2-D object recognition methods were not directly applicable to 3-D action recognition, and thus, completely separate approaches have been proposed for the latter. Indeed, even in terms of evaluation of performance, different criteria and methodologies have been employed for 2-D and 3-D.

There is a recent trend that better deals with visual recognition with the help of large database-driven nonparametric approaches [38, 39, 40]. These approaches are motivated by the realization that there is today a wealth of annotated image data available online. Instead of training sophisticated parametric models, these methods try to reduce the inference problem to matching a query to an existing set of annotated images. For example, these approaches can be very useful for such applications as image retrieval from the web where a single query image is compared with every gallery image in the annotated database, posing an image-to-image matching problem. More generally, by taking into account a set of images which represent intra-class variations, more robust recognition can be achieved. Such sets may consist of observations acquired from a video sequence or by multiple still shots. In other words, classifying a novel set of images into one of the training classes can be achieved through set-to-image or set-to-set matching. As a successful example of set-to-image matching, Boiman et al. [41] showed that a rather simple nearest-neighbor (NN) based image classifier in the space of the local image descriptors is efficient and even outperforms the leading learning-based image classifiers such as SVM-KNN [42], pyramid match kernel (PMK) [43, 31]. Action recognition methods such as those in [21, 22, 44, 45] which aim at recognizing actions based solely on one query support these ideas as well. In fact, companies such as Viewdle[1] and Videosurf[2] are currently providing a video search engine based on rather simple versions of such ideas.

1.2 Contributions

In this thesis, we propose a robust visual recognition system from a single (or a few) examples using a novel descriptor, *locally adaptive regression kernels* (LARK) [46]. LARK basically measures a pixel-level similarity in a local window. We extend the concept of pixel-level similarity used for LARK to patch-level similarity and image-level similarity (see Fig. **1.3**) to build a nonparametric detection framework. The proposed framework requires minimal assumptions with the least number of training examples. With LARKs, we can learn a wide variety of objects from relatively few examples and recognize them in real time. A key intuition behind the success of the proposed system with LARKs[3] is that we consistently use the data-adaptive kernel density estimation idea from LARK descriptor computation to nonparametric detection framework (see

[1] `http://www.viewdle.com`

[2] `http://www.videosurf.com`

[3] The comprehensive experimental results in each chapter will demonstrate superiority of the LARKs over other descriptors in our nonparametric detection framework.

Appendix A for more detail.) In the rest of this introductory chapter, we describe key ideas, properties of LARK and provide comparison to other state of the art descriptors. We develop applications for image and video in the following chapters. Specifically, this thesis is structured as follows:

▷ **Chapter 2 - Saliency Detection [47, 48]**

In this chapter, we propose a novel nonparametric saliency detection with excellent results in both static (2-D) and space-time (3-D) based on patch-level similarity.

▷ **Chapter 3 - Generic Object and Action Detection [49, 50, 51, 45]**

Extending the knowledge of the patch-level similarity to image-level similarity, we introduce a unified, generic object and action detection framework, which produces state of the art performance.

▷ **Chapter 4 - Real-time Robot Vision with Scalable LARK descriptors [52]**

In this chapter, we develop a real-time visual recognition system by speeding up the computation of LARK and employing coarse-to-fine pyramid search in conjunction with hierarchical clustering of multiple examples. The system can learn a wide variety of objects using relatively few examples and recognize them in real time.

▷ **Chapter 5 - Other Applications [53, 54]**

There are two more applications where LARKs are successfully applied to; 1) automatic change detection and 2) face verification.

Finally, in Chapter 6, we conclude the thesis and discuss possible topics for future research.

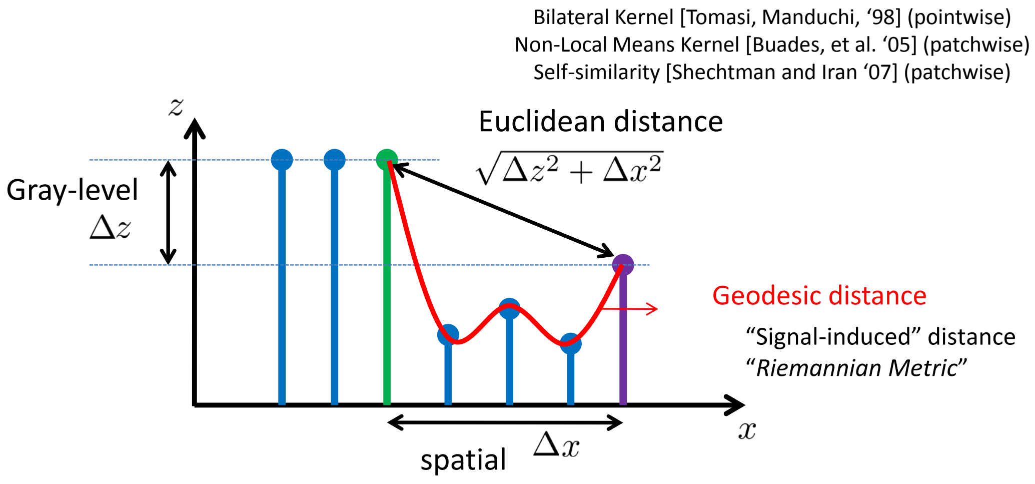

Bilateral Kernel [Tomasi, Manduchi, '98] (pointwise)
Non-Local Means Kernel [Buades, et al. '05] (patchwise)
Self-similarity [Shechtman and Iran '07] (patchwise)
z
Euclidean distance
$\sqrt{\Delta z^2 + \Delta x^2}$
Gray-level
Δz
Geodesic distance
"Signal-induced" distance
"*Riemannian Metric*"
x
spatial
Δx

Figure 1.4: Difference between Euclidean distance and geodesic distance (the shortest path along the manifold) in 1-D signal.

1.3 Locally Adaptive Regression Kernels (LARK)

1.3.1 LARK in 2-D

LARK effectively and efficiently captures local geometric structure by taking advantage of self-similarity based on gradients. In order to measure the similarity of two pixels, in general, we can naturally consider both the spatial distance (Δx) and the gray level distance (Δz) (See Fig. **1.4**.) The most simple way to incorporate the two Δ's is the Euclidean distance between points. However, a much more effective way to combine the two Δ's is to define a "signal-induced" distance [55] which basically stands for a distance between the points measured along the shortest path on the signal manifold (a.k.a. the geodesic distance).

Suppose that we consider the parameterized image surface $S(x_1, x_2) = \{x_1, x_2 , z(x_1, x_2)\}$, embedded in the Euclidean space $\mathbb{R}^3$ as shown in Fig. **1.5**: x_1, x_2 are spatial

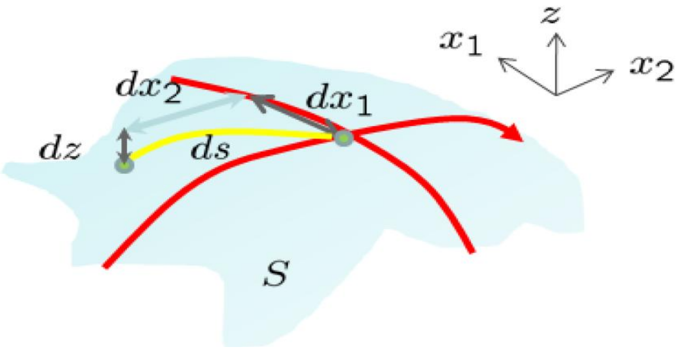


Figure 1.5: The geodesic distance in 2-D surface can be computed as squared arclength ($ds^2 = dx_1^2 + dx_2^2 + dz^2$).

coordinates. The differential arclength on the surface is given by $ds^2 = dx_1^2 + dx_2^2 + dz^2$. Applying the chain rule, we have

$$dz(x_1, x_2) = \frac{\partial z}{\partial x_1} dx_1 + \frac{\partial z}{\partial x_2} dx_2 = z_{x_1} dx_1 + z_{x_2} dx_2, \tag{1.1}$$

where z_{x_1}, z_{x_2} are first derivatives along x_1, x_2 respectively. Plugging $dz(x_1, x_2)$ into the arclength definition, we have

$$\begin{aligned} ds^2 &= dx_1^2 + dx_2^2 + dz^2, \\ &= dx_1^2 + dx_2^2 + (z_{x_1} dx_1 + z_{x_2} dx_2)^2, \\ &= [dx_1 \quad dx_2] \begin{bmatrix} z_{x_1}^2 + 1 & z_{x_1} z_{x_2} \\ z_{x_1} z_{x_2} & z_{x_2}^2 + 1 \end{bmatrix} \begin{bmatrix} dx_1 \\ dx_2 \end{bmatrix}, \\ &= \Delta\mathbf{x}^\top \mathbf{C} \Delta\mathbf{x} + \Delta\mathbf{x}^\top \Delta\mathbf{x}, \end{aligned} \tag{1.2}$$

where $\Delta\mathbf{x} = [dx_1, dx_2]^\top$, and $\mathbf{C}$ is the local gradient covariance matrix (a.k.a. structure tensor).

We measure this arclength between a center pixel and surrounding pixels in a local window (see Fig. **1.6**.)[4] The effect of $\Delta\mathbf{x}^\top \Delta\mathbf{x}$ in the local small window is trivial

[4] In a particular example of computing LARK of size 5×5 shown in Fig. **1.6**, $\Delta\mathbf{x}_{13}$ is $[0, 0]^T$ since $\mathbf{x}_{13}$

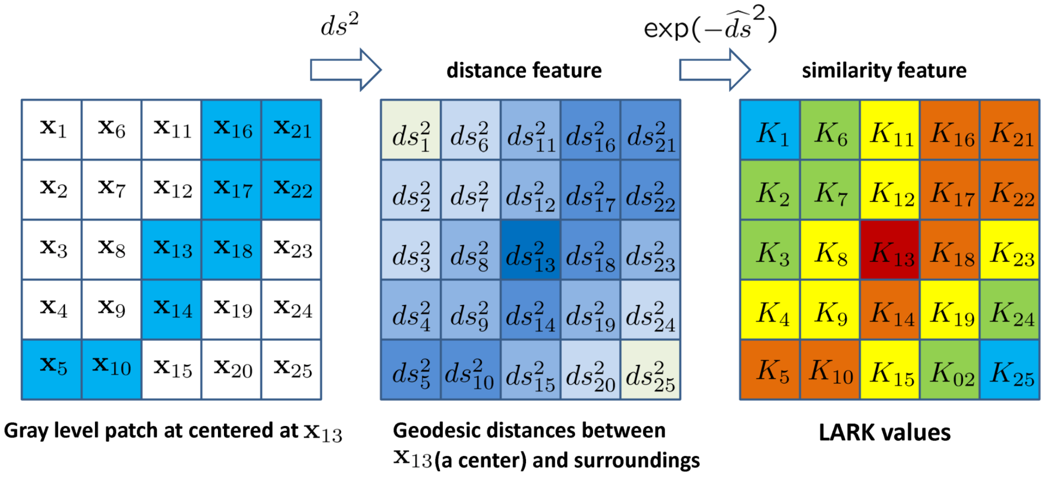

ds^2
$\exp(-\widehat{ds}^2)$
distance feature
similarity feature
$\mathbf{x}_1$ $\mathbf{x}_6$ $\mathbf{x}_{11}$ $\mathbf{x}_{16}$ $\mathbf{x}_{21}$
$\mathbf{x}_2$ $\mathbf{x}_7$ $\mathbf{x}_{12}$ $\mathbf{x}_{17}$ $\mathbf{x}_{22}$
$\mathbf{x}_3$ $\mathbf{x}_8$ $\mathbf{x}_{13}$ $\mathbf{x}_{18}$ $\mathbf{x}_{23}$
$\mathbf{x}_4$ $\mathbf{x}_9$ $\mathbf{x}_{14}$ $\mathbf{x}_{19}$ $\mathbf{x}_{24}$
$\mathbf{x}_5$ $\mathbf{x}_{10}$ $\mathbf{x}_{15}$ $\mathbf{x}_{20}$ $\mathbf{x}_{25}$
ds_1^2 ds_6^2 ds_{11}^2 ds_{16}^2 ds_{21}^2
ds_2^2 ds_7^2 ds_{12}^2 ds_{17}^2 ds_{22}^2
ds_3^2 ds_8^2 ds_{13}^2 ds_{18}^2 ds_{23}^2
ds_4^2 ds_9^2 ds_{14}^2 ds_{19}^2 ds_{24}^2
ds_5^2 ds_{10}^2 ds_{15}^2 ds_{20}^2 ds_{25}^2
K_1 K_6 K_{11} K_{16} K_{21}
K_2 K_7 K_{12} K_{17} K_{22}
K_3 K_8 K_{13} K_{18} K_{23}
K_4 K_9 K_{14} K_{19} K_{24}
K_5 K_{10} K_{15} K_{02} K_{25}
Gray level patch at centered at $\mathbf{x}_{13}$
Geodesic distances between $\mathbf{x}_{13}$(a center) and surroundings
LARK values

Figure 1.6: How to compute LARK (5 × 5) values centered at $\mathbf{x}_{13}$. First of all, geodesic distance (middle) between $\mathbf{x}_{13}$ and surrounding pixels are computed and transformed to a similarity (right). The darker blue colors are, the smaller distances are (middle). The red color means higher similarity whereas blue color represents smaller similarity (right).

and data-independent, thus we only consider $\widehat{ds}^2 \approx \Delta\mathbf{x}^\top\mathbf{C}\Delta\mathbf{x}$.

We define LARK as a self-similarity between a center and its surroundings as follows:

$$K(\mathbf{C}_l, \Delta\mathbf{x}_l) = \exp\left(-\widehat{ds}^2\right) = \exp\left\{-\Delta\mathbf{x}_l^\top\mathbf{C}_l\Delta\mathbf{x}_l\right\}, \tag{1.3}$$

where $l \in [1, \cdots, P]$, P is the total number of samples in a local analysis window around a sample position at the pixel of interest $\mathbf{x}$.

In theory, $\mathbf{C}_l$ is based on gradients (z_{x_1}, z_{x_2}) at one pixel. However, this $\mathbf{C}_l$ is unstable and prone to noise components in the data. Therefore, we use a collection of first derivatives of the visual signal $\mathbf{z}_l$, which contain the values of a patch Ω_l of pixels centered at position l, along spatial (x_1, x_2) axes. Then, the matrix $\mathbf{C}_l \in \mathbb{R}^{(2\times2)}$ can be

is the center pixel. $\mathbf{C}_{13}$ is an average 2×2 covariance matrix computed from the patch Ω_{13} of size 5×5 centered at $\mathbf{x}_{13}$.

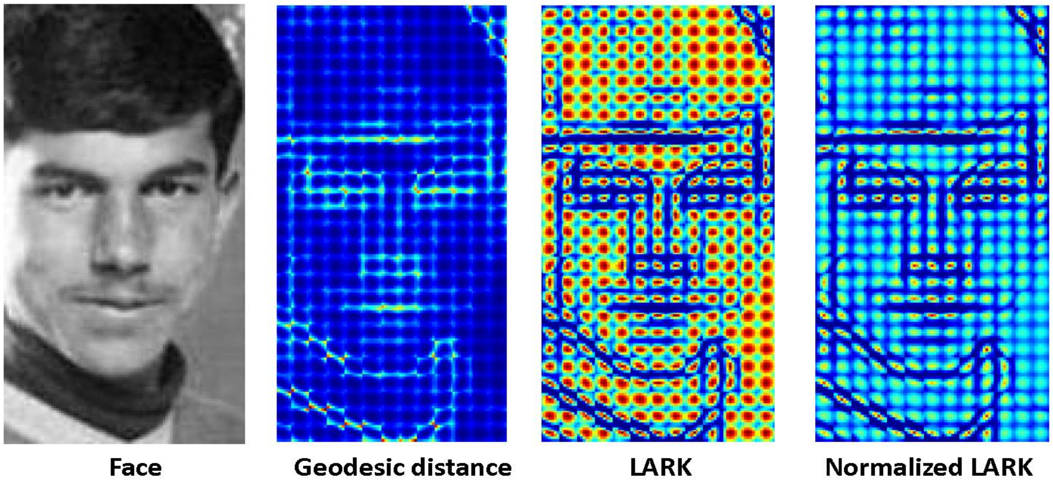


Figure 1.7: Examples of geodesic distance, LARK, and normalized LARK. We show these in non-overlapping patches of a face image for a graphical purpose.

written as follows:

$$\mathbf{C}_l = \sum_{m \in \Omega_l} \begin{bmatrix} z_{x_1}^2(m) & z_{x_1}(m) z_{x_2}(m) \\ z_{x_1}(m) z_{x_2}(m) & z_{x_2}^2(m) \end{bmatrix}. \tag{1.4}$$

This can be interpreted as averaging geodesic distances in a patch to obtain a robust estimation even in the presence of noise and other perturbations.

Another key aspect of LARK lies in the fact that we implicitly smooth the image surface so that the local geodesic distance can be computed in a stable way. Specifically, we perform eigen-decomposition on the (“average") covariance matrix $\mathbf{C}_l$ as follows:

$$\mathbf{C}_l = \lambda_1 \mathbf{u}_1^\top \mathbf{u}_1 + \lambda_2 \mathbf{u}_2^\top \mathbf{u}_2 = s_1 s_2 (\frac{s_1}{s_2} \mathbf{u}_1^\top \mathbf{u}_1 + \frac{s_2}{s_1} \mathbf{u}_2^\top \mathbf{u}_2), \tag{1.5}$$

where λ_1, λ_2 are eigenvalues, $\mathbf{u}_1, \mathbf{u}_2$ are eigenvectors, and $s_1 = \sqrt{\lambda_1}, s_2 = \sqrt{\lambda_2}$ are singular values.

Singular values s_1, s_2 are regularized to avoid numerical instabilities, while both eigenvectors remain the same. Namely,

$$\mathbf{C}_l^{reg} = (s_1 s_2 + \epsilon)^{\alpha} (\frac{s_1 + \tau}{s_2 + \tau} \mathbf{u}_1^{\top} \mathbf{u}_1 + \frac{s_2 + \tau}{s_1 + \tau} \mathbf{u}_2^{\top} \mathbf{u}_2), \tag{1.6}$$

where ϵ, τ, α are set to $10^{-7}, 1, 0.5$ respectively, and they are fixed throughout the thesis. This[5] can be thought of as a non-linear mapping of the eigenvalues in order to turn the structure tensor into a proper Riemannian metric [56]. Fig. **1.6** describes how to compute LARK descriptors of size 5×5. We compute LARKs densely from an image and LARKs are normalized[6] to a unit norm vector (**k**) as shown in Fig. **1.7**.

Invariance Property of LARK Normalized LARKs are robust to illumination changes and the presence of noise as shown in Fig. **1.8**. We also conducted an experiment in order to study the invariance properties of LARK under simple spatial transformations as similarly done in [57]. Specifically, we generated a dataset of 16x16 image patches from a large collection of natural images under different translations and rotations. We assume that descriptors computed from each patch are locally invariant if they do not change significantly under small transformations of the input. We compare the mean squared error (MSE) between the descriptor of the reference patch and the descriptor of the transformed version, averaged over 100 image patches. Fig. **1.9** shows comparisons of LARK against SIFT, SIFT with no invariance, IPSD, and IPSD with no invariance [57] (learned invariant features) under horizontal shift with 25 degree rotation and only horizontal shift. We observe that the normalized MSE of LARK changes more gradually than the MSE produced by other methods as the shift increases in both

[5] The intuition behind τ is to keep the shape of the kernel circular in flat areas ($s_1 \approx s_2 \approx 0$), and elongate it near edge areas ($s_1 \gg s_2$) while the scaling parameter $(s_1 s_2 + \epsilon)^{\alpha}$ is to result in large footprints in the flat (smooth) and smaller ones in the textured areas. The particular form of regularization is not critical in the sense that other non-linear mapping of the eigenvalues can be used as in [56].

[6] We can think of the normalized version of LARK as probability density in a local neighborhood.

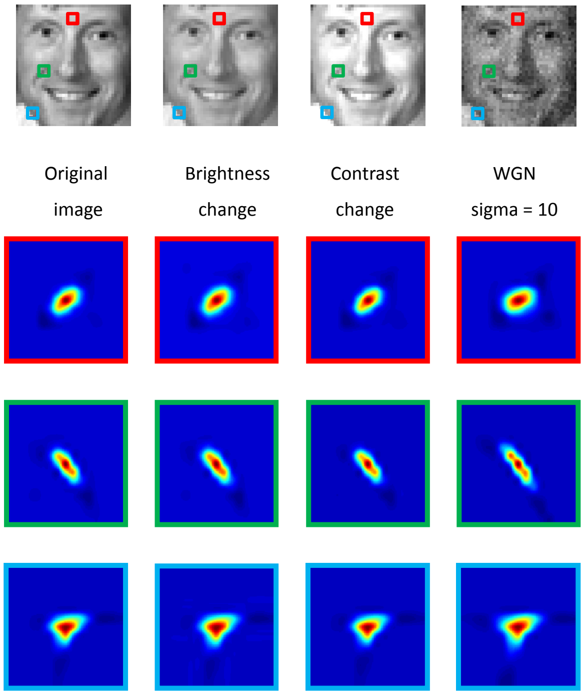


Figure 1.8: Robustness of LARK to illumination changes and the presence of white Gaussian noise (WGN).

cases.

LARKs Are Visual Geometric Units LARKs are closely related to visual geometric units that can serve as a key for visual perception. As shown in Fig. **1.10**, detailed inspection of Close's paintings[7] reveals a multiplicity of simple, but evocative geometric units, which at a distance convey a surprisingly accurate rendition of the subject. In fact, these paintings by Chuck Close go beyond art and lead to important questions regarding visual perception. Recently, Pelli [58], a psychologist and neuroscientist at New York University, identified a critical size of the geometric units in Close's works

[7] The contemporary American artist Chuck Close (1940 -) is well known for his block portraits of faces that are composed of non-overlapping local geometric forms.

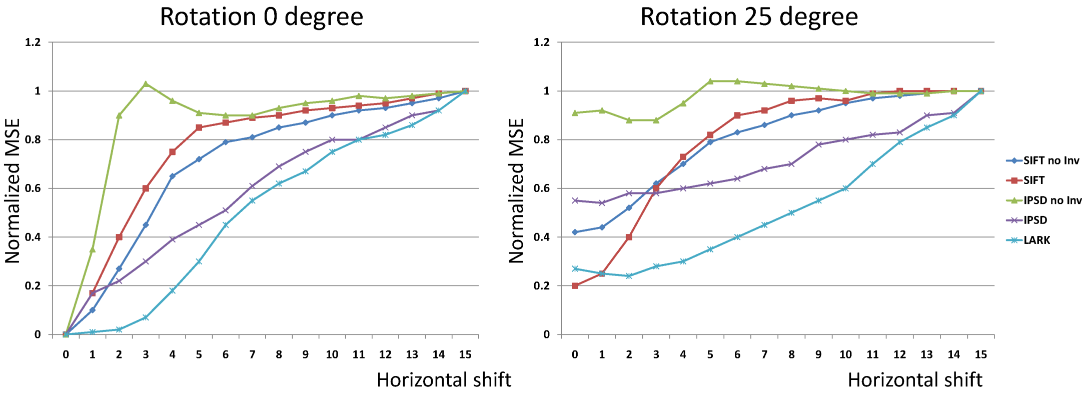


Figure 1.9: Mean squared error (MSE) comparison between descriptors computed from a patch and its transformed version. Left: the transformed patch is horizontally shifted. Right: the transformed patch is first rotated by 25 degrees and then horizontally shifted. The curves are an average over 100 patches from natural images. LARK appears to be more invariant to these transformations than others.

necessary for the image to take on an overall structure rather than to appear as an abstraction. Here, the point is that the size and type of the geometric units determine their power to convey globally perceived shapes [58, 59, 60]. In this thesis, we investigate the effect of LARK size and discriminativity of LARKs.

1.3.2 LARK in 3-D

Now, we introduce the time axis to the data model so that $\mathbf{x}_l = [x_1, x_2, t]_l^T$: x_1 and x_2 are the spatial coordinates, and t is the temporal coordinate. Similar to the 2-D case, the covariance matrix $\mathbf{C}_l \in \mathbb{R}^{(3\times3)}$ can be written as follows:

$$\mathbf{C}_l = \sum_{m\in\Omega_l} \begin{bmatrix} z_{x_1}^2(m) & z_{x_1}(m)z_{x_2}(m) & z_{x_1}(m)z_t(m) \\ z_{x_1}(m)z_{x_2}(m) & z_{x_2}^2(m) & z_{x_2}(m)z_t(m) \\ z_{x_1}(m)z_t(m) & z_{x_2}(m)z_t(m) & z_t^2(m) \end{bmatrix}, \tag{1.7}$$

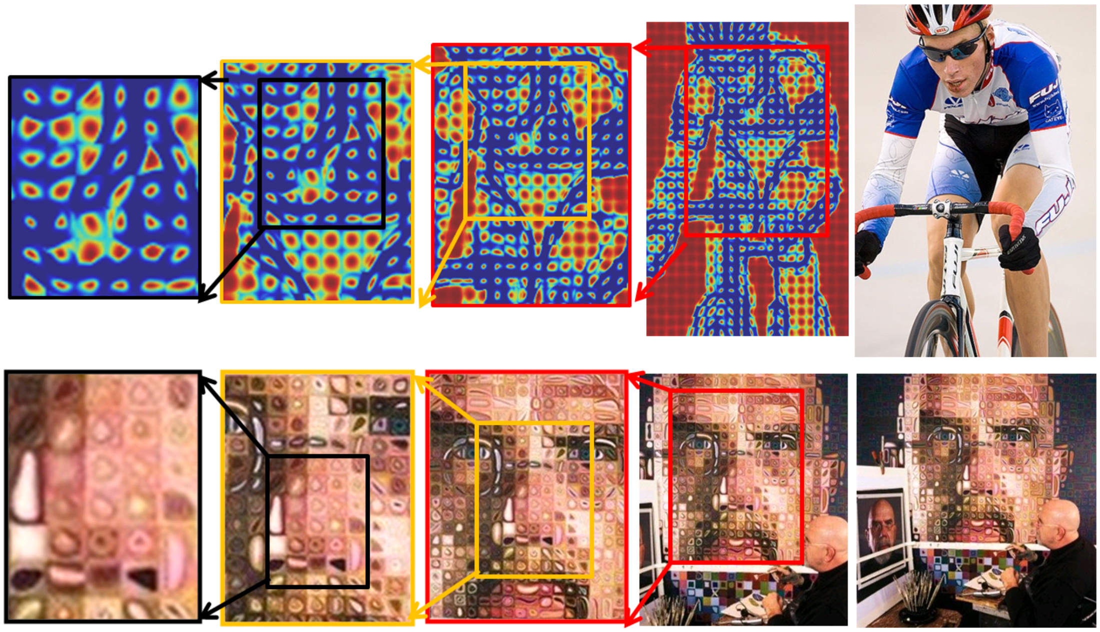


Figure 1.10: LARKs are similar to geometric units that artist Chuck Close employs for his block portraits. Top: LARKs computed from a bike image, Bottom: A self-portrait by Chuck Close, which consists of many geometric units.

where Ω_l is a cube instead of a patch. This enables us to obtain a robust estimation of $\mathbf{C}_l$ even in the presence of noise and other perturbations. Then, as similarly done in the 2-D case, we perform eigen-decomposition on the ("average") covariance matrix $\mathbf{C}_l$ as follows:

$$\mathbf{C}_l = \lambda_1 \mathbf{u}_1^\top \mathbf{u}_1 + \lambda_2 \mathbf{u}_2^\top \mathbf{u}_2 + \lambda_3 \mathbf{u}_3^\top \mathbf{u}_3 = s_1 s_2 s_3 (\frac{s_1}{s_2 s_3} \mathbf{u}_1^\top \mathbf{u}_1 + \frac{s_2}{s_1 s_3} \mathbf{u}_2^\top \mathbf{u}_2 + \frac{s_3}{s_1 s_2} \mathbf{u}_3^\top \mathbf{u}_3), \tag{1.8}$$

Singular values s_1, s_2, s_3 are regularized to avoid numerical instabilities, while both eigenvectors remain the same as follows:

$$\mathbf{C}_l^{reg} = (s_1 s_2 s_3 + \epsilon)^\alpha (\frac{s_1 + \tau}{s_2 s_3 + \tau} \mathbf{u}_1^\top \mathbf{u}_1 + \frac{s_2 + \tau}{s_1 s_3 + \tau} \mathbf{u}_2^\top \mathbf{u}_2 + \frac{s_3 + \tau}{s_1 s_2 + \tau} \mathbf{u}_3^\top \mathbf{u}_3), \tag{1.9}$$

where ϵ, τ, α are same as those in the 2-D case.

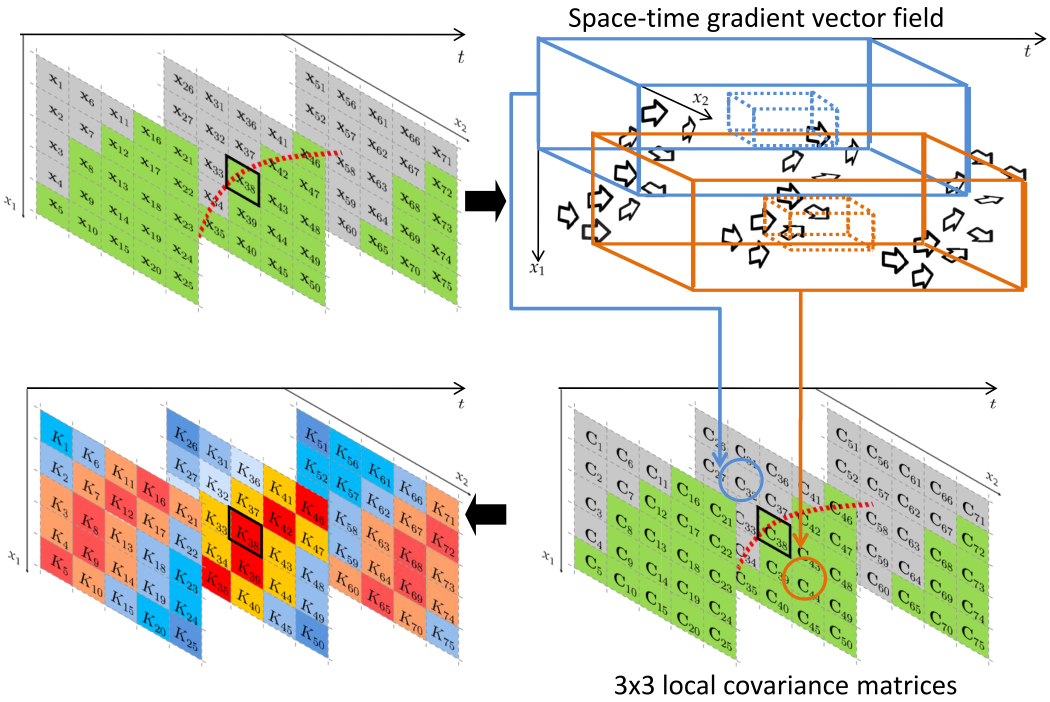

Space-time gradient vector field
3x3 local covariance matrices
t
x_1
x_2

Figure 1.11: Graphical description of how 3-D LARK values centered at voxel of interest $\mathbf{x}_{38}$ are computed in a space-time edge region. Note that each voxel location has its own $\mathbf{C}_l \in \mathbb{R}^{3\times3}$ computed from the space-time gradient vector field within a local space-time window.

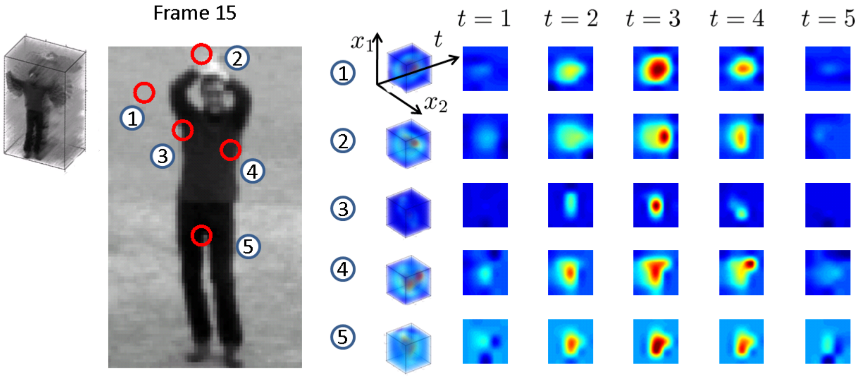


Figure 1.12: Examples of 3-D LARKs capturing 3-D local underlying geometric structure in various regions. In order to compute 3-D LARKs, 5 frames (frame 13 to frame 17) were used. 3-D LARKs are shown upsampled for illustration only.

In the 3-D case, orientation information captured in 3-D LARK contains the motion information implicitly [61]. It is worth noting that a significant strength of using this implicit framework (as opposed to the direct use of estimated motion vectors) is the flexibility it provides in terms of smoothly and adaptively changing descriptors. This flexibility allows the accommodation of even complex motions, so long as their magnitudes are not excessively large[8]. Fig. **1.12** shows examples of 3-D LARK capturing 3-D local underlying geometric structure in various space-time regions. The values of the kernel K in (1.3) are based on the covariance matrices $\mathbf{C}_l$ along with their space-time locations $\mathbf{x}_l$. Intuitively, $\mathbf{C}_l$'s computed from the local analysis cube Ω_l are similar to one another in the motion-free region (see Fig. **1.12** [1]). On the other hand, in the region where motion exists (see Fig. **1.12** [2,3,4,5]), the kernel size and shape depend on both $\mathbf{C}_l$ and its space-time location $\mathbf{x}_l$ in the local space-time window. Thus, if the pixel of interest (center pixel of kernel) is located in space-time edge region, high values in the kernel are yielded along the space-time edge region whereas the rest of kernel values are near zero. Fig **1.13** shows that 3-D LARKs are effective at capturing local space-time geometry individually, and global space-time geometry collectively.

1.3.3 Comparison to Other Descriptors

LARK is related to, but more general than bilateral kernels (BL) [1], non-local means kernels (NLM) [2], and local self-similarity (SSIM) [11]. LARK can be distinguished from BL, NLM, and SSIM in many useful ways which we explain below.

[8] When the magnitude of the motions is large (relative to the support of the LARKs, specifically,) a basic form of coarse but explicit motion compensation will become necessary. We refer the reader to [61] for more detail.

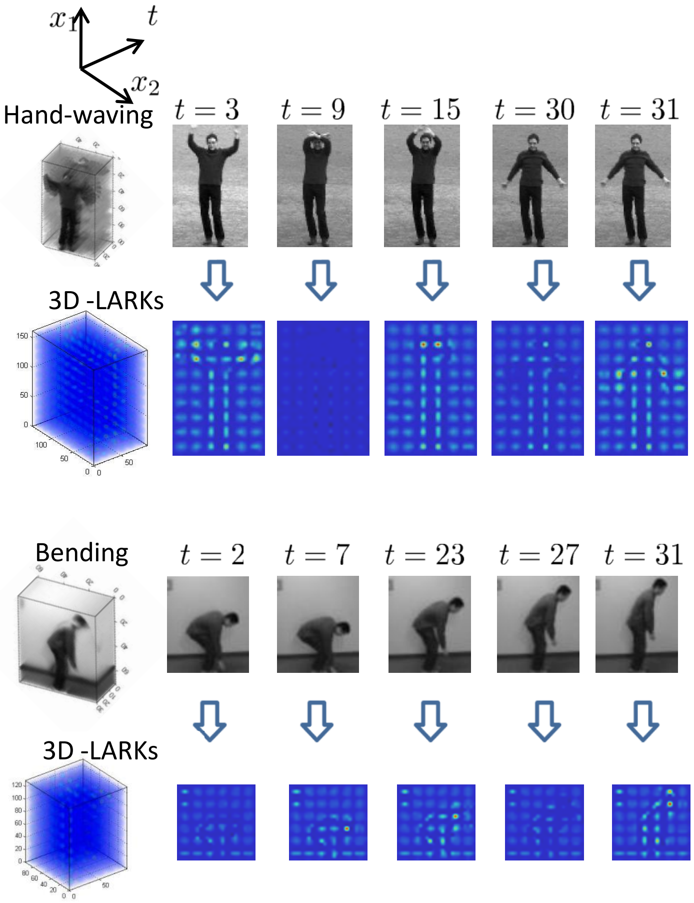


Figure 1.13: 3-D LARKs computed from a hand-waving action and a bending action are shown. For graphical description, we only computed 3-D LARKs at non-overlapping 5 × 5 × 5 cubes, even though we compute 3-D LARKs densely in practice.

Bilateral Kernels (BL) were originally developed for edge-preserving smoothing [1] and has been shown to be effective in tone-mapping [62] in computer graphics. BL is defined as follows:

$$K(y_l, y, \mathbf{x}_l, \mathbf{x}) = \exp\left\{-\frac{\|y_l - y\|^2}{h_r^2} - \frac{\|\mathbf{x}_l - \mathbf{x}\|^2}{h_s^2}\right\}, \tag{1.10}$$

where y_l is a pixel value at $\mathbf{x}_l$ near $\mathbf{x}$, h_r and h_s are global smoothing parameters for photometric and spatial distances respectively. BL captures underlying geomet-

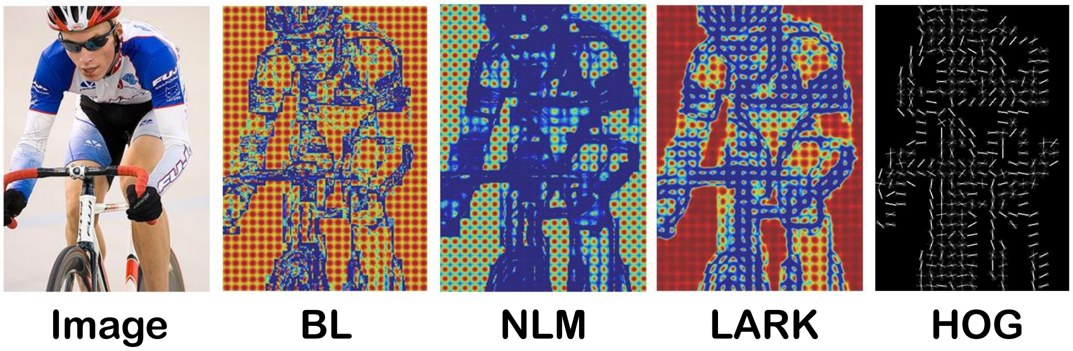


Figure 1.14: Comparison of BL [1], NLM [2], LARK, and HOG [3] computed in non-overlapping patches. This figure is better illustrated in color.

ric structures by separately computing radiometric and spatial similarities between a center pixel and its surroundings based on Euclidean distance. Since it is based on a direct similarity, the resulting geometric structure is not very informative and quite sensitive to noise and variation in illumination. This idea was generalized to measuring similarity between (not necessarily local) patches, in non-local means (NLM) [2].

Non-local Means Kernels (NLM) is defined as a weighted Gaussian kernel:

$$K(\mathbf{y}_l, \mathbf{y}, \mathbf{x}_l, \mathbf{x}) = \exp\left\{-\frac{\|\mathbf{y}_l - \mathbf{y}\|^2}{h_r^2} - \frac{\|\mathbf{x}_l - \mathbf{x}\|^2}{h_s^2}\right\}, \tag{1.11}$$

where $\mathbf{y}_l$ is a *patch* of pixels centered at $\mathbf{x}_l$, h_r and h_s are global smoothing parameters. Since NLM makes use of a patch instead of a pixel for similarity, it can capture more sophisticated geometric structure than BL. However NLM tends to fail in capturing accurate geometric shape in object boundaries. It is worth noting that NLM is still based on gray values and Euclidean distance. On the other hand, LARK utilizes a stable estimate of gradients covariance matrices which allow it to be robust in the presence of noise and certain illumination changes. Fig. **1.14** clearly demonstrates that LARK captures geometric structure in a more stable way than BL and NLM do.

Local Self-similarity Kernel (SSIM) Shechtman and Irani [11] proposed the so-called local self-similarity kernel for object detection. The SSIM kernel is defined as:

$$K(\mathbf{y}_l, \mathbf{y}) = \exp\left\{-\frac{\|\mathbf{y}_l - \mathbf{y}\|^2}{\max(v_{noise}, v)}\right\}, \tag{1.12}$$

where v_{noise} is a constant that stands for photometric variations (in color, illumination or due to noise), and v is the maximal variance of the difference of all patches within a very small neighborhood of $\mathbf{x}$.

As shown in equations (1.11) and (1.12), NLM and SSIM share similar forms. However SSIM is designed in a more sophisticated way than NLM because SSIM uses color patches, and log-polar representation which accounts for local affine deformations. Up to now, we have analyzed a relationship between LARK and other descriptors such as BL,NLM, and SSIM. Now we briefly discuss a relationship between LARK and histogram of gradients descriptors.

Relation to Histogram of Gradients Descriptor For the past few years, histogram of gradients based descriptors computed from interest points have become ubiquitous as local image descriptors. With the advent of SIFT [18], many researchers have developed such variants as PCA-SIFT [63], GLOH (Gradient Location and Orientation Histogram) [19], shape context [20], HOG [3] etc. We refer the reader to [19] for a comprehensive study on local image descriptors.

HOG and SIFT use histogram representation of quantized gradient orientations. It is interesting to note that 3-D LARKs[9] seem related to “HOG3D" introduced in [65]. However, our method is quite different in that our descriptors capture voxel relationships based on the locally measured distance between voxels using a natural signal

[9] HoG [3] and HoF [64] are also related to our 2-D LARKs ($x_1 - x_2$ axes) and 2-D LARKs (either $x_1 - t$ axes or $x_2 - t$ axes).

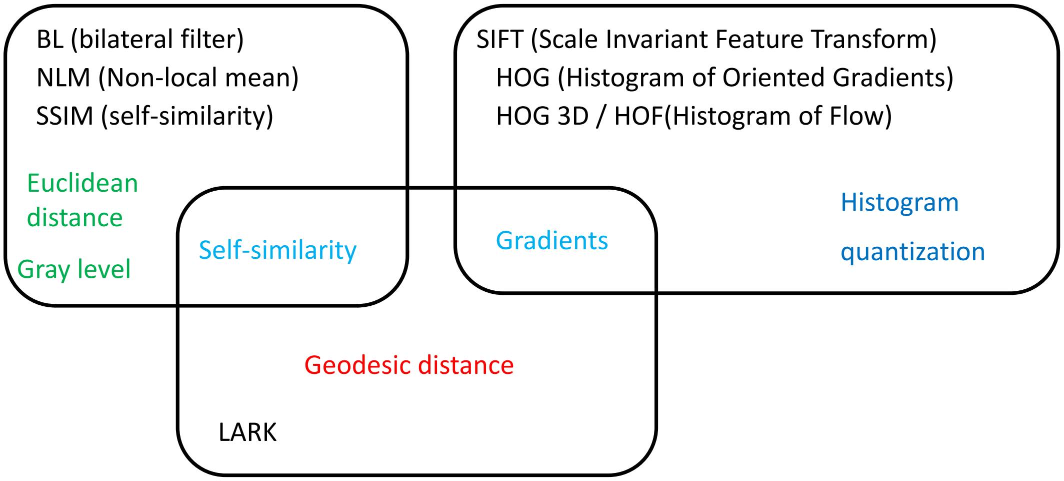


Figure 1.15: LARK shares *self-similarity* with BL, NLM, SSIM while sharing *gradients* with SIFT and its variants. However, LARK is the only one descriptor based on the *geodesic distance*.

induced metric, whereas HOG3D mostly makes use of the histogram of quantized local space-time gradients. We believe that quantization of oriented gradients, while useful in reducing computational complexity, can lead to a significant degradation in discriminative power of descriptors. This effect is particularly severe in the case where there is only a single positive example available without any prior information, which we will explain in Chapters II and III. Superior performance of LARKs in 2-D and 3-D over BL, NLM, SSIM, SIFT, HOG, and HOG3D is also demonstrated in Chapters II and III. The Discussion above is summarized in Fig. **1.15**.

Summary – In this chapter, we introduced LARK descriptors for visual recognition and noted some invariance properties of LARKs. LARKs are distinguished from other state of the art descriptors in the sense that LARK is based on the geodesic distance derived from the regularized covariance matrices. Since LARKs capture local geometric structure exceedingly well, we use LARKs to solve saliency detection problem in

Chapter 2. The concept of pixel-level similarity in LARKs is then extended to patch-level similarity, which leads to a nonparametric saliency detection framework without any prior information. Superiority of LARKs in 2-D and 3-D over other descriptors in saliency detection will be presented later in Chapter 2.

Chapter 2

Saliency Detection

Abstract – We present a novel, unified framework for both static and space-time saliency detection [47, 48]. The proposed method is a bottom-up approach with LARK descriptors extracted from the given image (or a video). Visual saliency is computed using the said “self-resemblance" measure derived from the concept of patch-level similarity. The framework results in a saliency map where each pixel (or voxel) indicates the statistical likelihood of saliency of a center patch given its surrounding patches. As a similarity measure, matrix cosine similarity (a generalization of cosine similarity) is employed. State of the art performance is demonstrated on commonly used human eye fixation data (static scenes [8] and dynamic scenes [66]) and some psychological patterns.

2.1 Introduction

The human visual system has a remarkable ability to automatically attend to only salient locations in static and dynamic scenes [67, 68, 69]. This ability enables us to allocate limited perceptual and cognitive resources on task-relevant visual input. In machine vision system, a flood of visual information fed into the system needs to be efficiently scanned in advance for relevance. In this chapter, we propose a computational model for selective visual attention, otherwise known as visual saliency. In recent years, visual saliency detection has been of great research interest [8, 4, 70, 5, 71, 39, 72, 73, 74, 75, 76]. Analysis of visual attention has benefited a wide range of applications such as object detection, action detection, video summarization [77], image quality assessment [78, 79] and more. There are two types of computational models for saliency according to what the model is driven by: a bottom-up saliency [8, 4, 5, 71, 72, 73, 75, 76] and a top-down saliency [70, 39, 74]. As opposed to bottom-up saliency algorithms that are fast and driven by low-level features, top-down saliency algorithms are slower and task-driven. In general, the plausibility of bottom-up saliency models is examined in terms of predicting eye movement data made by human observers in a task designed to minimize the role of top-down factors. Although some progress has been made by parametric saliency models [4, 71, 72, 39] in predicting fixation patterns and visual search, there is significant room to further improve the accuracy.

In this chapter, we develop a nonparametric bottom-up visual saliency method which exhibits state of the art performance. The problem of interest can be described as follow: Given an image or a video, we are interested in accurately detecting salient objects or actions from the data without any background knowledge. To accomplish this task, we propose to use, as features, *LARK descriptors* in 2-D and 3-D which capture local data structure exceedingly well. Our approach is motivated by a probabilistic framework, which is based on a nonparametric estimate of the likelihood of saliency.

As we describe below, this boils down to the local calculation of a “self-resemblance" map, which measures the similarity of a patch (feature matrix) at a pixel of interest to its neighboring patches (feature matrices).

Previous work Itti et al. [71] introduced a saliency model which was biologically inspired. Specifically, they proposed to use a set of feature maps from three complementary channels as intensity, color, and orientation. The normalized feature maps from each channel were then linearly combined to generate the overall saliency map. Even though this model has been successful in predicting human fixations, it is somewhat ad-hoc in that there is no objective function to be optimized and many parameters must be tuned by hand. With the proliferation of eye-tracking data, a number of researchers have recently attempted to address the question of what attracts human visual attention by being more mathematically and statistically precise [8, 4, 80, 70, 66, 72, 76].

Bruce and Tsotsos [8] modeled bottom-up saliency as the maximum information sampled from an image. More specifically, saliency is computed as Shannon’s self-information $-\log p(\mathbf{K})$, where $\mathbf{K}$ is a local visual feature (i.e., derived from independent component analysis (ICA) performed on a large sample of small RGB patches in the image.) The probability density function is estimated based on a Gaussian kernel density estimate in a neural circuit.

Gao et al. [4, 80, 70] proposed a unified framework for top-down and bottom-up saliency as a classification problem with the objective being the minimization of classification error. They first applied this framework to object detection [70] in which a set of features are selected such that a class of interest is best discriminated from all other classes, and saliency is then defined as the weighted sum of features that are salient for that class. In [4], they defined bottom-up saliency using the idea that pixel locations are salient if they are distinguished from their surroundings. They used difference of Gaussians (DoG) filters and Gabor filters, measuring the saliency of a point as the Kullback-Leibler (KL) divergence between the histogram of filter responses at the point and the histogram of filter responses in the surrounding region. Mahadevan and Vasconcelos [81] applied this bottom-up saliency to background subtraction in highly dynamic scenes.

Oliva and Torralba [82, 39] proposed a Bayesian framework for the task of visual search (i.e., whether a target is present or not.) They modeled bottom-up saliency as $\frac{1}{p(\mathbf{K}|\mathbf{K}_G)}$ where $\mathbf{K}_G$ represents a global feature that summarizes the appearance of the scene, and approximated this conditional probability density function by fitting to a multivariate exponential distribution. Zhang et al. [72] also proposed saliency detection using natural statistics (SUN) based on a similar Bayesian framework to estimate the probability of a target at every location. They also claimed that their saliency measure emerges from the use of Shannon's self-information under certain assumptions. They used ICA features as similarly done in [8], but their method differs from [8] in that natural image statistics were applied to determine the density function of ICA featuers. Itti and Baldi [66] proposed so-called "Bayesian Surprise" and extended it to the video case [10]. They measured KL-divergence between a prior distribution and posterior distribution as a measure of saliency.

For saliency detection in video, Marat et al. [75] proposed a space-time saliency detection algorithm inspired by the human visual system. They fused a static saliency map and a dynamic saliency map to generate the space-time saliency map. Gao et al. [4] adopted a dynamic texture model using a Kalman filter in order to capture the motion patterns even in the case when the scene is itself dynamic. Zhang et al.

[73] extended their SUN framework to a dynamic scene by introducing temporal filter (Difference of Exponential:DoE) and fitting a generalized Gaussian distribution to the estimated distribution for each filter response.

Most of the methods [4, 71, 82, 73] based on Gabor or DoG filter responses require many design parameters such as the number of filters, type of filters, choice of the nonlinearities, and a proper normalization scheme. These methods tend to emphasize textured areas as being salient regardless of their context. In order to deal with these problems, [8, 72] adopted non-linear features that model complex cells or neurons in higher levels of the visual system. Kienzle et al. [83] further proposed to learn a visual saliency model directly from human eyetracking data using a support vector machine (SVM).

Different from traditional image statistical models, a spectral residual approach based on the Fourier transform was recently proposed by Hou and Zhang [5]. The spectral residual approach does not rely on parameters and detects saliency rapidly. In this approach, the difference between the log spectrum of an image and its smoothed version is the spectral residual of the image. However, Guo and Zhang [84] claimed that what plays an important role for saliency detection is not spectral residual, but the image's phase spectrum. Recently, Hou and Zhang [76] proposed a dynamic visual attention model by setting up an objective function to maximize the entropy of the sampled visual features based on the incremental coding length.

Overview of the Proposed Approach Our contributions to the saliency detection task are three-fold. First we propose to use LARKs as features which, fundamentally differ from conventional filter responses, but capture the underlying local structure of the data exceedingly well, even in the presence of significant distortions. Second, in-

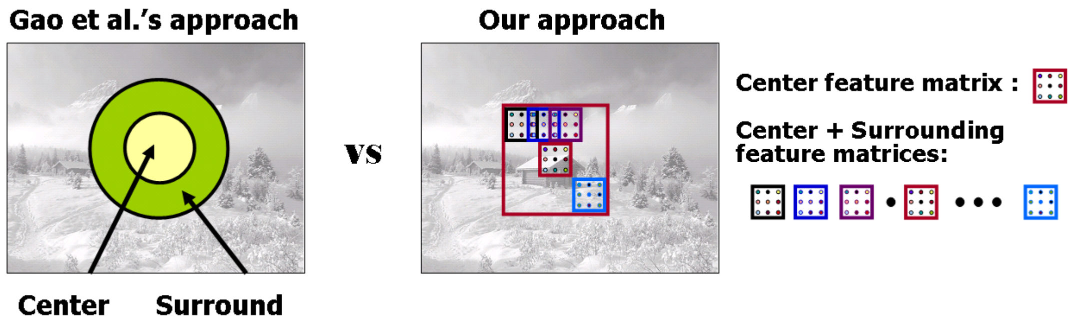


Figure 2.1: Illustration of difference between Gao el al. [4]'s approach and our approach about a center-surround definition.

stead of using parametric models, we propose to use a nonparametric kernel density estimation for such features, which results in a saliency map constructed from a local self-resemblance measure, indicating likelihood of saliency. Lastly, we provide a simple, but powerful unified framework for both static and space-time saliency detection. These contributions, which we will highlight at the end of this section, are evaluated in Section 2.3 in terms of predicting human eye fixation data in both commonly used image [8] and video [66] datasets.

As similarly done in Gao et al. [4], we measure saliency at a pixel in terms of how much it stands out from its surroundings. To formalize saliency at each pixel, we let the binary random variable y_i denote whether a pixel position $\mathbf{x}_i = [x_1, x_2]_i^\top$ is salient or not as follows:

$$y_i = \begin{cases} 1, & \text{if} \quad \mathbf{x}_i \quad \text{is} \quad \text{salient}, \\ 0, & \text{otherwise}, \end{cases} \tag{2.1}$$

where $i = 1, \cdots, M$, and M is the total number of pixels in the image. Motivated by the approach in [72, 82], we define saliency at pixel position $\mathbf{x}_i$ as a posterior probability

$Pr(y_i = 1|\mathbf{k})$ as follows:

$$S_i = Pr(y_i = 1|\mathbf{k}), \tag{2.2}$$

where the feature matrix, $\mathbf{k}_i = [\mathbf{k}_i^1, \cdots, \mathbf{k}_i^L]$ at pixel of interest $\mathbf{x}_i$ (what we call a center feature) contains a set of (normalized) LARK feature vectors ($\mathbf{k}_i$) in a local neighborhood where L is the number of features in that neighborhood (Note that if $L = 1$, we use a single feature vector. Using a feature matrix consisting of a set of feature vectors provides more discriminative power than using a single feature vector as also pointed out in [14, 85].) In turn, the larger collection of features $\mathbf{k} = [\mathbf{k}_1, \cdots, \mathbf{k}_N]$ is a matrix containing features not only from the center, but also a surrounding region (what we call a center+surround region; See Fig. **2.1**.) N is the number of feature matrices in the center+surround region. Using Bayes' theorem, Equation (2.2) can be written as

$$S_i = Pr(y_i = 1|\mathbf{k}) = \frac{p(\mathbf{k}|y_i = 1)Pr(y_i = 1)}{p(\mathbf{k})}. \tag{2.3}$$

By assuming[1] that 1) a-priori, every pixel is considered to be equally likely to be salient; and 2) $p(\mathbf{k})$ are uniform over features, the saliency we defined boils down to the conditional probability density $p(\mathbf{k}|y_i = 1)$.

Since we do not know the conditional probability density $p(\mathbf{k}|y_i = 1)$, we need to estimate it. Gao et al. [4] and Zhang et al. [72] fit the marginal density of local feature vectors $p(\mathbf{k})$ to a generalized Gaussian distribution. However, in this chapter, we approximate the conditional density function $p(\mathbf{k}|y_i = 1)$ based on nonparametric kernel density estimation which will be explained in detail in Section 2.2.

Before we begin a more detailed description, it is worthwhile to highlight some aspects of our proposed framework. While the state-of-the art methods [8, 4, 66,

[1] Tavakoli et al. [86] proposed to approximate $p(\mathbf{k})$ by estimating $p(\mathbf{k}|y_i = 1)$ and $p(\mathbf{k}|y_i = 0)$ separately while they learned $Pr(y_i = 1)$ from a training set.

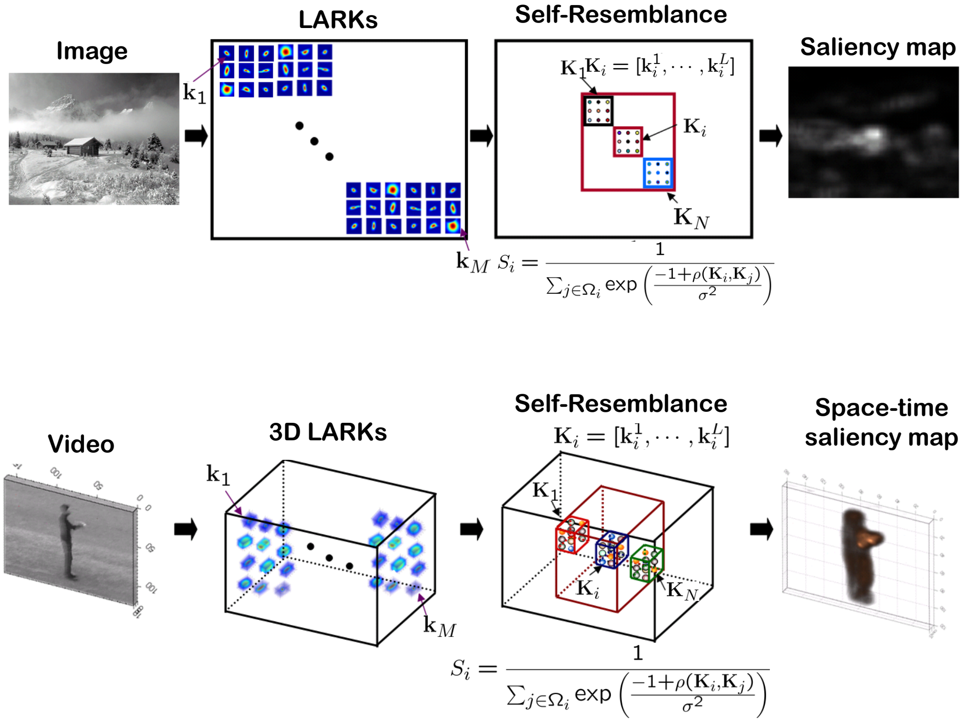


Image
LARKs
$\mathbf{k}_1$
$\mathbf{k}_M$
Self-Resemblance
$\mathbf{K}_1$
$\mathbf{K}_i = [\mathbf{k}_i^1, \cdots, \mathbf{k}_i^L]$
$\mathbf{K}_i$
$\mathbf{K}_N$
Saliency map
$S_i = \frac{1}{\sum_{j \in \Omega_i} \exp\left(\frac{-1+\rho(\mathbf{K}_i,\mathbf{K}_j)}{\sigma^2}\right)}$
Video
3D LARKs
$\mathbf{k}_1$
$\mathbf{k}_M$
Self-Resemblance
$\mathbf{K}_i = [\mathbf{k}_i^1, \cdots, \mathbf{k}_i^L]$
$\mathbf{K}_1$
$\mathbf{K}_i$
$\mathbf{K}_N$
Space-time saliency map
$S_i = \frac{1}{\sum_{j \in \Omega_i} \exp\left(\frac{-1+\rho(\mathbf{K}_i,\mathbf{K}_j)}{\sigma^2}\right)}$

(b)

Figure 2.2: Graphical overview of saliency detection system (a) static saliency detection (b) space-time saliency detection. Note that the saliency measure S_i is identical for both static and space-time saliency detection except that 3-D LARKs and cubes are employed for space-time saliency detection.

72] are related to our method, their approaches fundamentally differ from ours in the following respects: 1) While they use Gabor filters, DoG filters, or ICA to derive features, we propose to use LARKs which are highly nonlinear but stable in the presence of uncertainty in the data [46]. In addition, normalized local steering kernels provide a certain invariance as shown in Fig. **1.8** and Fig. **2.14**; 2) As opposed to [4, 72] which model marginal densities of band-pass features as a generalized Gaussian distribution, we estimate the conditional probability density $p(\mathbf{k}|y_i = 1)$ using nonparametric kernel density estimation (see Fig. **2.3**); 3) While Itti and Baldi [66] computed, as a measure of saliency, KL-divergence between a prior and a posterior distribution, we explicitly estimate the likelihood function directly using nonparametric kernel density estimation; 4) Our space-time saliency detection method does not require explicit motion estimation; 5) The proposed unified framework can handle both static and space-time saliency detection. Fig. **2.2** shows an overview of our proposed framework for saliency detection.

To summarize the operation of the overall algorithm, we first compute the normalized LARKs (space-time LARKs) from the given image (video) and vectorize them as $\mathbf{k}$'s. Then, we identify features $\mathbf{k}_i$ centered at a pixel of interest $\mathbf{x}_i$, and a set of feature matrices $\mathbf{k}_j$ in a center+surrounding region and compute the self-resemblance measure as shown in equations (2.7) and (2.8). The final saliency map is given as a density map as shown in Fig **2.2**. In the next section, we provide further technical details about the steps outlined above. In Section 2.3, we demonstrate the performance of the system with experimental results.

2.2 Saliency by Local Self-Resemblance

As we alluded to in Section 2.1, saliency at a pixel $\mathbf{x}_i$ is measured using the conditional density of the feature matrix at that position: $S_i = p(\mathbf{k}|y_i = 1)$. Hence, the task at hand is to estimate $p(\mathbf{k}|y_i = 1)$ over $i = 1, \cdots, M$. In general, the Parzen density estimator is a simple and generally accurate non-parametric density estimation method [87]. However, in higher dimensions and with an expected long-tail distribution, the Parzen density estimator with an isotropic kernel is not the most appropriate tool [88, 89, 90].

LARK features tend to generically come from long-tailed distributions [49], and as such, they do not form clusters in the feature space. When we estimate a prob-

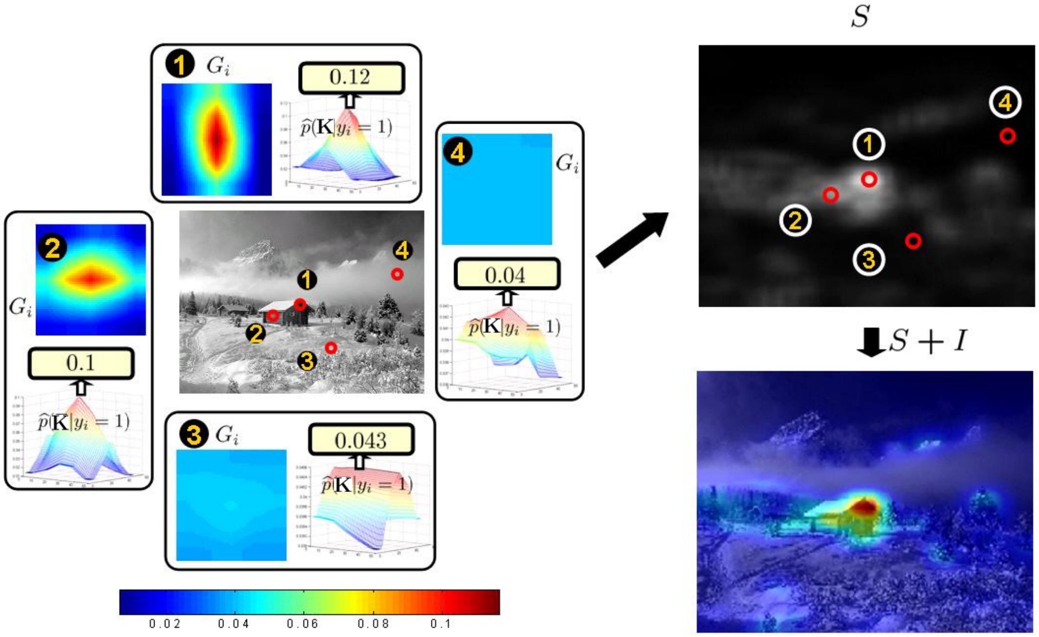


Figure 2.3: Example of saliency computation in natural gray-scale image. The estimated probability density $\widehat{p}(\mathbf{K}|y_i = 1)$ at the point 1 (0.12) is much higher than ones (0.043) and (0.04) at the point 3 and point 4, which depicts that the point 1 is more salient than point 3 and point 4. Note that red values in saliency map represent higher saliency, while blue values mean lower saliency.

ability density at a particular feature point, for instance $\mathbf{k}_i = [\mathbf{k}_i^1, \cdots, \mathbf{k}_i^L]$ (where L is the number of vectorized LARKs ($\mathbf{k}$'s) employed in the feature matrix), the isotropic kernel centered on that feature point will spread its density mass equally along all the feature space directions, thus giving too much emphasis to irrelevant regions of space and too little along the manifold. Earlier studies [88, 89, 90] also pointed out this problem. This motivates us to use *a locally data-adaptive kernel density estimator.* We define the conditional probability density $p(\mathbf{k}|y_i = 1)$ at $\mathbf{x}_i$ as a center value of a normalized adaptive kernel (weight function) $G(\cdot)$ computed in the center+surround region as follows:

$$S_i = \widehat{p}(\mathbf{k}|y_i = 1) = \frac{G_i(\overline{\mathbf{k}}_i - \overline{\mathbf{k}}_i)}{\sum_{j=1}^{N} G_i(\overline{\mathbf{k}}_i - \overline{\mathbf{k}}_j)}, \tag{2.4}$$

Inspired by earlier works such as [91, 92, 93, 49] that have shown the effectiveness of correlation-based similarity, the kernel function G_i in Equation (2.4) can be defined by using the concept of matrix cosine similarity [49] as follows:

$$G_i(\overline{\mathbf{k}}_i - \overline{\mathbf{k}}_j) = \exp\left(\frac{-||\overline{\mathbf{k}}_i - \overline{\mathbf{k}}_j||_F^2}{2\sigma^2}\right) = \exp\left(\frac{-1 + \rho(\mathbf{k}_i, \mathbf{k}_j)}{\sigma^2}\right), \quad j = 1, \cdots, N, \tag{2.5}$$

where $\overline{\mathbf{k}}_i = \frac{1}{\|\mathbf{k}_i\|_F}\left[\mathbf{k}_i^1, \cdots, \mathbf{k}_i^L\right]$ and $\overline{\mathbf{k}}_j = \frac{1}{\|\mathbf{k}_j\|_F}\left[\mathbf{k}_j^1, \cdots, \mathbf{k}_j^L\right]$, $||\cdot||_F$ is the Frobenious norm, and σ is a parameter (This parameter is set to 0.07 and fixed for all the experiments.) controlling the fall-off of weights. Here, $\rho(\mathbf{k}_i, \mathbf{k}_j)$ is the “Matrix Cosine Similarity (MCS)" between two feature matrices $\mathbf{k}_i, \mathbf{k}_j$ and is defined as the Frobenius inner product between two normalized matrices ($\rho(\mathbf{k}_i, \mathbf{k}_j) =< \overline{\mathbf{k}}_i, \overline{\mathbf{k}}_j >_F =$ trace$\left(\frac{\mathbf{k}_i^\top \mathbf{k}_j}{\|\mathbf{k}_i\|_F \|\mathbf{k}_j\|_F}\right) \in [-1, 1]$.) This matrix cosine similarity[2] can be rewritten as a weighted sum of the vector cosine similarities [91, 92, 93] $\rho(\mathbf{k}_i, \mathbf{k}_j)$ between each pair of corresponding feature vectors (i.e., columns) in $\mathbf{k}_i, \mathbf{k}_j$ as follows:

$$\rho(\mathbf{k}_i, \mathbf{k}_j) = \sum_{\ell=1}^{L} \frac{\mathbf{k}_i^{\ell\top} \mathbf{k}_j^{\ell}}{\|\mathbf{k}_i\|_F \|\mathbf{k}_j\|_F} = \sum_{\ell=1}^{L} \rho(\mathbf{k}_i^{\ell}, \mathbf{k}_j^{\ell}) \frac{\|\mathbf{k}_i^{\ell}\| \|\mathbf{k}_j^{\ell}\|}{\|\mathbf{k}_i\|_F \|\mathbf{k}_j\|_F}. \tag{2.6}$$

The weights are represented as the product of $\frac{\|\mathbf{k}_i^{\ell}\|}{\|\mathbf{k}_i\|_F}$ and $\frac{\|\mathbf{k}_j^{\ell}\|}{\|\mathbf{k}_j\|_F}$ which indicate the relative importance of each feature in the feature sets $\mathbf{k}_i, \mathbf{k}_j$. This measure not only generalizes the cosine similarity, but also overcomes the disadvantages of the conventional Euclidean distance which is sensitive to outliers (This measure can be efficiently computed by column-stacking the matrices $\mathbf{k}_i, \mathbf{k}_j$ and simply computing the cosine similarity between two long column vectors.) By inserting Equation (2.5) into Equation (2.4),

[2] We refer the reader to Section 3.2.2 for more detail

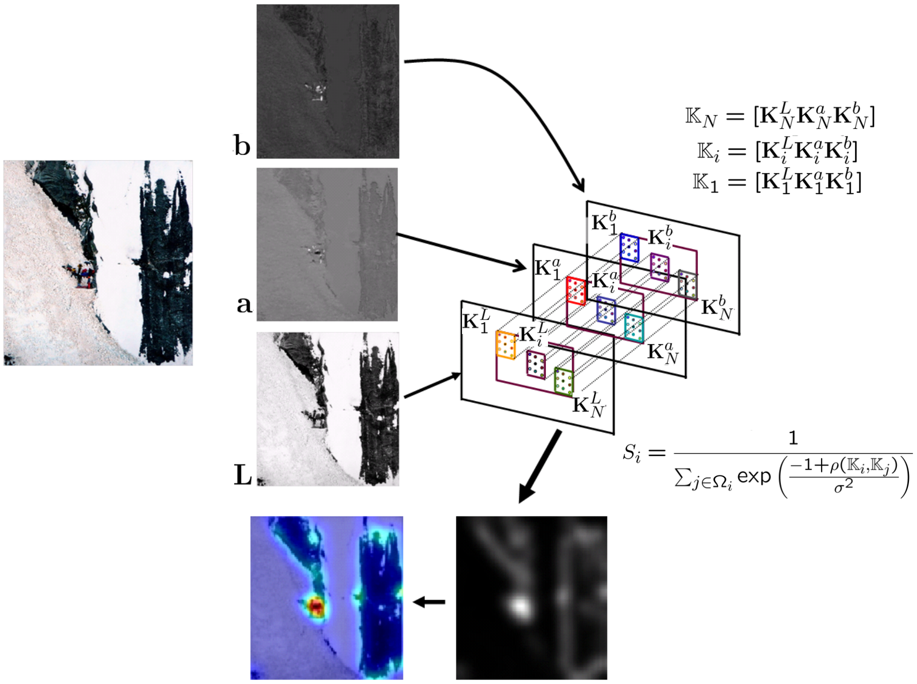

b
a
L
$\mathbb{K}_N = [\mathbf{K}_N^L \mathbf{K}_N^a \mathbf{K}_N^b]$
$\mathbb{K}_i = [\mathbf{K}_i^L \mathbf{K}_i^a \mathbf{K}_i^b]$
$\mathbb{K}_1 = [\mathbf{K}_1^L \mathbf{K}_1^a \mathbf{K}_1^b]$
$\mathbf{K}_1^b$
$\mathbf{K}_i^b$
$\mathbf{K}_N^b$
$\mathbf{K}_1^a$
$\mathbf{K}_i^a$
$\mathbf{K}_N^a$
$\mathbf{K}_1^L$
$\mathbf{K}_i^L$
$\mathbf{K}_N^L$
$S_i = \frac{1}{\sum_{j \in \Omega_i} \exp\left(\frac{-1+\rho(\mathbb{K}_i,\mathbb{K}_j)}{\sigma^2}\right)}$

Figure 2.4: As an example of saliency detection in a color image (in this case, CIE L*a*b*), we show how saliency is computed using matrix cosine similarity.

S_i can be rewritten as follows:

$$S_i = \frac{1}{\sum_{j=1}^{N} \exp\left(\frac{-1+\rho(\mathbf{k}_i,\mathbf{k}_j)}{\sigma^2}\right)}. \qquad (2.7)$$

Fig. **2.3** describes what normalized kernel functions G_i look like in various regions of a natural image. As shown in Fig. **2.3**, at $\mathbf{x}_i$ (that is, $S_i = \widehat{p}(\mathbf{k}|y_i = 1)$) can be explained by the peak value of the normalized weight function G_i which contains contributions from all the surrounding feature matrices. In other words, $\widehat{p}(\mathbf{k}|y_i = 1)$ reveals how salient $\mathbf{k}_i$ is given all the features $\mathbf{k}_j$'s in a neighborhood.

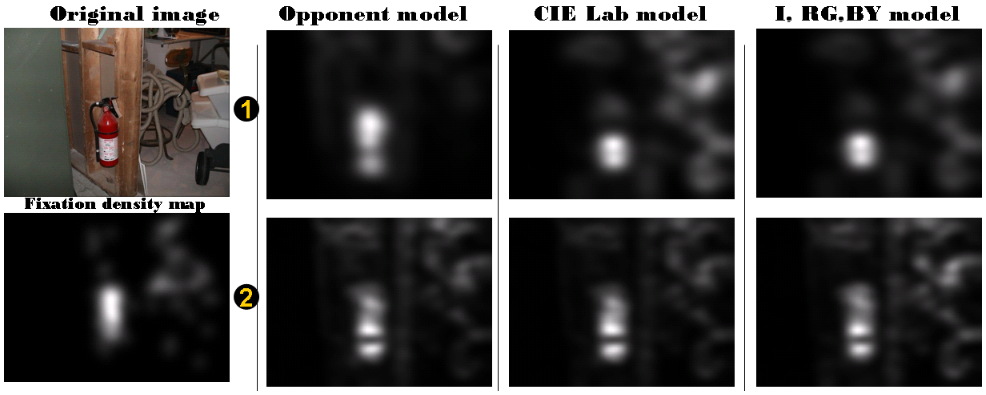


Figure 2.5: Comparisons between (1) Simple normalized summation and (2) The use of matrix cosine similarity without any fusion in three different color spaces. Simple normalized summation method tends to be dominated by a particular chrominance information. It is clearly shown that using matrix cosine similarity provides consistent results than the simple normalized summation fusion method.

Handling Color Images Up to now, we only dealt with saliency detection in a grayscale image. If we have color input data, we need an approach to integrate saliency information from all color channels. To avoid some drawbacks of earlier methods [71, 94], we do not combine saliency maps from each color channel linearly and directly. Instead we utilize the idea of matrix cosine similarity. More specifically, we first identify feature matrices from each color channel c_1, c_2, c_3 as $\mathbf{k}_i^{c_1}, \mathbf{k}_i^{c_2}, \mathbf{k}_i^{c_3}$ as shown in Fig. **2.4**. By collecting them as a larger matrix $\mathbb{K}_i = [\mathbf{k}_i^{c_1}, \mathbf{k}_i^{c_2}, \mathbf{k}_i^{c_3}]$, we can apply matrix cosine similarity between $\mathbb{K}_i$ and $\mathbb{K}_j$. Then, the saliency map from color channels can be analogously defined as follows:

$$S_i = \widehat{p}(\mathbb{K} | y_i = 1) = \frac{1}{\sum_{j=1}^{N} \exp\left(\frac{-1+\rho(\mathbb{K}_i, \mathbb{K}_j)}{\sigma^2}\right)}. \tag{2.8}$$

In order to verify that this idea allows us to achieve a consistent result and leads us to a better performance than using fusion methods, we have compared three different color spaces; namely opponent color channels [95], CIE L*a*b* [49, 11] channels, and I

R-G B-Y channels [72][3].

Fig. **2.5** compares saliency maps using simple normalized summation of saliency maps from different channels as compared to using matrix cosine similarity. It is clearly seen that using matrix cosine similarity provides consistent results regardless of color spaces and helps to avoid some drawbacks of fusion-based methods. To summarize, the overall pseudo-code for the algorithm is given in **Algorithm** 1.

Algorithm 1 Visual Saliency Detection Algorithm

```
$I$ : input image or video, $P$ : size of LARK or 3-D LARK window, $h$ : a global smoothing parameter for LARK, $L$ : number of LARK or 3-D LARK used in the feature matrix, $N$ : size of a center+surrounding region for computing self-resemblance, $\sigma$ : a parameter controlling fall-off of weights for computing self-resemblance.
Stage1 : Compute Features
if I is an image then
    Compute the normalized LARK $K_i$ and vectorize it to $\mathbf{k}_i$, where $i = 1, \cdots, M$.
else
    Compute the normalized 3-D LARK $K_i$ and vectorize it to $\mathbf{k}_i$, where $i = 1, \cdots, M$.
end if
Stage2 : Compute Self-Resemblance
for $i = 1, \cdots, M$ do
    if I is a grayscale image (or video) then
        Identify feature matrices $\mathbf{k}_i, \mathbf{k}_j$ in a local neighborhood.
        $S_i = \frac{1}{\sum_{j=1}^{N} \exp\left(\frac{-1+\rho(\mathbf{k}_i, \mathbf{k}_j)}{\sigma^2}\right)}$
    else
        Identify feature matrices $\mathbb{K}_i = [\mathbf{k}_i^{c1}, \mathbf{k}_i^{c3}, \mathbf{k}_i^{c3}]$ and $\mathbb{K}_j = [\mathbf{k}_j^{c1}, \mathbf{k}_j^{c3}, \mathbf{k}_j^{c3}]$
        in a local neighborhood from three color channels.
        $S_i = \frac{1}{\sum_{j=1}^{N} \exp\left(\frac{-1+\rho(\mathbb{K}_i, \mathbb{K}_j)}{\sigma^2}\right)}$
    end if
end for
Output : Saliency map $S_i$,   $i = 1, \cdots, M$
```

[3] Opponent color space has proven to be superior to RGB, HSV, normalized RGB, and more in the task of object and scene recognition [95]. [11] and [49] showed that CIE L*a*b* performs well in the task of object detection.

2.3 Experimental Results

In this section, we demonstrate the performance of the proposed method with comprehensive experiments in terms of 1) interest region detection; 2) prediction of human fixation data; and 3) performance on psychological patterns. Comparison is made with other state of the art methods both quantitatively and qualitatively.

2.3.1 Interest Region Detection

2.3.1.1 Detecting Proto-objects in Images

In order to efficiently compute the saliency map, we downsample an image I to an appropriate coarse scale[4] (64×64). We then compute LARK of size 3×3 as features and generate feature matrices $\mathbf{k}_i$ in a 5×5 local neighborhood. The number of LARK used in the feature matrix $\mathbf{k}_i$ is set to 9. For all the experiments, the smoothing parameter h for computing LARK was set to 1 and the fall-off parameter σ for computing self-resemblance was set to 0.07. We obtained an overall saliency map by using CIE L*a*b* color space throughout all the experiments. A typical run time takes about 1 second at scale (64×64) on an Intel Pentium 4, 2.66 GHz core 2 PC with 2 GB RAM.

From the point of view of object detection, saliency maps can explicitly represent proto-objects. We use the idea of non-parametric significance testing to detect

[4] Changing the scale leads to a different result in the saliency map. Assume that we use a 3×3 LARK and 5×5 local analysis window for $\mathbf{k}$. If the visual search is performed at a fine scale, finer detail will be captured as salient whereas at a coarser scale, larger objects will be considered to be salient. As expected, computing saliency map at a finer scale takes longer. In fact, we have tried to combine saliency maps from multi-scale, but this idea did not improve performance even at the expense of time-complexity. This brings up an interesting question worth considering for future research; namely; what is the optimal resolution for saliency detection? Clearly, higher resolution images do not imply better saliency maps. Recent publication [96] by Judd et al. reveals that working with fixations on images of mid-level resolutions $(16 - 64$ pixels) could be perceptually adequate and computationally attractive.

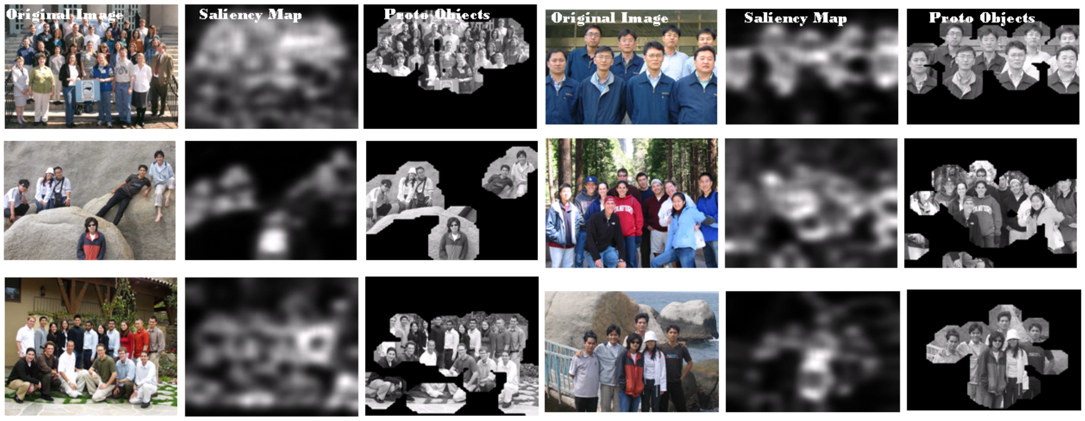


Figure 2.6: Some examples of proto-objects detection in face images.

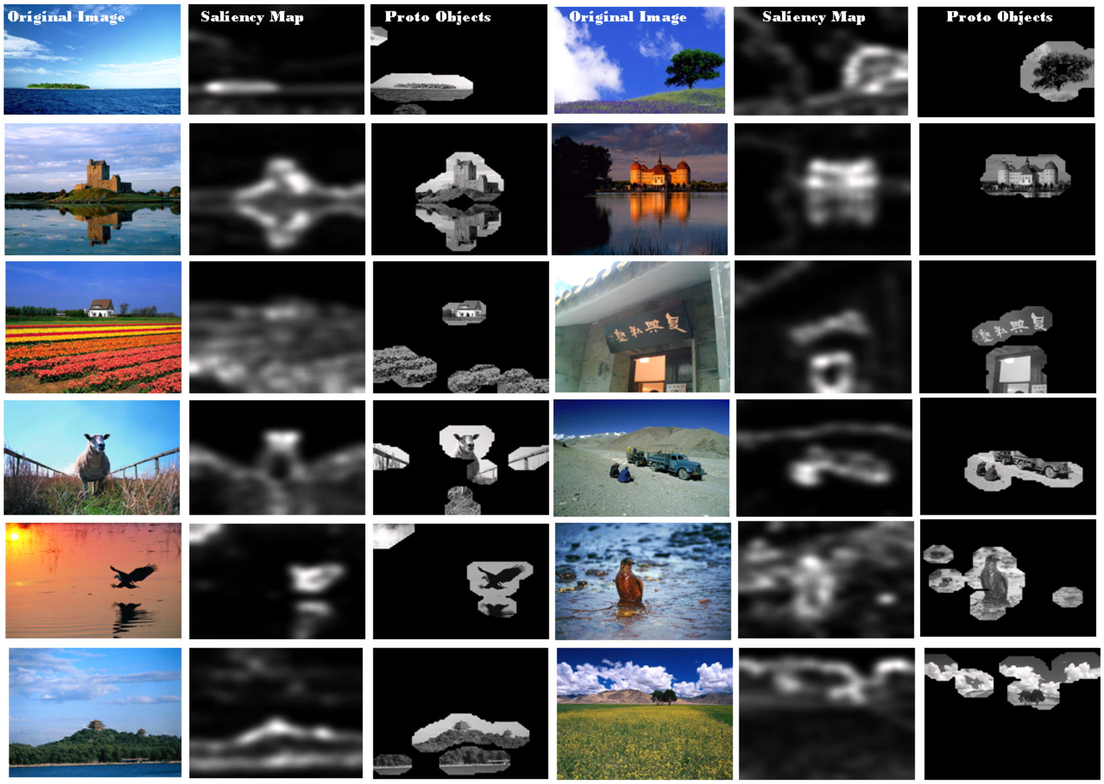


Figure 2.7: Some examples of proto-objects detection in natural scene images [5]

proto-objects. Namely, we compute an empirical PDF from all the saliency values and set a threshold so as to achieve, for instance, a 95 % significance level in deciding whether the given saliency values are in the extreme (right) tails of the empirical PDF. The approach is based on the assumption that in the image, a salient object is a relatively rare object and thus results in values which are in the tails of the distribution of saliency values. After making a binary object map by thresholding the saliency map, a morphological filter is applied. More specifically, we dilate the binary object map with a disk shape of size 5×5. Proto-objects are extracted from corresponding locations of the original image. Multiple objects can be extracted sequentially. Fig. **2.6** shows that the proposed method works well in detecting proto-objects in the images[5] which contain a group of people in a complicated cluttered background. In order to quantitatively evaluate the performance of our method in terms of finding proto-objects, we also tested our method on Hou and Zhang's dataset [5]. This dataset contains 62 natural scene images and binary ground truth images ($\mathcal{G}$) labeled by 4 naive human subjects. Fig. **2.7** also illustrates that our method accurately detects salient objects in natural scenes [5]. For the sake of completeness, we compute the *Hit Rate* (HR) and the *False Alarm Rate* (FAR) as follows:

$$HR = E(\prod_{l} \mathcal{G}_i^l \times O_i), \tag{2.9}$$

$$FAR = E(\prod_{l} (1 - \mathcal{G}_i^l) \times O_i), \tag{2.10}$$

where O is a proto-objects map, l is the image index.

From Table. **2.1**, we observe that our method overall outperforms Hou and Zhang's method [6] [5] and Itti's method [7] [71].

[5] Downloadable from http://www.facedetection.com/facedetection/datasets.htm

[6] Downloadable from http://bcmi.sjtu.edu.cn/~houxiaodi/

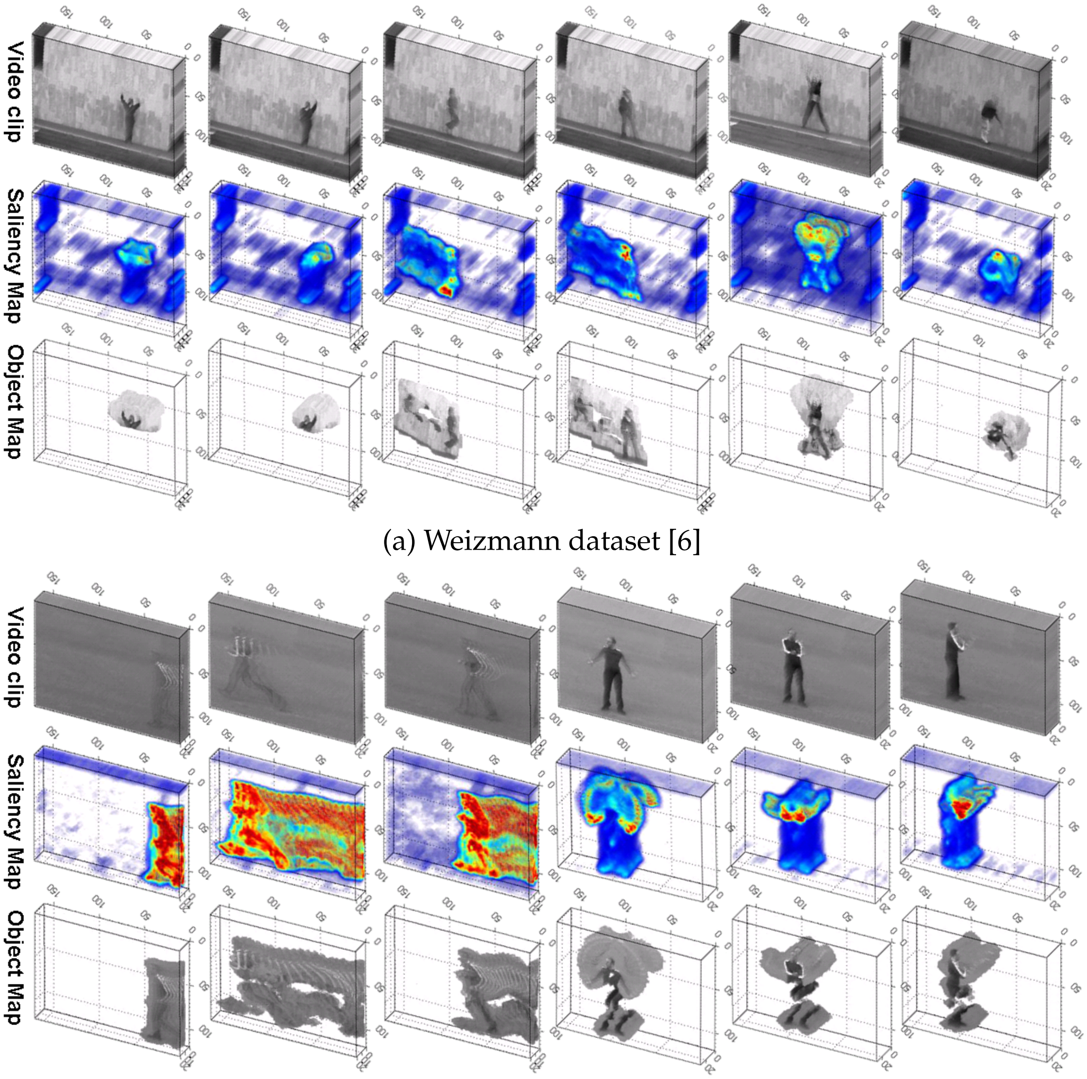
[7] Downloadable from http://www.saliencytoolbox.net/

Table 2.1: Performance comparison of the methods on finding proto-objects in Hou and Zhang's dataset [5]. We compare HR and FAR of three methods at a fixed FAR and a fixed HR respectively. Our method overall outperforms others.

	Our method	Hou and Zhang [5]	Itti et al [71]
HR	**0.5933**	0.4309	0.2482
Fixed FAR	0.1433	0.1433	0.1433
Fixed HR	0.5076	0.5076	0.5076
FAR	**0.1048**	0.1688	0.2931

2.3.1.2 Detecting Actions in Videos

The goal of action recognition is to classify a given action query into one of several pre-specified categories. Here, a query video may include a complex background which deteriorates recognition accuracy. In order to deal with this problem, it is necessary to have a procedure which automatically segments from the query video a small cube that only contains a valid action. Space-time saliency can provide such a mechanism. In order to compute the space-time saliency map, we only use the illumination channel because color information does not play a vital role in detecting motion saliency. We downsample each frame of input video I to a coarse spatial scale (64×64) in order to reduce the time-complexity (we do not downsample the video in the time domain.) We then compute 3-D LARK of size 3 × 3 × 3 as features and generate feature matrices $\mathbf{k}_i$ in a (3 × 3 × 7) local space-time neighborhood. The number of 3-D LARK used in the feature matrix $\mathbf{k}_i$ is set to 1 for time efficiency. The procedure for detecting space-time proto-objects and the rest of parameters remain the same as in the 2-D case. A typical run of space-time saliency detection takes about 52 seconds on 50 frames of a video at spatial scale (64×64) on an Intel Pentium 4, 2.66 GHz core 2 PC with 2 GB



(a) Weizmann dataset [6]

(b) KTH dataset [7]

Figure 2.8: Some examples of detecting salient human actions in the video (a) the Weizmann dataset [6] and (b) the KTH dataset [7].

RAM.

Fig. **2.8** shows that the proposed space-time saliency detection method successfully detects only salient human actions in both the Weizmann dataset [6] and the KTH dataset [7]. Our method is also robust to the presence of fast camera zoom in and

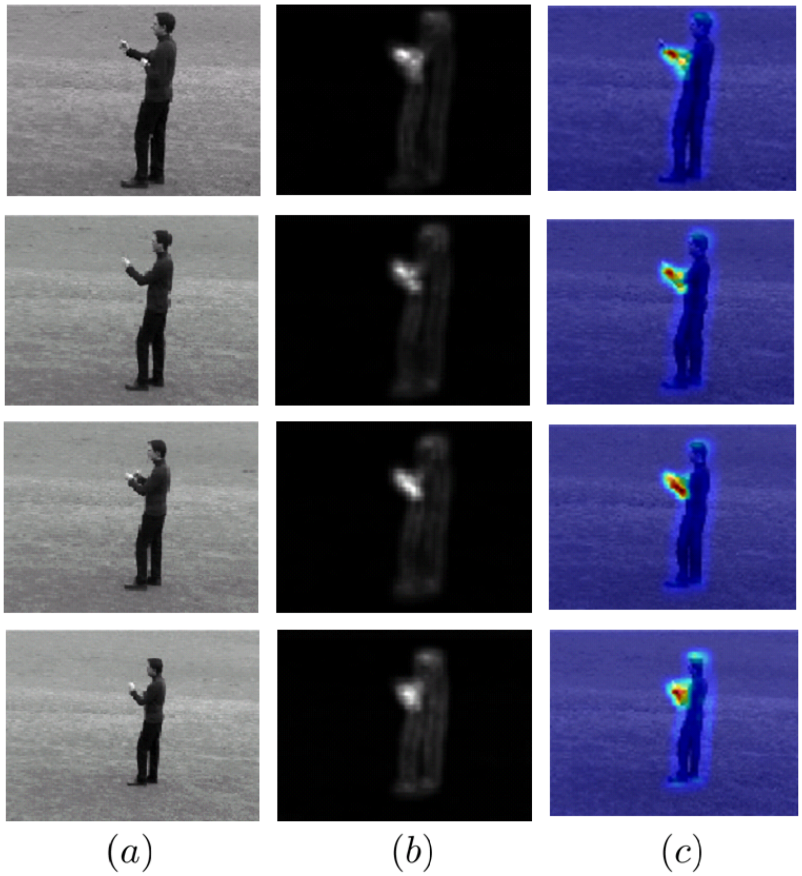


Figure 2.9: Space-time saliency detection even in the presence of fast camera zoom-in. Note that a man is performing a boxing action while a camera zoom is activated.

out as shown in Fig. **2.9** where a man is performing a boxing action while a camera zoom is activated.

2.3.2 Predicting Human Visual Fixation Data

2.3.2.1 Static Images

In this section, we used an image database and its corresponding fixation data collected by Bruce and Tsotsos [8] as a benchmark for quantitative performance

Table 2.2: Prediction of human eye fixations when viewing color images. SE means standard errors.

Model	KL (SE)	ROC (SE)
Itti *et al.* [71]	0.1130 (0.0011)	0.6146 (0.0008)
Bruce and Tsotsos [8]	0.2029 (0.0017)	0.6727 (0.0008)
Gao *et al.* [4]	0.1535 (0.0016)	0.6395 (0.0007)
Zhang *et al.* [72]	0.2097 (0.0016)	0.6570 (0.0008)
Hou and Zhang [5]	0.2511 (0.0019)	0.6823 (0.0007)
Our method (HOG)	0.1533 (0.0015)	0.4427 (0.0006)
Our method (SIFT)	0.0857 (0.0009)	0.5548 (0.0007)
Our method (NLM)	0.2174 (0.002)	0.6376 (0.0008)
Our method (BL)	0.2286 (0.0019)	0.651 (0.0007)
Our method (SSIM)	0.2337 (0.0021)	0.6474 (0.0008)
Our method (LARK)	**0.2779** (0.002)	**0.6896** (0.0007)

analysis and comparison. This dataset contains eye fixation records from 20 subjects for a total of 120 images of size 681 × 511. The parameter settings are the same as explained in Section 2.3.1. Some visual results of our model are compared with state of the art methods in Figs. **2.10** and **2.11**. As opposed to Bruce's method [8] which is quite sensitive to textured regions, and SUN [72] which is somewhat better in this respect, the proposed method is much less sensitive to background texture.

To compare the methods quantitatively, we computed the area under receiver operating characteristic (ROC) curve, and KL-divergence by following the experimental protocol of [72].

In [72], Zhang et al. pointed out that the dataset collected by Bruce [8] is center-biased and the methods by Itti et al. [71], Bruce et al. [8] and Gao et al. [4] are all corrupted by edge effects which resulted in relatively higher performance than they should have (See Fig. **2.12**.). We compare our model against Itti et al.[8] [71], Bruce and

[8] Downloadable from `http://ilab.usc.edu/toolkit/home.shtml`

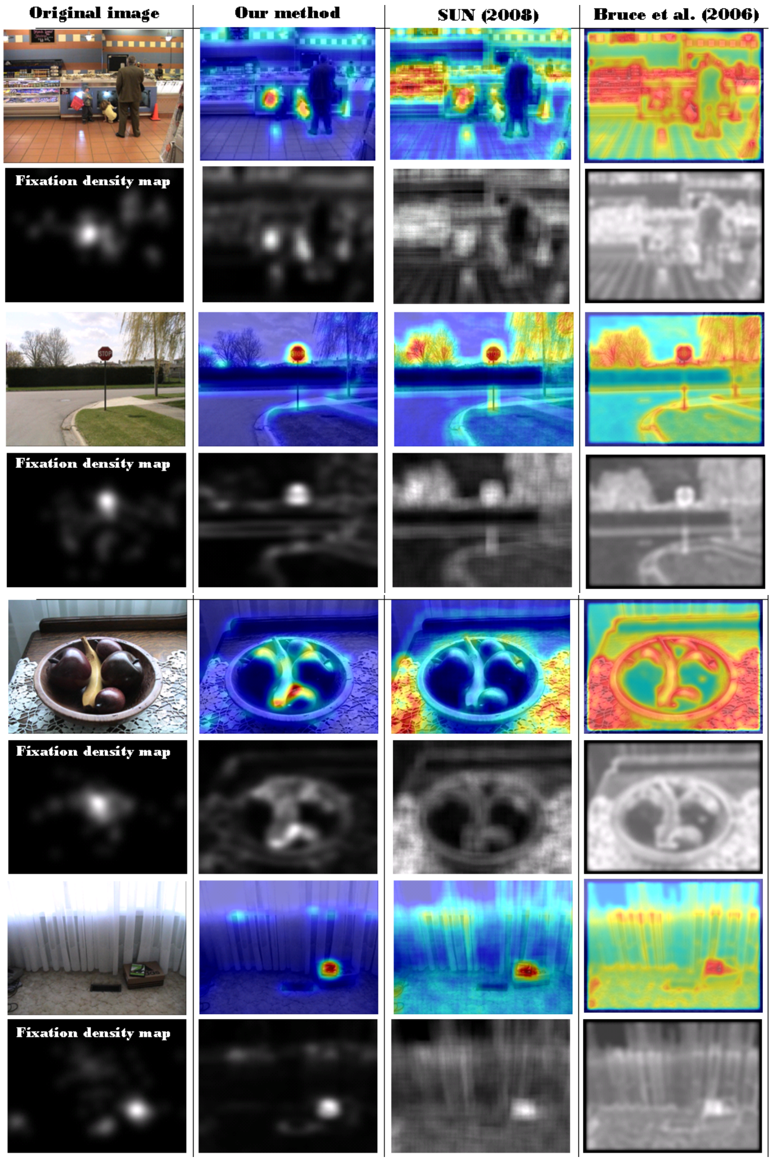


Figure 2.10: Examples of saliency maps with comparison to the state of the art methods. Human fixation density maps are derived from human eye fixation data and are shown right below the original images. Visually, our method outperforms other state of the art methods.

Tsotsos[9] [8], Gao et al. [4], and SUN[10] [72]. For the evaluation of the algorithm, we used the same procedure as in [72]. More specifically, we first compute true positives

[9] Downloadable from `http://web.me.com/john.tsotsos/VisualAttention/ST_and_Saliency.html`

[10] Downloadable from `http://www.roboticinsect.net/index.htm`

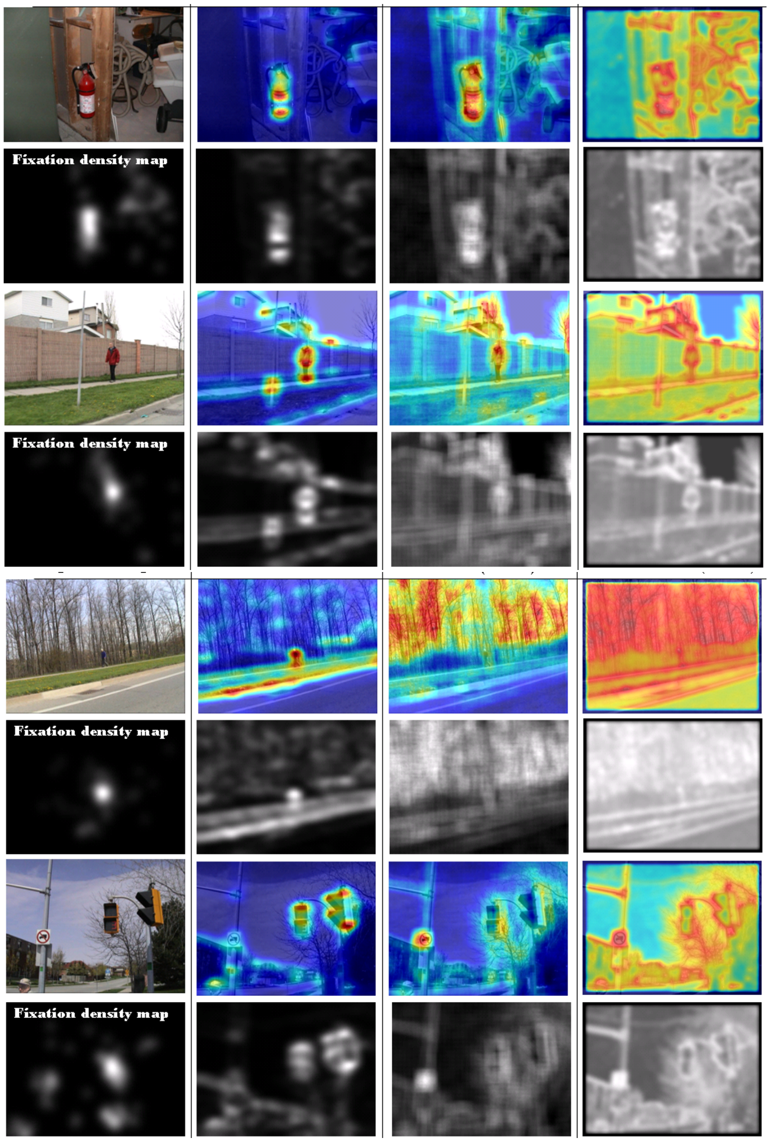


Figure 2.11: Examples of saliency maps with comparison to the state of the art methods. Human fixation density maps are derived from human eye fixation data and are shown right below the original images. Visually, our method outperforms other state of the art methods.

from the saliency maps based on the human eye fixation points. In order to calculate false positives from the saliency maps, we use the human fixation points from other images by permuting the order of images. This permutation of images is repeated 100

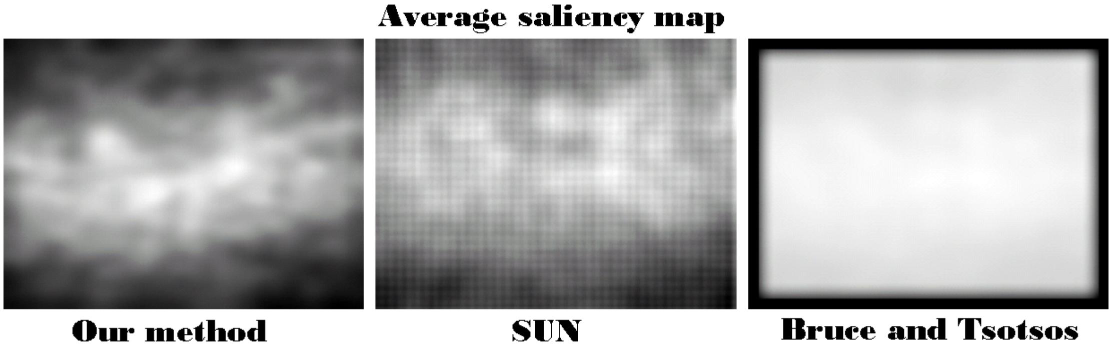


Figure 2.12: Comparison of average saliency maps on human fixation data by Bruce and Tsotsos [8]. Averages were taken across the saliency maps for a total of 120 color images.

times. Each time, we compute KL-divergence between the histograms of true positives and false positives and average them over 100 trials. When it comes to calculating the area under the ROC curve, we compute detection rates and false alarm rates by thresholding histograms of true positives and false positives at each stage of shuffling. The final ROC area is the average value over 100 permutations. The mean and the standard errors are also reported in Table **2.2**. As we alluded to earlier, we also compare the superiority of LARKs to other descriptors described in Chap. 1. We replaced LARKs with other descriptors (HOG[11], SIFT[12], NLM, BL[13], and SSIM[14]) in the proposed saliency detection framework while remaining the rest of the step same. Our model outperforms all the other state of the art methods, and LARKs perform better than all the other descriptors in terms of both KL-divergence and ROC area.

- As we alluded to in Section 1.3 earlier, our LARK features are robust to the presence of noise and changes in brightness and contrast. Fig. **2.14** well demonstrates

[11] `http://www.robots.ox.ac.uk/~vgg/research/caltech/phog.html`

[12] `http://people.csail.mit.edu/ceilu/ECCV2008`

[13] For NLM and BL, we used our own Matlab implementation

[14] `http://www.robots.ox.ac.uk/~vgg/software/SelfSimilarity`

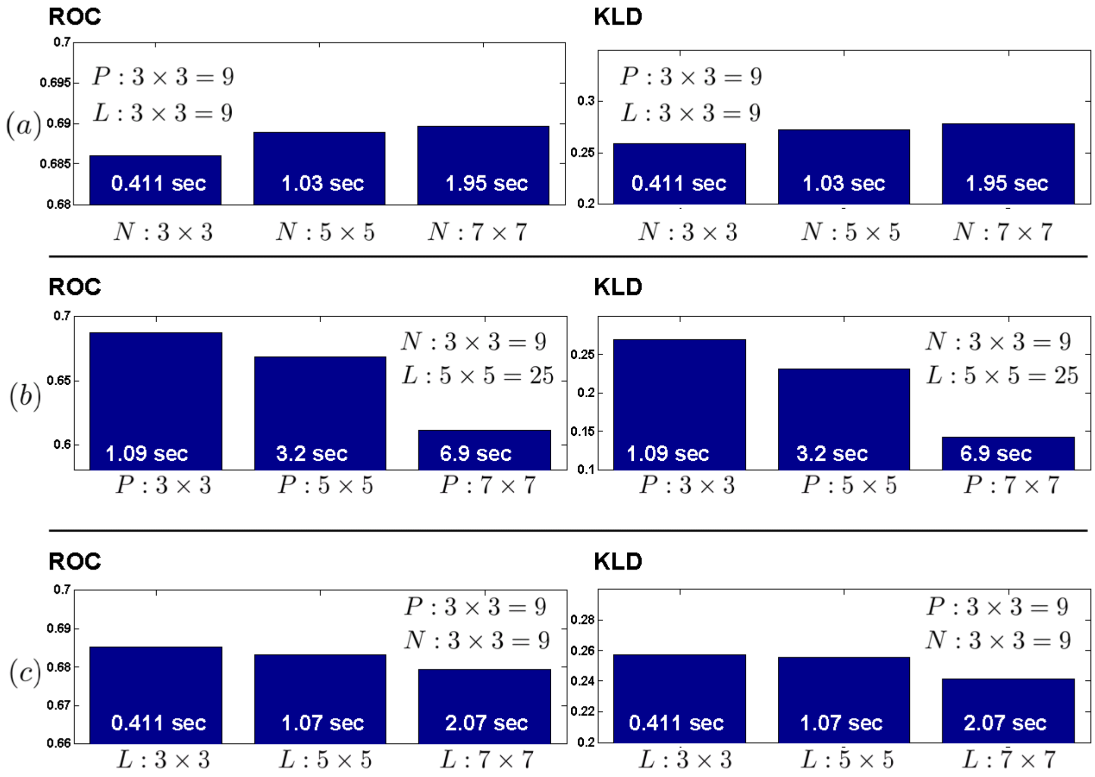


Figure 2.13: Performance comparison on human fixation data by Bruce and Tsotsos [8] with respect to the choice of 1) *N*: size of cneter+surrouding region for computing self-resemblance 2) *P*: size of LARK; and 3) *L*: number of LARK used in the feature matrix. Run time on one image is shown on top of each bar.

that the self-resemblance maps based on LARK features are not influenced by various distortions such as white-color noise, contrast change, and brightness change.

- We further examined how the performance of the proposed method is affected by the choice of parameters such as 1) *N*: size of center+surrouding region for computing self-resemblance 2) *P*: size of LARK; and 3) *L*: number of LARK used in the feature matrix. As shown in Fig. **2.13**, it turns out that as we increase *N*, the overall performance is improved while increasing *P* and *L* rather deteriorates the performance. Overall, the best performance was achieved with the choice of $P = 9 = (3 \times 3), L = 9 =$

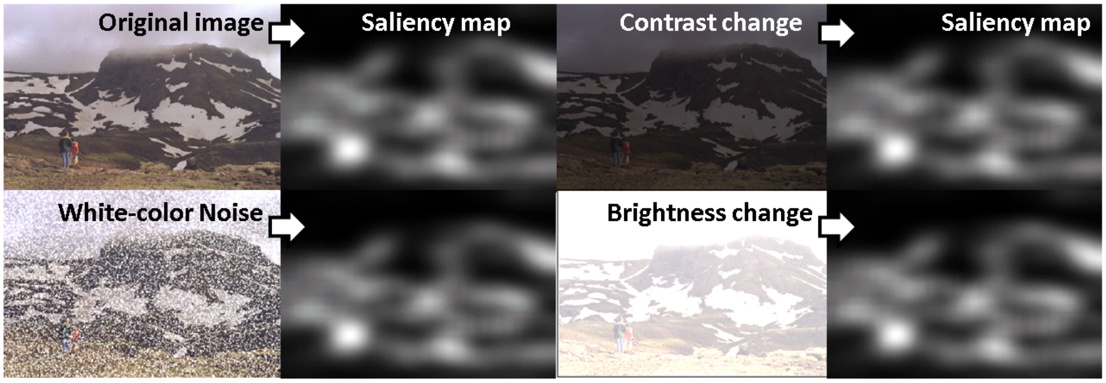


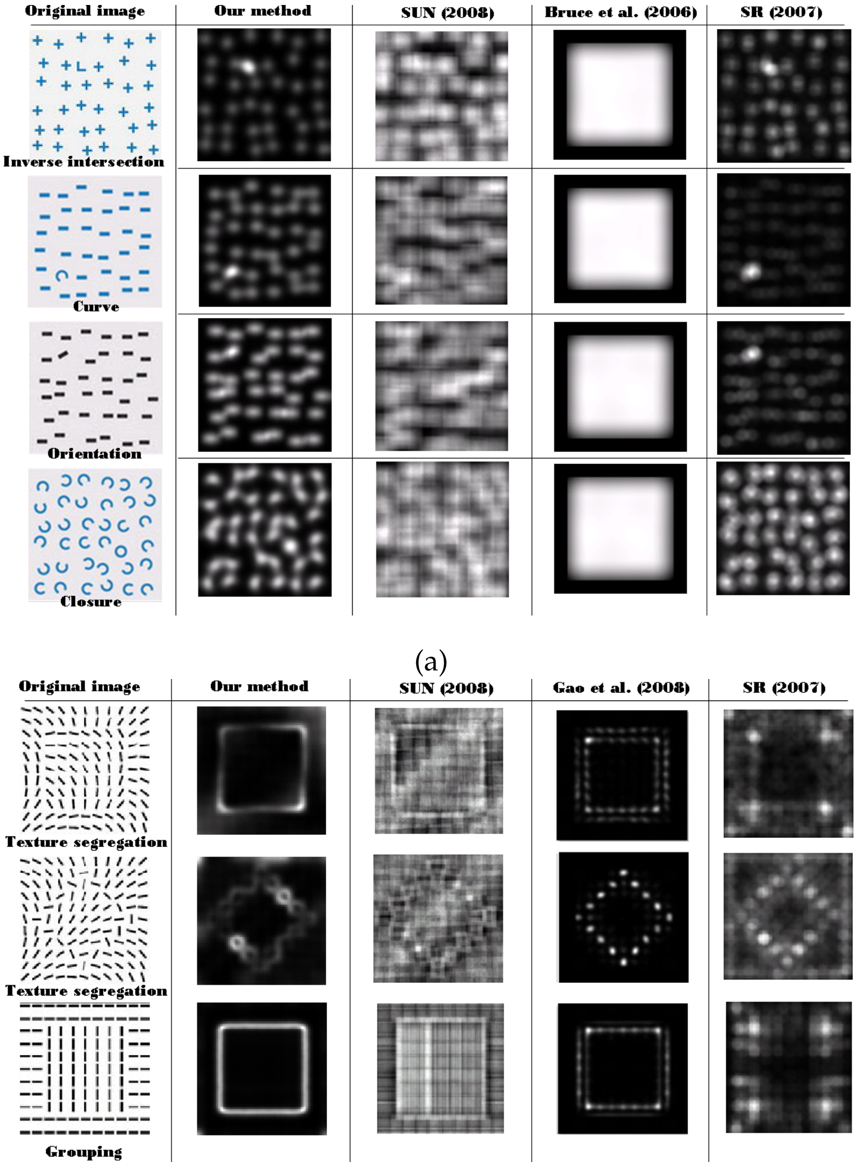
Figure 2.14: Our saliency model is largely unaffected by various distortions such as white-color noise, brightness change, and contrast change.

(3 × 3), and $N = 49 = (7 \times 7)$ at the expense of increased runtime.

It is important to note that while the LARK size (P) and the number of LARK (L) determine a feature dimension, the surrounding size (N) affects how many surrounding feature matrices would be compared with the center feature matrix. We do not wish to increase feature dimensions unnecessarily. Instead, we keep the surrounding size large enough so that we could get a reasonable self-resemblance value.

2.3.2.2 Response to Psychological Pattern

We also tested our method on psychological patterns. Psychological patterns are widely used in attention experiments not only to explore the mechanism of visual search, but also to test effectiveness of saliency maps [9, 97]. As shown in Fig. **2.15**, whereas SUN [72] and Bruce's method [8] failed to capture perceptual differences in most cases, Gao's method [4] and Spectral Residual [5] tend to capture perceptual organization rather better. Overall, however, the proposed saliency algorithm outperforms other methods in all cases including closure pattern (see Fig. **2.15** a) and texture segregation (see Fig. **2.15** b) which seem to be very difficult even for humans to distinguish.



(a)

(b)

Figure 2.15: Examples of Saliency map on psychological patterns. (a) images are from [5] (b) images are from [4].

The proposed method also predicts search asymmetry [9]. As shown in Fig. **2.16**, it is evident that our method mimics the human tendency of finding a Q (or a plus)

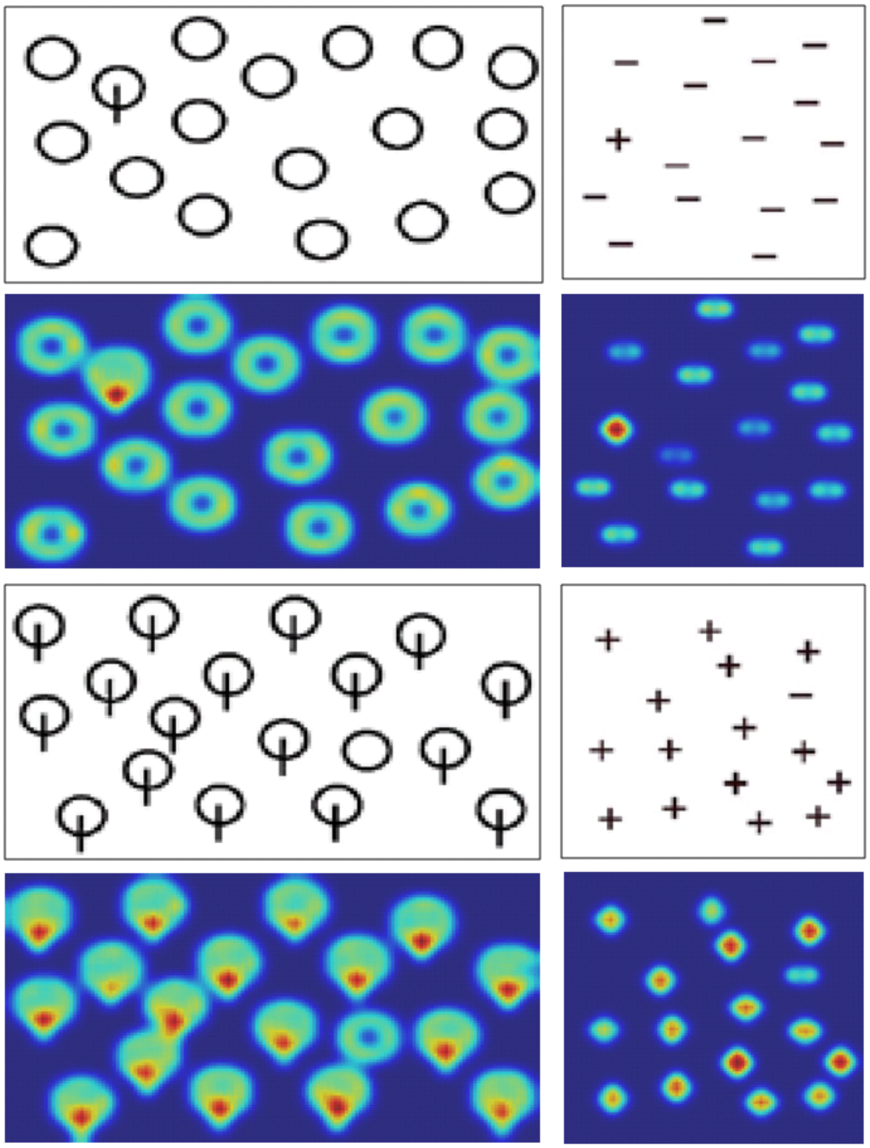


Figure 2.16: Top: The task of finding a Q among Os is easier than finding an O among Qs. Bottom: The task of finding a plus among dashes is easier than finding a dash among plus. This effect demonstrates a specific example of search asymmetry discussed in [9].

among Os (or dashes) to be easier than finding an O (or a dash) among Qs (pluses).

2.3.2.3 Dynamic Scenes

In this section, we quantitatively evaluate our space-time saliency algorithm on the human fixation video data from Itti et al. [10]. This dataset consists of a total of 520 human eye-tracking data traces recorded from 8 distinct subjects watching 50 different videos (TV programs, outdoors, test stimuli, and video games: about 25 minutes of total playtime). Each video has a resolution of size 640 × 480. Eye movement data was collected using an ISCAN RK-464 eye-tracker. For evaluation, two hundred (four subjects × fifty video clips) eye movement traces were used (see [10] for more details.) As similarly done earlier, we computed the area under ROC curve, and the KL-divergence. We compare our model against Bayesian Surprise [71] and SUNDAy [73]. Note that human eye movement data collected by Itti et al. [10] is also center-biased, and Bayesian Surprise [10] is corrupted by edge effects which resulted in relatively higher performance than it should have.

For the evaluation of the algorithm, we first compute true positives from the saliency maps based on the human eye movement fixation points. In order to calculate false positives from the saliency maps, we use the human fixation points from frames of other videos by permuting the order of video. This permutation of images is repeated 10 times. Each time, we compute KL-divergence between the histograms of true positives and false positives and average them over 10 trials. When it comes to calculating the area under the ROC curve, we compute detection rates and false alarm rates by thresholding histograms of true positives and false positives at each time of shuffling. The mean ROC area and the mean KL-divergence are reported in Table **2.3**. Some visual results of our model are shown in Fig. **2.17**. Our model outperforms Bayseian Surprise and SUNDAy in terms of both KL-divergence and ROC area.

Table 2.3: Prediction of human eye fixations when viewing videos [10].

Model	KL(SE)	ROC(SE)
Bayesian Surprise [10]	0.034	0.581
SUNDAy [73]	0.041	0.582
Our method	**0.262**(0.0085)	**0.589**(0.0031)

It seems at first surprising that our KL-divergence value is much higher than Bayesian Surprise[10] and SUNDAy[73] while there is a rather smaller difference between ROC areas. However, this phenomenon can be explained as follows. While the range of ROC area is limited from 0 to 1, the range of KL-divergence is from 0 to ∞. The KL-divergence is asymptotically related to the probability of detection and false alarm rate, and provides an upper bound on the detection performance [98, 99]. Namely, as the number of samples increases, $P_f(1 - P_d) \to \exp(-\alpha \mathscr{J})$, where α is a constant and $\mathscr{J}$ is KL-divergence. Even though there is a large difference between KL-divergence values, the difference in ROC area can be relatively small.

Our model is simple, but very fast and powerful. In terms of time complexity, a typical run time takes about 8 minutes (Zhang et al. [73] reported that their method runs in Matlab on a video of about 500 frames in minutes on a Pentium 4, 3.8 GHz dual core PC with 1 GB RAM.) on a video of size of 640 × 480 with about 500 frames while Bayesian Surprise requires hours because there are 432,000 distributions that must be updated with each frame.

2.3.3 Discussion

In the previous section, we have provided comprehensive experimental results which show that our method consistently outperforms other state-of-the art methods. We estimate saliency by using non-parametric density estimation, while other competing methods [71, 39, 4, 72] focused on fitting the conditional probability density function to a parametric distribution. In other words, we do not assume a distributional form or model for the data. As such, we call our method non-parametric. Even though we have a few parameters such as h, σ, and λ, these parameters are mostly set and fixed for all the experiments.

Our model is somewhat similar to Gao et al. [4] in the sense that a center-surround notion is used to compute saliency. One of the most important factors which makes the proposed method more effective is the use of LARKs as features. LARKs can capture local geometric structure exceedingly well even in the presence of signal uncertainty. In addition, unlike standard fusion methods which linearly and directly combine saliency maps computed from each color channel, we used the matrix cosine similarity to combine information from three color spaces. Our comprehensive experimental results indicate that the self-resemblance measure derived from a locally data-adaptive kernel density estimator is much more effective and simpler than other existing methods and does not require any training. Although our method is built entirely on computational principles, the resulting model structure exhibits considerable agreement with fixation behavior of the human visual system. With very good features like LARKs, the center-surround model is arguably an effective computational model of how the human visual system works. The proposed saliency detection based on the patch-level similarity can effectively reduce search space for object detection in

Summary – In this chapter, we have proposed a unified framework for both static and space-time saliency detection algorithm by employing 2-D and 3-D *LARKs*; and by using a nonparametric kernel density estimation based on Matrix Cosine Similarity (MCS). The proposed method can automatically detect salient objects in the given image and salient moving objects in videos. The proposed method is practically appealing because it is nonparametric, fast, and robust to uncertainty in the data. Experiments on challenging sets of real-world human fixation data (both images and videos) demonstrated that the proposed saliency detection method achieves a high degree of accuracy and improves upon state of the art methods. Due to its robustness to noise and other systemic perturbations, we also expect the present framework to be quite effective in other applications such as image quality assessment, background subtraction in dynamic scene, and video summarization. In the next chapter, we extend the concept of patch-level similarity within one image to image (video)-level similarity across images (videos) for object (action) detection.

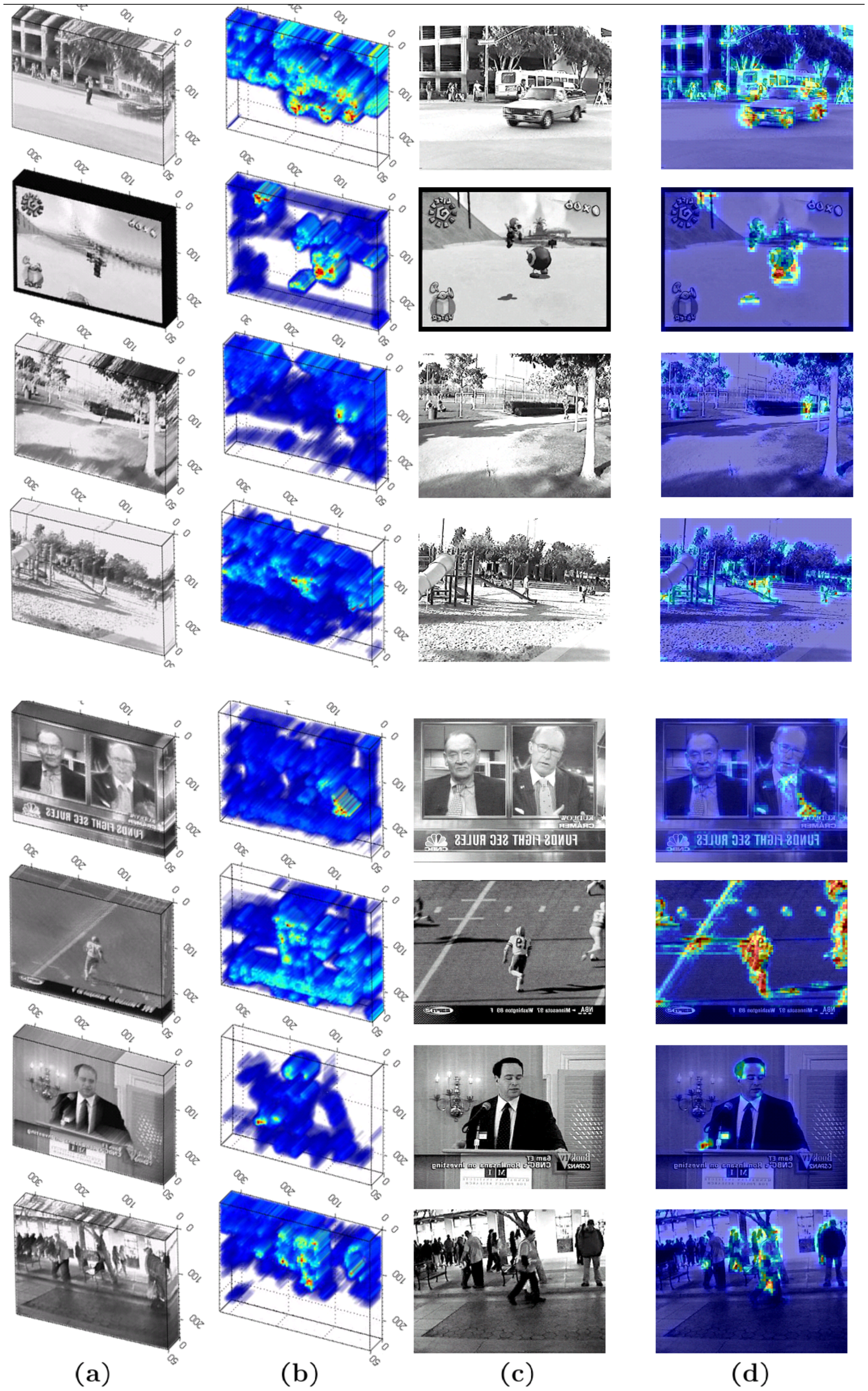


Figure 2.17: Some results on the video dataset [10] (a) video clips (b) space-time saliency map (c) a frame from (a) (d) a frame superimposed with corresponding saliency map from (b).

Chapter 3

Generic Object and Action Detection

Abstract – In this chapter, we introduce image (video)-level similarity to find matches between a query and a target. The proposed detection method is a *unified* framework that can deal with both objects and actions, and operates using a *single* query; does not require prior knowledge about objects (actions) being sought; and does not require any pre-processing step or segmentation of a target. As similarly done in saliency detection in Chapter 2, LARK descriptors in 2-D and 3-D are used to reliably capture local geometric structure. By employing Matrix Cosine Similarity (MCS), the proposed method yields a scalar resemblance map, indicating the likelihood of similarity between the query and all patches (cubes) in the target image (video). Our method detects the presence and location of objects (actions) similar to the given query through nonparametric significance tests. We provide optimality properties of the algorithm using a naive-Bayes framework. High performance is demonstrated on several challenging datasets, indicating successful detection of objects (actions) in diverse contexts and under different imaging conditions.

3.1 Introduction

Visual recognition (identifying objects/actions) is the main goal of this thesis as we alluded to in Chapter 1. According to the literature [100], visual recognition can also be divided into two parts: category recognition (classification) and detection/localization. The goal of object (action) detection is to separate objects (actions) of interest from the background in a target image (video) while object (action) classification is to classify a given object (action) into one of the pre-specified categories. In this chapter, we focus on solving object/action *detection* problems based on the concept of image-level similarity. Before we describe our proposed method, we briefly review related works.

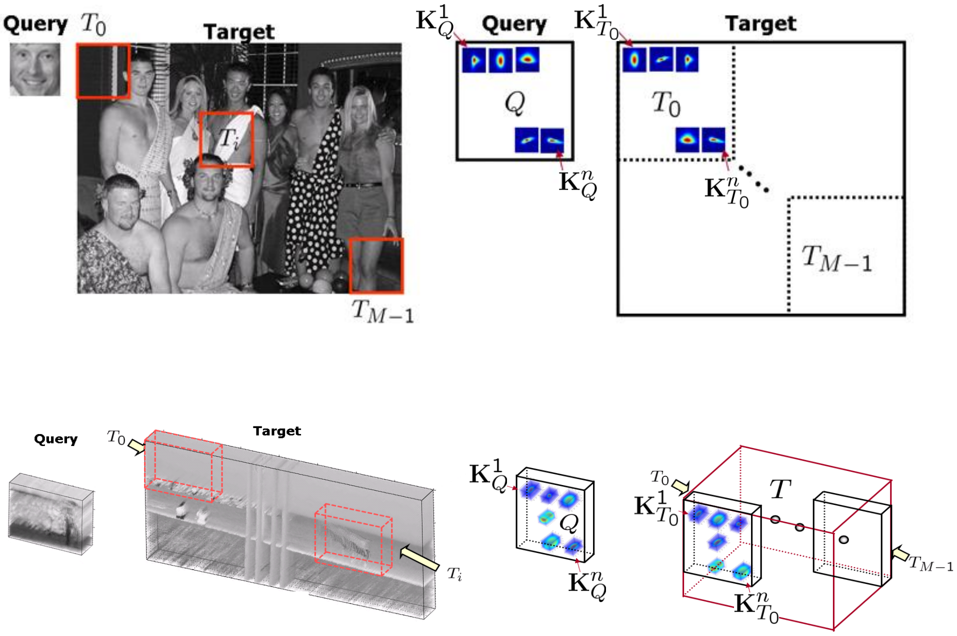
3.1.1 Related Works

Object Detection Object detection is a critical part in many applications such as image retrieval, scene understanding, and surveillance system; however it is still an open problem because the intra-class variation makes a generic detection very complicated, requiring various types of pre-processing steps. In the current literature, a popular object detection paradigm is *probabilistic constellation* [27] or *parts-and-shape models* [32] that represent not only the statistics of individual parts, but also their spatial layout. These are based on learning-based classifiers, that require an intensive learning/training phase of the classifier parameters and thus are called parametric methods. For the purpose of localization, a sliding window scheme is usually used by taking the peak confidence values as an indication of the presence of an objet in a given region. Recently, [101] proposed an efficient sub-window search method based on branch and bound scheme and attained a huge speed-up. To make a real-time object detection system[1] while achieving high detection rates, methods combining classifiers in a cascade [29, 102] have also been proposed. In PASCAL 2009 object detection challenge [35], Felzenszwalb et al. [28] gained state of the art object detection performance based on mixtures of multi-scale deformable part models relying on HOG [3] and latent SVM for discriminative training.

Action Detection In the literature of action detection, the term “action" refers to a simple motion pattern as performed by a single subject, and represents mostly a physical human body motion. Recent approaches can be categorized on the basis of action *representation*; namely, appearance-based representation [103, 104], shape-based representation [105, 6], optical-flow-based representation [106, 107], interest-point-based representation [7, 108, 109, 110], and volume-based representation [111, 21, 112, 113, 22]. We refer the interested reader to [114] and references therein for a good summary.

As examples of the interest-point-based approach, Niebles et al. [115] considered videos as spatiotemporal bag-of-words by extracting space-time interest points and clustering the features, and then used a probabilistic Latent Semantic Analysis (pLSA) model to localize and categorize human actions. Yuan et al. [116] also used spatiotemporal features as proposed by [109]. They extended the naive Bayes nearest neighbor classifier [41], which was developed for object recognition, to action recognition. By modifying the efficient searching method based on branch-and-bound [101] for the 3-D case, they provided a very fast action detection method. However, the performance of these methods can degrade due to 1) the lack of enough training samples; 2) misdetections and occlusions of the interest points since they ignore global space-time information.

[1] We refer the reader to Chapter 4 for how we realize a real-time object detection system.



(a) (b)

Figure 3.1: Object and action detection problem (a) Given a query image (video) Q, we wish to detect/localize objects (actions) of interest in a target image (video) T. T is divided into a set of overlapping patches (cubes) (b) LARKs (3-D LARKs) capture the local (space-time) geometric structure of underlying data.

Shechtman and Irani [21] employed a three dimensional correlation scheme for action detection. They focused on subvolume matching in order to find similar motion between the two space-time volumes, which can be computationally heavy. Ke et al. [112] presented an approach which uses boosting on 3-D Haar-type features inspired by similar features in 2-D object detection [29]. While these features are very efficient to compute, many examples are required to train an action detector in order to achieve good performance. They further proposed a part-based shape and flow matching framework [117] and showed good action detection performance in crowded videos. Ning et al. [22] proposed a system to search for human actions using a coarse-to-fine approach with a five-layer hierarchical space-time model. These volumetric methods do not require background subtraction, motion estimation, or complex models of body configuration and kinematics. They tolerate modest variations in appearance, scale, rotation, and movement.

3.1.2 Overview of the Proposed Method

In this chapter, we propose to use LARKs in 2-D and 3-D for the problem of localizing objects (actions) of interest given a query and a target. Referring to Fig. **3.1**, by denoting the target (T), and the query (Q), we compute a dense set of LARKs (3-D LARKs) from each. These densely computed descriptors are highly informative, but taken together tend to be over-complete (redundant). Therefore, we derive features by applying dimensionality reduction (namely PCA) to these resulting arrays, in order to retain only the salient characteristics of the LARKs.

Generally, T is bigger than the query Q. Hence, we divide the target T into a set of overlapping patches (cubes for video) which are the same size as Q and assign a class to each patch (cube) (T_i). The feature vectors which belong to a patch (cube) are thought of as training examples in the corresponding class. The feature collections from Q and T_i form feature matrices $\mathbf{F}_Q$ and $\mathbf{F}_{T_i}$. We compare the feature matrices $\mathbf{F}_{T_i}$ and $\mathbf{F}_Q$ from i^{th} patch (cube) of T and Q to look for matches (image/video-level similarity). Inspired in part by the many studies [92, 118, 93] which took advantage of cosine similarity over the conventional Euclidean distance, we employ Matrix Cosine Similarity (MCS) as a similarity measure which generalizes the vector cosine similarity [119, 120]. We illustrate the optimality properties of the proposed approach using a naive Bayes framework, which leads to the use of the MCS measure in Appendix A.

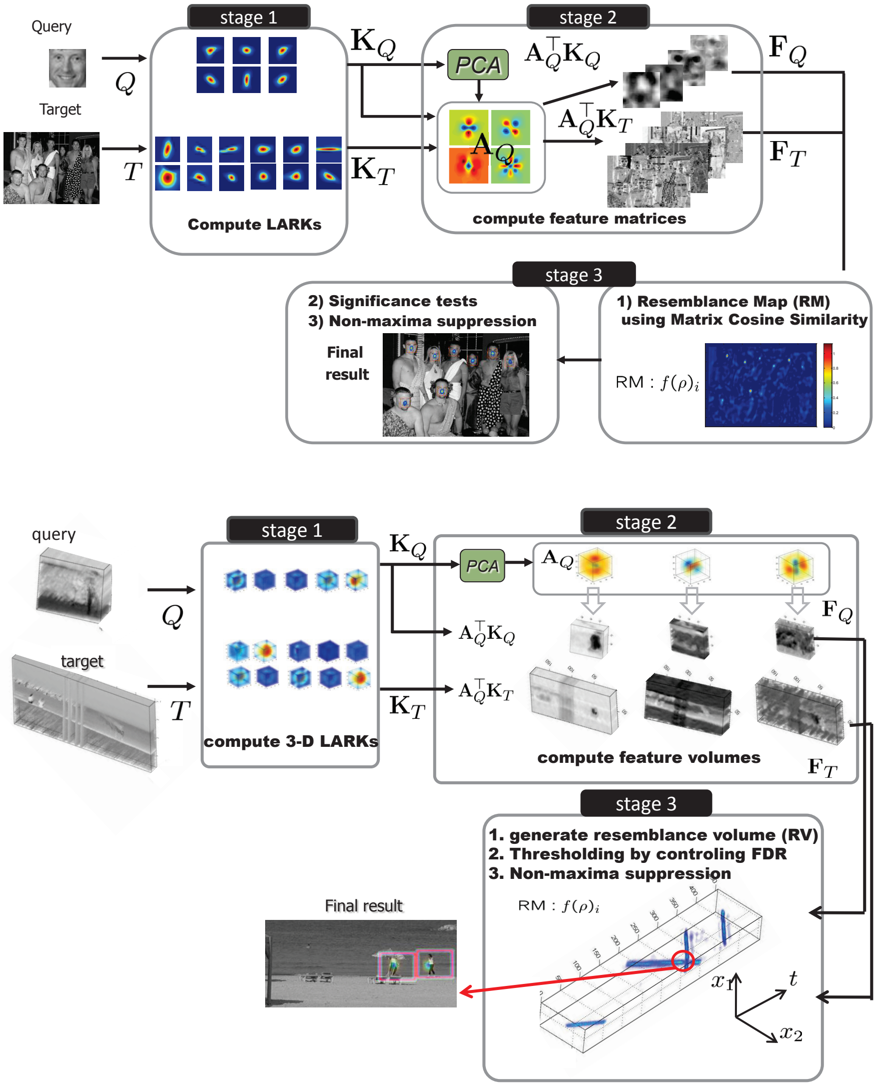


Figure 3.2: Object and action detection system overview (There are broadly three stages.)

In order to deal with the case where the target may not include any objects (actions) of interest or when there are more than one object (action) in the target, we also adopt the idea of a significance test.

For action detection, it is generally assumed that the query video is smaller than target video. However, this is not true in practice and a query video may indeed include a complex background which degrades recognition accuracy. In order to deal with this problem, it is necessary to have a procedure which automatically segments from the query video a small cube that only contains a valid human action. For this, we employ space-time saliency detection [48], also described in Chapter 2. This idea not only allows us to extend the proposed detection framework to action category classification, but also improve both detection and classification accuracy by automatically removing irrelevant background from the query video.

Fig. **5.1** shows an overview of our proposed framework. The first stage consists of computing the normalized LARKs $\mathbf{K}_Q, \mathbf{K}_T$ and obtaining the salient feature matrices $\mathbf{F}_Q, \mathbf{F}_T$. In the second stage, we compare the feature matrices $\mathbf{F}_{T_i}$ and $\mathbf{F}_Q$ using the MCS. The final output is given after a sequence of significance tests, followed by non-maxima suppression [12].

3.2 Unified Training-free Detection Framework

3.2.1 Feature Representation in Image and Video

As shown in Fig. **3.1**, at a position $\mathbf{x}_i$, we use a normalized LARK as a local feature to represent inherent local geometry. Many studies [41, 121, 122] have shown that densely computed local image features give better results in classification tasks than key-point based local image features such as SIFT [18] which are designed for mainly invariance and compact coding. According to these studies, the distribution of the local image feature both in natural images as well as images of a specific object

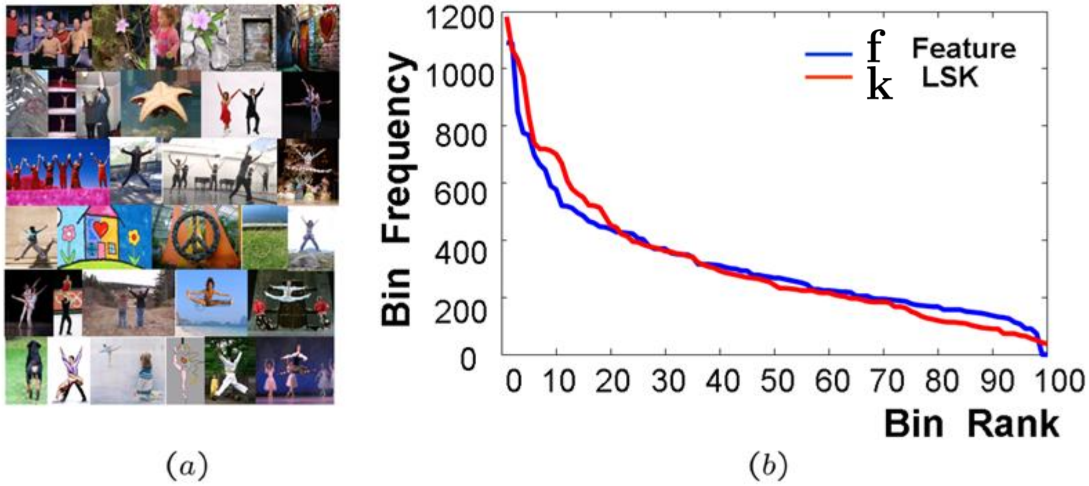


Figure 3.3: LARKs follow a power-law distribution. (a) Some example images (Shechtman's object dataset [11]) where LARKs were computed. (b) Plots of the bin density of LARKs and their corresponding low-dimensional features.

class follows a power-law (i.e., a long-tail) distribution [41, 121, 122]. In other words, the features are scattered out in a high dimensional feature space, and thus there basically exists no dense cluster in the feature space. In order to illustrate and verify that the normalized LARKs also satisfy this property as described in [11, 41] and follow a power-law distribution, we computed an empirical bin density (100 bins) of the normalized LARKs (using a total of $31,319$ LARKs) densely computed from 60 images (from Shechtman's general object dataset [11]) using the K-means clustering method. (See Fig. **3.3** for an example.) The same principle applies to 3-D LARKs[2].

Boiman et al. [41] observed that while an ensemble of local features with little discriminative power can together offer a significant discriminative power, both quantization and informative feature selection on a long-tail distribution can lead to

[2] We computed an empirical bin density (100 bins) of the normalized 3-D LARKs (using a total of $50,000$ 3-D LARKs) extracted from 90 videos of the Weizmann action dataset [6].

a precipitous drop in performance. Therefore, instead of any quantization and informative feature selection, we focus on reducing the dimension of densely computed LARKs using PCA to enhance the discriminative power and reduce computational complexity. It is worth noting that this approach was also taken by Ke et al. in [63] where PCA was applied to SIFT features, leading to enhanced performance. Ali and Shah [107] also applied PCA to derive salient kinematic features from optical flow in the action recognition task. This idea results in a new feature representation with a moderate dimension which inherits the desirable discriminative attributes of LARK. The distribution of the resulting features sitting on the low dimensional manifold also tends to follow a power-law distribution as shown in Fig. **3.3** (b) and this attribute of the features will be utilized in applying a nearest-neighbor approximation in the theoretical formulation in Appendix A. In order to organize K_Q and K_T, which are densely computed from Q and T, let $\mathbf{K}_Q$,$\mathbf{K}_T$ denote matrices whose columns are vectors $\mathbf{k}_Q$,$\mathbf{k}_T$, which are column-stacked (rasterized) versions of K_Q,K_T respectively:

$$\begin{aligned} \mathbf{K}_Q &= [\mathbf{k}_Q^1, \cdots, \mathbf{k}_Q^n] \in \mathbb{R}^{P \times n}, \\ \mathbf{K}_T &= [\mathbf{k}_T^1, \cdots, \mathbf{k}_T^{n_T}] \in \mathbb{R}^{P \times n_T}, \end{aligned} \qquad (3.1)$$

where P is the dimension of LARK, and n and n_T are the total number of LARKs in Q and T respectively.

As described in Fig. **5.1**, the next step is to apply PCA[3] to $\mathbf{K}_Q$ for dimensionality reduction and to retain only its salient characteristics. Applying PCA to $\mathbf{K}_Q$ we can retain the first (largest) d principal components[4] which form the columns of a matrix

[3] It is worth noting that the use of the PCA here may not be critical in the sense that any unsupervised subspace learning method such as Kernel PCA, LLE [123], LPP [124] CDA [93], CPCA [92], and CEA [92] can be used.

[4] Typically, d is selected to be a small integer such as 3 or 4 so that 80 to 90% of the “information" in the

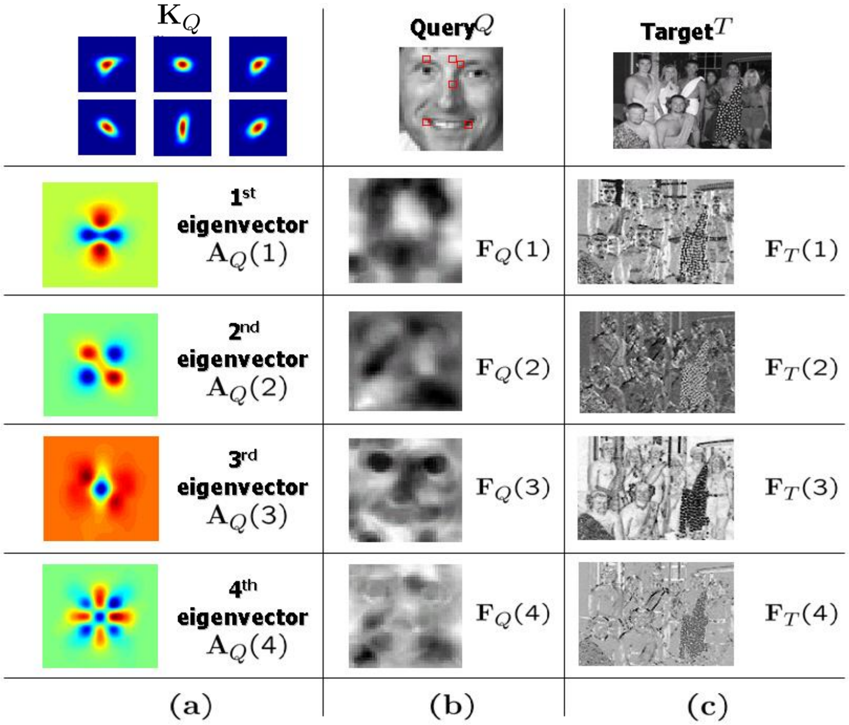


Figure 3.4: Face and car examples (a) : $\mathbf{A}_Q$ learned from a collection of LARKs $\mathbf{K}_Q$, (b): Feature row vectors of $\mathbf{F}_Q$ from query Q, (c) : Feature row vectors $\mathbf{F}_T$ from target image T. Eigenvectors and feature vectors were reshaped into image and up-scaled for illustration purposes.

$\mathbf{A}_Q \in \mathbb{R}^{P\times d}$. Next, the lower dimensional features are computed by projecting $\mathbf{K}_Q$ and $\mathbf{K}_T$ onto $\mathbf{A}_Q$:

$$\begin{aligned} \mathbf{F}_Q &= [\mathbf{f}_Q^1, \cdots, \mathbf{f}_Q^n] = \mathbf{A}_Q^\top \mathbf{K}_Q \in \mathbb{R}^{d\times n}, \\ \mathbf{F}_T &= [\mathbf{f}_T^1, \cdots, \mathbf{f}_T^{n_T}] = \mathbf{A}_Q^\top \mathbf{K}_T \in \mathbb{R}^{d\times n_T}. \end{aligned} \tag{3.2}$$

Figs. **5.6** and **3.5** illustrate the principal components in $\mathbf{A}_Q$ and shows what the features $\mathbf{F}_Q, \mathbf{F}_T$ look like for some examples such as face and walking action.

LARKs would be retained. (i.e., $\frac{\sum_{i=1}^{d}\lambda_i}{\sum_{i=1}^{P}\lambda_i} \geq 0.8$ (to 0.9) where λ_i are the eigenvalues.)

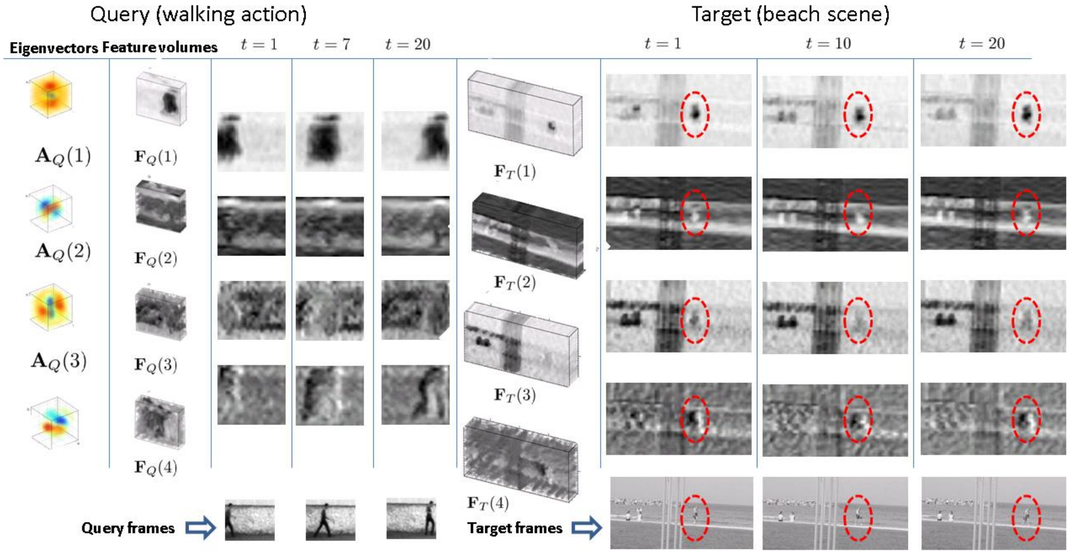


Figure 3.5: Examples of top 3 principal components in $\mathbf{A}_Q$ for walking action. Note that these eigenvectors reveal geometric characteristic of queries in both space and time domain, and thus they are totally different from linear 3-D Gabor filters. Feature row vectors of $\mathbf{F}_Q$ and $\mathbf{F}_T$ are computed from query Q and target T respectively. Eigenvectors and feature vectors were transformed to volume and upscaled for illustration purposes.

3.2.2 Resemblance Map

The next step in the proposed framework is to generate a resemblance map (RM)[5] based on the measurement of a distance between the computed features $\mathbf{F}_Q, \mathbf{F}_{T_i}$ (a chunk of $\mathbf{F}_T$). Earlier works such as [92, 91, 93] have shown that correlation based metrics outperforms the conventional Euclidean and Mahalanobis distances for the classification and subspace learning tasks. Motivated by the effectiveness of correlation-based similarity measure, we explore the idea behind Matrix Cosine Similarity in this section. In general, “correlation" indicates the strength and direction of a linear rela-

[5] We use resemblance volume (RV) for action detection.

tionship between two random variables. But the idea of correlation is quite malleable. Indeed, according to Rodgers et al. [120], there are at least thirteen distinct ways to look at correlation! However, we are interested in two main types of correlation: Pearson's correlation coefficient which is the familiar standard correlation coefficient, and the cosine similarity (so-called non-Pearson-compliant). Note that the cosine similarity coincides with the Pearson's correlation when each vector is centered to have zero-mean. In several earlier papers including [119, 125], it has been shown that the Pearson correlation is less discriminating than the cosine similarity due to the fact that centered values are less informative than the original values, and the computation of centered values is sensitive to zero or small values in the vectors. Since the discriminative power is critical in our detection framework, we focus on the cosine similarity. The cosine similarity is defined as the inner product between two normalized vectors as follows:

$$\rho(\mathbf{f}_Q, \mathbf{f}_{T_i}) =< \frac{\mathbf{f}_Q}{\|\mathbf{f}_Q\|_2}, \frac{\mathbf{f}_{T_i}}{\|\mathbf{f}_{T_i}\|_2} >= \frac{{\mathbf{f}_Q}^\top \mathbf{f}_{T_i}}{\|\mathbf{f}_Q\|_2 \|\mathbf{f}_{T_i}\|_2} = \cos\theta_i \in [-1, 1], \tag{3.3}$$

where $\mathbf{f}_Q, \mathbf{f}_{T_i} \in \mathbb{R}^d$ are column vectors. The cosine similarity measure therefore focuses only on the angle (phase) information while discarding the scale information.

If we deal with the features $\mathbf{F}_Q, \mathbf{F}_{T_i}$ which consist of a set of vectors, MCS can be defined as a natural generalization using the "Frobenius inner product" between two normalized matrices as follows[6]:

$$\rho(\mathbf{F}_Q, \mathbf{F}_{T_i}) =< \overline{\mathbf{F}}_Q, \overline{\mathbf{F}}_{T_i} >_F = \text{trace}\left(\frac{\mathbf{F}_Q^\top \mathbf{F}_{T_i}}{\|\mathbf{F}_Q\|_F \|\mathbf{F}_{T_i}\|_F} \right) \in [-1, 1], \tag{3.4}$$

where, $\overline{\mathbf{F}}_Q = \left[\frac{\mathbf{f}_Q^1}{\|\mathbf{F}_Q\|_F}, \cdots, \frac{\mathbf{f}_Q^n}{\|\mathbf{F}_Q\|_F} \right]$ and $\overline{\mathbf{F}}_{T_i} = \left[\frac{\mathbf{f}_{T_i}^1}{\|\mathbf{F}_{T_i}\|_F}, \cdots, \frac{\mathbf{f}_{T_i}^n}{\|\mathbf{F}_{T_i}\|_F} \right]$.

This generalization is also known as "vector correlation" in the statistics literature [126]. Fu et al. [92] also used a generalized cosine similarity tensor case for

[6] Note that we introduced MCS in Chapter 2.

subspace learning, and showed performance improvement in the task of image classification. Returning to our definition, if we look at Equation (3.4) carefully, we note that one can rewrite it as a weighted average of the cosine similarities $\rho(\mathbf{f}_Q,\mathbf{f}_{T_i})$ between each pair of corresponding feature vectors (i.e., columns) in $\mathbf{F}_Q,\mathbf{F}_{T_i}$ as follows:

$$\rho(\mathbf{F}_Q,\mathbf{F}_{T_i}) = \sum_{\ell=1}^{n} \frac{\mathbf{f}_Q^{\ell\top}\mathbf{f}_{T_i}^{\ell}}{\|\mathbf{F}_Q\|_F\|\mathbf{F}_{T_i}\|_F} = \sum_{\ell=1}^{n}\rho(\mathbf{f}_Q^{\ell},\mathbf{f}_{T_i}^{\ell})\frac{\|\mathbf{f}_Q^{\ell}\|\|\mathbf{f}_{T_i}^{\ell}\|}{\|\mathbf{F}_Q\|_F\|\mathbf{F}_{T_i}\|_F}. \tag{3.5}$$

The weights are represented as the product of $\frac{\|\mathbf{f}_Q^{\ell}\|}{\|\mathbf{F}_Q\|_F}$ and $\frac{\|\mathbf{f}_{T_i}^{\ell}\|}{\|\mathbf{F}_{T_i}\|_F}$ which indicate the relative importance of each feature in the feature sets $\mathbf{F}_Q,\mathbf{F}_{T_i}$. We see here an advantage of the MCS in that it takes care of the strength and angle similarity of vectors at the same time. Hence, this measure not only generalizes the vector cosine similarity, but also overcomes the disadvantages of the conventional Euclidean distance which is sensitive to outliers. We compute $\rho(\mathbf{F}_Q,\mathbf{F}_{T_i})$ over all the target patches to generate resemblance map (RM)[7].

In Appendix 3A, we further generalize the cosine similarity to a "Canonical Cosine Similarity" which is a corresponding version of the canonical correlation analysis (CCA) [127] for the vector data case where we have a set of features separately computed from multiple sources (for instance, color image (YCbCr or CIE L*a*b*) or a sequence of images). In a similar vain as Boiman et al. [41], we show in Appendix A that a particular version of optimal naive-Bayes decision rule can actually lead to the

[7] This can be efficiently implemented by column-stacking the matrices $\mathbf{F}_Q,\mathbf{F}_{T_i}$ and simply computing the cosine similarity between two long column vectors.

$$\begin{aligned} \rho_i \quad \equiv \quad \rho(\mathbf{F}_Q,\mathbf{F}_{T_i}) &= \sum_{\ell=1}^{n} \frac{\mathbf{f}_Q^{\ell\top}\mathbf{f}_{T_i}^{\ell}}{\|\mathbf{F}_Q\|_F\|\mathbf{F}_{T_i}\|_F} = \sum_{\ell=1,j=1}^{n,d} \frac{f_Q^{(\ell,j)} f_{T_i}^{(\ell,j)}}{\sqrt{\sum_{\ell=1,j=1}^{n,d}|f_Q^{(\ell,j)}|^2}\sqrt{\sum_{\ell=1,j=1}^{n,d}|f_{T_i}^{(\ell,j)}|^2}}, \\ &= \rho(\text{colstack}(\mathbf{F}_Q),\text{colstack}(\mathbf{F}_{T_i})) \in [-1,1], \end{aligned} \tag{3.6}$$

where $f_Q^{(\ell,j)}, f_{T_i}^{(\ell,j)}$ are elements in ℓ^{th} vector $\mathbf{f}_Q^{\ell}$ and $\mathbf{f}_{T_i}^{\ell}$ respectively, and colstack$(\cdot)$ means an operator which column-stacks (rasterizes) a matrix.

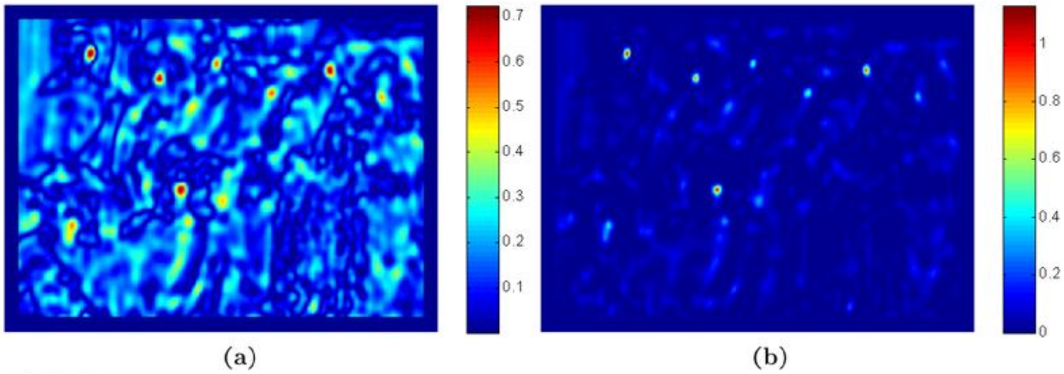


Figure 3.6: (a) Resemblance map (RM) which consists of $|\rho_i|$ (b) Resemblance map (RM) which consists of $f(\rho_i)$. Note that Q and T are the same examples shown in Fig **3.1**.

use of MCS measure.

Each pixel value of RM indicates the likelihood of similarity between the Q and T. When it comes to interpreting the value of correlation, it is noted in [128, 129] that $\rho_i^2 \in [0,1]$ describes the proportion of variance in common between the two feature sets as opposed to ρ_i which indicates a linear relationship between two feature matrices $\mathbf{F}_Q, \mathbf{F}_{T_i}$. At this point, we can use ρ_i directly as a measure of resemblance between the two feature sets. However, the shared variance interpretation of ρ_i^2 has several advantages. In particular, as for the final test statistic comprising the values in the resemblance map, we use the *proportion* of shared variance (ρ_i^2) to that of the "residual" variance ($1-\rho_i^2$). More specifically, RM is computed using the mapping function f as follows:

$$\text{RM}: f(\rho_i) = \frac{\rho_i^2}{1-\rho_i^2}. \tag{3.7}$$

In Fig. **3.6**, examples of RM based on $|\rho_i|$ and $f(\rho_i)$ are presented. Red color represents higher resemblance. As is apparent from these typical results, qualitatively, the resemblance map generated from $f(\rho_i)$ provides better contrast and dynamic range in the

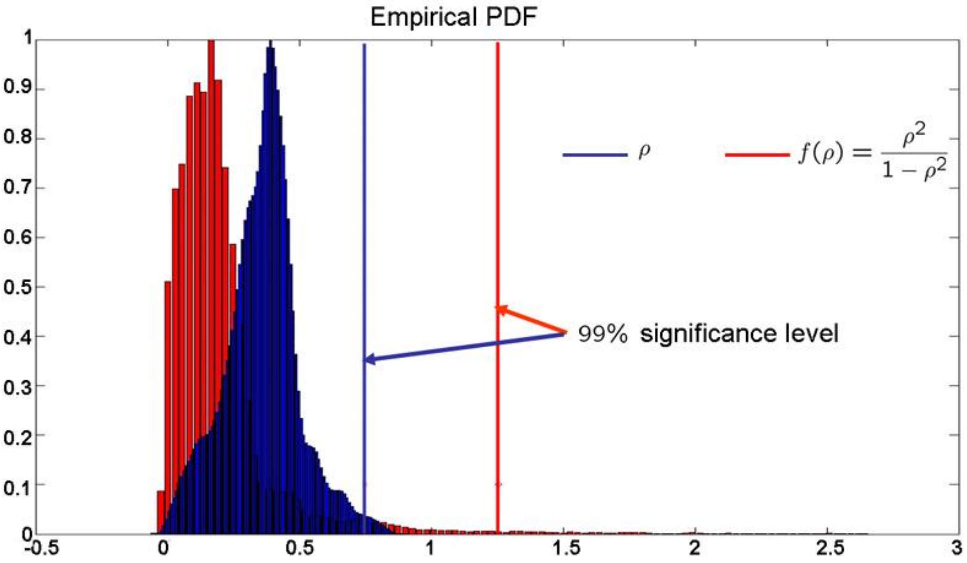


Figure 3.7: Comparison of empirical PDF between ρ and $\frac{\rho^2}{1-\rho^2}$.

result ($f(\rho_i) \in [0,\infty]$). More importantly, from a quantitative point of view, we note that $f(\rho_i)$ is essentially the Lawley-Hotelling Trace statistic [127, 130], which is used as an efficient test statistic for detecting correlation between two data sets. Furthermore, it is worth noting that historically, this statistic has been suggested in the pattern recognition literature as an effective means of measuring the separability of two data clusters (e.g. [131].)

3.2.3 Significance Test

If the task is to find the most similar patch (cube) T_i to the query Q in the target, one can choose the patch (cube) which results in the largest value in the RM (RV) (i.e., $\max f(\rho_i)$) among all the patches (cubes), no matter how large or small the value is in the range of $[0,\infty]$. This, however, is not wise because there may not be *any* object (action) of interest present in the target. We are therefore interested in two

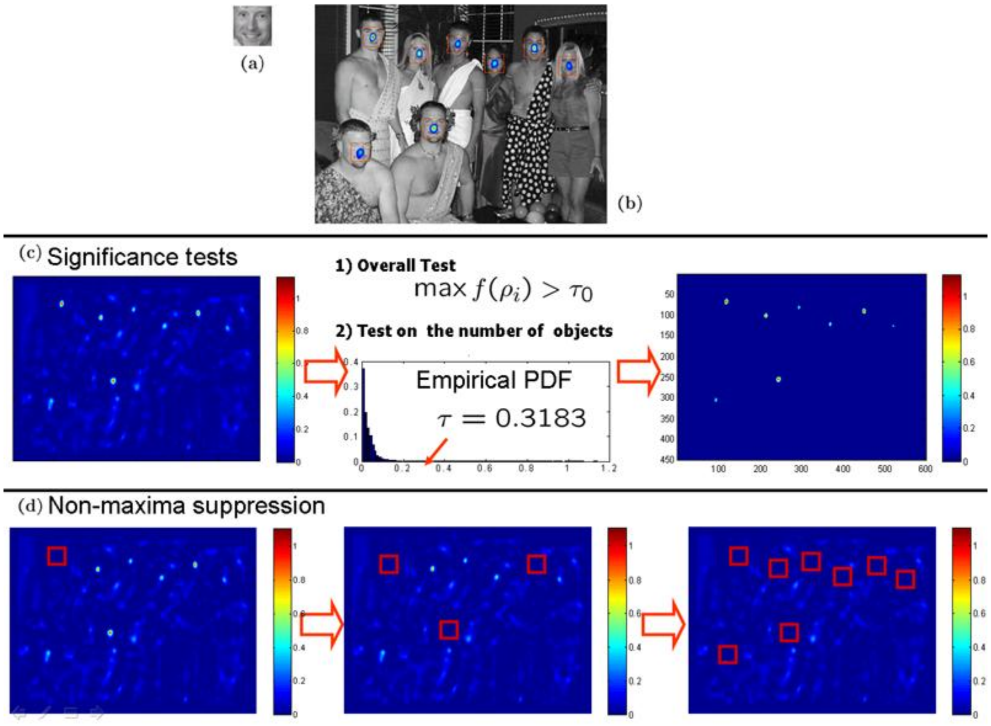


Figure 3.8: (a) Query (b) Target with detection (c) Two significance tests (d) Non-maxima suppression [12].

types of significance tests. The first is an overall test to decide whether there is any sufficiently similar object (action) present in the target at all. If the answer is yes, we would then want to know how many objects (actions) of interest are present in the target and where they are. Therefore, we need two thresholds: an overall threshold τ_o and a threshold τ to detect the possibly multiple similar objects (actions) present in the target.

In a typical scenario, we set the overall threshold τ_o to be, for instance, 0.96 which is about 50% of variance in common (i.e., $\rho^2 = 0.49$). In other words, if the maximal $f(\rho_i)$ is just above 0.96, we decide that there exists at least one object (action) of interest. The next step is to choose τ based on the properties of $f(\rho_i)$. When it comes to choosing the τ, there is need to be more careful. If we have a basic knowledge of the underlying distribution of $f(\rho_i)$, then we can make predictions about how this particular statistic will behave, and thus it is relatively easy to choose a threshold which will indicate whether the pair of features from the two images are sufficiently similar. But, in practice, we do not have a very good way to model the distribution of $f(\rho_i)$. Therefore, instead of assuming a type of underlying distribution, we employ the idea of nonparametric testing. We compute an empirical PDF from all the given samples of $f(\rho_i)$ and we set τ so as to achieve, for instance, a 99 % confidence level in deciding whether the given values are in the extreme (right) tails of the distribution[8]. This approach is based on the assumption that in the target, most of patches (cubes) do not contain the object (actions) of interest, and therefore, the few matches will result in values which are in the tails of the distributions of $f(\rho_i)$. This is also known as controlling the False Discovery Rate (FDR) [132]. We refer the reader to Appendix B for more details.

After the two significance tests with τ_o, τ are performed, we employ the idea of non-maxima suppression [12] for the final detection. We take the region with the highest $f(\rho_i)$ value and eliminate the possibility that any other object (action) is detected within some radius[9] of the center of that region again. This enables us to avoid multiple false detections of nearby objects (actions) already detected. Then we iterate this process until the local maximum value falls below the threshold τ. Fig. **3.8** shows the graphical illustration of significance tests and the non-maxima suppression idea.

[8] Yet another justification for using $f(\rho_i)$ instead of ρ_i is the observation that the empirical PDF of ρ_i is itself heavy-tailed, making the detection of rare events more difficult. The use of $f(\rho_i)$ instead tends to alleviate this problem (see Fig. **3.7**.)

[9] The size of this "exclusion" region will depend on the application at hand and the characteristics of the query.

Algorithm 2 Pseudo-code for the non-parametric object and action detection algorithm

```
$Q$ : Query, $T$ : Target, $\tau_o$ : Overall threshold, $\alpha$ : Confidence level, $P$ : Size of LARK (3-D LARK) window.
Stage1 : Feature representation
1) Construct $\mathbf{K}_Q, \mathbf{K}_T$ (a collection of normalized LARK associated with $Q, T$)
2) Apply PCA to $\mathbf{K}_Q$ and obtain projection space $\mathbf{A}_Q$ from its top $d$ eigenvectors.
3) Project $\mathbf{K}_Q$ and $\mathbf{K}_T$ onto $\mathbf{A}_Q$ to construct $\mathbf{F}_Q$ and $\mathbf{F}_T$.
Stage2 : Compute Matrix Cosine Similarity
for every target patch (cube) $T_i$, where $i \in [0, \cdots, M-1]$ do
    $\rho_i = < \frac{\mathbf{F}_Q}{\|\mathbf{F}_Q\|_F}, \frac{\mathbf{F}_{T_i}}{\|\mathbf{F}_{T_i}\|_F} >_F$ and compute (RM) : $f(\rho_i) = \frac{\rho_i^2}{1-\rho_i^2}$.
end for
Then, find $\max f(\rho_i)$.
Stage3 : Significance tests and Non-maxima suppression
1) If $\max f(\rho_i) > \tau_o$, go to the next test. Otherwise, there is no object of interest in $T$.
2) Threshold RM by $\tau$ which is set to achieve 99 % confidence level ($\alpha = 0.99$) from the empirical PDF of $f(\rho_i)$.
3) Apply non-maxima suppression to RM (RV) until the local maximum value is below $\tau$.
```

To summarize, the overall pseudo-code for the algorithm is given in **Algorithm** 2.

3.3 Experimental Results

3.3.1 Object Detection

In this section, we demonstrate the object detection performance of the proposed method with comprehensive experiments on four datasets; namely, the UIUC car dataset [13], MIT-CMU face dataset [17], a subset of the Bao face dataset[10] and Caltech face dataset[11], and Shechtman's general object dataset [11]. The proposed algorithm provides a series of bounding boxes around objects of interest using the criterion described in [13]. More specifically, if the detected region by the proposed method lies

[10] http://www.facedetection.com/facedetection/datasets.htm

[11] http://www.vision.caltech.edu/html-files/archive.html

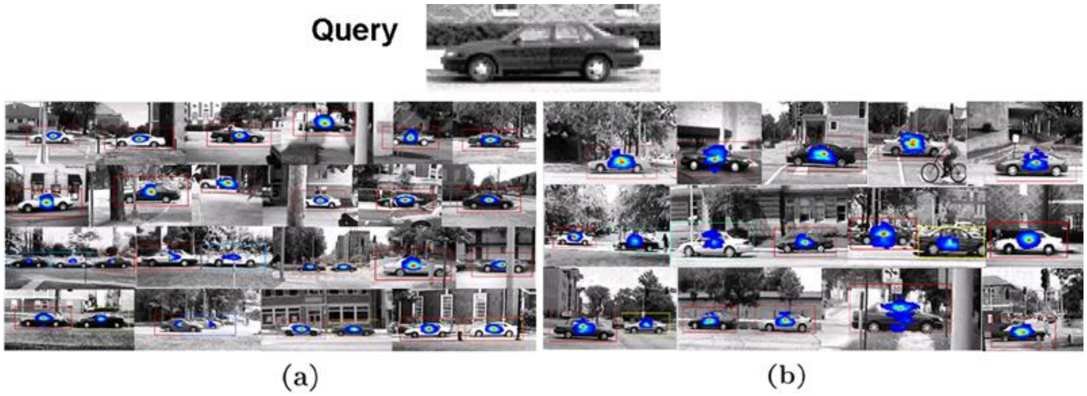


Figure 3.9: (a) Examples of correct detections on the UIUC single-scale car test set [13]. (b) Examples of correct detections on the UIUC multi-scale car test set. Confidence level α was set to 0.99 and RM only above the threshold τ corresponding to α is embedded in test images. Bounding boxes are drawn at the correct locations. In case of a multiple detection, a red bounding box indicates higher resemblance to query than a blue bounding box.

within an ellipse of a certain size centered around the ground truth, we evaluate it as a correct detection. Otherwise, it is counted as a false positive. Eventually, we compute *Precision* and *Recall* defined as

$$\text{Recall} = \frac{TP}{nP}, \qquad \text{Precision} = \frac{TP}{TP+FP}, \tag{3.8}$$

where, TP is the number of true positives, FP is the number of false positives, nP is the total number of positives in dataset, and $1-\text{Precision} = \frac{FP}{TP+FP}$.

Experimental results on each dataset will be presented as recall versus (1-precision) curve and detection equal-error rate[12] in the following sections.

Table 3.1: Detection equal-error rates on the UIUC single-scale car test set [13]

The proposed method w/o PCA	Query 1	Query 2	Query 3	Query 4	Query 5	Agarwal et al. [13] (1)	Wu and Nevatia [14]	Mutch and Lowe [16]
detection rates	79.29 %	**88.12** %	81.11 %	80.41 %	87.11 %	77.08 %	97.5 %	99.94 %
The proposed method	Query 1	Query 2	Query 3	Query 4	Query 5	Agarwal et al. [13] (2)	Kapoor and Winn [15]	Lampert et al. [101]
detection rates	85.26 %	**87.27** %	87.13 %	80.57 %	86.73 %	76.72 %	94.0 %	98.5 %

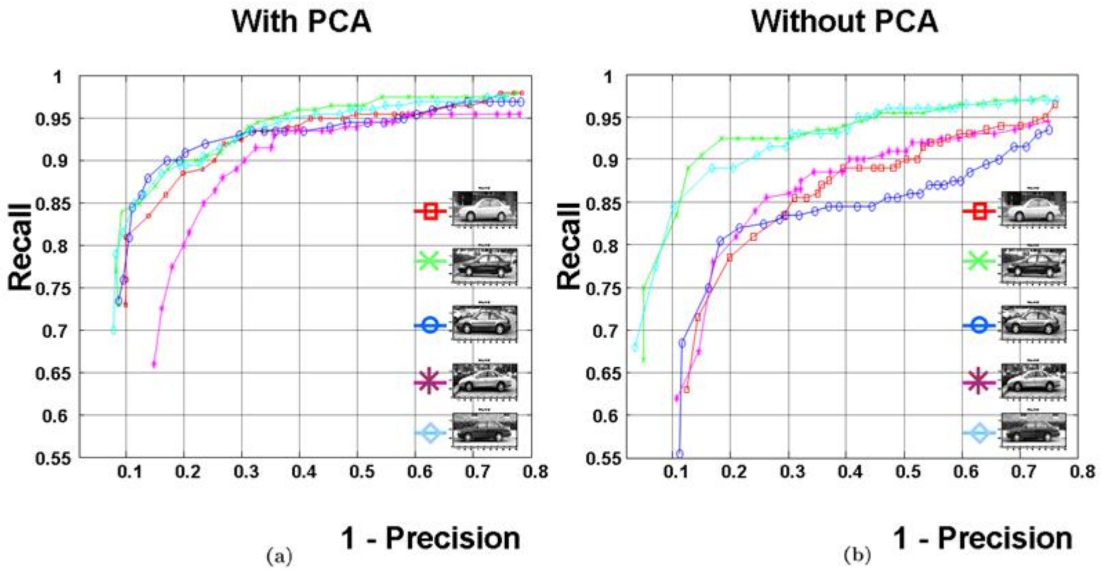


Figure 3.10: (a) Recall versus 1-Precision curves of the proposed method (b) Recall versus 1-Precision curves of the proposed method without PCA on the UIUC single-scale car test set [13] using 5 different query images.

3.3.1.1 Car detection

The UIUC car dataset [13] consists of learning and test sets. The learning set contains 550 positive (car) images and 500 negative (non-car) images. The test set is divided into two parts: 170 gray-scale images containing 200 side views of cars with

[12] Note that detection equal-error rate is a detection (recall) rate when a recall rate is the same as the precision rate.

Table 3.2: Detection equal-error rates on the UIUC multi-scale car test set [13]

The proposed method	Query 1	Query 2	Query 3	Query 4	Query 5	Agarwal et al. [13]	Mutch and Lowe [16]	Kapoor and Winn [15]	Lampert et al. [101]
Detection rates	75.47 %	**77.66** %	70.21 %	75.00 %	74.22 %	43.77 ~ 44.00 %	90.6 %	93.5 %	98.6 %

size of 100×40, and 108 gray-scale images containing 139 cars at various sizes with a ratio between the largest and smallest car of about 2.5. Since our method is training-free, we use only one query image at a time from the 550 positive examples.

Single-scale test set We compute LARK of size 9×9 as descriptors, as a consequence, every pixel in Q and T yields an 81-dimensional local descriptor $\mathbf{K}_Q$ and $\mathbf{K}_T$ respectively. The smoothing parameter h for computing LARKs was set to 2.1. We end up with $\mathbf{F}_Q, \mathbf{F}_T$ by reducing dimensionality from 81 to $d = 4$ and then, we obtain RM by computing the MCS measure between $\mathbf{F}_Q, \mathbf{F}_{T_i}$. The threshold τ for each test example was determined by the confidence level $\alpha = 0.99$. Fig. **3.9** (a) shows the output of the proposed method on single-scale test images. We conducted an experiment by computing RM without performing PCA in order to verify that the use of dimensionality reduction step (PCA) plays an important role in extracting only salient features and improving the performance. We also repeated these experiments by changing the query image and computing precision and recall. In Fig. **3.10**, recall-precision curves represent a performance comparison between the proposed method and the proposed method without PCA using 5 different query images. We can clearly see that the performance of our system is not terribly affected by a choice of the query images, but is quite consistent. Furthermore, PCA consistently contributes to a performance improvement. The detection equal-error rates comparison is provided in Table **3.1** as

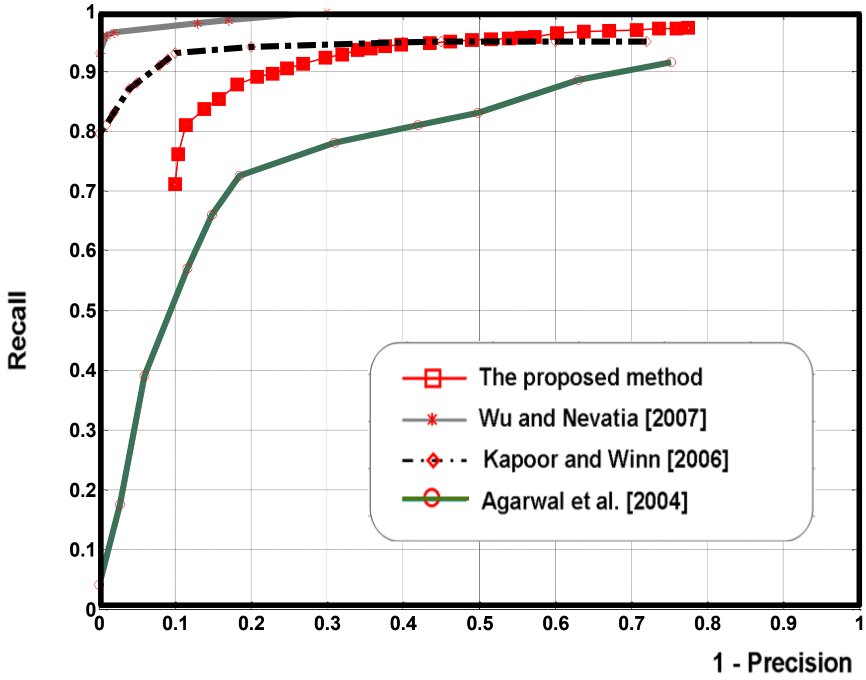
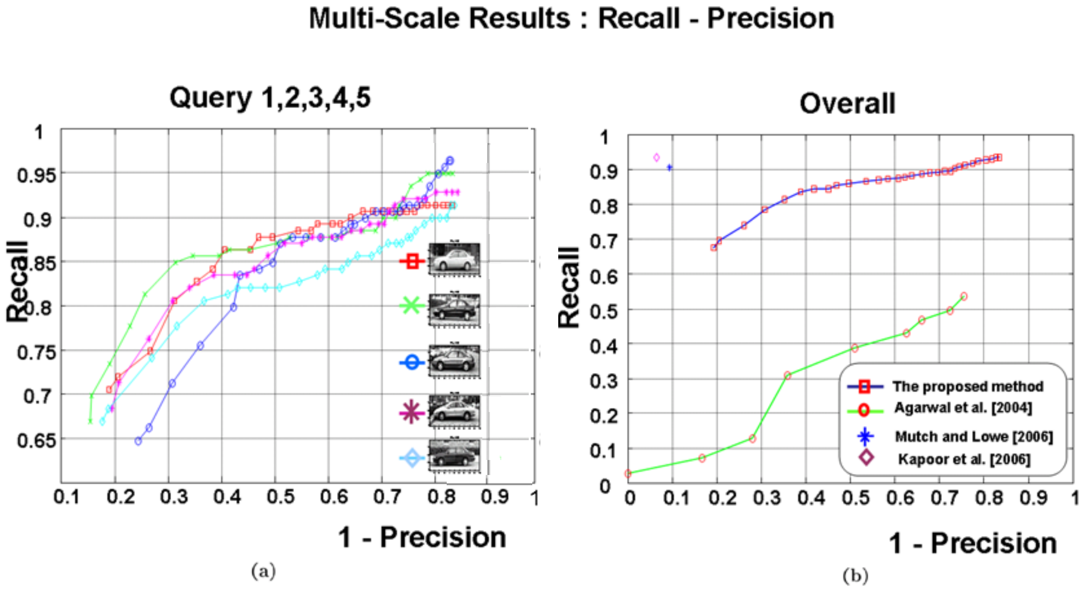


Figure 3.11: Comparison of Recall versus 1-Precision curves between the proposed method and state-of-the-art methods [13, 14, 15] on the UIUC single-scale test set [13].

well.

To show the overall performance of the purposed method on five different query images, we summed up TP and FP over the entire experiment, then computed recall and precision at various steps of the threshold value τ according to the confidence level α. Note that, to the best of our knowledge, there are no other training-free methods evaluated on the UIUC dataset [13], and thus, comparison is only made with state-of-the-art training-based methods. The proposed method which is training-free performs favorably against state-of-the-art training-based methods [13, 14, 15] which use extensive training as shown in Fig. **3.11**.



(a)

(b)

Figure 3.12: (a) Recall versus 1-Precisions curve using 5 different query images (b) Comparison of Recall versus 1-Precision curves between the proposed method and state-of-the art methods [15, 16, 13] on the UIUC multi-scale test set [13].

Multi-scale test set We construct a multi-scale pyramid of the target image T : 5 scales with scale factors 0.4, 0.6, 0.8, 1, and 1.2 as explained in Section 3A. More specifically, we reduce the target image size by steps of 20% up to 40% of the original size and upscale the target image by 20% so that we can deal with both cases of either the size of objects in the target images being bigger or smaller than the query. The rest of the process is similar to the single-scale case. Fig. **3.12** (b) shows examples of correct detections using τ corresponding to $\alpha = 0.99$.

The overall performance improvement of the proposed method (using 5 different query images) over Agarwal et al. [13] is even greater (over 30%) on the multi-scale test set as shown in Table **3.2** and Fig. **3.12**. As for the interpretation of the performance on the UIUC car dataset (both single-scale and multi-scale cases), our methods show performance that is not far from the state-of-the-art training-based methods, ex-

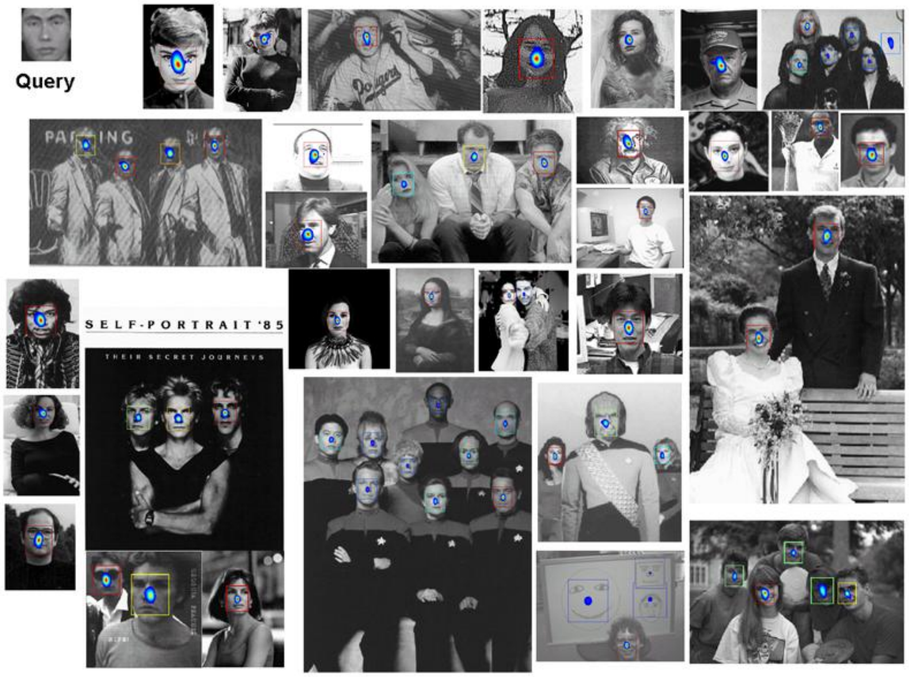


Figure 3.13: Detection Results on the MIT-CMU multi-scale test set [17]. α was set to 0.99. Hand-drawn faces on the white board were also detected using a real face query image.

cept that it requires *no training* at all.

3.3.1.2 Face detection

We showed the performance of the proposed method in the presence of moderate scale variation (a ratio between the largest and smallest object of about 2.5) in the previous section. In this section, we further evaluate our method on more general scenario where the scale ratio between the largest and smallest is over 10 and large rotations of objects may exist. Therefore, a test set is chosen from a subset of the MIT-

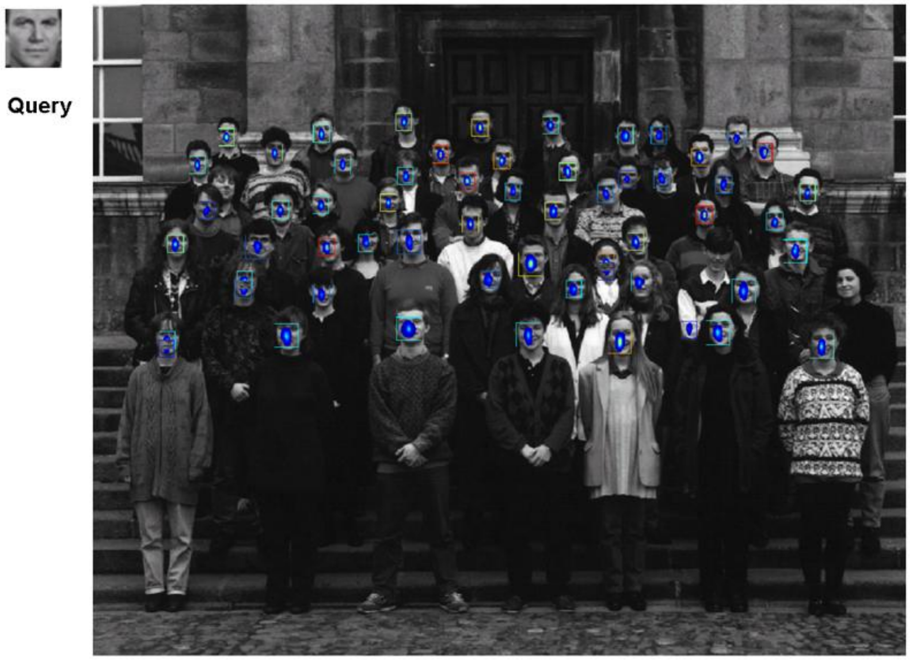


Figure 3.14: Detection Results on the MIT-CMU multi-scale test set [17]. α was set to 0.99. Among 57 faces present, we detected 54 faces at a correct location with 4 false alarms.

CMU face dataset [17]. The test set is composed of 43 gray-scale images[13] containing 149 frontal faces at various sizes and 20 gray-scale images [14] containing 30 faces with various rotations. A query face image of size 35×36 was employed as shown in Fig.

[13] The 43 images (from http://vasc.ri.cmu.edu/idb/html/face/index.html) are listed as follows: aerosmith-double.gif, blues-double.gif, original2.gif, audrey1.gif, audrey2.gif, baseball.gif, cfb.gif, cnn1714.gif, cnn2020.gif, cnn2600.gif, crimson.gif, ew-courtney-david.gif, gpripe.gif, hendrix2.gif, henry.gif, john.coltrane.gif, kaari1.gif, kaari2.gif, kaari-stef.gif, knex0.gif, lacrosse.gif, married.gif, police.gif, sarah4.gif, sarah_live_2.gif, tammy.gif, tori-crucify.gif, tori-entweekly.gif, tp.gif, voyager2.gif, class57.gif, trek-trio.gif, albert.gif, madaboutyou.gif, frisbee.gif, me.gif, speed.gif, ysato.gif, wxm.gif, torrance.gif, mona-lisa.gif, karen-and-rob.gif, and Germany.gif.

[14] The 20 images (from http://vasc.ri.cmu.edu/idb/html/face/index.html) are listed as follows: 3.gif, 217.gif, 221.gif, af2206b.gif, am4945a.gif, am5528a.gif, am6227a.gif, bm5205a.gif, bm6290a.gif, boerli01.gif, cast1.gif, dole2.gif, jprc.gif, pict_6.gif, pict_28.gif, sbCelSte.gif, siggi.gif, tf5189a.gif, tf5581a.gif, and tm6109a.gif

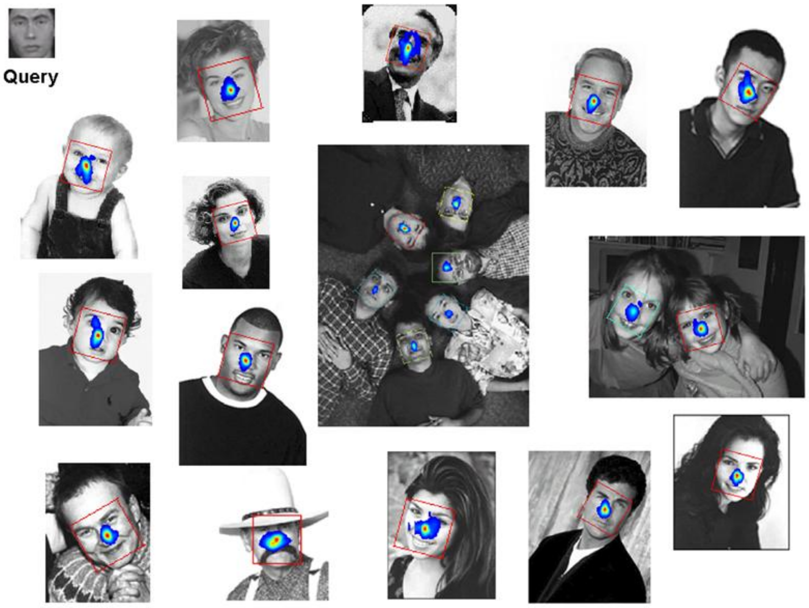


Figure 3.15: Detection Results on the MIT-CMU multi-rotation test set [17]. α was set to 0.99.

3.13, and images for a rotation experiment were resized so that faces are about the same size as the query face. Such parameters as the smoothing parameter (h), LARK size (P), confidence level (α) remain same as the ones used in the UIUC car test sets. However, we increased scale steps for the multi-scale pyramid up to 29, and rotation steps were set to 24 (i.e., rotate the query image by 15 degrees) to achieve an accurate rotation estimation (see Appendix 3A for more detail.) Fig. **3.13**, Fig. **3.14**, and Fig. **3.15** show that the proposed method is capable of detecting and localizing faces at distinct scale and rotation angle even in the presence of large variations in scale and rotation. We repeated the experiment by changing the query image. Fig. **3.16** shows Recall versus 1-Precision curves and (for the sake of completeness) corresponding receiver operating

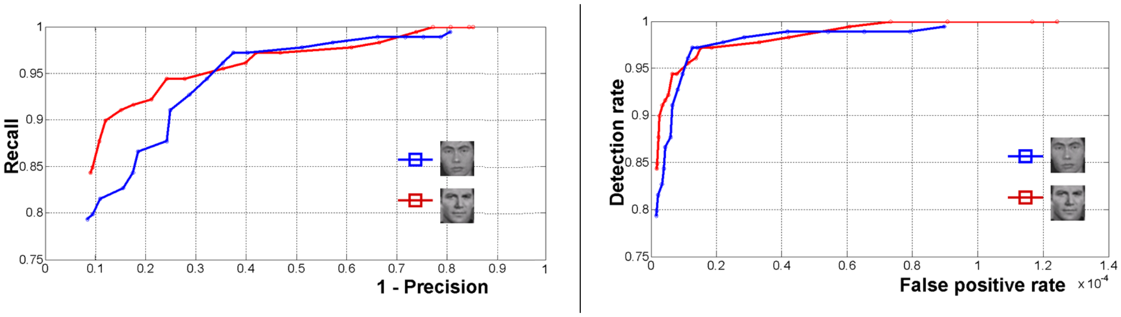


Figure 3.16: Left: Precision-Recall curves, Right: ROC curves on the MIT-CMU test set [17] using 2 different query images. Note that detection rate P_d and false alarm rate P_f are defined as $\frac{TP}{nP}(= recall)$ and $\frac{FP}{FP+TN}$ respectively, where TN is the number of true negatives.

characteristic (ROC) curves with respect to two different queries. Note that, in the ROC curve, detection rate P_d and false alarm rate P_f are defined as $\frac{TP}{nP}(= recall)$ and $\frac{FP}{FP+TN}$ respectively, where TN is the number of true negatives. As seen in Fig. **3.16**, the performance of our method on this test set is consistent with the results in the UIUC car test sets. More specifically, the performance of the proposed method is little affected by the choice of similar query images and is quite stable.

We also compare LARK with other descriptors explained in Chapter 1 when used within this framework. A test set is chosen from the Bao face dataset[15], Caltech face dataset[16], and group photos from Google search. The test set is composed of 72 images containing 266 frontal faces at various sizes. As shown in Fig. **3.17**, in this experiment, we only use a single query which is a female face image of size 60 × 60 while the test set contains faces with occlusion, different gender, ethnic group, in different lighting condition and faces in different context such as hand drawn faces,

[15] `http://www.facedetection.com/facedetection/datasets.htm`

[16] `http://www.vision.caltech.edu/html-files/archive.html`

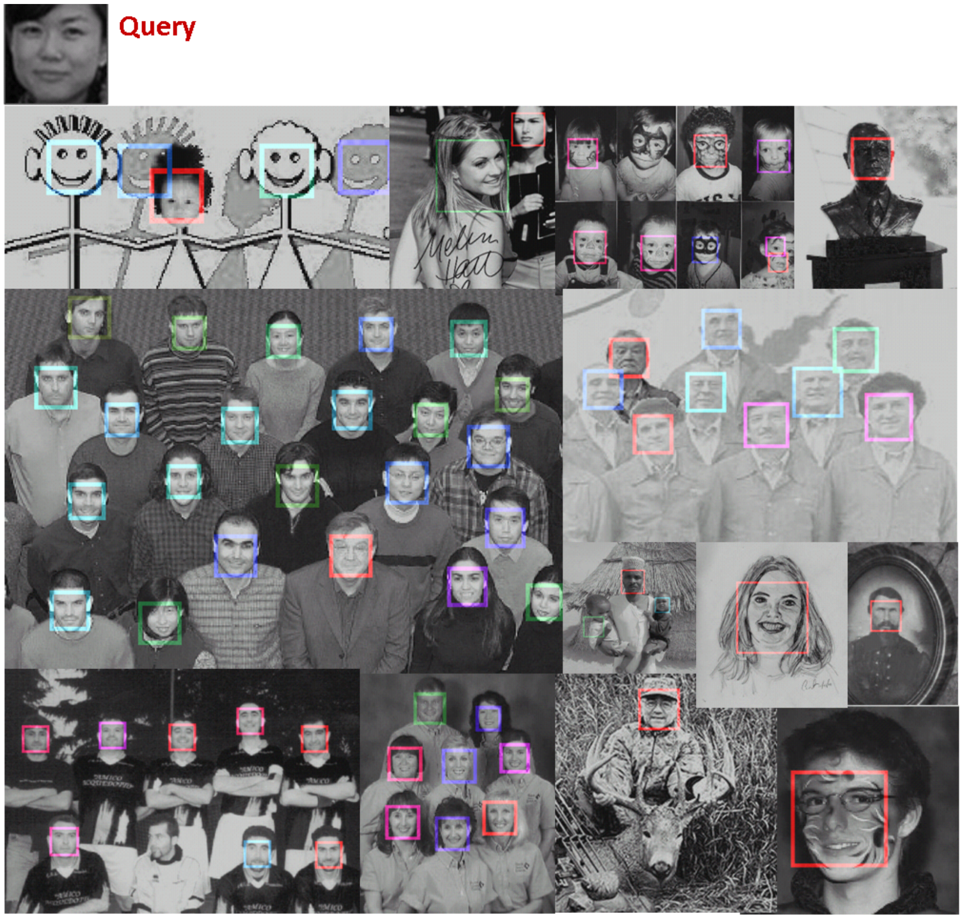


Figure 3.17: Object detection results by using LARK. Note that a single query of female face led to high detection performance on the challenging dataset.

sculpture faces. We compute LARK, NLM, BL [17], SSIM[18], SIFT[19], HOG[20] densely from the query and the target. Then we plug in these descriptors into the object detection framework. To compare detection performance of these six different descriptors, we show recall versus (1-precision) curves. As seen in Fig. **3.18**, the proposed LARK descriptors achieves 0.85 recall rate at 0.9 precision rate and outperforms all the other

[17] For NLM and BL, we used our own Matlab implementation

[18] `http://www.robots.ox.ac.uk/~vgg/software/SelfSimilarity`

[19] `http://people.csail.mit.edu/ceilu/ECCV2008`

[20] `http://www.robots.ox.ac.uk/~vgg/research/caltech/phog.html`

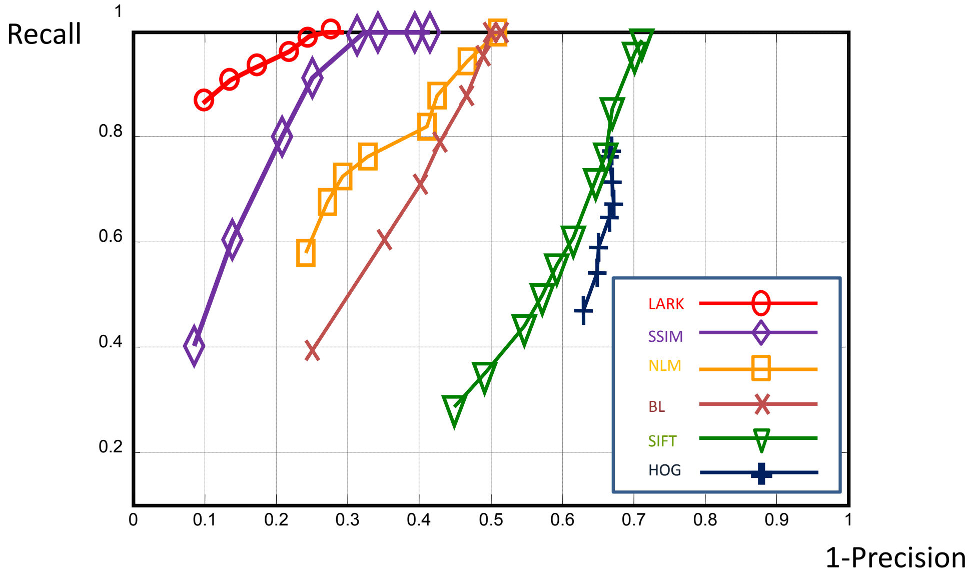


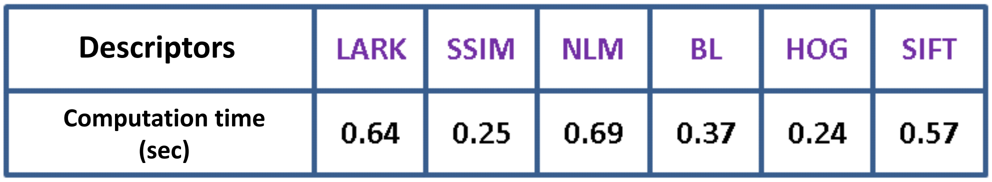
Figure 3.18: Recall vs. 1-Precision curves of descriptors on the challenging dataset shown in Fig. **3.17**. LARK outperforms other descriptors.

descriptors. Note that a single query of female face led to high detection performance on this challenging dataset.

Fig. **3.19** illustrates the computational cost of descriptors. LARK descriptors have a higher computational cost than other descriptors. Despite somewhat higher computational cost we pay, our process for computing LARK descriptors is stable, yet showing rather high specificity at the same time, resulting in overall very good performance. In Chapter 4, we illustrate how to speed up the computation of LARK for real-time object detection while maintaining detection accuracy.

3.3.1.3 General object detection

We have shown the performance of the proposed method on data sets composed of gray-scale images which contain specific objects such as car and face. In this



Descriptors	LARK	SSIM	NLM	BL	HOG	SIFT
Computation time (sec)	0.64	0.25	0.69	0.37	0.24	0.57

Figure 3.19: Comparison of computational cost among descriptors. Due to the robust estimation of **C**, LARK descriptors have a higher computational cost than other descriptors. The size of query is 60 × 60 pixels. We introduce how to speed up the computation of LARK in Chapter 4.

section, we have applied our method to a more difficult scenario where general real-world images containing flowers, hearts, and human poses are considered. Furthermore, *rough hand-drawn sketches* are used as a query instead of real images. Shechtman et al.'s general object dataset [11] consists of many challenging pairs of color images (60 pairs with queries such as flowers, hearts, peace symbols, face, and human poses; see Fig. **3.3**). In order to justify the usefulness of the MCS measure for this dataset and to further verify the advantage of the "Canonicl Cosine Simialrity" (CCS) defined in Appendix 3A over the MCS measure, we begin with evaluating the proposed method on the luminance channel only. In Fig. **3.20**, some examples of RM are shown. Fig. **3.21** and Fig. **3.22** show that the proposed method is able to detect and localize reliably.

We further justify the use of LARKs on this dataset by comparing the performance against state-of-the-art local descriptors evaluated in [19] as similarity done in [11]. We densely computed such local descriptors as *gradient location-orientation histogram* (GLOH) [19], *Shape Context* [20], and SIFT [18] using the implementation in [19]. By replacing LARKs with these descriptors, but keeping the rest of the steps the same, we repeated the experiment on this test set. The Precision-Recall curve in Fig. **3.23** verifies that our LARKs have more discriminative power than other local descriptors.

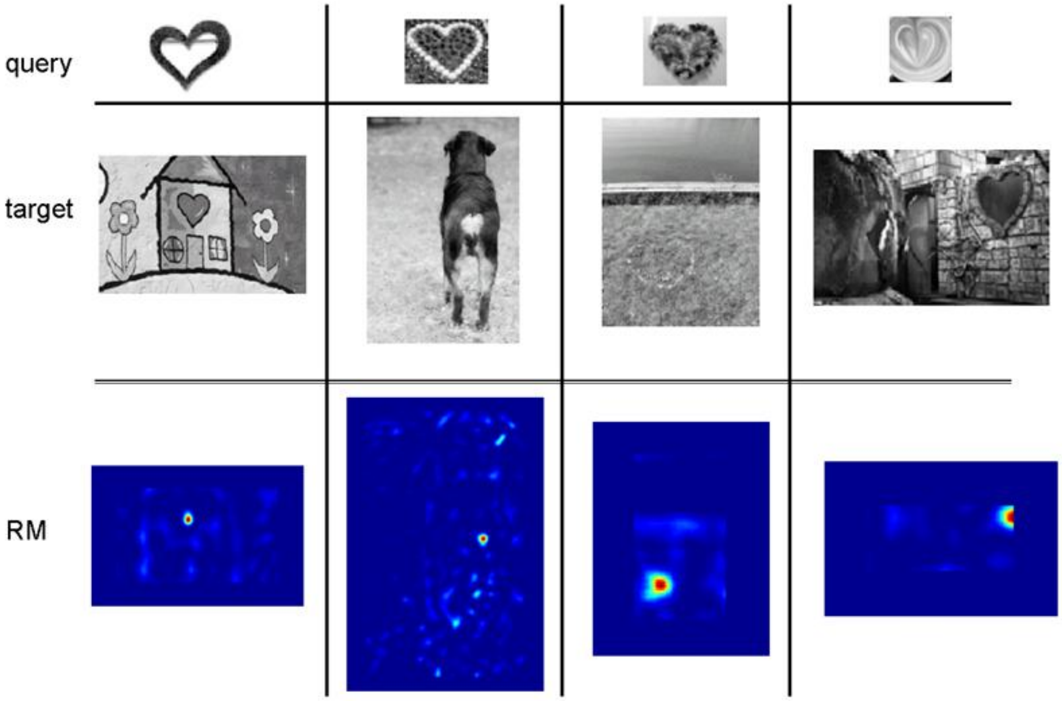


Figure 3.20: Some examples of detection results with RMs in Shechtman's object test set [11]. RMs are shown in bottom row.

The proposed method is also evaluated on full CIE L*a*b* data. If we look at recall rates in the range of $0 \leq$ (1-precision) ≤ 0.1 in Fig. **3.23**, we can see that full CIE L*a*b* data provide more information, and thus CCS outperforms the MCS measure as also observed in [11]. Consistent with these results, Shechtman and Irani [11] also showed that their local self-similarity descriptor clearly outperformed other state-of-the-art descriptors in their ensemble matching framework. However, the performance figures they provide are rather incomplete. Namely, they mentioned 86% detection rate without specifying either any precision rates or false alarm rates. Therefore, we claim that our proposed method is more general and practical than the training-free detection method in [11] .

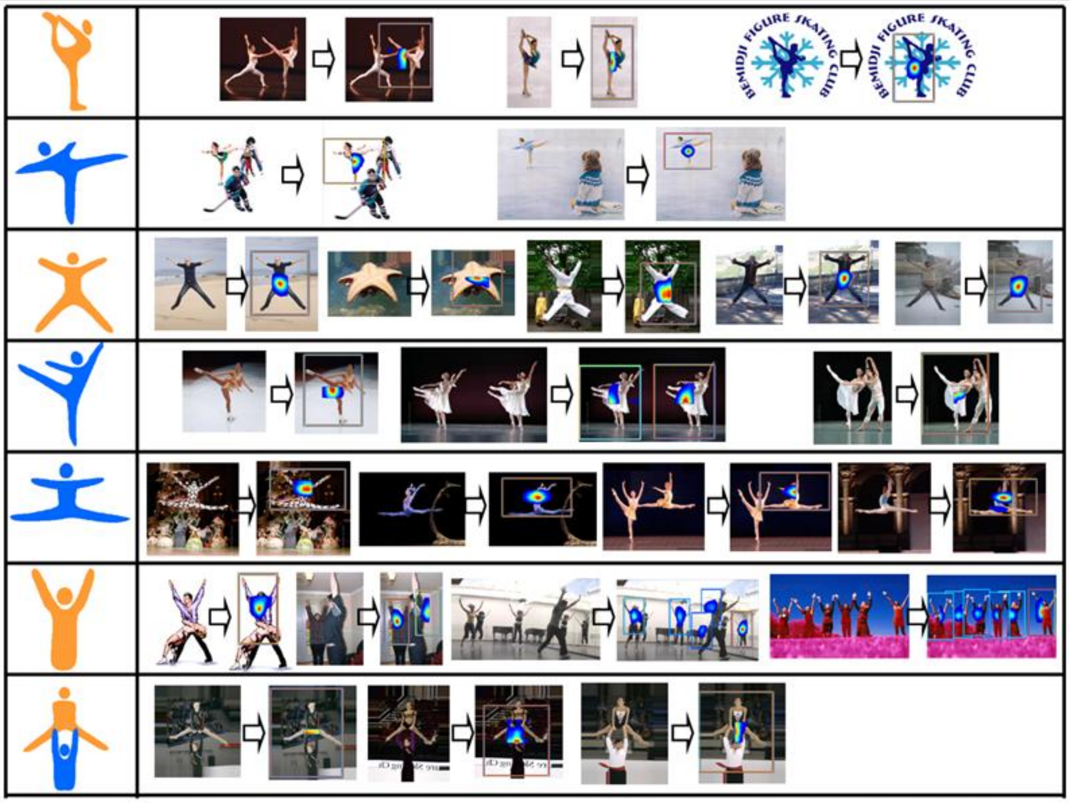


Figure 3.21: Left: hand-drawn sketch query (human poses) Right: targets and examples of correction detections/ localizations in Shechtman's object test set [11]. α was set to 0.98.

Discussion The CCS has shown to be more effective than MCS when vector-valued images are available though this requires further study. Challenging sets of real-world object experiments have demonstrated that the proposed approach achieves a high detection accuracy of objects of interest even in completely different context and under different imaging conditions. Unlike other state-of-the-art learning-based detection methods, the proposed framework operates using a *single* example of an image of interest to find similar matches; does not require any prior knowledge (learning) about objects being sought; and does not require any segmentation or pre-processing step of the target image. Since the proposed method is designed with detection accuracy

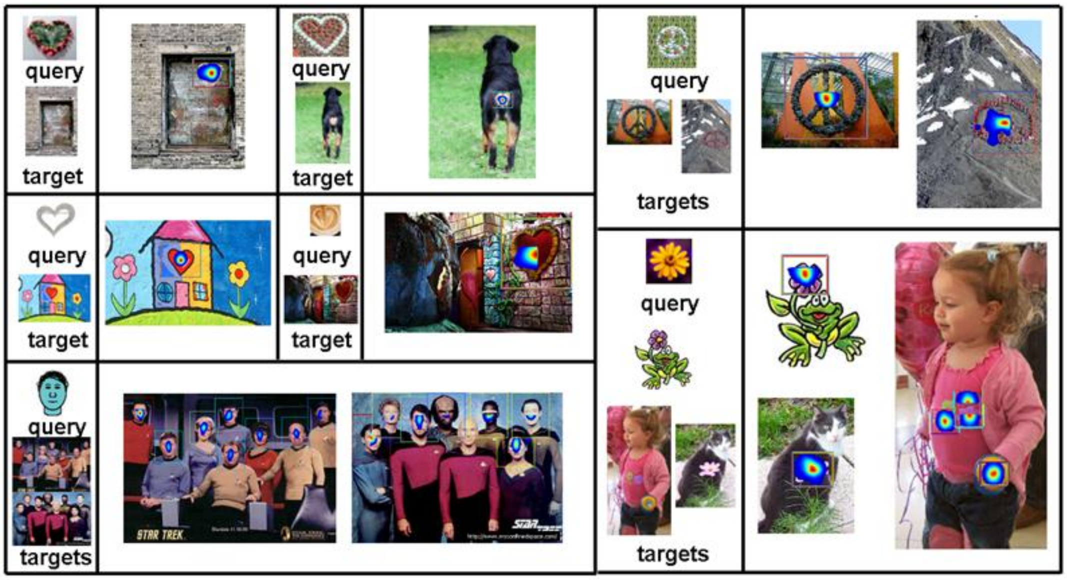


Figure 3.22: Query: hearts, hand-drawn face, peace symbol and flower. Some targets and examples of correction detections/ localizations in Shechtman's object test set [11] are shown. Some false positives appeared in a girl's T-shirt and candle. α was set to 0.98.

as a high priority, extension of the method to a large-scale dataset requires a significant improvement of the computational complexity of the proposed method. Toward this end, we could benefit from an efficient searching method (coarse-to-fine search)[21] and/or a fast nearest neighbor search method (e.g., vantage point tree [133]). Recently, large database-driven approaches [38, 39] have shown potential for nonparametric detection. For instance, [39] showed that with a database of 80 million images, even simple matching based on the sum of squared differences (SSD) can provide semantically meaningful classification performance for 32×32 images. Thus, we could use a fast indexing techniques such as spatial pyramid matching (SPM) [31] or GIST matching [82] in order to reduce the search space and rapidly, and accurately, limit the number

[21] In Chapter 4, we realize a real-time object and action detection system based on coarse-to-fine pyramid search in conjunction with hierarchical clustering.

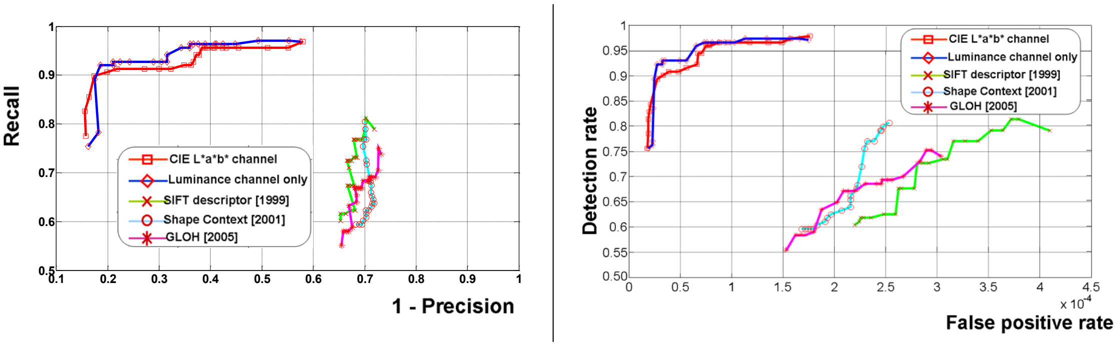


Figure 3.23: Left: Comparison of Recall versus 1-Precision curves between luminance channel only and CIE L*a*b* channel on the Shechtman's test set [11]. It is clearly shown that such descriptors as SIFT [18], GLOH [19], Shape Context [20] turn out to be inferior to LARKs in terms of discriminative power. Right: Comparison of ROC curves. Note that detection rate P_d and false alarm rate P_f are defined as $\frac{TP}{nP}(= recall)$ and $\frac{FP}{FP+TN}$ respectively, where TN is the number of true negatives.

of candidate images. Subsequently, we can apply the proposed method for more accurate detection. Additionally, for the proposed method to be feasible for scalable image retrieval, we may adopt the idea of encoding the features as proposed in [134, 135]. In Chapter 5, we will show that the proposed method can be also applied to other challenging problems such as face verification and automatic change detection in medical imaging applications.

In the following section, we demonstrate the action detection performance of the proposed method with comprehensive experiments on four action datasets: namely, the general action dataset [21], the drinking dataset, [23], the Weizmann action dataset [6], and the KTH action dataset [7]. The general action dataset and the drinking dataset are used to evaluate detection performance of the proposed method, while the Weizmann action dataset and the KTH action dataset are employed for action categorization. Comparison is made with state-of-the-art methods that have reported their results on these datasets.

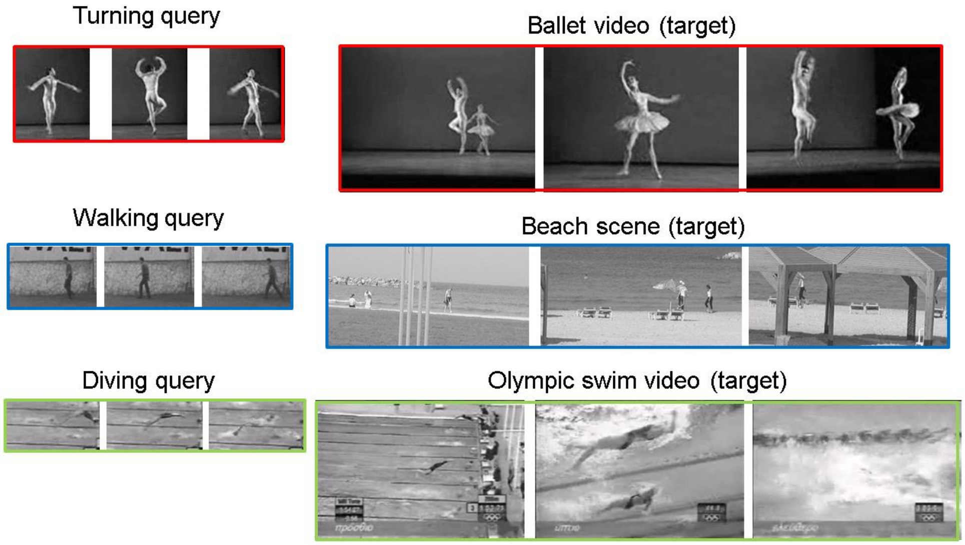


Figure 3.24: Examples of general action dataset [21]: 1) a turning query and ballet video, 2) a walking query and beach scene video, and 3) a diving query and Olympic swim relay video.

3.3.2 Action Detection

In this section, we show several experimental results on searching with a short query video clip against a (typically longer and larger) target video. Our method detects the presence and location of actions similar to the given query and provides a series of bounding cubes with resemblance volume embedded around detected actions. Note again that no background/foreground segmentation or explicit motion estimation are required in the proposed method. Our proposed method can also handle modest variations in rotation (up to ±15 degree), and spatial and temporal scale change (up to ±20%). For larger variations in scale, we use a multi-scale approach as similarly done in Section 3.3.1.1 and show below that this results in improvement over the single-scale implementation.

Given Q and T, we spatially blur and downsample both Q and T by a factor

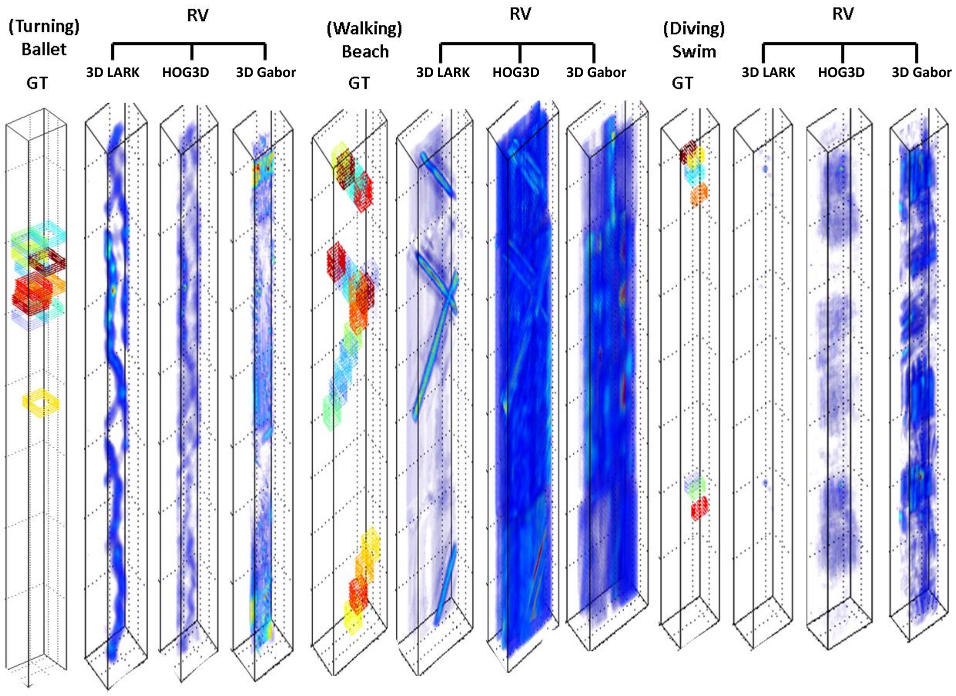


Figure 3.25: Comparison of resemblance volumes (RV) among 3-D LARK, HOG3D, and 3-D Gabor for three pairs of videos (Ballet with a turning query, Beach with a walking query, and Swim with a diving query). HOG3D was computed densely for a fair comparison. Note that colors in the ground truth volume are used to distinguish individual actions from each other. This figure is better viewed in color.

of 3 in order to reduce the time-complexity. We then compute 3-D LARK of size 3×3 (space) $\times 7$ (time) as descriptors so that every space-time location in Q and T yields a 63-dimensional local descriptor $\mathbf{K}_Q$ and $\mathbf{K}_T$ respectively. The reason why we choose a larger time axis size than space axis of the cube is that we focus on detecting similar actions regardless of different appearances. Thus we give a higher priority to temporal evolution information than spatial appearance. We end up with $\mathbf{F}_Q$ and $\mathbf{F}_T$ by further reducing the dimension of descriptors[22] to d using PCA. Finally, we obtain a resemblance volume (RV) by computing the MCS measure between $\mathbf{F}_Q$ and $\mathbf{F}_T$. After

[22] Note that $d = 4$ for the walking query whereas $d = 7$ for the ballet turning and diving queries.

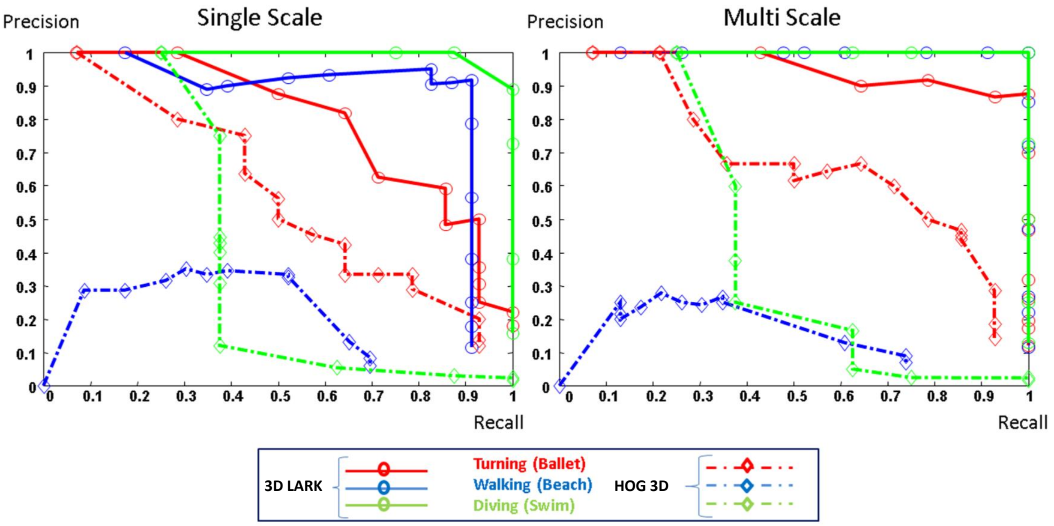


Figure 3.26: Left: Comparison of Precision-Recall curves between 3-D LARK and HOG3D for three different actions (walking, ballet turning, and diving) in single-scale implementation Right: multi-scale comparison. Note that other state-of-the art action detection methods in [21, 11, 22] did not provide any quantitative performance on these examples. This figure is better viewed in color.

significance testing by controlling the FDR with a specified α value[23] and non-maxima suppression explained in Section B, the proposed method localizes actions of interest[24].

The General Action Dataset [11] This dataset contains three pairs of action query and target videos. Note that the in all cases, the query video is not from the target video sequence.

a. The query video contains a single turn of a male dancer (13 frames of 90×110 pixels) while the target video (766 frames of 144×192 pixels) includes ballet actions

[23] In our experiments, $\alpha = 0.01$ works well.

[24] The localization is considered to be correct when detected region is 50% overlapped with the ground truth.

from a male and a female dancer.

b. The query video contains a very short walking action moving to the left (14 frames of 60×70 pixels) with a stationary stone wall in the background while the target video has walking people in a beach scene (456 frames of 180×360 pixels) with crashing waves in the background.

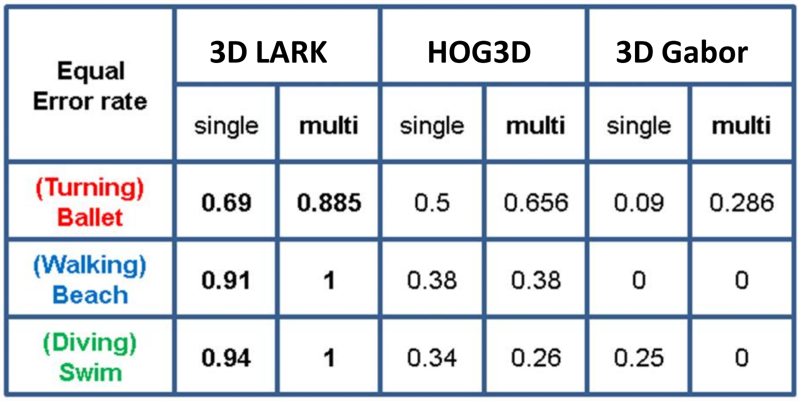
c. The query video contains a swimmer's dive into a pool (16 frames of 70×140 pixels) while the target is an Olympic relay-match video (757 frames of 240×360 pixels) which was severely MPEG compressed.

As we alluded to in Section 2.1, we compare our 3-D LARK with 3-D Gabor filter response [136] and HOG3D [65] both qualitatively and quantitatively[25]. Fig. **3.25** shows a comparison of resemblance volumes with 3-D LARK, HOG3D, and 3-D Gabor filter for three datasets. Note that we plugged in HOG3D and 3-D Gabor instead of 3-D LARK while the rest of the process in the proposed action detection framework remains exactly same. Red value in RVs signifies higher resemblance to the given query

[25] We set parameters for HOG3D and 3-D Gabor filters as follows:

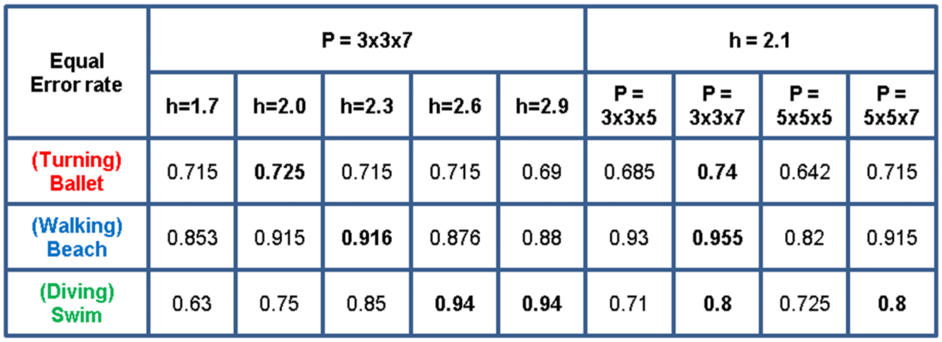
1 HOG3D [65]: A 3-D patch of interest is divided into 3x3x2 space-time cells. The corresponding descriptor concatenates oriented gradient (10 orientations) histograms of all cells and is then normalized. With dense sampling (x_1x_2-stride: 6 pixels apart and t-stride:1 pixel apart), the resulting descriptors have 180 dimensions at every sampled position. We use the executable binary from the authors' website (downloadable from `http://lear.inrialpes.fr/people/klaser/software_3d_video_descriptor`. We set the parameters for this method to achieve its best performance. These parameters were not the same as those setting recommended at the website. This is because the recommended settings were not best suited for the general action dataset.)

2 3-D Gabor [136]: We used 16 of 3-D Gabor filter responses (0, $\pi/4$, $\pi/2$, $3\pi/4$: preferred direction of motion) and (1,2,3,4: preferred speed of the filter (in pixels per frame)). We use a matlab code from the website (downloadable from `http://www.cs.rug.nl/~imaging/spatiotemporal_Gabor_function/GaborApp.html`.)



Equal Error rate	3D LARK		HOG3D		3D Gabor	
	single	multi	single	multi	single	multi
(Turning) Ballet	**0.69**	**0.885**	0.5	0.656	0.09	0.286
(Walking) Beach	**0.91**	**1**	0.38	0.38	0	0
(Diving) Swim	**0.94**	**1**	0.34	0.26	0.25	0

Figure 3.27: Comparison of equal error rates between 3-D LARK, HOG3D, and 3-D Gabor filter for three different actions (walking, ballet turning, and diving).



Equal Error rate	P = 3x3x7					h = 2.1			
	h=1.7	h=2.0	h=2.3	h=2.6	h=2.9	P = 3x3x5	P = 3x3x7	P = 5x5x5	P = 5x5x7
(Turning) Ballet	0.715	**0.725**	0.715	0.715	0.69	0.685	**0.74**	0.642	0.715
(Walking) Beach	0.853	0.915	**0.916**	0.876	0.88	0.93	**0.955**	0.82	0.915
(Diving) Swim	0.63	0.75	0.85	**0.94**	**0.94**	0.71	**0.8**	0.725	**0.8**

Figure 3.28: Equal error rates with respect to different parameter settings on three datasets where equal error rate means a recall rate when a recall rate is the same as the precision rate.

actions while blue means lower resemblance. 3-D LARKs provide the most consistent results with the ground truth. We observe that RVs with 3-D LARKs reveal most relevant actions with a few false positives whereas HOG3D results in many false positives and 3-D Gabor filter misses most actions. Actions in target videos vary in scale. This can be better dealt with multi-scale approach as described below.

Multiscale Action Detection We construct a multi-scale pyramid of the target feature volume $\mathbf{F}_T$. We resize the target feature volume size by steps of 10 %, so that a relatively fine quantization of spatial scales are taken into account. By using 5 scale factors from 0.9 ~ 1.3, we obtain five resemblance volumes. These resemblance volumes represent the likelihood functions $p(f(\rho_i)|S_i)$ where S_i is the scale at $\mathbf{x}_i$. However the sizes of the respective resemblance volumes are naturally different. Therefore, we simply rescale all the resemblance volumes by voxel replication so that they match the dimensions of the original target volume. Next, the maximum likelihood estimate of the scale at each position is arrived at by comparing the rescaled resemblance volumes as follows[26]:

$$\hat{S}_i = \arg\max_{S_i} p(\underline{\mathrm{RV}}|S_i). \tag{3.9}$$

Action detection methods [21, 11, 22] which also tested on this dataset only presented qualitative results with either empirically chosen threshold values or no description about how the threshold values are determined. On the other hand, the threshold values are automatically chosen in our algorithm by controlling the FDR with respect to the specified α (see Appendix B). Unlike [21, 11, 22], we provide the precision-recall curves in Fig. **3.26** for quantitative evaluation. For these experiments, we used the entire frames while [21, 11, 22] used a part of video frames. The detection result of the proposed method on this video outperforms those in [21, 22] and compares favorably to that in [11] in terms of visual detection accuracy. As shown in Figs. **3.26**, **3.27** and expected from qualitative comparison in Fig. **3.25**, 3-D LARK clearly outperforms HOG3D and 3-D Gabor.

[26] By $\underline{\mathrm{RV}}$ we mean a collection of RV indexed by i at each position.

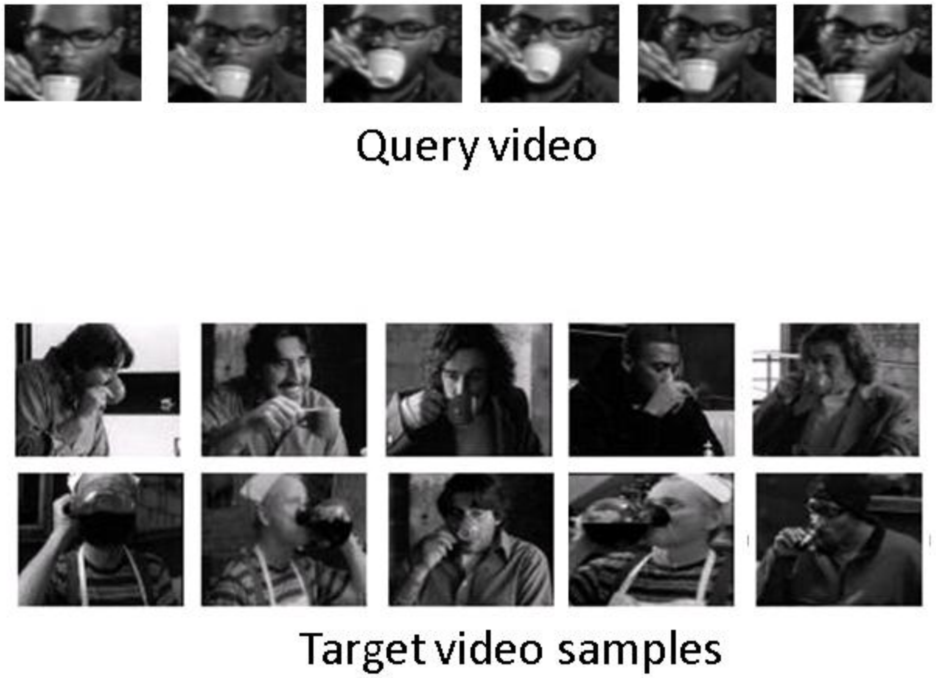


Figure 3.29: The drinking dataset [23]: Top: a query video chosen from the episode “No problem". Bottom: Some target video samples from the episode “Cousin?" and “Delirium".

Effect of Parameters We examined how the performance of the proposed method is affected by the choice of parameters P (the size of 3-D LARK) and h (the smoothing parameter). Fig. **3.28** illustrates equal error rates for 3-D LARKs in single-scale implementation. As shown in Fig. **3.28**, the overall performance of the proposed method changes gracefully with the particular choice of parameter h and P. It appears that best performance can be achieved with the fixed choice of $P = 3 \times 3 \times 7$ and $h = 2.3$ across three video dataset.

The Drinking Action Dataset [23] In this section, we further evaluate our method on more challenging scenarios such as real movie scenes. The drinking action dataset comprises a total of 36,000 frames from two episodes of the movie “Coffee and Cigarette".

The dataset includes 37 drinking actions from the episodes “Cousins?" and “Delirium". Fig. **3.29** illustrates how drinking actions in target video samples largely vary in scales and view-points as well as the background clutter. Furthermore, there are abrupt scene changes, and the size and appearance of cups also vary. We chose one drinking action (55 frames of 107×101 pixels) as a query (see Fig. **3.29**) from the episode called “No problem". Thus, there is no overlap between the query and the target videos. We take the multiscale approach in temporal axis as well as in spatial axis because temporal extents of drinking actions in the test set vary from 30 to 200 frames with the mean length of 70 frames. More specifically, we used 9 spatial scales from $0.7 \sim 1.5$ and 6 temporal scales from $0.8 \sim 1.3$. As explained in Appendix 3.3.2, we take a maximum value across all scales at each voxel and end up with one RV. In order to deal with variations in view points, we used mirror-reflected version of the query as well. By voting the higher score among values from two RVs at every space-time location, we arrive at one RV which includes correct locations of drinking action. The performance of our method on this testset in comparison to Laptev’s methods [23] is illustrated in Fig. **3.30** in terms of precision-recall curves and average precision (AP) values. Note that Laptev 1, 2, and 3 are based on discrete AdaBoost using 106 positive examples for training. As discussed in [23], Laptev 1 uses HOF with additional keyframe priming while Laptev 2 and 3 use HOG3D. Even though we use a single frontal view query, the proposed method performs favorably with Laptev 1 and 2. Twenty strongest detections (sorted in decreasing order of resemblance volume score) with the proposed method are illustrated in Fig. **3.31**. In spite of a substantial variation in subject appearance, motion, surrounding scenes, view points and scales, and also abrupt scene change in the video, the proposed method retrieved most of actions at the correct locations. We expect that

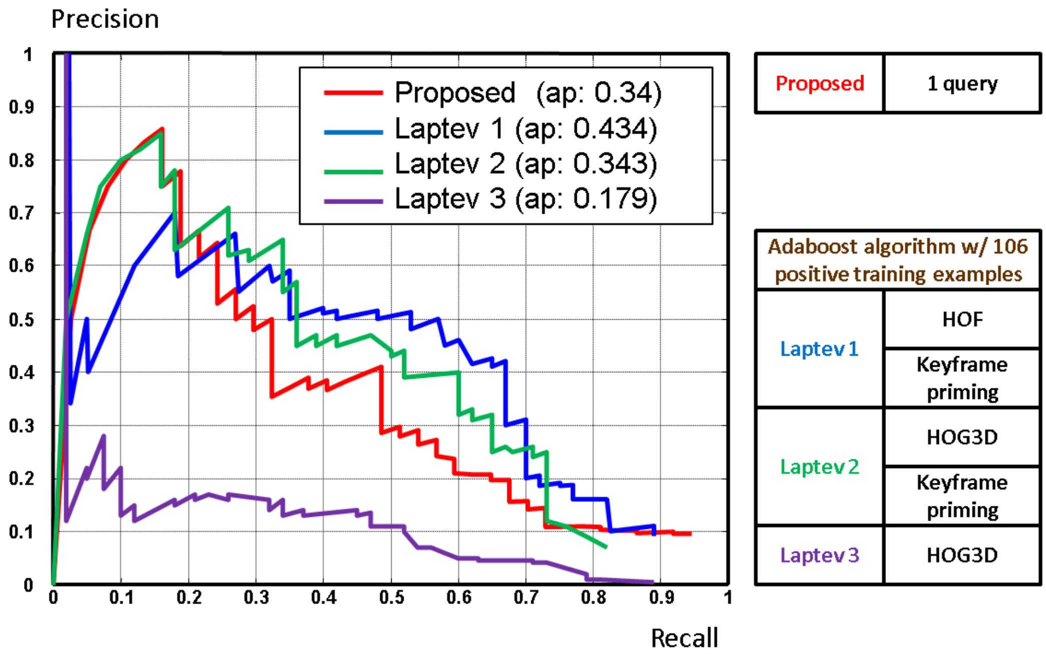


Figure 3.30: Precision-Recall curves comparison between the proposed method and three action detection methods by [23]. The proposed method performs favorably with Laptev 1 and 2 even though there is a single query video used. The average precision (ap) means an average precision over the entire range of recall. This figure is better viewed in color.

our method might also benefit from keyframe priming as discussed in [23].

3.3.3 Action Category Classification

As opposed to action detection, action category classification aims to classify a given action query into one of several pre-specified categories. In earlier discussion on action detection, we assumed that in general the query video is smaller than the target video. Now we relax this assumption, and thus we need a preprocessing step which selects a valid human action from the query video. This idea allows us to not only extend the proposed detection framework to action category classification, but to also improve both detection and classification accuracy by removing unnecessary

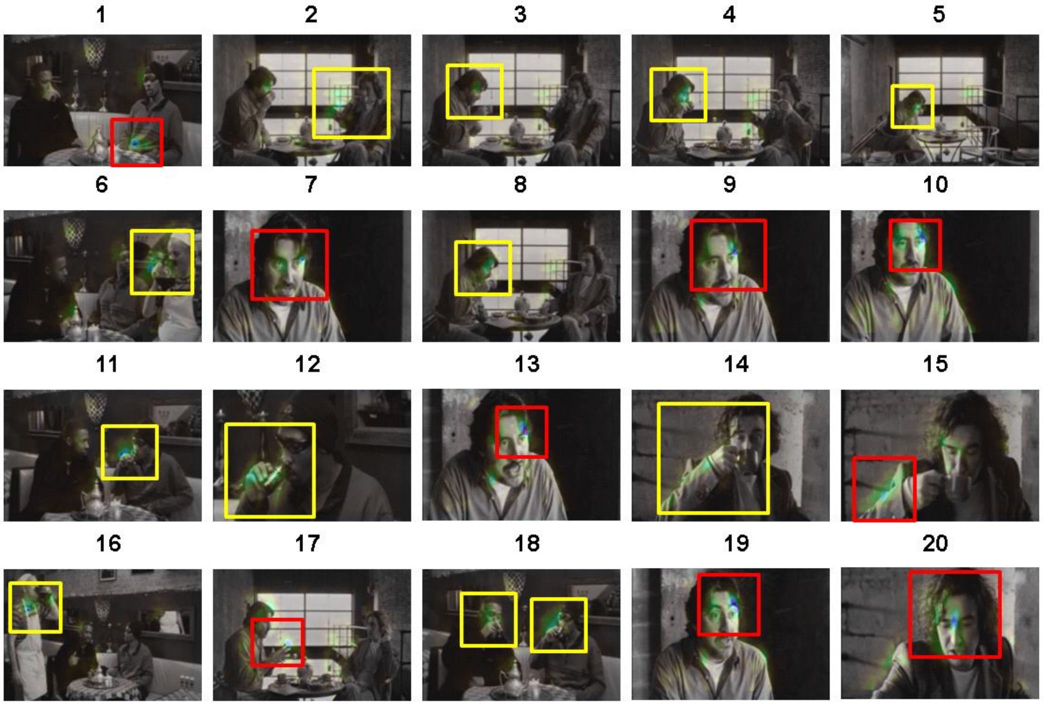


Figure 3.31: Detection of drinking actions (yellow: true positives, red: false positives) sorted in the decreasing confidence order by the proposed method. This figure is better viewed in color.

background from the query video.

Once the query video is cropped to a short action clip by using space-time saliency detection, as described in Chapter 2, the cropped query is searched against each labeled video in the database, and the value of the resemblance volume (RV) is viewed as the likelihood of similarity between the query and each labeled video. Then we classify a given query video as one of the predefined action categories using a nearest neighbor (NN) classifier.

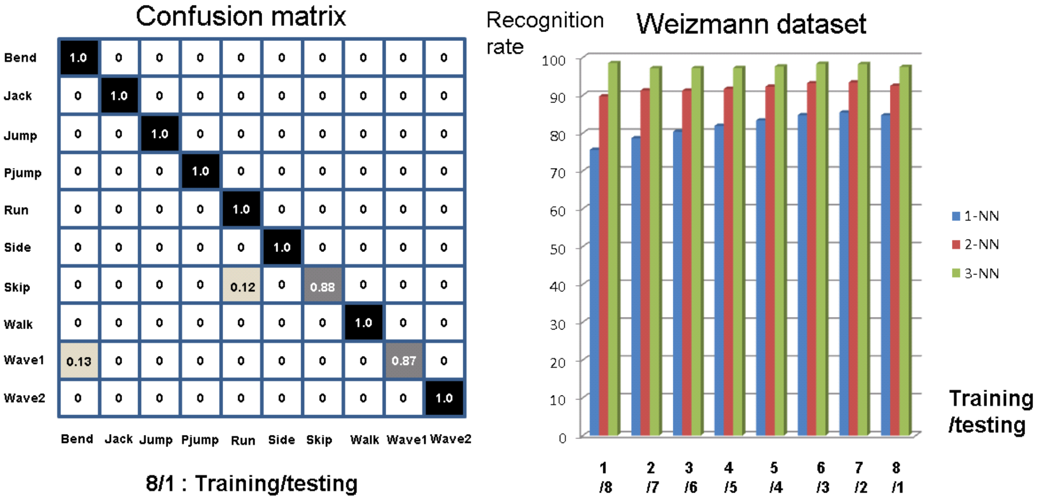
The Weizmann Action Dataset [6] The Weizmann action dataset contains 10 actions (bend, jumping jack, jump forward, jump in place, jump sideways, skip, run, walk, wave with two hands, and wave with one hand) performed by 9 different subjects. This dataset contains videos with static cameras and simple background, but it provides a good testing environment to evaluate the performance of the algorithm when the number of categories are large compared to the KTH dataset (a total of 6 categories). We conducted experiments on the Weizmann dataset under various data split setups. For example, the videos of m subjects are randomly drawn for testing (query) and the videos of the remaining $9 - m$ subject are labeled for each run where $m \in [1, \cdots, 8]$. We applied the automatic action cropping method introduced in the previous section to the query video. Then the resulting short action clip is matched against the remaining labeled videos using the proposed method. We classify each testing video as one of the 10 action types by 3-NN (nearest neighbor) as similarly done in [22]. The results are reported as the average of 100 runs. To begin, we achieved a recognition rate of 97.5% for all ten actions in the leave-one-out setting ($m = 1$). The recognition rate comparison is provided in Table **3.3** as well. The proposed method performs favorably against state-of-the-art methods [115, 137, 138, 139, 107, 140, 141, 65]. We observe that these results also compare favorably to several state-of-the-art methods even though our method involves no training phase, and requires no background/foreground segmentation. As an added bonus, our method provides localization of actions as a side benefit. Fig. **3.32** (left) shows the confusion matrix for our method.

Next, we provide further results using 1-NN and 2-NN in comparison to 3-NN in Fig. **3.32** (right) with respect to various split setups. The recognition rates are quite stable regardless of the split used.

The KTH Action Dataset [7] In order to further quantify the performance of our algorithm, we also conducted experiments on the KTH dataset. The KTH action dataset

Table 3.3: Comparison of average recognition rate on the Weizmann dataset [6]

Our approach	1-NN	2-NN	3-NN
Recognition rate	**84.7%**	**92.5%**	**97.5%**
Method	Junejo et al. [137]	Liu et al. [138]	Klaser et al. [65]
Recognition rate	95.33%	90%	84.3%
Method	Niebles et al. [115]	Fathi and Mori [142]	Zhang et al. [141]
Recognition rate	90%	100%	92.89%
Method	Jhuang et al. [139]	Batra et al. [140]	Bregonzio et al. [85]
Recognition rate	98.8%	92%	96.6%
Method	Ali et al. [107]	Sun et al. [143]	Schindler and van Gool [144]
Recogniton rate	95.75%	97.8%	100%



Confusion matrix

	Bend	Jack	Jump	Pjump	Run	Side	Skip	Walk	Wave1	Wave2
Bend	1.0	0	0	0	0	0	0	0	0	0
Jack	0	1.0	0	0	0	0	0	0	0	0
Jump	0	0	1.0	0	0	0	0	0	0	0
Pjump	0	0	0	1.0	0	0	0	0	0	0
Run	0	0	0	0	1.0	0	0	0	0	0
Side	0	0	0	0	0	1.0	0	0	0	0
Skip	0	0	0	0	0.12	0	0.88	0	0	0
Walk	0	0	0	0	0	0	0	1.0	0	0
Wave1	0.13	0	0	0	0	0	0	0	0.87	0
Wave2	0	0	0	0	0	0	0	0	0	1.0

8/1 : Training/testing

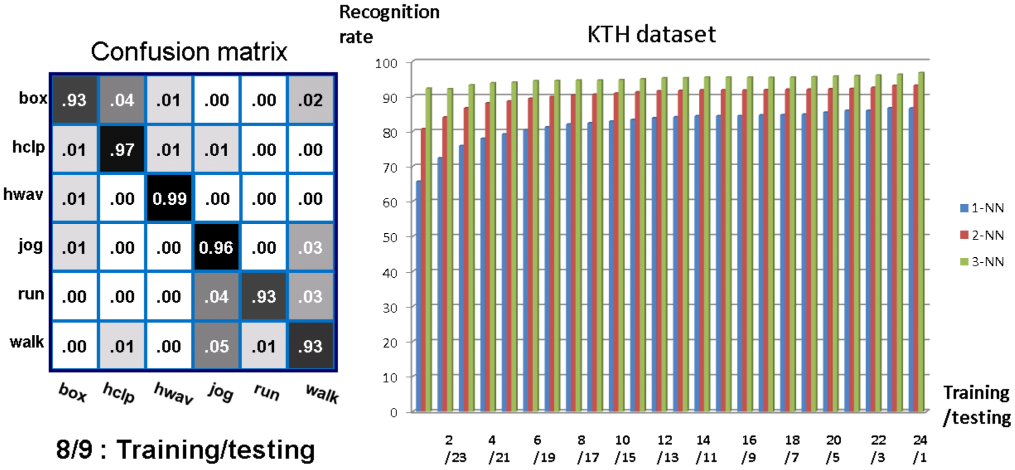
Figure 3.32: Left: Confusion matrix on the Weizmann dataset for the leave-one-out setting, Right: Average recognition rate according to various data split setups. (Weizmann dataset)

contains six types of human actions (boxing, hand waving, hand clapping, walking, jogging, and running), performed repeatedly by 25 subjects in 4 different scenarios: outdoors (c_1), outdoors with camera zoom (c_2), outdoors with different clothes (c_3), and indoors (c_4). This dataset seems more challenging than the Weizmann dataset because there are large variations in human body shape, view angles, scales, and appearance. We also evaluate our method on the KTH dataset under various split setups. First, we use the same setup as in [7], *i.e.,* 8 people for training[27] and 9 for testing for each run. The recognition rates are reported as the average of 100 runs for this setup. We were able to achieve a recognition rate of 95.1% on these six actions. Fig. **3.33** (left) shows the average confusion matrix across all scenarios for this setup. The recognition rate comparison with competing methods is provided in Table **3.4** as well. Our method outperforms all the other state-of-the-art methods and is fully automatic. We further tried other data-split setups as similarly done in the previous section. The videos of m subjects are randomly drawn for testing (query) and the videos of the remaining subject $25 - m$ are labeled for each run, where $m \in [1, \cdots, 24]$. As shown in Fig. **3.33** (right), it is consistent with the results on the Weizmann dataset that the recognition rates are quite stable regardless of the split used as similarly stated in [145].

Table 3.4: Comparison of average recognition rate on the KTH dataset

Our approach	1-NN	2-NN	3-NN
Recognition rate	**82.7**%	**91**%	**95.1**%
Method	Kim et al. [113]	Ning et al. [22]	Klaser et al. [65]
Recognition rate	95.33%	92.31% (3-NN)	91.4%
Method	Laptev et al. [64]	Niebles et al. [115]	Liu and Shah [145]
Recognition rate	91.8%	81.5%	94.2%
Method	Dollar et al. [146]	Wong et al. [147]	Rapantzikos et al. [148]
Recogniton rate	81.17%	84%	88.3%
Method	Ali et al. [107]	Sun et al. [143]	Schindler and van Gool [144]
Recogniton rate	87.7%	94%	92.7%

[27] We use the term "training" here to be consistent with notation used in the literature even though our method does not require training mechanisms.



Confusion matrix

	box	hclp	hwav	jog	run	walk
box	.93	.04	.01	.00	.00	.02
hclp	.01	.97	.01	.01	.00	.00
hwav	.01	.00	0.99	.00	.00	.00
jog	.01	.00	.00	0.96	.00	.03
run	.00	.00	.00	.04	.93	.03
walk	.00	.01	.00	.05	.01	.93

8/9 : Training/testing

Figure 3.33: Left: Confusion matrix on the KTH dataset for the 8 training/ 9 testing setup, Right: Average recognition rate according to different data split setup. (KTH dataset)

Discussion It is important to note that our features computed using the PCA process are a function of the input query video, and therefore are adapted to each changing query. As such, one would expect them to serve better in identifying actions that are similar to the given query in a way that is more accurate than would a generic basis. Indeed, the tradeoff between having a fixed basis for all input queries and a basis that is extracted from each query manifests itself as a tradeoff between stability and specificity. Despite the higher computational cost we pay, our process for extraction of features appear to be stable, yet showing rather high specificity at the same time, resulting in overall very good performance.

Our system is designed with recognition accuracy as a high priority. A typical run of the action detection system implemented in Matlab takes a little over 1 minute on a target video T (50 frames of 144×192 pixels, Intel Pentium CPU 2.66 Ghz machine) using a query Q (13 frames of 90×110). Most of the run-time is taken up by the computation of MCS (about 9 seconds, and 16.5 seconds for the computation of 3-D LARKs from Q and T respectively, which needs to be computed only once.) There are many factors that affect the precise timing of the calculations, such as query size, complexity of the video, and 3-D LARK size. We can speed up the proposed method by applying coarse-to-fine search [149] as done in Chapter 4. Even though our method is stable in the presence of moderate amount of camera motion, our system can benefit from camera stabilization methods as done in [150, 151] in case of large camera movements.

In the Weizmann dataset and the KTH dataset, target videos contain only one type of action. However, a target video may contain multiple actions in practice. In this case, simple nearest neighbor classifiers can possibly fail. Therefore, we might benefit from contextual information to increase accuracy of action recognition systems as similarly done in [152]. In fact, there is a broad agreement in the computer vision community about the valuable role that context plays in any image understanding task [153, 154].

Summary – In this chapter, we have proposed a unified framework for both object and action detection by employing *LARKs* and *MCS*. The proposed method can automatically detect in the target the presence, the number, as well as location of similar objects (actions) to the given *single* query. To deal with more general scenarios, accounting for large variations in scale and rotation, we further proposed multiscale and multirotation approaches. Experiments on challenging sets of real-world object and action data have demonstrated that the proposed approach achieves a high detection accuracy in varied contexts and under different imaging conditions. However, due to the heavy computational complexity, it is difficult for the proposed method to be extended to real-time applications. In the next chapter, we introduce how to speed-up the computation of LARK and realize a real-time object and action detection system by employing coarse-to-fine pyramid search in conjunction with a tree-structure.

3A Handling Variations

Although our detection framework can handle modest scale and rotation variations by adopting a sliding window scheme, robustness to larger scale and rotation changes (for instance above ±20% in scale, 30 degrees in rotation) are desirable. Furthermore, the use of color images as input should be also considered from a practical point of view. In this section, the approach described in the previous sections for detecting objects at a single scale is extended to detect objects at different scales[28] and at different orientations in an image. In addition, we deal with a color image by defining and using "Canonical Cosine Similarity".

Multi-Scale approach In order to cope with large scale variations, we construct a multi-scale pyramid of the target T. This is a non-standard pyramid as we reduce the target size by steps of $10 \sim 15\%$, so that a relatively fine quantization of scales are taken into account. Fig. **3.34** (a) shows the block diagram of the multi-scale approach. The first step is to construct the multi-scale pyramid $T^0, T^1, \cdots, T^S$ where S is the coarsest scale of the pyramid. As shown in Fig. **3.34** (a), $\mathbf{F}_Q, \mathbf{F}_{T^0}, \mathbf{F}_{T^1}, \mathbf{F}_{T^2} (S = 2)$ are obtained by projecting $\mathbf{K}_Q$ and $\mathbf{K}_{T^0}, \mathbf{K}_{T^1}, \mathbf{K}_{T^2}$ onto the principal subspace defined by $\mathbf{A}_Q$ as follows:

$$\begin{aligned} \mathbf{F}_Q &= \mathbf{A}_Q^\top \mathbf{K}_Q, \quad \mathbf{F}_{T^0} = \mathbf{A}_Q^\top \mathbf{K}_{T^0}, \\ \mathbf{F}_{T^1} &= \mathbf{A}_Q^\top \mathbf{K}_{T^1}, \quad \mathbf{F}_{T^2} = \mathbf{A}_Q^\top \mathbf{K}_{T^2}. \end{aligned} \tag{3.10}$$

[28] Note that multi-scale approach of saliency detection in Chapter 2 did not improve overall performance while multi-scale approach of object detection is effective. This is in part due to the fact that we rescale a center patch (which corresponds to a query in object detection) and surrounding patches at the same time, while we only rescale target images without changing the size of query image. Another reason is that we use LARKs of size 3 × 3 in saliency detection whereas 7 × 7 LARKs are employed in conjunction with PCA in object detection.

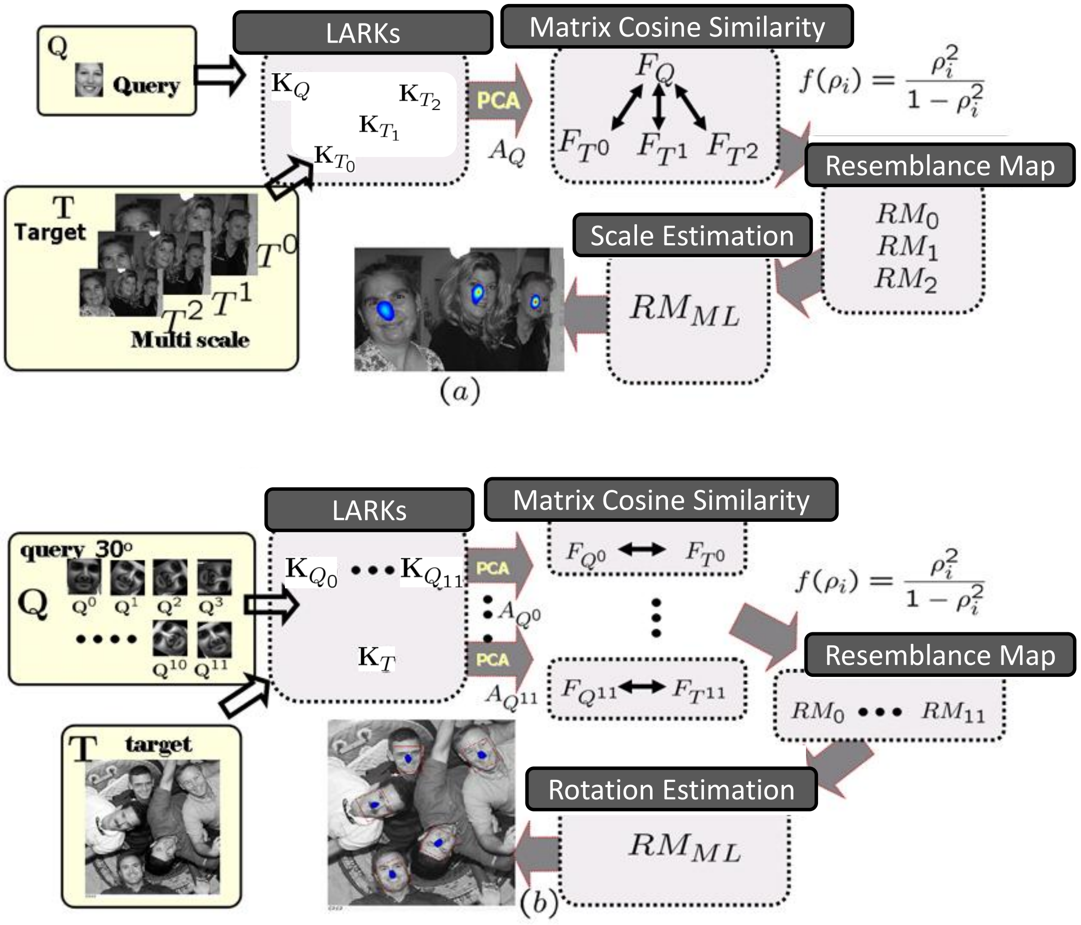


Figure 3.34: Block diagrams of (a) multi-scale object detection system and (b) multi-rotation object detection system.

We obtain three resemblance maps $\mathrm{RM}_0, \mathrm{RM}_1, \mathrm{RM}_2$ by computing $f(\rho_i) = \frac{\rho_i^2}{1-\rho_i^2}$. These resemblance maps represent the likelihood functions $p(f(\rho_i)|S_i)$ where S_i is the scale at i^{th} point. However the sizes of the respective resemblance maps $\mathrm{RM}_0, \mathrm{RM}_1, \mathrm{RM}_2$ are naturally different. Therefore, we simply upscale all the resemblance maps by pixel replication so that they match the dimensions of the finest scale map RM_0. Next, the maximum likelihood estimate of the scale at each position is arrived at by comparing the upscaled resemblance maps as follows[29]:

$$\hat{S}_i = \arg\max_{S_i} p(\underline{\mathrm{RM}}|S_i). \tag{3.11}$$

Multi-Rotation approach In order to cope with large rotations, we take a similar approach and generate rotated images (this time of the query Q) in roughly 30 degree steps. As seen in Fig. **3.34** (b), $\mathbf{F}_{Q^0}, \mathbf{F}_{Q^1}, \cdots, \mathbf{F}_{Q^{11}}$ and $\mathbf{F}_T$ are obtained by projecting $\mathbf{K}_{Q^0}, \cdots, \mathbf{K}_{Q^{11}}$ and $\mathbf{K}_T$ onto the principal subspace defined by $\mathbf{A}_{Q^0}, \cdots, \mathbf{A}_{Q^{11}}$. After computing $f(\rho_i) = \frac{\rho_i^2}{1-\rho_i^2}$ from 12 pairs by employing the sliding window scheme, we obtain twelve resemblance maps $\mathrm{RM}_0, \cdots, \mathrm{RM}_{11}$. We compute the maximum likelihood estimate of the best matching pattern accounting for rotation as follows:

$$\hat{R}_i = \arg\max_{R_i} p(\underline{\mathrm{RM}}|R_i). \tag{3.12}$$

Canonical Cosine Similarity Now, we define Canonical Cosine Similarity (CCS) to extend the proposed framework with a single gray-scale query image to vector-valued images. In particular, suppose at each pixel, the image has q values. As per the earlier discussion (Section 3.2.2), we generate q feature sets $\mathbf{F}_Q^\ell, \mathbf{F}_{T_i}^\ell (\ell = [1, \cdots, q])$ by projecting $\mathbf{K}_Q^\ell, \mathbf{K}_{T_i}^\ell$ onto the subspaces $\mathbf{A}_Q^\ell$ respectively and form the overall feature set as follows:

$$\mathbf{F}_I = [\mathrm{colstack}(\mathbf{F}_I^1), \cdots, \mathrm{colstack}(\mathbf{F}_I^q)] \in \mathbb{R}^{(d \times n) \times q}, I \in \{Q, F_i\}. \tag{3.13}$$

The key idea is to find the vectors $\mathbf{u}_Q$ and $\mathbf{u}_{T_i}$ which maximally correlate two data sets ($\mathbf{F}_Q$,$\mathbf{F}_{T_i}$).

$$\mathbf{v}_I = \mathbf{F}_I \mathbf{u}_I = u_{I_1} \mathrm{colstack}(\mathbf{F}_I^1) + \cdots + u_{I_q} \mathrm{colstack}(\mathbf{F}_I^q) \in \mathbb{R}^{(d \times n)}, \tag{3.14}$$

where $\mathbf{u}_Q = [u_{Q_1}, \cdots, u_{Q_q}]^\top \in \mathbb{R}^q$ and $\mathbf{u}_{T_i} = [u_{T_1}, \cdots, u_{T_q}]^\top \in \mathbb{R}^q$.

[29] By $\underline{\mathrm{RM}}$ we mean a collection of RM indexed by i at each position.

Then, the objective function we are maximizing is the cosine similarity between $\mathbf{v}_Q$ and $\mathbf{v}_{T_i}$ as follows

$$\begin{aligned} \rho &= \max_{\mathbf{u}_Q, \mathbf{u}_{T_i}} \frac{\mathbf{v}_Q^\top \mathbf{v}_{T_i}}{\|\mathbf{v}_Q\| \|\mathbf{v}_{T_i}\|} = \max_{\mathbf{u}_Q, \mathbf{u}_{T_i}} \frac{\mathbf{u}_Q^\top \mathbf{F}_Q^\top \mathbf{F}_{T_i} \mathbf{u}_{T_i}}{\|\mathbf{F}_Q \mathbf{u}_Q\| \|\mathbf{F}_{T_i} \mathbf{u}_{T_i}\|}, \\ & \text{such} \quad \text{that} \quad \|\mathbf{F}_Q \mathbf{u}_Q\| = \|\mathbf{F}_{T_i} \mathbf{u}_{T_i}\| = 1, \end{aligned} \tag{3.15}$$

where $\mathbf{u}_Q$ and $\mathbf{u}_{T_i}$ are called canonical variates and ρ is the canonical cosine similarity. The above is inspired by canonical correlation analysis (CCA) [127].

The Lagrangian objective function to the minimization problem in Equation (3.15) is

$$f(\lambda_Q, \lambda_T, \mathbf{u}_Q, \mathbf{u}_{T_i}) = \mathbf{u}_Q^\top \mathbf{F}_Q^\top \mathbf{F}_{T_i} \mathbf{u}_{T_i} - \frac{\lambda_Q}{2}(\mathbf{u}_Q^\top \mathbf{F}_Q^\top \mathbf{F}_Q \mathbf{u}_Q - 1) - \frac{\lambda_{T_i}}{2}(\mathbf{u}_{T_i}^\top \mathbf{F}_{T_i}^\top \mathbf{F}_{T_i} \mathbf{u}_{T_i} - 1). \tag{3.16}$$

Taking derivatives with respect to $\mathbf{u}_Q$ and $\mathbf{u}_{T_i}$, we obtain

$$\frac{\partial f}{\partial \mathbf{u}_Q} = \mathbf{F}_Q^\top \mathbf{F}_{T_i} \mathbf{u}_{T_i} - \lambda_Q (\mathbf{F}_Q^\top \mathbf{F}_Q \mathbf{u}_Q) = 0, \tag{3.17}$$

$$\frac{\partial f}{\partial \mathbf{u}_{T_i}} = \mathbf{F}_{T_i}^\top \mathbf{F}_Q \mathbf{u}_Q - \lambda_{T_i} (\mathbf{F}_{T_i}^\top \mathbf{F}_{T_i} \mathbf{u}_{T_i}) = 0. \tag{3.18}$$

We pre-multiply $\mathbf{u}_{T_i}^\top$ to Equation (3.18) and also pre-multiply $\mathbf{u}_Q^\top$ to Equation (3.17). By subtracting these two equations, we have

$$\mathbf{u}_Q^\top \mathbf{F}_Q^\top \mathbf{F}_{T_i} \mathbf{u}_{T_i} - \lambda_Q (\mathbf{u}_Q^\top \mathbf{F}_Q^\top \mathbf{F}_Q \mathbf{u}_Q) - \mathbf{u}_{T_i}^\top \mathbf{F}_{T_i}^\top \mathbf{F}_Q \mathbf{u}_Q - \lambda_{T_i} (\mathbf{u}_{T_i}^\top \mathbf{F}_{T_i}^\top \mathbf{F}_{T_i} \mathbf{u}_{T_i}) = 0, \tag{3.19}$$

where $(\mathbf{u}_Q^\top \mathbf{F}_Q^\top \mathbf{F}_{T_i} \mathbf{u}_{T_i})^\top = \mathbf{u}_{T_i}^\top \mathbf{F}_{T_i}^\top \mathbf{F}_Q \mathbf{u}_Q$ is a scalar. Enforcing the constraints $(\mathbf{u}_Q^\top \mathbf{F}_Q^\top \mathbf{F}_Q \mathbf{u}_Q)^\top = (\mathbf{u}_{T_i}^\top \mathbf{F}_{T_i}^\top \mathbf{F}_{T_i} \mathbf{u}_{T_i})^\top = 1$, we are led to the conclusion that $\lambda_Q = \lambda_{T_i}$. We define $\rho = \lambda_Q = \lambda_{T_i}$. Assuming $\mathbf{F}_{T_i}^\top \mathbf{F}_{T_i}$ is invertible from Equation (3.18),

$$\mathbf{u}_{T_i} = \frac{(\mathbf{F}_{T_i}^\top \mathbf{F}_{T_i})^{-1} \mathbf{F}_{T_i} \mathbf{F}_Q \mathbf{u}_Q}{\rho}, \tag{3.20}$$

and so plugging in Equation (3.17), we have

$$\frac{(\mathbf{F}_Q^\top \mathbf{F}_{T_i})(\mathbf{F}_{T_i}^\top \mathbf{F}_{T_i})^{-1}(\mathbf{F}_{T_i}\mathbf{F}_Q)\mathbf{u}_Q}{\rho} = \rho(\mathbf{F}_Q^\top \mathbf{F}_Q)\mathbf{u}_Q. \tag{3.21}$$

Assuming $\mathbf{F}_Q^\top \mathbf{F}_Q$ is also invertible, we are left with

$$(\mathbf{F}_Q^\top \mathbf{F}_Q)^{-1}(\mathbf{F}_Q^\top \mathbf{F}_{T_i})(\mathbf{F}_{T_i}^\top \mathbf{F}_{T_i})^{-1}(\mathbf{F}_{T_i}^\top \mathbf{F}_Q)\mathbf{u}_Q = \rho^2 \mathbf{u}_Q. \tag{3.22}$$

Similarly, we have

$$(\mathbf{F}_{T_i}^\top \mathbf{F}_{T_i})^{-1}(\mathbf{F}_{T_i}^T \mathbf{F}_Q)(\mathbf{F}_Q^\top \mathbf{F}_Q)^{-1}(\mathbf{F}_Q^\top \mathbf{F}_{T_i})\mathbf{u}_{T_i} = \rho^2 \mathbf{u}_{T_i}. \tag{3.23}$$

The canonical cosine similarity ρ and canonical variates $\mathbf{u}_Q, \mathbf{u}_{T_i}$ can be obtained by solving the above coupled eigenvalue problems. The positive square root of eigenvalues ρ^2 is the "Canonical Cosine Similarity". If $\mathbf{F}_Q, \mathbf{F}_{T_i}$ are each composed of a single vector (colstack($\mathbf{F}_Q$), colstack($\mathbf{F}_{T_i}$)), the above equations reduce to $\frac{(\text{colstack}(\mathbf{F}_Q)^\top \text{colstack}(\mathbf{F}_{T_i}))^2}{\|\text{colstack}(\mathbf{F}_Q)\|^2 \|\text{colstack}(\mathbf{F}_{T_i})\|^2} = \rho^2$ which is just the squared cosine similarity defined earlier in Section 3.2.2.

Now, we take a closer look at the particular case of color images where $q = 3$. A natural question here is whether we can gain more if we use the color information instead of using only the luminance channel as we have so far. The answer to this question is positive. There exist many color spaces such as RGB, YCbCr, CIE L*a*b* etc. We observe that CIE L*a*b color model provides the most discriminative information among all as also observed by Shechtman and Irani [11]. We define the respective RM[30] as the summation of mapping function $f(\rho_i(\ell))$ of CCS $\rho_i(\ell)$ between a set of features which are calculated from each channel ($\ell = 1, \cdots, q$), where $\sum_{\ell=1}^{d_c} \frac{\rho_i^2(\ell)}{1-\rho_i^2(\ell)}$) ($d_c$ is the

[30] Again as mentioned earlier, note that $\sum_{\ell=1}^{d_c} \frac{\rho_i^2(\ell)}{1-\rho_i^2(\ell)}$ is analogous to the Lawley-Hotelling trace test statistic $\sum \frac{\rho^2}{1-\rho^2}$ which is used in the significance test of canonical variates in canonical correlation analysis [127, 130].

number of canonical cosine similarity values $\rho_i(\ell)$ greater than zero). Also illustrated in Section 3.3, the color approach based on CCS not only provides better discriminative power, but also gives more accurate localization results than the luminance channel only does.

Chapter 4

Real-time Robot Vision with Scalable LARK Descriptors

Abstract – Although LARKs are flexible enough to recognize a wide spectrum of objects, humans and actions with high accuracy, they have not been applied to a real-time application due to their heavy computational complexity. In this chapter, we aim at developing a real-time detection system with (scalable) LARKs that will support human-robot interaction (HRI). To allow interaction, it must run in real time and only need a few examples to learn with. In this chapter, given these requirements, we find an efficient method to compute LARKs in real time. We then construct a feature pyramid that allows for rapid coarse-to-fine object search. Finally, we employ tree-based hierarchical feature clustering so that recognition time grows slower than linear with the number of objects/examples learned, thus achieving our aims.

4.1 Introduction

In the last decade, we have witnessed substantial progress in the problem of object detection/classification. One popular paradigm for visual object classification is based on histogram of gradients descriptors such as SIFT and its variants such as HOG, SURF, etc., computed from an image either at interest points or densely. The image is then represented as a histogram of codewords generated by applying vector quantization to a collection of descriptors. The resulting histogram is used as an input to a standard classifier, for example, a support vector machine (SVM). While these approaches often provide acceptable accuracy for object categorization tasks, most of these approaches are not appropriate for interactive object recognition from few examples because general classifiers are often learned off-line and when a new instance is added, classifiers often must be trained from scratch using many training examples for each category in order to cope with intra-class variations.

For effective human-robot interaction, recognition of many objects should run in real time. The generic object detection framework with LARKs described in Chapter 3 achieved high detection accuracy on challenging datasets (including face, car, and on various generic objects) using a single example even in the presence of variations in scale, rotation, local deformation, and illumination. In our case, for a robot detecting man-made rigid objects and their view based pose, we need to use several examples per object but we hope to keep this at a minimum. We take advantage of LARKs which are highly discriminative and tolerant to viewpoint changes so that only a few views are sufficient for learning a new object, person or action. LARK is a good candidate for our needs, but it is computationally complex and thus too slow for real-time object detection. In this chapter, we develop a sped-up version of LARK.

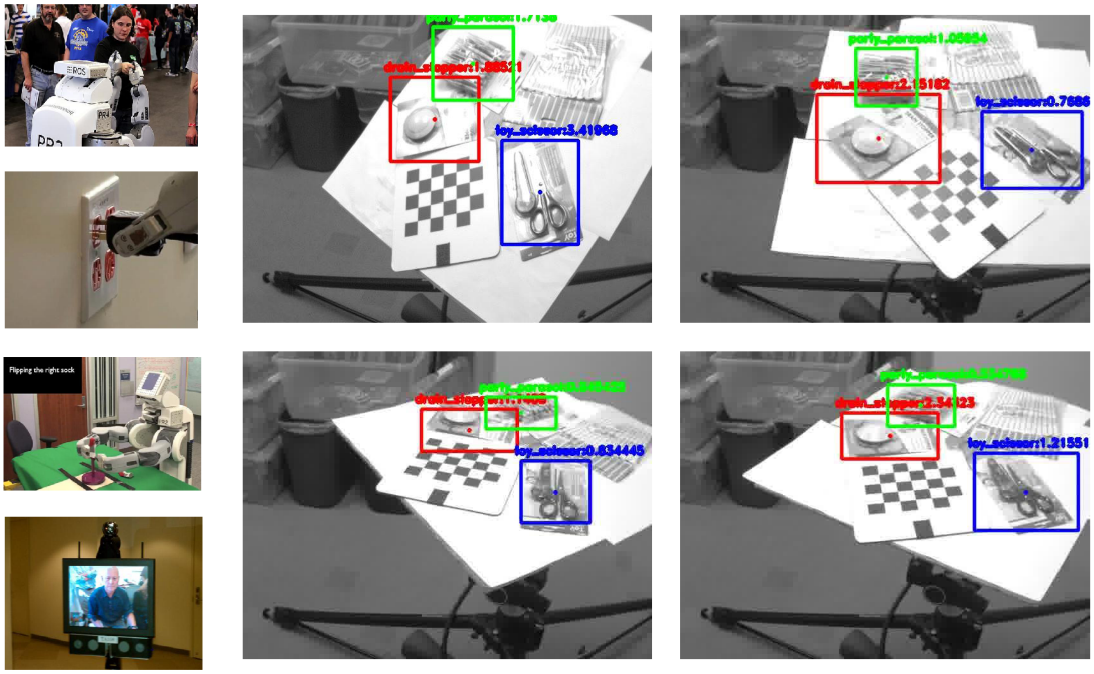


Figure 4.1: Left: Robots are increasingly working with humans (HRI). For HRI we desire quick learning from few examples followed by recognition and pose in real time. Right: The proposed detection system can reliably detect textured objects in cluttered backgrounds in real-time from just a few learned examples.

Related Works Earlier approaches such as [155] used the Chamfer distance between example and input image contours. However, contours are sensitive to the presence of blur, noise, and illumination changes. Even though a recent approach [156] based on a Hough-style voting scheme with a non-rigid shape matching on the contour image achieved very good classification results. It is not applicable to real-time object detection due to heavy computational cost.

The method proposed in [33] tried to overcome the limitations of using contours by considering a two level histogram of image gradients descriptors which provide invariance to local transformations. Recently, [57] proposed learning locally-invariant features through topographic filter maps. The discriminative power of these invariant types of features heavily relies on a large training set making learning quite slow and one-shot learning impractical.

[135] proposed CHoG descriptors by compressing gradient histograms with a tree structure. However, CHoG was only tested for a descriptor matching task. A new binarized gradient example matching method, dominant orientation examples (DOT) [24] achieved very fast detection. DOT is designed to be tolerant to small image transformations (small shifts and rotations). However, this representation tends to have many false positives in highly textured areas as it is based directly on the image gradients. Since DOT has limited invariance, it requires many examples to learn an object and so is not suited to one-shot learning.

Recently, [143] proposed a depth-encoded Hough voting detection scheme in order to localize objects and estimate their pose. They incorporate depth information into the process of learning distributions of image patches. The requirement of 3-D training information along with a heavy computational cost keeps this method from real-time detection.

In this chapter, we generalize and build on [49] to derive a fast generic object detection system that can scale to multiple objects and run in real-time. We also utilize the saliency detection of [48] (also described in Chapter 2) with these improved LARKs to allow us to rapidly focus on regions of the visual scene that are most likely to contain objects. While the baseline saliency detection method was quite slow, saliency detection based on our new LARK feature can compute saliency maps at 60 frames per second on a core two 2GHz desktop for 640×480 images.

Contributions The contributions in this chapter serve to make LARK practical for fast learning, real time, scalable object and human action recognition. We first develop

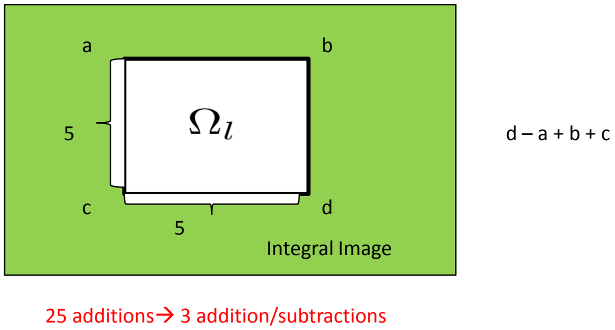


Figure 4.2: Integral image representation. The summation over a local window Ω_l of size 5×5 reduces down to 2 addition and 2 subtractions.

an efficient method to speed up the computation of LARKs. By using salient region detection based on the faster version of LARK, we reduce search space. We then develop a coarse-to-fine pyramid approach to allow for rapid object search and combine it with a tree based structure for sharing multiple examples. Search time of the resulting approach grows slower than linear (logarithmic on average) in learning multiple objects. Our algorithm uses only a few examples to learn each object allowing for quick, online learning. In training, examples are automatically added as they are needed. We demonstrate accurate detection performance of LARK on a standard datasets [19] and show the utility of LARK for robotic object recognition and action recognition in cluttered backgrounds. In the following section, we introduce how to accelerate the computation of LARKs.

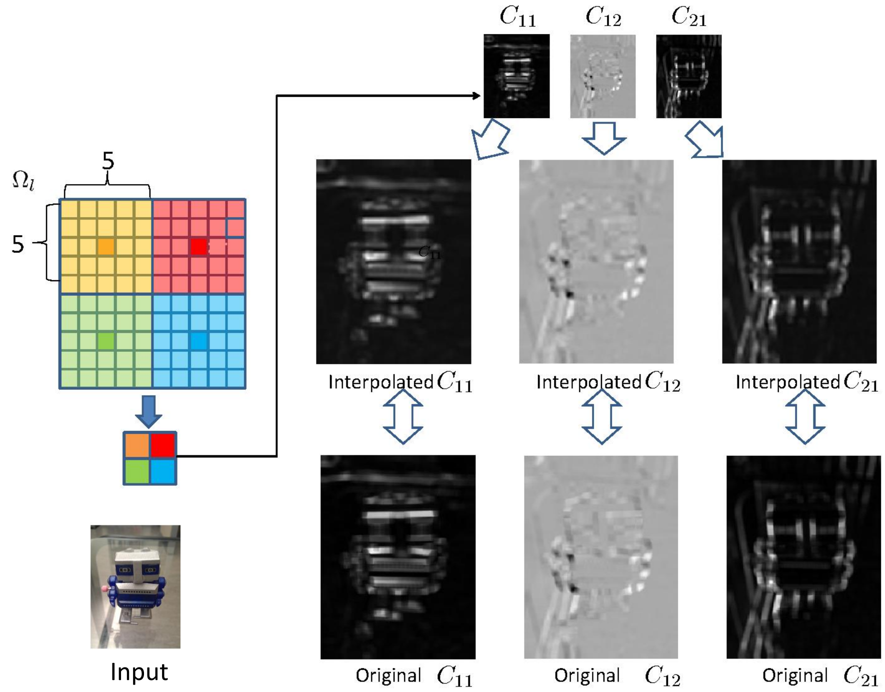


Figure 4.3: $\mathbf{C}_l^{reg}$ is computed in a grid of 5 pixels and upsampled to the original scale by using lanczos interpolation over 8×8 neighborhood. Due to redundancy of Ω_l, resulting LARKs between down-scale interpolated $\mathbf{C}_l^{reg}$ and the full-scale $\mathbf{C}_l^{reg}$ computed at the original scale make little difference, but gives 16× speedup.

4.2 LARK speed-up

Integral Image: In order to efficiently compute “average" $\mathbf{C}_l$ in Equation (1.4), we employ the idea of integral images [29] to the components of $\mathbf{C}_l$: $z_{x_1}^2$, $z_{x_1}z_{x_2}$, $z_{x_2}^2$ respectively. Then summation over a local window Ω_l of size 5×5 reduces down to 2 addition and 2 subtractions as shown in Fig. **4.2**.

Avoid Eigenvalue decomposition: Eigenvalues and eigenvectors of the covariance matrix in Equation (1.5) can be efficiently computed in closed form:

$$\begin{aligned} \lambda_{1,2} &= \frac{(C_{11}+C_{22}) \pm \sqrt{(C_{11}-C_{22})^2 + 4C_{12}C_{21}}}{2}, \\ \mathbf{u}_1 &= [\cos\theta, \sin\theta]^\top, \quad \mathbf{u}_2 = [-\sin\theta, \cos\theta]^\top, \end{aligned} \tag{4.1}$$

where $\mathbf{u}_1, \mathbf{u}_2$ are eigenvectors, $\theta = \tan^{-1}(-\frac{C_{11}+C_{21}-\lambda_1}{C_{22}+C_{12}-\lambda_1})$, and λ_1, λ_2 are eigenvalues. This provides 4 × speedup.

Interpolation of $\mathbf{C}_l^{reg}$: While having a stable estimation of $\mathbf{C}_l^{reg}$ is crucial, most computation is consumed in calculating $\mathbf{C}_l$. We take advantage of redundancy of local patch Ω_l. Instead of computing $\mathbf{C}_l^{reg}$ at every pixel, we interpolate $\mathbf{C}_l^{reg}$ after computing them in a grid of 5 pixels which in turn results in 10 × speed-up (see Fig. **4.3**.) Note that we use the Lanczos interpolation over 8×8 pixel neighborhood (OpenCV implementation[1].)

4.3 Detection speed-up

Salient Region Filter The generic object detection method [49] (also described in Chapter 3) relies on a sliding window scheme which is computationally expensive and in not scalable to large images. In this chapter, we employ saliency detection[2] to reduce the sliding window search spaces. We obtain a saliency map by measuring MCS between a collection of (a faster version of) LARKs in a center patch vs. surrounding patches as described in Chapter 2.

[1] http://opencv.willowgarage.com/documentation/cpp/index.html.

[2] The fact that both saliency detection and object detection share the idea of data-adaptive kernel density estimation naturally leads us to use saliency detection to reduce search space as a pre-processing.

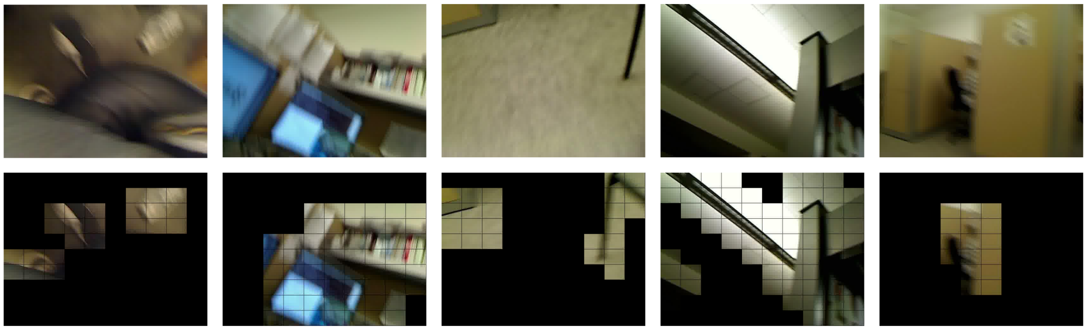


Figure 4.4: Saliency detection results. Images are divided into 10 × 10 blocks. We test each block to see if they are salient or not. Black blocks are regions which are not salient.

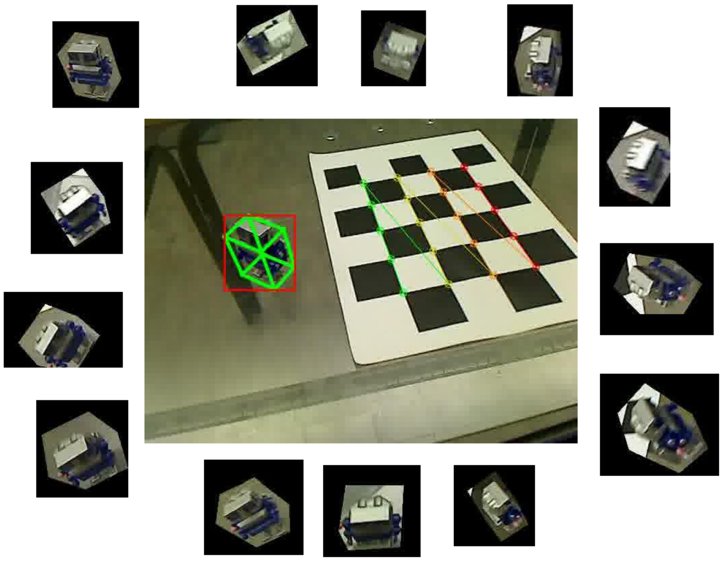


Figure 4.5: examples are automatically registered to the system by applying a chessboard detection and selecting 3-D box around the object. Each example is registered with its mask and pose. We used OpenCV's select3dobj function for these processes.

We divide the saliency map into 10 × 10 blocks, and then compute average values from each block. We declare a block as salient if its score is greater than a fixed threshold (= 0.3). As shown by the masked regions in Fig. **4.4**, this can result in significant computational savings by allowing us to skip searching in non-salient blocks.

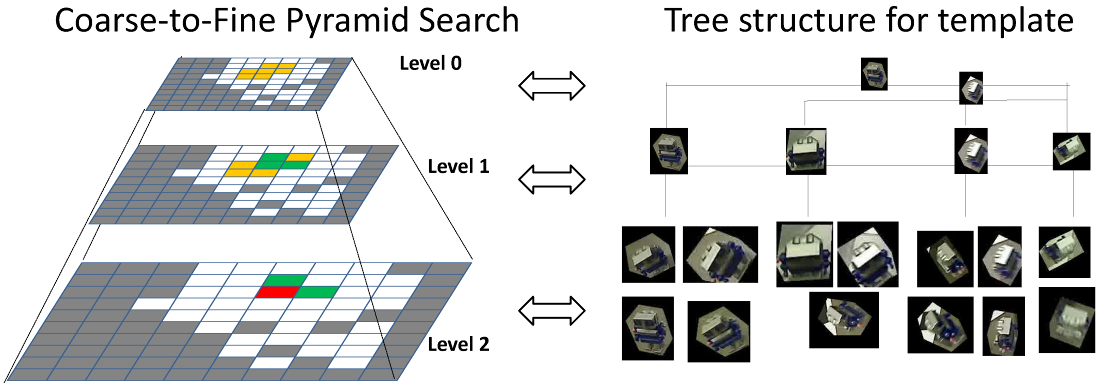


Figure 4.6: By constructing a tree structure of examples and a pyramid of LARK feature images, we can perform efficient a coarse-to-fine search in order to accurately localize an object with its view based pose. This figure is better viewed in color.

Automatic Training In order to register multiple examples that represent varying appearance/pose of man-made rigid objects with respect to camera view points, we use a chessboard detection method. The detected chessboard defines a ground plane coordinate system where the user draws a box around and object and adjusts the box height to cover the object. The process is as follows. The user draws a 3-D box around the object of interest. Once the first example is registered with its mask and pose, multi-scale object detection described in Chapter 3 is performed to the 3-D bounding box in the following frames and new examples are added if the MCS score between the registered examples and the candidate[3] falls below 0.4 (see Fig. **4.5**.) This is repeated until we reach to a maximum number of examples (20~30).

Pyramid Search and Clustering of examples We perform PCA on the collections of LARKs from the all the registered examples in order to construct a common subspace per object category. This differs from Chapter 3 where only one example is used and

[3] We can also use a pan-tilt table to gather examples from multiple viewpoints as shown in Fig **4.1**.

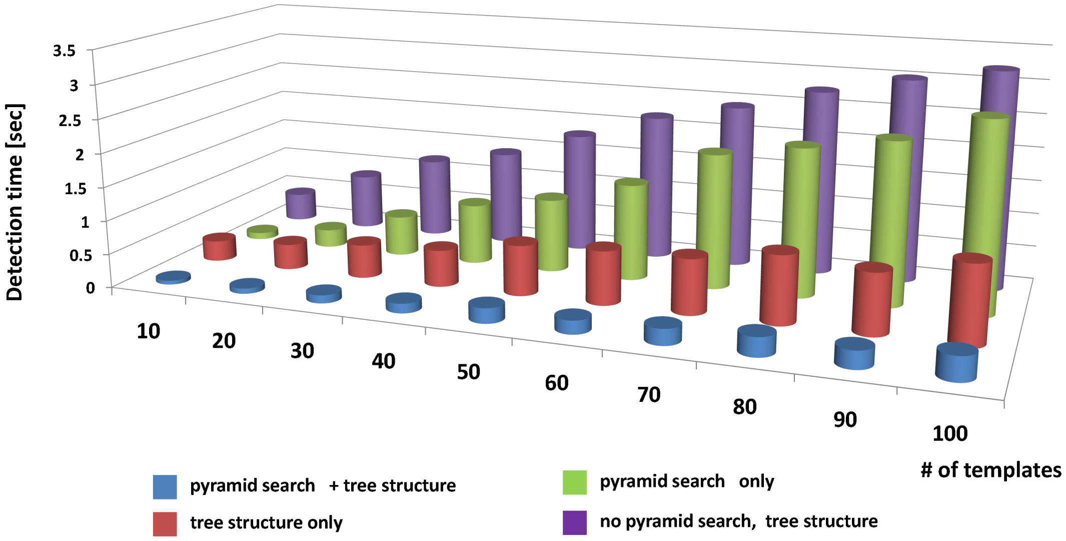


Figure 4.7: The tree based approach grows slower than linear. The use of pyramid search and tree-structured examples results in 10 × speedup on average detection time over 188 frames.

each example possesses its own PCA subspace[4]. We employ a coarse-to-fine search in conjunction with a hierarchical clustering of examples. By constructing a feature pyramid of PCA reduced LARK features and starting search at the coarsest level with a few examples, we find candidate sub-blocks (yellow blocks in Fig. **4.6** Left) which are likely to have objects inside. Indices of these blocks are propagated to the higher level (level 1) and refined (green blocks) with a search using a larger number of examples. This process is repeated until we reach to the finest level (level 2) of the pyramid and detect the object's location (red block) and its pose. In our implementation, we use two levels for both feature pyramid and a hierarchy of example clusters as shown in Fig. **4.6**. This allows for scaling since recognition time grows slower than linear with increasing number of objects/examples (see Fig. **4.7**).

[4] This idea also is applied to face verification problem in Chapter 5. We refer the reader to Chapter 5 for more detail.

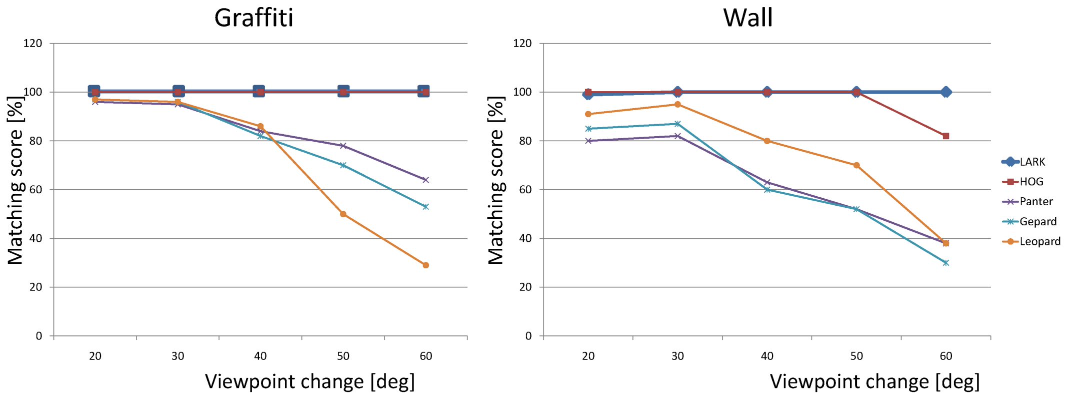


Figure 4.8: Matching score comparisons on the Graffiti and Wall Oxford datasets [24]. Left: Matching scores for Graffiti in terms of viewpoint angles. Right: Matching scores for Wall in terms of viewpoint angles. LARK outperforms the other approaches.

4.4 Experiments

4.4.1 Detection Accuracy

In order to study detection accuracy of the proposed method, we conducted experiments on the Oxford Graffiti dataset and the Wall dataset [19] to find out matching regions between two images as similarly done in [24]. We randomly selected 100 patches from the first image and synthesized 5 different example patches by scaling and rotating the first image of the dataset for changes in viewpoint angle.

The matching score is defined as a ratio of the number of correct matches to the smaller number of regions detected in one image following [19]. It is considered to be correctly matched if the overlap of two regions is smaller than 40%. In Fig. **4.8**, we compare our result with HOG and other patch rectification approaches (Leopar, Panter, Gepard)[24] which appear to perform better than affine region detectors proposed in [19]. Note that other methods[5] used many more examples (around a couple

[5] Note that DOT [24] also achieves 100 % on both dataset, but requires a couple of hundred examples.

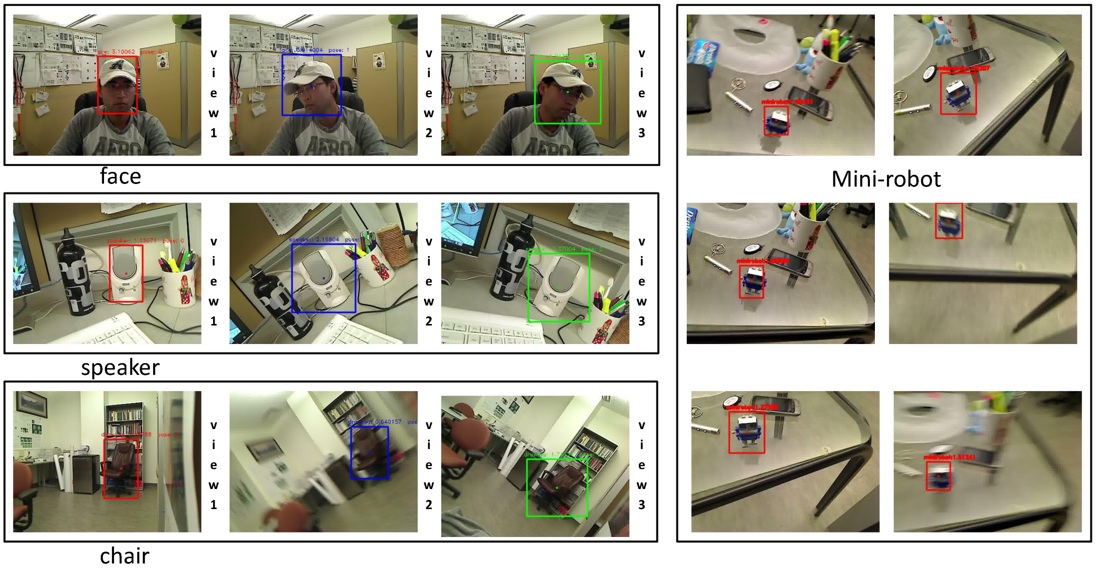


Figure 4.9: Detection of different objects such as face, speaker, chair and mini-robot at about 8 fps in a cluttered background. The color of bounding boxes represents object's rough view angle (or rough pose).

of hundred) than our 5 examples. LARK clearly outperforms the other approaches by achieving 100% matching rate on the Graffiti image set. For the Wall image set, LARK also gets 100% while HOG performs worse for large viewpoint changes.

4.4.2 Real-time Recognition

Real-time Object Recognition LARK can detect generic objects such as face, speaker, chair, and mini-robot as shown in Fig. **4.9**. In case of face, speaker, and chair, we used three different examples corresponding to center, left, right shown by the red, blue, and green boxes respectively. For mini-robot, we collected 12 examples by using chessboard detection as described in section 4.3. The proposed detection system provides a single object's location and rough pose at 8 frames per second on a core two 2GHz desktop for 640 × 480 images. The system can also detect multiple objects simultane-

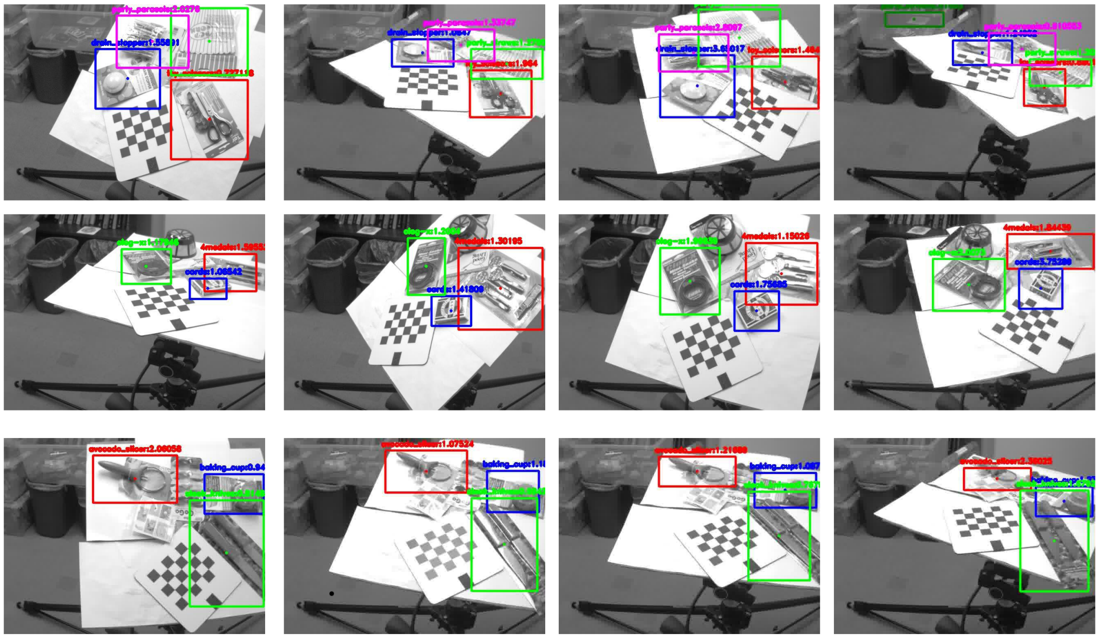


Figure 4.10: Detection of 10 different objects with partial clear plastic covers in multiple view points. 1st row: drain stopper, party straw, party parasol, and toy scissors, 2nd row: clog-x, cards, 4 medals, 3rd row: baking cups, avocado slicer, two knives.

ously as shown in Fig. **4.10**. In this case, recognition was accurate and stable over 20 degrees of table pan and 10 degrees of tilt. In another test, we took 20 common office items such as mugs, plastic cups, stapler, mouse, CD cases etc, and learned an average of 11 examples for each one. Each example seemed to be robust to ±20% scale change and to ±25 degrees of rotation. Data covered the span of typical robot viewpoints. We tested each object with a 6 second “look around" with the object in cluttered desk scenes such as in Fig. **4.9**. Since the robot tracks objects, we defined any solid half second of recognition as a confirmed recognition. In this case, all objects were reliably recognized. We thus have further indication that LARK can learn a wide variety of objects using relatively few examples and recognize them in real time.

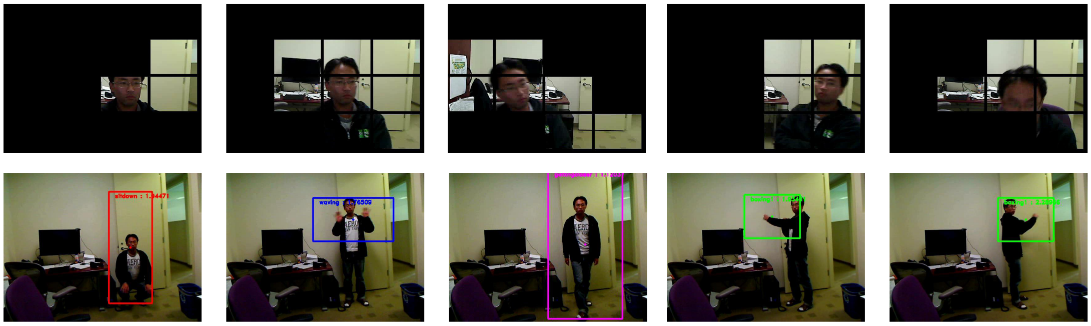


Figure 4.11: Space-time saliency detection and 4 different action recognition: sit-down, waving, boxing, and getting-closer. Recognition is done at about 10 fps in a cluttered background. While most actions were detected and recognized correctly over 20 times, some of waving actions (6) were missed due to a threshold that was set to ensure that all actions are correctly classified as well.

Real-time Action Recognition The extension of fast LARK to action recognition is straightforward. By applying space-time saliency detection with fast 3-D LARK of size $3 \times 3(space) \times 5(time)$, we significantly reduce the search space for actions of interest in real-time. Fig. **4.11** shows that fast 3-D LARK can reliably detect actions such as sit-down, waving, boxing, and getting-closer at about 10 fps in a cluttered background. Over 20 examples of each action, recognition worked completely for all but waving where it worked 70 percent of the time. This is further indication of the flexibility of LARK while maintaining real time recognition speeds.

Summary – In this chapter, we have taken advantage of the discriminative but view-point tolerant properties of LARKs and made them practical for robotic vision systems by speeding them up to run in real time. We have also designed their recognition time to scale slower than linear (logarithmic on average) with increasing numbers of objects. This scaling was done by adopting a coarse-to-fine search in conjunction with a tree structure of examples. LARK learns coarse pose (according to how many views the user wants to learn), runs in real time and works across face, object and action recognition and scales well with increasing numbers of learned items thus making LARK a promising feature to use in robot vision systems. In the next chapter, we further examine the efficacy of LARKs in two more visual recognition applications: face verification and automatic change detection.

Chapter 5

Other Applications of LARKs

Abstract – In Chapters 2 and 3, we have shown that LARKs are very useful for such applications as saliency detection and object/action detection that are core problems in visual recognition. In this chapter, we successfully apply LARKs to two more applications: 1) automatic change detection and 2) face verification. Comprehensive experiments demonstrate that the proposed methods yield state of the art results.

5.1 Automatic Change Detection

5.1.1 Introduction

The automatic analysis of subtle change between images of the same subject over time is a very important component in a large number of applications in diverse disciplines. Areas where such analyses are deployed include computer-aided diagnosis (CAD), video surveillance, and remote sensing, to mention just a few. In particular, change detection in medical diagnosis may be applicable to a broad range of diseases including cancers, Multiple Sclerosis, Alzheimer's and more. In general, a change detection method consists of three stages: 1) geometric registration of images, 2) intensity adjustments, and 3) image comparison to identify changes. We refer the interested reader to [157] and references therein for a good summary.

The generic problem of interest addressed in this section focuses on the third component and can be briefly described as follows: We are given a set of brain Magnetic Resonance Imaging (MRI) scans of the same subject acquired over time, and we are interested in identifying pixels which are "significantly" different between the two MRI scans. Even in the absence of registration errors, estimating diagnostically significant changes is still challenging due to such factors as signal nonuniformity or presence of noise. A variety of MRI artifacts also introduce a wide range of confounding factors, making standard change detection methods unreliable. In order to deal with these problems, multispectral MRI scans were employed for the purpose of lesion detection by many researchers. For example, there are at least five different MRI modalities including T1 weighed, inversion recovery (IR), proton-density-weighted (PD), T2-weighted, and fluid attenuation inversion recovery (FLAIR). For statistical change detection in multispectral MRI scans, Bosc et al. [158] used the Generalized Likelihood Ratio Test (GLRT) followed by nonlinear joint histogram normalization. However, their approach tends to fail when noise is non-stationary. Patriarche et al. [159] also used multispectral MRI scans to detect progression of brain tumors. Recently, Rousseau et al. [160] proposed an *a contrario* approach to detect Multiple Sclerosis in multispectral MRI scans. However, in the majority of clinical situations, only one type of anatomical MRI scan is collected, since the acquisition of multispectral MRI scans is more time consuming and costly. Longer scanning times are further not feasible in many patients due to the severity of their conditions.

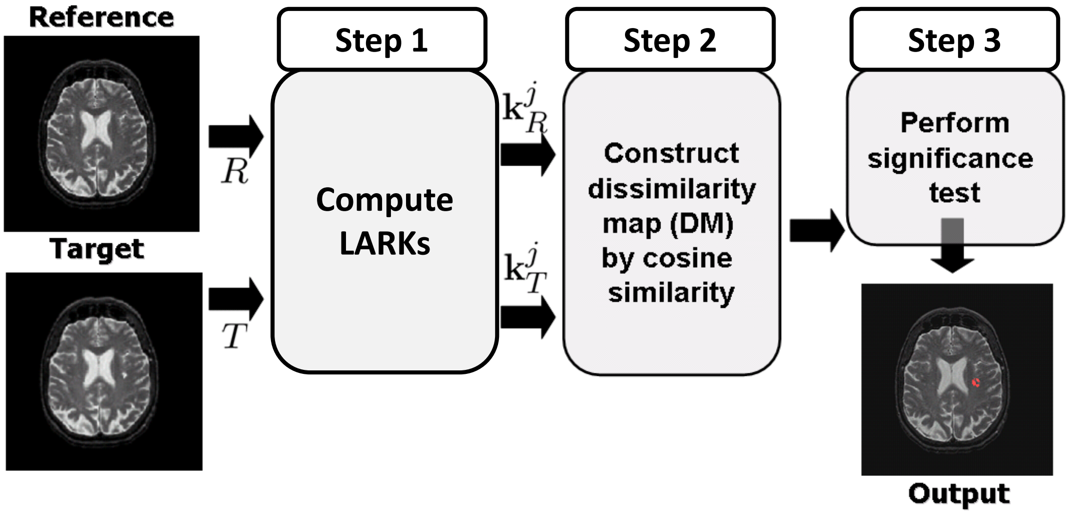


Figure 5.1: System overview of automatic change detection. (There are three steps.)

Very recently, Pecot et al. [161] introduced a change detection framework based on the so-called “patch-based Markov models" in image sequence analysis. Their method is to detect pixels with meaningful change for several frames by first constructing a difference image while our method directly computes LARKs from the reference image and the target image. The proposed method has an advantage over their method in that the calculation of LARKs is stable even in the presence of uncertainty in the data and is not sensitive to relatively large variations in illumination as described in Chapter 1. To summarize the operation of the overall algorithm, given the reference image and the target image, we first calculate LARKs from both the reference image and the registered target image at all pixel locations. Comparison between LARKs computed from two images is carried out using the vector cosine similarity measure. This step produces a “dissimilarity map" showing the likelihood of dissimilarity between the reference and target images. The final output is given after a significance test. (See Fig.

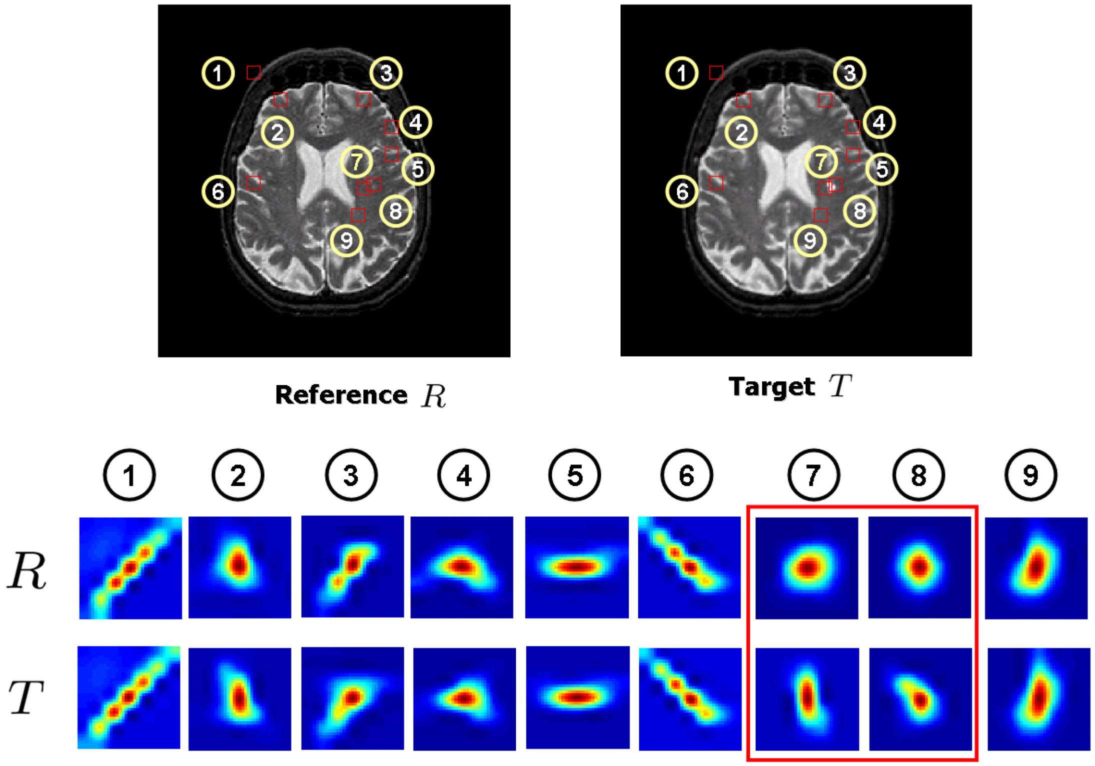


Figure 5.2: Examples of LARK in various regions. Note that LARKs computed from various regions look alike except for regions 7 and 8 where small lesions exist in the target.

5.1 for a graphical overview.)

In the next section, we provide further details about the various steps outlined above. In Section 5.1.3, we demonstrate the performance of the system with some experimental results.

5.1.2 Technical Details

Assume that we are given a target MRI scan T and that we have a reference MRI scan R. The first step in the proposed algorithm is to calculate the LARKs K_i measuring the relationship between a center pixel and its neighboring pixels, at each pixel from both R and T. Fig. **5.2** shows some examples of LARK in various regions of both the reference and the target. Note that LARKs computed from various regions in both reference and target look essentially identical except for regions 7 and 8 where small lesions exist.

At each pixel $\mathbf{x}_i$, we arrive at an array of P numbers by column-stacking (rasterizing) K_i as $\mathbf{k}_I^i (I \in \{R, T\})$ as done in Chapters 2 and 3. The next step in the algorithm is the measurement of a "distance" between the computed features, $\mathbf{k}_R^i$ and $\mathbf{k}_T^i$. As we alluded to earlier in Chapters 2 and 3, correlation based metrics perform better than the conventional Euclidean and Mahalanobis distances for classification and learning tasks. We employ vector cosine similarity as a similarity measure as follows:

$$\rho(\mathbf{k}_R^i, \mathbf{k}_T^i) = < \frac{\mathbf{k}_R^i}{\|\mathbf{k}_T^i\|}, \frac{\mathbf{k}_R^i}{\|\mathbf{k}_T^i\|} > = \frac{\mathbf{k}_R^{i\,\prime} \mathbf{k}_T^i}{\|\mathbf{k}_R^i\| \|\mathbf{k}_T^i\|} = \cos\theta_i, \tag{5.1}$$

where $\cos\theta_i \in [-1, 1]$. The cosine similarity measure therefore focuses only on the angle (phase) information while discarding the scale information. As for the final test statistic comprising the values in the dissimilarity map, we use the *proportion* of "residual" variance $(1 - \rho_i^2)$ to the shared variance ρ_i^2, as similarly done in Chapter 3. More specifically, the test statistic at each point in the image is computed. And the dissimilarity map (DM) is generated at each point as follows:

$$\text{DM} : f(\rho_i) = \frac{1 - \rho_i^2}{\rho_i^2}. \tag{5.2}$$

From a quantitative point of view, we note that $f(\rho_i)$ is essentially the inverse of the Lawley-Hotelling trace statistic [127], which is used as an efficient test statistic for detecting correlation between two data sets.

In order to detect salient and significant changes using the DM, we need a threshold τ. If we have a basic knowledge of the underlying distribution of $f(\rho_i)$, then

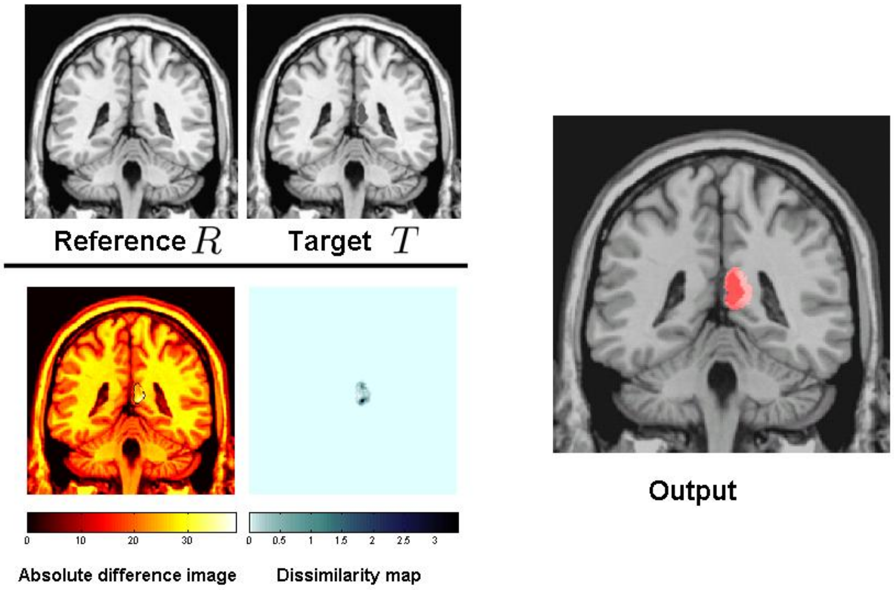


Figure 5.3: Coronal view: detected lesion on the simulated image. We used the parameters $P = 25, h = 1.0, \tau = 0.99$. Note that absolute difference image can not identify lesions at all while the proposed method detected simulated lesions stably. Note that images are better illustrated in color.

we can make predictions about how this particular statistic will behave, and thus it is relatively easy to choose a threshold which will indicate whether the pair of features from the two images are sufficiently dissimilar. But, in practice, we do not have a very good way to model the distribution of $f(\rho_i)$. Therefore, instead of assuming a type of underlying distribution, we employ the idea of nonparametric testing[1]. We compute an empirical PDF from the values of $f(\rho_i)$ across the image and we set τ so as to achieve, for instance, a 99 % confidence level in deciding whether a given value is in the extreme (right) tail of the distribution. This approach is based on the assumption

[1] Namely, we control the false discovery rate (FDR) [132]. We refer the reader to Appendix B for more details.

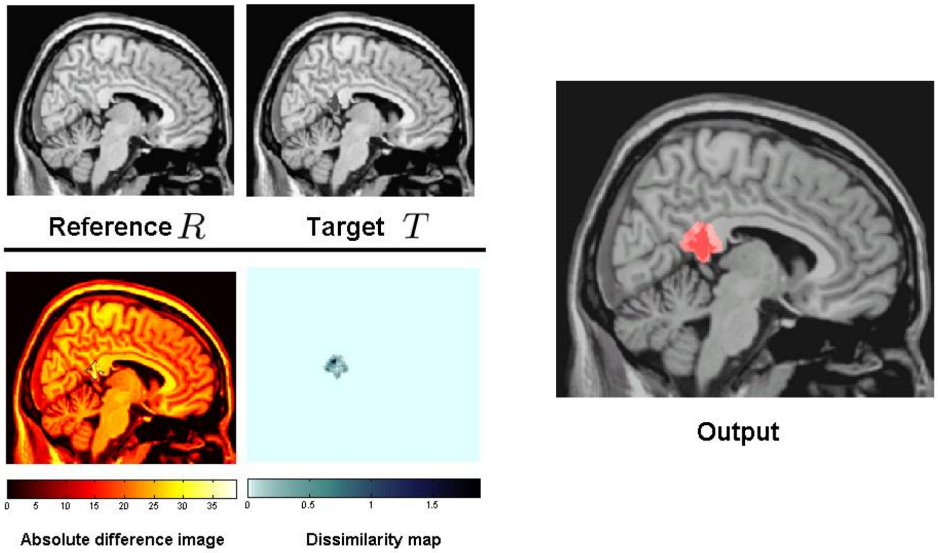


Figure 5.4: Sagittal view. We used the same parameters $P = 25, h = 1.0, \tau = 0.99$.

that in the target image, most of pixels are not involved with significant change, and therefore, the few outliers will result in values which are in the tail of the distributions of $f(\rho_i)$.

5.1.3 Experimental Results

In order to validate the proposed method quantitatively, we simulated lesions in normal brain MRI slices (sagittal, coronal, and axial views). These simulated lesions were generated using a 3-D region of interests (ROI) creation tool provided in MRIcro[2]. Exact sizes and locations of irregular shapes of simulated lesions were stored and treated as the ground truth. In order to cover a variety of lesions, we constructed

[2] http://www.sph.sc.edu/comd/rorden/mricro.html

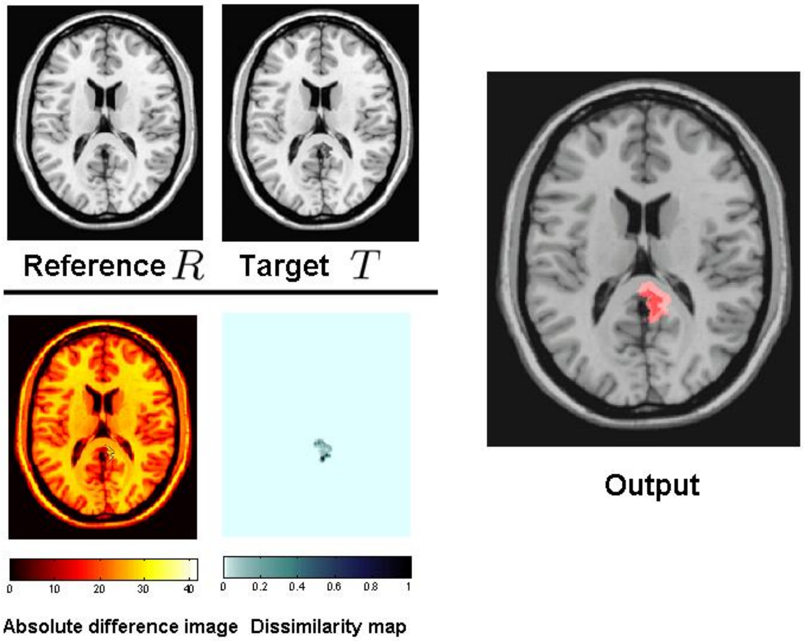


Figure 5.5: Axial view. We used the same parameters $P = 25, h = 1.0, \tau = 0.99$.

a total of 168 (14 ROIs in different sizes × 3 different views × 4 intensity reduction of $0\%, 20\%, 40\%$, and 60%, respectively) target slices by following the procedure as in [162]. Besides, we further made the intensity range of targets (T) different from the reference (R). We compute LARKs of size 5×5 as descriptors from both R and T. As a consequence, each pixel in R and T yields a 25-dimensional local descriptor respectively[3]. By performing significance test on the resulting dissimilarity map with confidence level $\tau = 0.99$, we detected regions with anomalous and statistically significant changes. Figs. **5.3**, **5.4**, and **5.5** illustrate three examples of the detected results at the simulated lesions with 20 % intensity reduction (i.e., degree of lesion transparency).

[3] Performance of our change detection system is not particularly sensitive to the choice of LARK size because $\mathbf{C}_l$ plays a role in automatically determining the shape and size of kernels.

As an overall measure of performance[4], we were able to achieve sensitivity= 0.877, specificity=0.998, and similarity index (SI)=0.879.

Rousseau et al. [160] evaluated their method on simulated lesion images and reported their SI value around 0.75. Shen et al. [162] tested their lesion detection method based on segmentation to lesions generated from MRIcro [5] and reported their SI values on the simulated target slices with $20\%, 40\%$, and 60% intensity reduction as 0.867, 0.879, and 0.724 respectively. Our method which obtained an average of 0.879 for SI performs comparably with the method in [162] and outperforms the method in [160] even though [160] used multispectral MR images[6].

5.2 Face Verification

5.2.1 Introduction

Face recognition has been of great research interest [163, 164, 165, 166, 167, 168, 169, 133, 170] in recent years. Face recognition is mainly divided into two tasks: 1) face identification and 2) face verification. The goal of face identification is to place a given test face into one of several predefined sets in a database, whereas face verification is to determine if two face images belong to the same person. In general, the face verification task is more difficult than face identification because a global threshold is required to make a decision. There are also many papers on face detection such

[4] sensitivity = $\frac{A_{gt} \bigcap A_{dt}}{A_{gt}}$, specificity = $\frac{(I-A_{gt}) \bigcap (I-A_{dt})}{I-A_{gt}}$, SI = $2 \times \frac{A_{gt} \bigcap A_{dt}}{A_{gt} \bigcup A_{dt}}$, where A_{gt} represents the ground truth, which is regions with true lesions (i.e., simulated lesions in this chapter). A_{dt} represents detected lesions. I refers to the whole image.

[5] `http://www.sph.sc.edu/comd/rorden/mricro.html`

[6] As pointed out in [160], it is difficult to provide a fair comparison among automatic change detection algorithms due to the fact that there is no gold standard and codes of state-of-the art methods are not publicly unavailable.

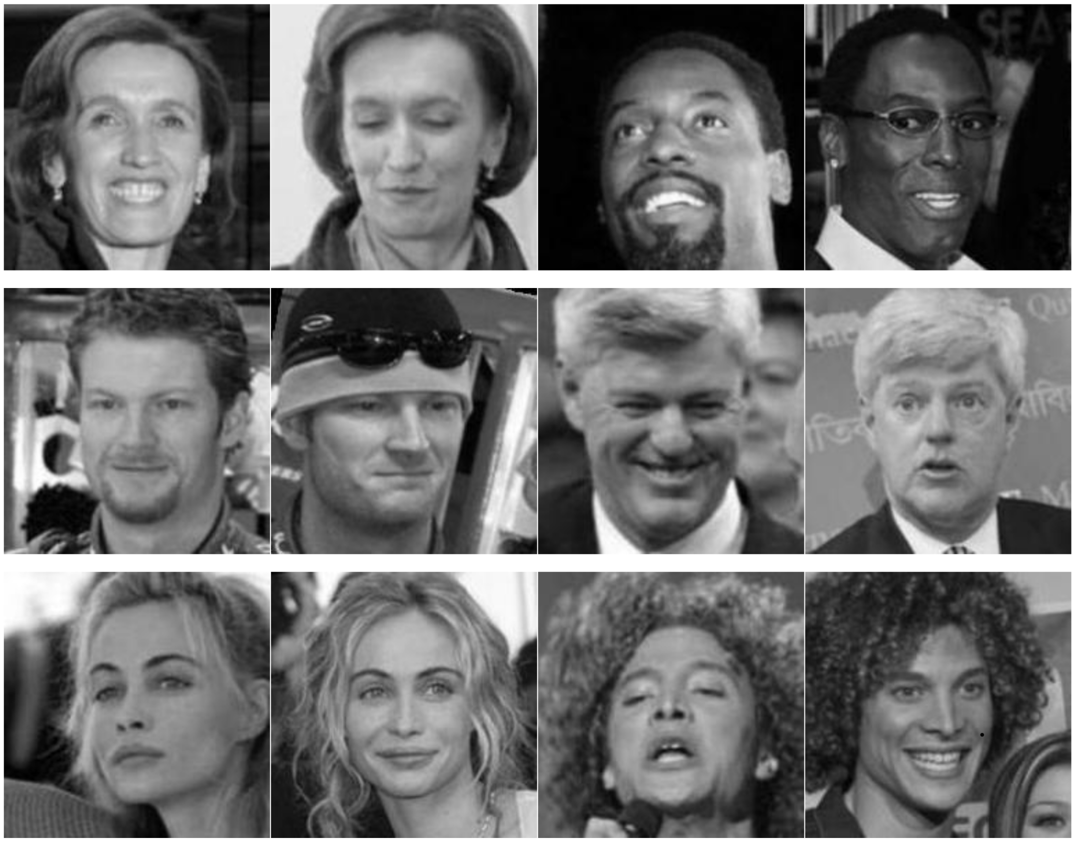


Figure 5.6: Example faces from Labeled Faces in the Wild (LFW) [25]: faces belonging to the same person may look very different from each other due to the large variation caused by different poses, light conditions, facial expressions, and etc.

as [29, 17] and [171], which is considered as a pre-processing step for face recognition.

According to the face recognition grand challenge (FRGC) [172], face identification rates under well-constrained environments have been saturated (almost perfect with a small false alarm rate.) Nevertheless, face recognition in uncontrolled settings is still an open problem due to the large variations caused by different poses, lighting conditions, facial expression, occlusions, misalignments, etc. With the advent of a standard benchmark dataset “Labeled Faces in the Wild (LFW) [25]", the face verification problem in unconstrained settings has recently attracted much research effort [165, 173, 174, 133, 170]. This challenging dataset contains a collection of annotated faces captured from news articles, and exhibits all the variations mentioned above. There are three evaluation protocols for this dataset: 1) the image *un*restricted training setting, 2) the image restricted training setting, and 3) the unsupervised (no training) setting.

In this section, we address the face verification problem in uncontrolled environments (on LFW dataset and FRGC dataset). The main task is to decide whether the images of two faces belong to the same individual. Among three evaluation settings for LFW dataset, we focus on the last two (the unsupervised and the image restricted training) which are more realistic in practice. In Chapter 3, we have tackled the generic object detection problem by employing LARKs in conjunction with MCS measure. This combination has led to state of the art detection performance from a single query, and without any further training. In fact, face detection is in nature very similar to face verification in the sense that both are binary classification problems and require two major components: 1) a face representation and 2) a similarity measure. In this section, we provide some insights into how the face detection method in [49] can be extended to face verification.

Recently, face representations based on local image descriptors such as local binary pattern (LBP) and its variants [175, 165] and histogram of gradient descriptors (SIFT [18] and HOG [3]) have been proven to be effective for face verification. These descriptors encode local geometric structures by using either a quantized version of local gray level patterns or quantized codes of the image gradients. In this section, we propose to use LARKs which provide much more rich and detailed information than other local descriptors [3, 18, 11]. After reducing the dimension of LARK by performing PCA, we apply a logistic function to the result. The role of the logistic function here is to make LARK become more or less binarized by stretching values to extreme ends. We demonstrate that the use of MCS, combined with our feature representation results in the best performance in unsupervised settings of LFW benchmark.

For the image restricted setting, we employ one-shot similarity (OSS) measure [26] based on linear discriminative analysis (LDA). The OSS with the proposed feature representation achieves state of the art performance as a single descriptor and obtains results comparable with [26] when jointly used with other descriptors (with many fewer distances: 14 (ours) vs. 30 [26]). A block diagram of the proposed face verification system is given in Fig. **5.7**.

Related Work A texture descriptor called local binary patterns (LBP) [175] has been shown to be effective for face recognition. Ever since LBP was introduced, such variants of LBP as three-patch LBP (TPLBP), and four-patch LBP (FPLBP) have been proposed by Wolf et al. [165]. These descriptors were combined with the one-shot similarity (OSS) [173] measure motivated by the growing body of "One-Shot Learning" techniques [176]. Wolf et al. [173] also applied the OSS in the framework of support vector machine (SVM) [131] by modifying the OSS to a conditional positive definite kernel. The OSS was extended to two-shot similarity (TSS) in [177]. In [165, 173, 26], it has been further shown that combining multiple descriptors and multiple measures can boost the overall verification performance.

Guillaumin et al. [168] proposed two methods called 1) logistic discriminant metric learning and 2) marginalized k-nearest neighbor. They focused on finding a metric based on learning the Mahalanobis distance. Independently from [165, 173], they also showed that combination of descriptors and metrics improves upon using

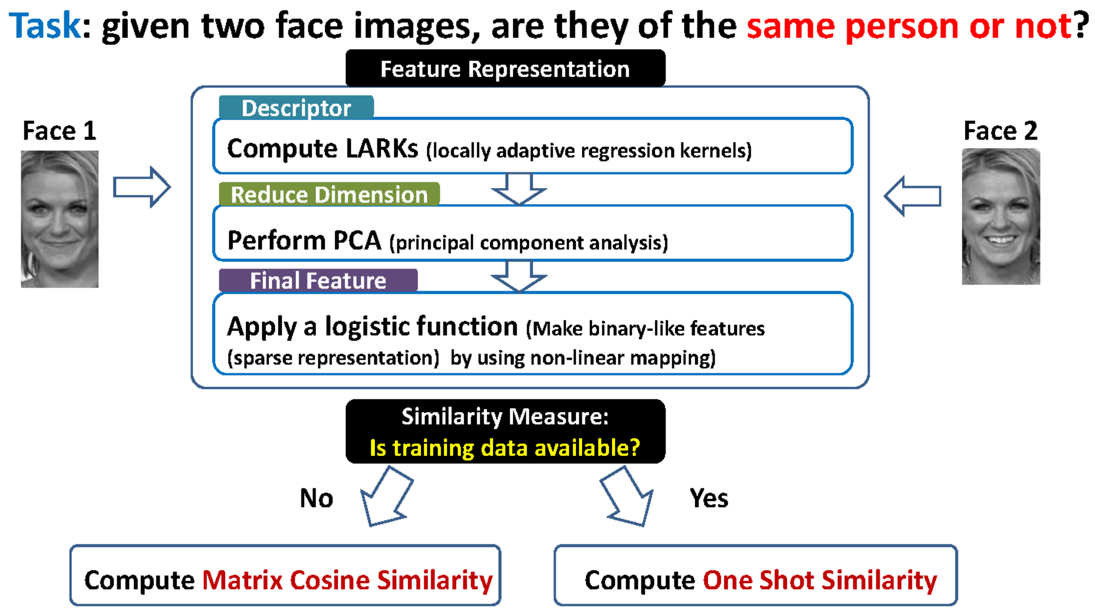

Task: given two face images, are they of the same person or not?
Feature Representation
Face 1
Face 2
Descriptor
Compute LARKs (locally adaptive regression kernels)
Reduce Dimension
Perform PCA (principal component analysis)
Final Feature
Apply a logistic function (Make binary-like features (sparse representation) by using non-linear mapping)
Similarity Measure:
Is training data available?
No
Yes
Compute Matrix Cosine Similarity
Compute One Shot Similarity

Figure 5.7: Block diagram of face verification system. The system mainly consists of two stages: feature representation and similarity measure.

only one metric and one descriptor.

Hua and Akbarzadeh [174] recently proposed an elastic and partial matching metric which robustly measures distance between two sets of descriptors by using a nonparametric significance test. In their work, they revealed that a simple difference of Gaussian (DoG) filtering on face images works better than the more often utilized photometric rectification methods (such as self-quotient image [178]) in handling lighting variations.

Motivated by the observation that humans perform very well on the LFW dataset, Kumar et al. [133] proposed two classifiers called “attribute” and “simile” classifiers for face verification. While the attribute classifiers are binary classifiers trained to recognize the presence or absence of visual aspects such as gender, race, age, and hair color, simile classifiers are binary classifiers trained to recognize the similarity of faces. This method achieved state of the art performance (85.29% verification rates on the LFW dataset), but requires a combination of many (more than 70) classifiers.

Distinguished from aforementioned works, Cao et al. [170] introduced a learning-based encoding method based on unsupervised learning techniques such as k-means, kd-tree, and random projection tree [179]. In [170], they focused on learning uniform descriptors from a collection of histogram-based low-level descriptors. They claimed that the uniformity of features is important when L_2 or L_1 distances are used as similarity metrics. By using multiple learning-based descriptors from nine fiducial areas and pose-adaptive matching, they have achieved a verification rate of 84.45% on the LFW dataset.

While all the works [165, 173, 133, 170, 174] above evaluated their methods in the image restricted training setting, Ruiz-del-solar et al. [180] carried out a comprehensive study of existing image matching methods such as LBP matching, Gabor-Borda count, and SIFT matching in the unsupervised setting (without any training). They empirically compared these methods by changing the image crop size, parameter settings of descriptors, and image block size.

In our earlier generic object detection work described in Chapter 3, after employing principal component analysis (PCA), LARKs were transformed to compact feature vectors. MCS between two matrices composed of resulting sets of feature vectors, $\mathbf{F}^{(i)} = [\mathbf{f}_1, \cdots, \mathbf{f}_n]^{(i)}, \mathbf{F}^{(j)} = [\mathbf{f}_1, \cdots, \mathbf{f}_n]^{(j)}$ was defined as:

$$\begin{aligned} \rho(\mathbf{F}^{(i)}, \mathbf{F}^{(j)}) &= \sum_{\ell=1}^{n} \frac{\mathbf{f}_\ell^{(i)\top} \mathbf{f}_\ell^{(j)}}{\|\mathbf{F}^{(i)}\| \|\mathbf{F}^{(j)}\|}, \\ &= \sum_{\ell=1}^{n} \underbrace{\rho(\mathbf{f}_\ell^{(i)}, \mathbf{f}_\ell^{(j)})}_{\text{vector cosine}} \underbrace{\frac{\|\mathbf{f}_\ell^{(i)}\| \|\mathbf{f}_\ell^{(j)}\|}{\|\mathbf{F}^{(i)}\| \|\mathbf{F}^{(j)}\|}}_{\text{relative weights}}, \end{aligned} \tag{5.3}$$

where n is the number of features. This MCS is a weighted sum of the vector cosine similarities of local features $\mathbf{f}_\ell$. Robustness to local deformation, presence of noise, and occlusion is implicitly attained by the relative weights which play a key role in finding interest points in the face image (see Fig. **5.9**). This is particularly useful for face detection where the goal is to separate faces from the background in a given image. In order for this framework to be extended to face verification, the role of the relative weights should be adjusted accordingly, as we will describe in Section 5.2.2.

Our contributions to the face verification task are three-fold. First, we employ LARK which robustly captures local geometric structures. This LARK in conjunction with PCA provides a very compact face representation, desirable for real-time applications. Second, we extend face detection to face verification by introducing a binary-like face representation. The proposed representation along with both MCS and OSS achieves the best performance on the unsupervised and image restricted settings as a single descriptor. Lastly, we show that a very simple idea (namely the addition of a mirror image of a query) remarkably boosts the overall performance[7].

5.2.2 Technical Details

As we alluded to earlier in Chapter 3, densely computed LARKs from images are highly informative, but taken together are be over-complete (redundant). Therefore, we derive features by applying dimensionality reduction (namely PCA) to **K**, in order to retain only the salient characteristics of the LARKs. Applying PCA to **K** we can retain the top d principal components which form the columns of a matrix $\mathbf{V} = [\mathbf{v}_1, \cdots, \mathbf{v}_d] \in \mathbb{R}^{P \times d}$. Since we focus on finding stable (but less specific) bases which can represent basic characteristics of general faces, we used LARKs collected from 120

[7] This is a computational strategy that takes advantage of the fact that objects in our world frequently present mirror-symmetric views [181]

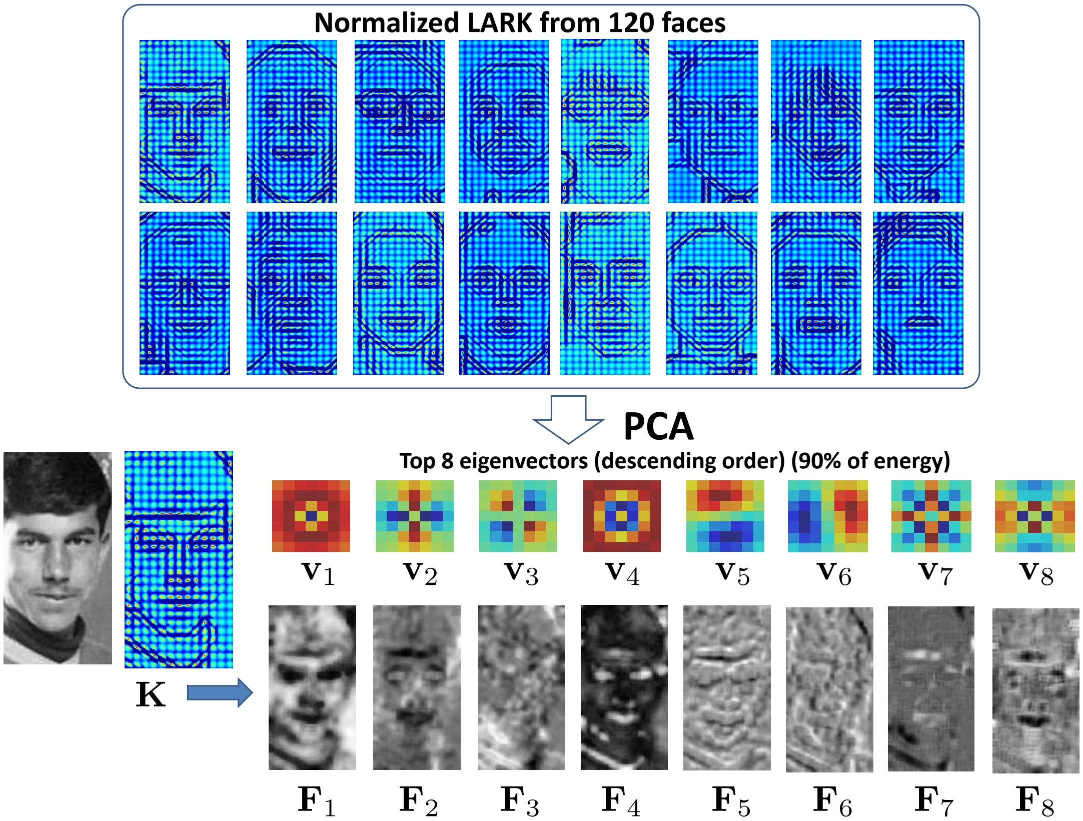


Figure 5.8: A collection of normalized LARK descriptors are very informative, but contains a redundant information as well. PCA is applied to not only reduce the dimensionality of LARK, but also to retain only the salient characteristics of the LARKs. After applying PCA to LARKs of size 7 × 7 collected from 120 face images, we obtained 8 eigenvectors corresponding to the top 8 eigenvalues which preserve 90 % of energy.

face images to learn an overall, *fixed*, PCA basis for faces whereas the goal of the object detection system described in Chapter 3 was to find similar images to a particular query, thus PCA bases were learned from only one image. (see Fig. **5.8**.)

For the fixed PCA basis, d is selected to be a small integer such as 7 or 8 so that 80 to 90% of the information in the LARKs would be retained. (i.e., $\frac{\sum_{i=1}^{d}\lambda_i}{\sum_{i=1}^{P}\lambda_i} \geq 0.8$) where λ_i are the eigenvalues.) Next, the lower dimensional features are computed by

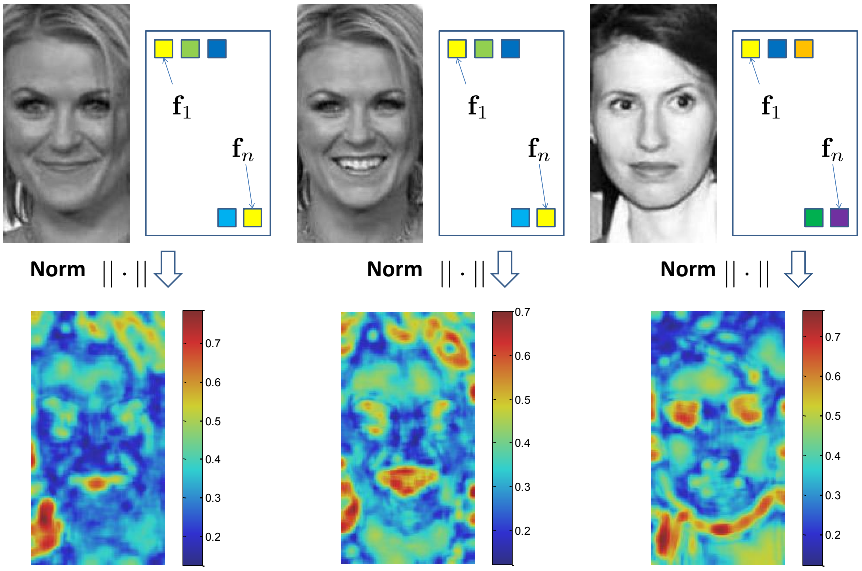


Figure 5.9: $||\mathbf{f}||$: magnitude of **f** reveals interest points in the face images.

projecting **K** onto **V** as follow:

$$\mathbf{F} = [\mathbf{f}_1, \cdots, \mathbf{f}_n] = \mathbf{V}^\top \mathbf{K} \in \mathbb{R}^{d \times n}. \tag{5.4}$$

The use of the PCA here is not critical in the sense that any unsupervised subspace learning method such as Kernel PCA, LLE [123], LPP [124] CDA [93], CEA [92], kd-tree, and random projection tree [179] can be used.

Denoting **F** as the feature representation, we measure the similarity score by MCS between two images. As shown in Fig. **5.9**, $||\mathbf{f}||$ reveals interest points (e.g., eyes, mouth, hairs, jaws, etc.) in the face images. If the task is to detect faces from background, focusing on interest points would be helpful for robustness to occlusion, misalignment, pose, and local deformation. Even though these properties are also important for face verification, relying too much on these relative weights tends to weaken the ability to discriminate *between* two faces. In order to alleviate this problem, we

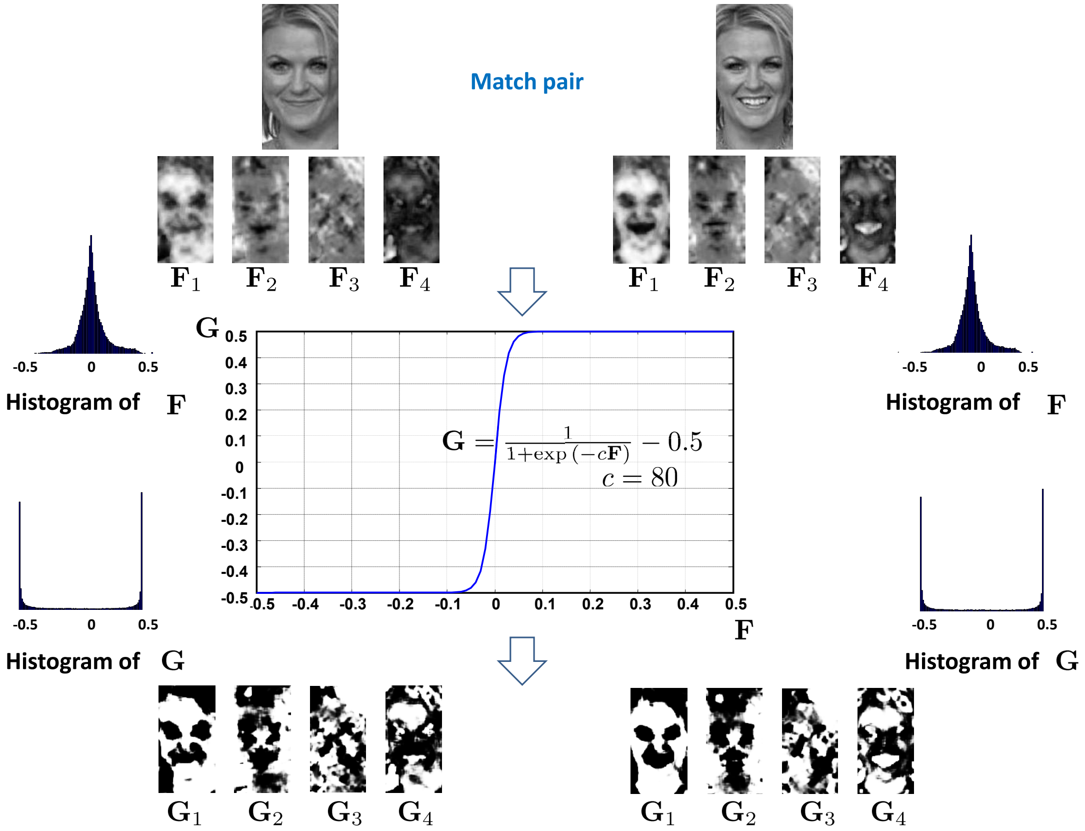

Match pair
$\mathbf{F}_1$ $\mathbf{F}_2$ $\mathbf{F}_3$ $\mathbf{F}_4$
$\mathbf{F}_1$ $\mathbf{F}_2$ $\mathbf{F}_3$ $\mathbf{F}_4$
-0.5 0 0.5
Histogram of $\mathbf{F}$
-0.5 0 0.5
Histogram of $\mathbf{F}$
$\mathbf{G}$
0.5 0.4 0.3 0.2 0.1 0 -0.1 -0.2 -0.3 -0.4 -0.5
-0.5 -0.4 -0.3 -0.2 -0.1 0 0.1 0.2 0.3 0.4 0.5
$\mathbf{F}$
$\mathbf{G} = \frac{1}{1+\exp(-c\mathbf{F})} - 0.5$
$c = 80$
-0.5 0 0.5
Histogram of $\mathbf{G}$
-0.5 0 0.5
Histogram of $\mathbf{G}$
$\mathbf{G}_1$ $\mathbf{G}_2$ $\mathbf{G}_3$ $\mathbf{G}_4$
$\mathbf{G}_1$ $\mathbf{G}_2$ $\mathbf{G}_3$ $\mathbf{G}_4$

Figure 5.10: We employ a logistic function $\frac{1}{1+\exp(-c\mathbf{F})} - 0.5$ in order to make binary-like representation.

need to make these weights relatively spread out (somewhat uniform). This can be realized by applying a nonlinear mapping to features **F**. Specifically, we apply a logistic function element-by-element[8] to the feature matrices **F** as follows:

$$\mathbf{G} = \frac{1}{1+\exp(-c\mathbf{F})} - 0.5. \tag{5.5}$$

The role of this logistic function is to make the features (**F**) become more or less binary-like by stretching values to extreme ends (-0.5,0.5). As shown in Fig. **5.10**, histograms of **G** have two peaks around (-0.5,0.5) whereas histograms of **F** are centered around

[8] This nonlinear mapping plays a role in making features sparse [57].

0. After applying the logistic function, the dominance of large relative weights in **F** is removed and discriminative power of **G** compared to **F** is increased as shown in Fig. **5.11**. This idea is somewhat related to [170] in which the uniformity of histogram-based feature values is considered important, and [26] in which the Hellinger distance outperforms L_2 distance. That is, ensuring that feature values stay in a small range enhances the discriminative power of any distance measure.

As defined in (5.3), the MCS between two matrices $\mathbf{G}^{(i)} = [\mathbf{g}_1, \cdots, \mathbf{g}_n]^{(i)}, \mathbf{G}^{(j)} = [\mathbf{g}_1, \cdots, \mathbf{g}_n]^{(j)}$ is as follows:

$$\rho(\mathbf{G}^{(i)}, \mathbf{G}^{(j)}) = \sum_{\ell=1}^{n} \underbrace{\rho(\mathbf{g}_\ell^{(i)}, \mathbf{g}_\ell^{(j)})}_{\text{cosine similarity}} \underbrace{\frac{\|\mathbf{g}_\ell^{(i)}\| \|\mathbf{g}_\ell^{(j)}\|}{\|\mathbf{G}^{(i)}\|_F \|\mathbf{G}^{(j)}\|_F}}_{\text{relative weights}}. \tag{5.6}$$

This $\rho(\mathbf{G}^{(i)}, \mathbf{G}^{(j)})$ can be efficiently implemented by column-stacking the matrices $\mathbf{G}^{(i)}, \mathbf{G}^{(j)}$ and simply computing the cosine similarity between two long column vectors as follows:

$$\rho(i, j) = \rho(\text{colstack}(\mathbf{G}^{(i)}), \text{colstack}(\mathbf{G}^{(j)})) \in [-1, 1], \tag{5.7}$$

where colstack$(\cdot)$ means an operator which column-stacks (rasterizes) a matrix.

The MCS measure computed on **G** provides robustness to many small deformations, but tends to fail when there are large variations due to out-of-plane rotation, which is common in the LFW dataset. To deal with off-frontal (out-of-plane rotated) faces, we use a very simple (but novel) idea of additionally using mirror-reflect version of **G** (see Fig. **5.12**.) We take a maximum value between the two resulting MCS scores as a final MCS score[9].

$$\text{MCS}(i, j) = \max(\rho(\mathbf{G}^{(i)}, \mathbf{G}^{(j)}), \rho(\overline{\mathbf{G}}^{(i)}, \mathbf{G}^{(j)})), \tag{5.8}$$

[9] Interestingly, we found that this simple idea has not been utilized before, but remarkably boosts the overall performance.

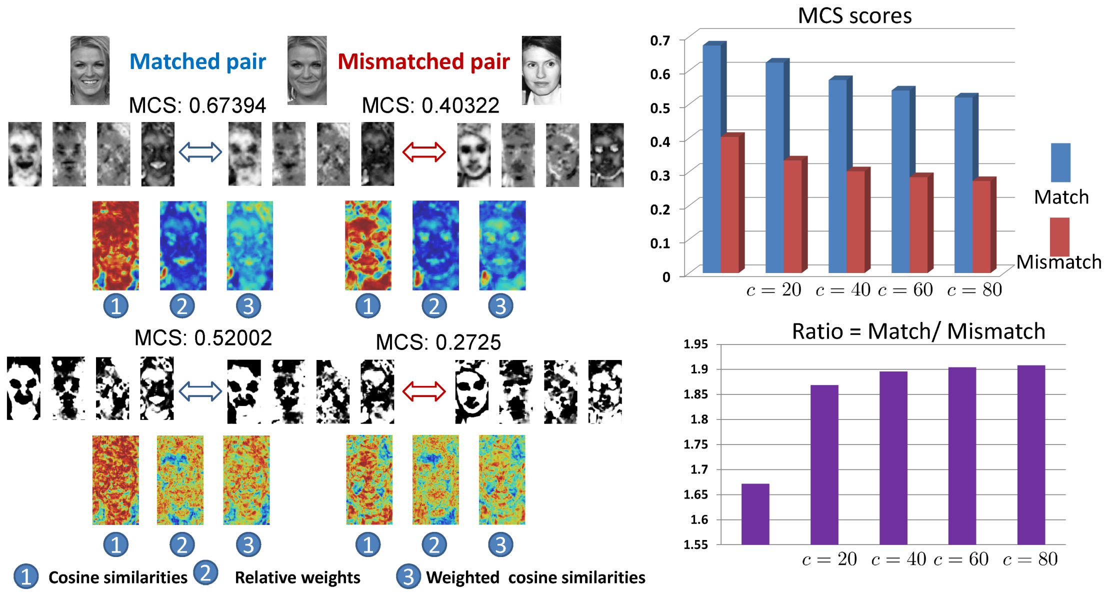


Figure 5.11: MCS scores for a matched pair and a mismatched pair. Top-Left: MCS by using **F** 1) $\rho(\mathbf{f}_\ell^{(i)},\mathbf{f}_\ell^{(j)})$, 2) $\frac{\|\mathbf{f}_\ell^{(i)}\|\|\mathbf{f}_\ell^{(j)}\|}{\|\mathbf{F}^{(i)}\|\|\mathbf{F}^{(j)}\|}$, 3) $\rho(\mathbf{f}_\ell^{(i)},\mathbf{f}_\ell^{(j)})\frac{\|\mathbf{f}_\ell^{(i)}\|\|\mathbf{f}_\ell^{(j)}\|}{\|\mathbf{F}^{(i)}\|\|\mathbf{F}^{(j)}\|}$, Bottom-Left: MCS by using **G**, 1) $\rho(\mathbf{g}_\ell^{(i)},\mathbf{g}_\ell^{(j)})$, 2) $\frac{\|\mathbf{g}_\ell^{(i)}\|\|\mathbf{g}_\ell^{(j)}\|}{\|\mathbf{G}^{(i)}\|\|\mathbf{G}^{(j)}\|}$, 3) $\rho(\mathbf{g}_\ell^{(i)},\mathbf{g}_\ell^{(j)})\frac{\|\mathbf{g}_\ell^{(i)}\|\|\mathbf{g}_\ell^{(j)}\|}{\|\mathbf{G}^{(i)}\|\|\mathbf{G}^{(j)}\|}$. The relative weights of **G** are more useful than those of **F** because they are not just focusing on texture region. Right: The effect of coefficient c in the logistic function. The ratio of MCS scores of match pair to those of mismatch pair is maximized at $c = 80$. Also see Fig. **5.14**.

where $\overline{\mathbf{G}}$ is a mirror-reflect version[10] of **G**.

5.2.3 Experimental Results

Up to now, we have described the proposed face representation for the face verification task. In this section, we demonstrate the performance of the proposed method with comprehensive experiments on the challenging labeled faces in the wild (LFW) [164] dataset.

[10] $\overline{\mathbf{G}}$ is not computed from mirror-reflected face images, but is the reflected version of **G**. This helps us compute LARK features just once.

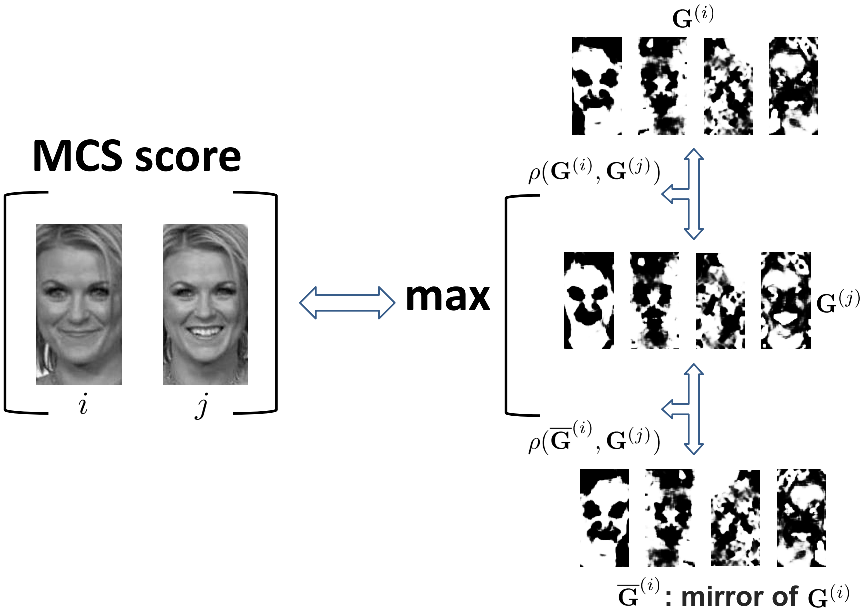

$\mathbf{G}^{(i)}$
MCS score
$\rho(\mathbf{G}^{(i)}, \mathbf{G}^{(j)})$
max
$\mathbf{G}^{(j)}$
i
j
$\rho(\overline{\mathbf{G}}^{(i)}, \mathbf{G}^{(j)})$
$\overline{\mathbf{G}}^{(i)}$: mirror of $\mathbf{G}^{(i)}$

Figure 5.12: MCS score between face images (i) and (j) is a maximum score of two MCS scores computed between $\mathbf{G}^{(j)}$ and $\mathbf{G}^{(i)}$, and $\overline{\mathbf{G}}^{(i)}$: mirror-reflect version of $\mathbf{G}^{(i)}$ respectively.

5.2.3.1 Labeled Faces in the Wild (LFW) Dataset

The LFW database [164] consists of 13,233 face images of 5,749 different persons, obtained from news articles on the web. The images in the LFW database have a very large degree of variability in the facial expression, age, race, pose, occlusion, and illumination conditions (see Fig. **5.6**). The task is to determine if a pair of face images belong to the same individual or not. We test on the “View 2" which includes 3,000 matched pairs and 3,000 mismatched pairs. The data are equally divided into 10 sets. The final verification performance is reported as the mean recognition rate and standard error about the mean over 10-fold cross-validation.

We also provide the receiver operating characteristic (ROC) curves for the sake of completeness. The true positive rate (TPR), the false positive rate (FPR), and the verification rate (VR) are defined as follows:

$$\text{TPR} = \frac{\sharp \text{ correctly accepted matched pairs}}{\sharp \text{ total matched pairs}}, \tag{5.9}$$

$$\text{FPR} = \frac{\sharp \text{ incorrectly accepted mismatched pairs}}{\sharp \text{ total mismatch pairs}}, \tag{5.10}$$

$$\text{VR} = \frac{\sharp \text{ correctly classified pairs}}{\sharp \text{ total pairs}}. \tag{5.11}$$

We compute the TPR and FPR by changing the threshold values to draw the ROC curves and report the best VR across the ROC curves.

As mentioned earlier, there are three evaluation settings : 1) the image unrestricted training setting, 2) the image restricted training setting, and 3) the unsupervised setting. The unsupervised setting is the most difficult one among these because there are no training examples available. On the other hand, the other two settings allow us to utilize available image pair information in the training set. The image unrestricted setting further provides the identity information of each pair. The official LFW website[11] provides all the state of the art results on the three settings.

In this section, we only focus on the two most challenging settings: the unsupervised setting and the image restricted setting, because these scenarios are more realistic in practice. We use the aligned version of the LFW dataset available from the website[12]. The images were cropped to a size of 184 × 97 so that images include more or less faces only[13].

Unsupervised Setting In this section, we examine the efficacy of the proposed method in the unsupervised setting where we do not use any training examples. We compute

[11] http://vis-www.cs.umass.edu/lfw/results.html

[12] http://www.openu.ac.il/home/hassner/data/lfwa/

[13] A slight difference in crop size makes no difference for the overall performance. For example, consider 184 × 97 vs. 186 × 94. More detailed discussion about the choice of image crop size in LFW dataset can be found in [180].

LARKs (K) of size 7 × 7 densely from each face image. We end up with features **G** by reducing dimensionality from 49 to 8 and employing a logistic function with $c = 80$ (performance converges as we increase c value (see Fig. **5.11**)). The MCS score described in Fig. **5.12** is computed from each of 6,000 pairs. [180] conducted comprehensive experiments to find the best combination among various state of the art descriptors (i.e., LBP, PCALBP, Gabor jets, and SIFT) and similarity measures (i.e, histogram intersection, Chi-square, Borda count, and Euclidean distance). They reported that LBP with Chi-square achieves the best performance (69.45% VR). We computed TPR and FPR by changing the threshold to draw a ROC curve. The proposed method achieves (72.23% VR) and outperforms previous state of the art methods reported in the LFW website as shown in Fig. **5.13**. Even before employing the logistic function, the proposed approach outperforms state of the art methods. We can see in Fig. **5.14** that the higher parameter c is, the better performance is. It clearly shows that binary-like features **G** are superior to the direct use of **F** for face verification task. We observe that there is no further improvement above $c = 80$. We also analyzed the effect of using mirror-reflect version of **G**. This simple idea[14] led to a nontrivial improvement (1 ~ 2%) which is more pronounced in the range of smaller c.

Image Restricted Setting In this section, we deal with the case where there are training image pairs available. More specifically, in the training set, it is known whether an image pair belongs to the same person or not, while identity information is not used at all. We employ one-shot similarity (OSS) [26] based on linear discriminative analysis (LDA). We briefly review OSS and explain how we use the proposed feature

[14] Faces are generally not symmetric. Face asymmetry has been studied in [182]. Their results supported previous work in Psychology that facial asymmetry contributes to human identification. This also justifies the idea of mirror-reflection that improves overall performance.

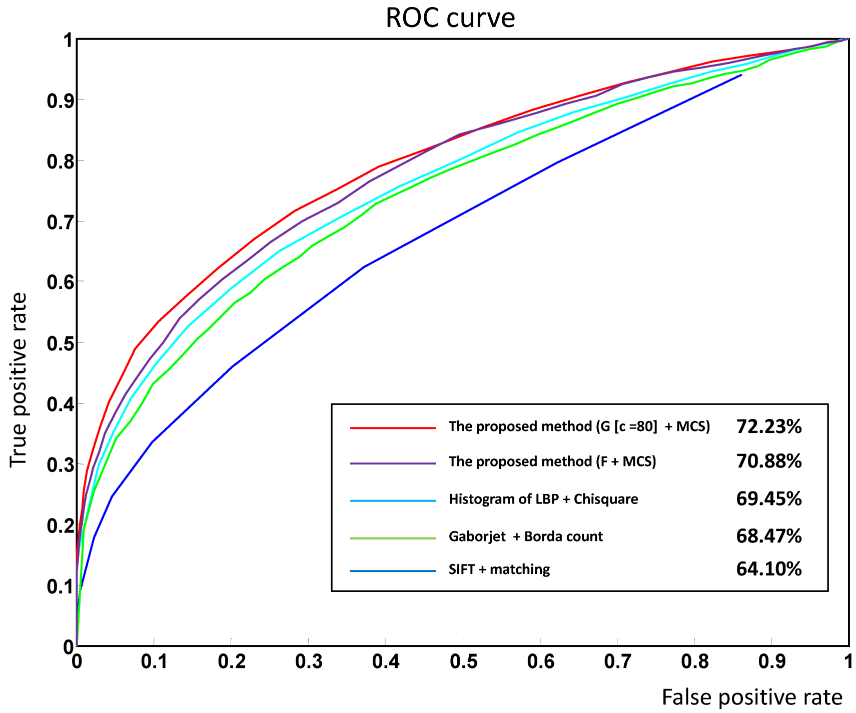


Figure 5.13: ROC curves and VR computed from 10 folds of View 2 (the unsupervised setting). The proposed method performs the best among all.

representation in the OSS framework.

5.2.3.2 One Shot Similarity (OSS)

The key idea behind the OSS is to use negative examples. Suppose that there are two classes (positive (+) and negative (-)) and we have many negative examples while there is only one positive example. In binary LDA case, the goal is to find out a projection direction **w** which maximizes the Raleigh quotient:

$$\widehat{\mathbf{w}} = \arg\max_{\mathbf{w}} \frac{\mathbf{w}^\top S_B \mathbf{w}}{\mathbf{w}^\top S_W \mathbf{w}}, \tag{5.12}$$

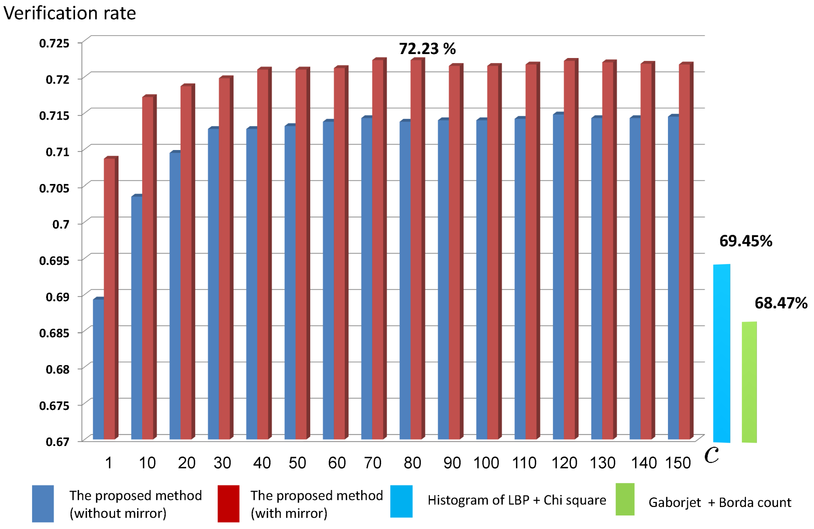


Figure 5.14: Verification rates as a function of the parameter c in the proposed representation with comparison to state of the art methods in the unsupervised setting of the LFW dataset (view 2). Adding mirror-reflect improves the overall performance ($1 \sim 2\%$).

where S_B is the "between-class scatter matrix" and S_W is the "within-class scatter matrix." The definitions of the scatter matrices are as follows:

$$\begin{aligned} S_B &= (\mathbf{m}_+ - \mathbf{m}_-)(\mathbf{m}_+ - \mathbf{m}_-)^\top, \\ S_W &= S_+ + S_-, \\ S_k &= \sum_k (\mathbf{G}_k - \mathbf{m}_k)(\mathbf{G}_k - \mathbf{m}_k)^\top, \end{aligned}$$

where $k \in \{+, -\}$ and $\mathbf{m}_+, \mathbf{m}_-$ are mean sample vectors of the positive class and the negative class respectively. It can be shown [131] that this maximization leads to a generalized eigenvalue problem:

$$S_B \mathbf{w} = \lambda S_W \mathbf{w}. \tag{5.13}$$

This problem can be solved very easily because $S_B\mathbf{w}$ is always in the direction of $(\mathbf{m}_+ - \mathbf{m}_-)$. Since there is only one positive example, that is, $S_+ = 0$, the within-class scatter

Table 5.1: Test set, Negative set, and Training sets in 10-fold validation (view 2)

Fold	1	2	3	4	5	6	7	8	9	10
Test set	3~10	1, 4~10	1,2, 5~10	1~3, 6~10	1~4, 7~10	1~5, 8~10	1~6, 9~10	1~7, 10	1~8	2~9
Negative set	2	3	4	5	6	7	8	9	10	1
Train set	1	2	3	4	5	6	7	8	9	10

matrix boils down to S_- which can be precalculated. $\mathbf{w}$ can be computed as follow:

$$\mathbf{w} \propto S_W^{-1}(\mathbf{m}_+ - \mathbf{m}_-) = S_-^{-1}(\mathbf{m}_+ - \mathbf{m}_-). \tag{5.14}$$

The benefit of using LDA is that the training step consists mainly of a vector difference followed by a matrix multiplication.

Fig. **5.15** describes how an OSS score between a pair of images (i) and (j) is computed as given in [26]. As negative examples, we used 1,200 faces which are not included in the test image pairs. First, by learning model 1 between (j) and negative set (N) and classifying (i) based on model 1, we obtain a score 1. Then we switch the role of (i) and (j) and learn model 2 and classify (j) on model 2 in order to get a score 2. The final score is an average of these two scores. We used the Matlab code available from the website[15].

One Shot Similarity (OSS) with the Proposed Representation We use the same parameters for binary-like face representation as the ones explaned in Section 5.2.3.1. By using the mirror-reflect version of $\mathbf{G}^{(i)}, \mathbf{G}^{(j)}$, we construct 4 models instead of 2 models and obtain three scores instead of a single score (see Fig. **5.16**). We treat these three scores as a single vector and feed these vectors into a support vector machine

[15] http://www.openu.ac.il/home/hassner/projects/Ossk/

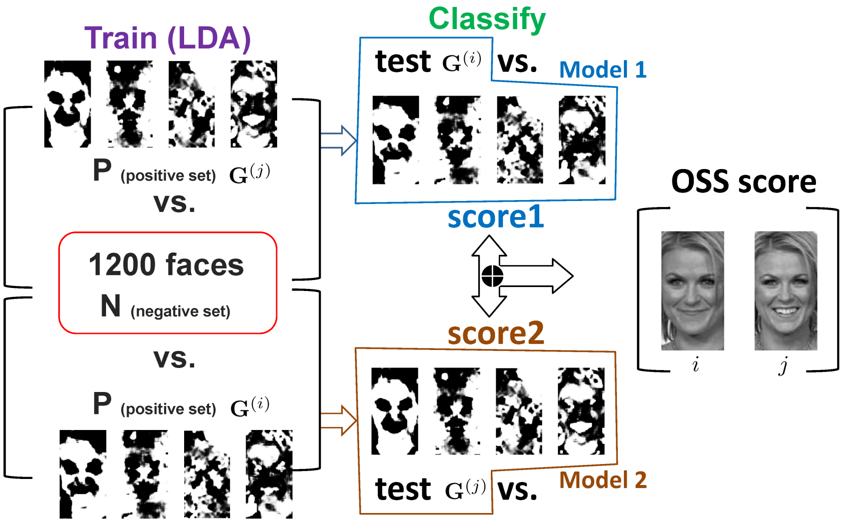


Figure 5.15: The original OSS score between two images (i,j) is an average of two scores from model 1 and model 2. We use a negative set composed of 1,200 faces which are exclusive to the test image pairs.

Table 5.2: Mean verification rates (10-fold) on the LFW dataset (view 2). sqrt means $\sqrt{\text{descriptors}}$ which is Hellinger distance and Mirror means that mirror-reflect version is added.

Descriptors	L2 distance	L2 + sqrt	MCS	MCS + sqrt	OSS	OSS+ sqrt
LBP	67.86%	68.53%	67.98%	68.18%	74.48%	74.41%
LBP (Mirror)	68.33%	69.08%	71.0%	67.61%	75.65%	76.05%
TPLBP	68.28%	68.78%	68.35%	67.76%	74.7%	74.58%
TPLBP (Mirror)	68.98%	69.38%	71.66%	68.6%	77.08%	76.1%
SIFT	71.01%	71.05%	70.65%	70.96%	73.13%	76.4%
SIFT (Mirror)	71.3%	71.08%	71.26%	71.3%	73.2%	78.2%
	L2 distance	L2 + logistic(c=80)	MCS	MCS + logistic(c=80)	OSS	OSS+ logistic(c=80)
Ours	65.81%	70.98%	68.25%	71.08%	75.81%	76.45%
Ours (Mirror)	66.28%	**73.23%**	71.26%	**73.3%**	76.38%	**78.9%**

(SVM) [131]. As shown in Table **5.1**, we use (1 set) as a negative set, train the support vector machine (SVM) with 4,800 (8 sets) OSS scores, and test 600 OSS scores (1 set).

We compare our feature representation with state of the art descriptors such

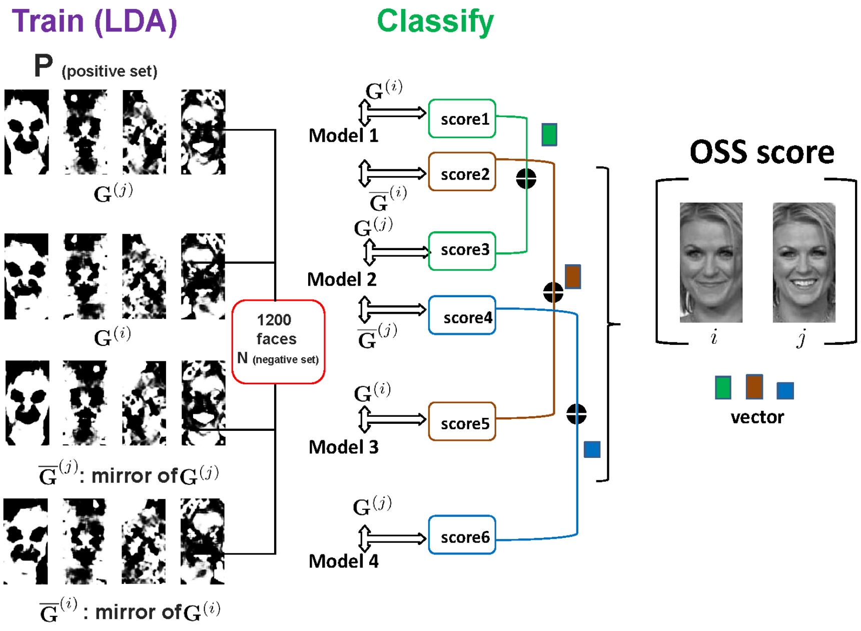


Figure 5.16: We have 4 models that are learned from $\mathbf{G}^{(i)}$, $\mathbf{G}^{(j)}$, $\overline{\mathbf{G}}^{(i)}$, $\overline{\mathbf{G}}^{(j)}$ respectively. In this case, the OSS score is not a scalar, but a vector of three elements.

Table 5.3: Mean verification rate (10-fold) comparison between [26] and our best result. TSS means the two shot similarity [26]. Numbers mean the number of descriptors used.

Method	L_2	L_2 + sqrt	TSS	TSS + sqrt	OSS	OSS + sqrt
Wolf et al. [26] (30) 85.13 ±0.37%	LBP, Gabor FPLBP, TPLBP SIFT (5)	LBP, Gabor FPLBP, TPLBP SIFT (5)	LBP, Gabor FPLBP, TPLBP SIFT (5)	LBP, Gabor FPLBP, TPLBP SIFT (5)	LBP, Gabor, FPLBP, TPLBP SIFT (5)	LBP, Gabor FPLBP, TPLBP SIFT (5)
Method	L2 distance	L2 + logistic(c=80)	MCS	MCS + logistic(c=80)	OSS	OSS+ logistic(c=80)
Ours (14) 85.10 ±0.59%	(0)	(0)	TPLBP (3) SIFT, pcaLARK	TPLBP (3) SIFT, pcaLARK	LBP, TPLBP SIFT, pcaLARK (4)	LBP, TPLBP SIFT, pcaLARK (4)

as TPLBP[16], LBP[17], and SIFT[18]. The parameters of all descriptors were copied from [26]. We used either the descriptor vectors or their square roots (i.e., the Hellinger distance)

[16] http://www.openu.ac.il/home/hassner/projects/Patchlbp/

[17] http://www.ee.oulu.fi/research/imag/texture/download/lbp.m

[18] http://people.csail.mit.edu/ceilu/ECCV2008

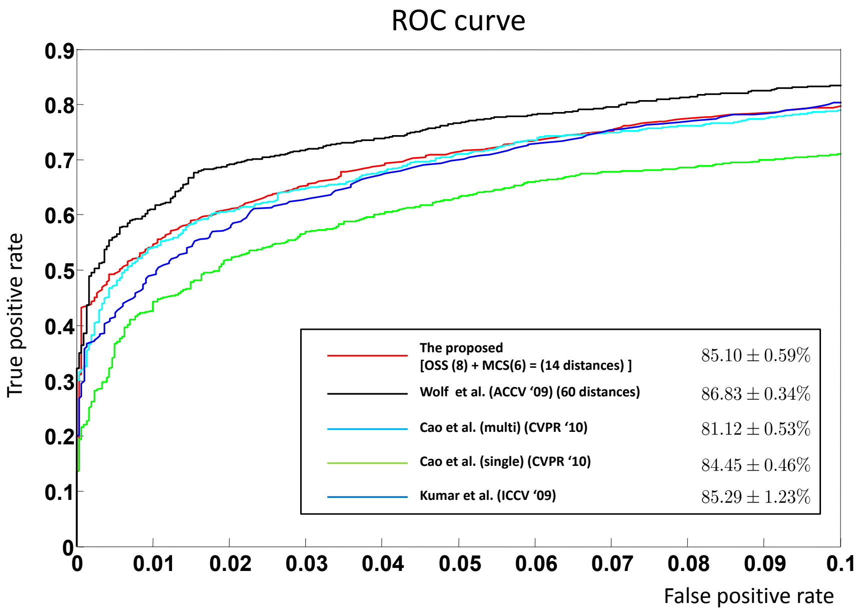


Figure 5.17: ROC curves and VR averaged over 10 folds of View 2. We achieve state of the arts performance with the much less number of distances than 30 distances in [26].

for other descriptors. L_2 distance and MCS were also shown in Tables **5.2** and **5.3** for a comparison.

Consistent with the unsupervised setting in Fig. **5.14**, the use of mirror-reflect lead to 1 ~ 3% improvement to all descriptors as described in Table **5.2**. As we can see from Table **5.2**, the proposed representation ($c = 80$) outperforms all the other (single) descriptors when used with OSS. Consistent with the results in the previous section (unsupervised setting), addition of mirror-reflect boosts the overall performance as well. Wolf et al. [26] reported that they achieve 85.13% with a total of 30 distances, but we are able to get the same performance with only 14 distances (vectors).

Discussion State of the art descriptors such as LBP, TPLBP, and SIFT use preprocessing steps as suggested in [26]. Accordingly, we applied a noise-removal filter (Matlab's wiener2 function) to the cropped images and saturated 1% of values at the low and high intensities for these descriptors. After computing descriptors from preprocessed images, descriptors were normalized to unit length. Then, these values are truncated at 0.2 and once again normalized to unit length. On the other hand, the proposed LARK descriptor does not require any preprocessing steps and is directly normalized to a unit vector[19].

The MCS (0.01 sec per pair) and OSS (0.37 sec per pair: Matlab implementation on Intel Pentium CPU 2.66 GHz machine) in conjunction with the proposed features is computationally efficient. Since the proposed method is based on a fixed set of bases, the extension of this methods to a large-scale face dataset would be straightforward. To this end, we could benefit from an efficient searching method (coarse-to-fine search) and/or a fast nearest neighbor search method (e.g., vantage point tree [133] and kernelized locality-sensitive hashing [183].)

Summary –

In this chapter, we have proposed 1) a simple, but effective statistical change detection framework to detect meaningful changes between two MRI images, 2) a novel binary-like representation for the face verification task. The proposed change detection framework in Section 5.1 is general enough as to be extendable to 3-D for other applications such as tumor detection in serial MRI scans using analogous 3-D LARKs [61]. Due to its robustness to noise and other systemic perturbations, we also

[19] We acknowledge that recognition rates of LBP, TPLBP, and SIFT in Table **5.2** do not coincide with ones in [26]. This slight difference may come from the image crop size, the sizes of the blocks, and how they are distributed within the crop size. However, we believe that the results shown in Table **5.2** in the same image crop size are a fair comparison because we followed the optimal parameter settings the authors reported.

expect the present framework to be quite effective in other imaging modalities such as CT, PET, etc. In Section 5.2, we developed a binary-like representation by applying PCA to LARK to develop a *fixed* basis, followed by a logistic function in order to make LARK as compact as possible and adapted to face verification. Experiments on the LFW dataset demonstrated that the proposed method yields state of the art results. We expect that face detection and verifications problems can be dealt with in a unified framework. In the following chapter, we conclude this thesis with some future directions.

Chapter 6

Conclusion and Future works

6.1 Summary of Contributions

In this thesis, we studied the effectiveness of LARKs in visual recognition and applied the proposed nonparametric detection framework to a wide variety of problems such as saliency detection, object/action recognition, automatic change detection, and face verification. The experimental results on challenging data demonstrated that the proposed framework outperforms state of the art methods.

▷ **Chapter 1** – We reviewed visual recognition problems and illustrated LARK descriptors that have desirable invariance properties. LARKs (3-D LARKs) capture local (space-time) geometric structure exceedingly well by taking advantage of the pixel-level similarity in a local patch (cube). LARKs are distinguished from other state of the art descriptors in the sense that LARK is based on the geodesic distance derived from the regularized covariance matrices. The concept of pixel-level similarity in LARKs can be extended to patch (cube)-level similarity by comparing a collection of LARKs (3-D LARKs) in a patch (cube) within an image.

- ▷ **Chapter 2** – The patch-level similarity by employing a nonparametric kernel density estimation based on 2-D/3-D *LARKs* and *MCS* led us to a unified framework for both static and space-time saliency detection. The proposed saliency detection method can automatically detect salient objects in the given image and salient moving objects in videos. Experiments on challenging sets of real-world human fixation data (both images and videos) demonstrated that the proposed saliency detection method achieves a high degree of accuracy and improves upon state of the art methods. We have tried to combine saliency maps from multiscale, but this idea did not improve performance even at the expense of time-complexity. This brings up an interesting question worth considering for future research; namely; what is the optimal resolution for saliency detection? Clearly, higher resolution images do not imply better saliency maps.

- ▷ **Chapter 3** – We extended the concept of patch-level similarity within one image to image (video)-level similarity across images (videos) for object (action) detection task. The image (video)-level similarity by employing *LARKs* and *MCS* in a naive Bayes framework led us to a unified framework for both object and action detection algorithm. The proposed method can automatically detect in the target the presence, the number, as well as location of similar objects (actions) to the given *single* query. To deal with more general scenarios, accounting for large variations in scale and rotation, we further proposed multiscale and multirotation approach. Challenging sets of real-world object and action experiments have demonstrated that the proposed approach achieves a high detection accuracy in completely different context and under different imaging conditions.

- ▷ **Chapter 4** – In order to make the training-free detection system run in real-time, we described how to speed-up the computation of LARKs while maintaining the discriminative but viewpoint tolerant properties of LARKs. By adopting a coarse-to-fine search in conjunction with a tree structure of examples, the recognition time of the proposed system grows logarithmically on average. The sped-up LARKs can learn coarse pose (according to how many views the user wants to learn), run in real time and work across face, object and action. This makes LARK a promising feature to be used in robot vision systems. Experiments on real-time object and action recognition showed that our proposed method runs in a real-time with high detection accuracy.

- ▹ **Chapter 5** – In order to further examine the efficacy of LARKs in visual recognition, we tackled two more problems: face verification and automatic change detection. We proposed 1) an effective statistical change detection framework to detect meaningful changes between two MRI images, 2) a binary-like representation for the face verification task. Experiments on the challenging datasets demonstrated that the proposed methods with LARKs yield state of the art results in both face verification and automatic change detection.

6.2 Future Directions

The future directions we discuss below are mainly categorized into (i) how to solve a large scale image classification problem with LARKs, (ii) how to extend the proposed detection framework to classification, and (iii) joint restoration and recognition from degraded examples.

6.2.1 Image Classification with LARKs

As we alluded to in Chapter 3, image classification is distinguished from object detection in the sense that the goal of image classification is to classify a given object into one of the pre-specified categories while object detection is to separate objects of interest from the background in a target image. Recently SVMs using spatial pyramid matching (SPM) kernel and SIFT have been highly successful in image classification. Despite its popularity, quantization used in both SIFT and K-means clustering is known to lead to severe degradation of discriminative power. The use of nonlinear SVMs with the expense of heavy complexity somewhat did "undo" the quantization damage. However, these nonlinear SVMs have a complexity $O(N^2 \sim N^3)$ in training and $O(N)$ in testing, where N is the training size, implying that it is nontrivial to scale-up the algorithms to handle more than thousands of training images. Yang et al. [184] developed an efficient SPM method which works well with linear SVM by generalizing vector quantization to sparse coding and applying it to SIFT along with multi-scale max pooling. This approach remarkably reduces the complexity of SVMs to $O(N)$ in training and a constant in testing.

In this section, we test LARK in the image classification framework [184] by replacing SIFT with LARK as shown in Fig. **6.1**. In the following image categorization experiment, we find that, in terms of classification accuracy, the linear SPM with the sparse coding of LARK descriptors leads to state-of-the-art performance on Caltech-101 dataset [32].

Preliminary Results on The Caltech-101 Dataset The Caltech-101 dataset [32] contains 101 classes (including animals, vehicles, flowers, etc.) with high shape variability.

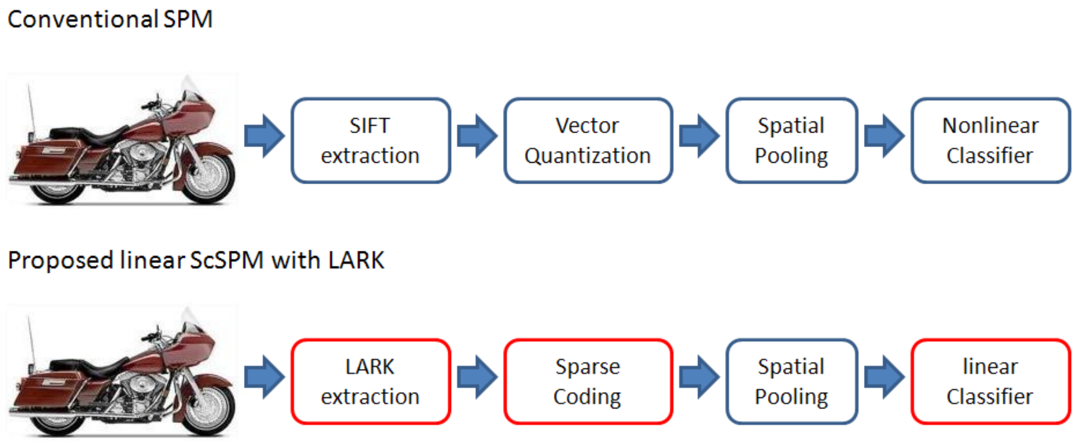


Figure 6.1: Schematic comparison of the original nonlinear SPM with our proposed linear SPM based on sparse coding of LARK descriptors. The underlying spatial pooling function for nonlinear SPM is *averaging* (leading to a histogram), while the spatial pooling function in sparse coding SPM is *max pooling* (which is not a histogram anymore).

Table 6.1: Classification rate (%) comparison on the Caltech-101 Dataset.

Algorithms	5	10	15	20	25	30
Zhang et al. [72]	46.6	55.8	59.1	62.0	–	66.20
Lazebnik [31]	–	–	56.40	–	–	64.60
Boiman [41]	–	–	65.00	–	–	70.40
Griffin [185]	44.2	54.5	59.0	63.3	65.8	67.60
Gemert [186]	–	–	–	–	–	64.16
Wang et al. [187]	51.15	59.77	65.43	67.74	70.16	73.44
Sparse coding SPM (LARK)	**52.16 ± 0.87**	**62.27 ± 0.56**	**66.45 ± 0.34**	**68.79 ± 0.45**	**71.35 ± 0.19**	**73.56 ± 0.23**

The number of images per category varies from 31 to 800. Most images are medium resolution , i.e. about 300 × 300 pixels. We followed the common experiment setup for Caltech-101, training on 5, 10, $\cdots$, 30 images per category and testing on the rest (no more than 50 testing images per class). The LARK descriptors extracted from 13 × 13 pixel patches were densely sampled from each image on a grid with stepsize 6 pixels.

The images were all preprocessed into gray scale. To train the codebooks[1], we used the sparse coding scheme, and fixed the codebook size as 1,024. We repeated the experimental process by 10 times with different random selected training and testing images to obtain reliable results. The average of per-class recognition rates were recorded for each run. And we report our final results by the mean recognition rates over 10 runs. Detailed comparison results are shown in Table **6.1**. As shown, our LARK descriptors with the sparse coding scheme outperforms the nonlinear SPM [31] by a large margin (about 11 percent for 15 training and 9 percent for 30 training per category) and a recent method by Wang et al. [187].

Direction Recently, a NEC-UIUC team won the first place in Large Scale Visual Recognition Challenge 2010 (ImageNet [188]). They focused on 1) fast descriptor coding based on local coordinate coding (LCC) [187], and 2) large-scale SVM classification. The LCC is basically the sparse coding with a locality constraint, and large-scale SVM is realized via average stochastic gradient descent [189]. We believe that LARK can efficiently be employed in the same framework for larger scale databases such as ImageNet [36] and Pascal VOC [35, 190]. Also we expect that the use of 3-D LARKs in this framework can lead to state of the art action classification results on the challenging action datasets such as HOLLYWOOD2 [152] and UCF50 dataset[2].

6.2.2 Extension of The Proposed Detection Framework to Classification

In this section, we approach classification problem differently from the ones described in the previous section, by extending object and action detection with LARK

[1] For training the linear classifiers, we used the implementation of SVM [184].

[2] `http://www.cs.ucf.edu/~kreddy/Datasets.html`

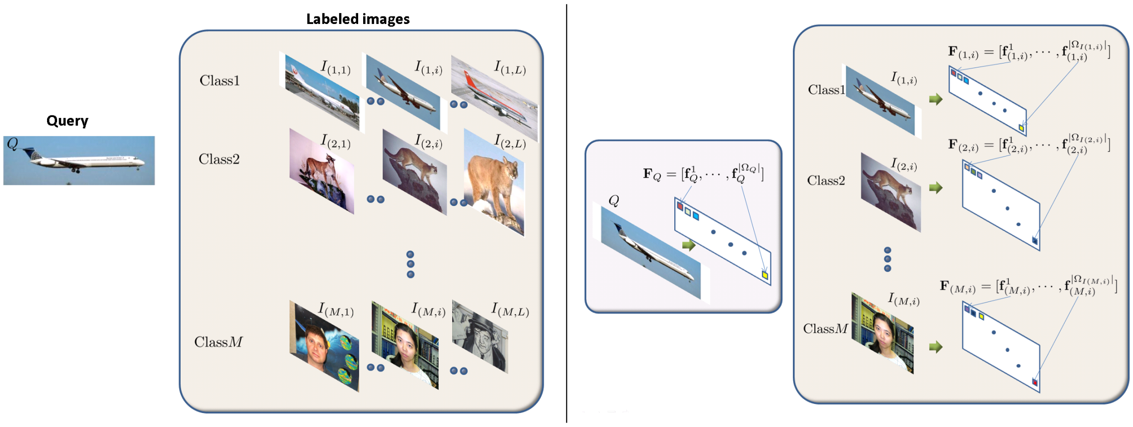

Query
Q
Labeled images
Class1
$I_{(1,1)}$
$I_{(1,i)}$
$I_{(1,L)}$
Class2
$I_{(2,1)}$
$I_{(2,i)}$
$I_{(2,L)}$
Class$M$
$I_{(M,1)}$
$I_{(M,i)}$
$I_{(M,L)}$
$\mathbf{F}_Q = [\mathbf{f}_Q^1, \cdots, \mathbf{f}_Q^{|\Omega_Q|}]$
Q
$\mathbf{F}_{(1,i)} = [\mathbf{f}_{(1,i)}^1, \cdots, \mathbf{f}_{(1,i)}^{|\Omega_{I(1,i)}|}]$
Class1
$I_{(1,i)}$
$\mathbf{F}_{(2,i)} = [\mathbf{f}_{(2,i)}^1, \cdots, \mathbf{f}_{(2,i)}^{|\Omega_{I(2,i)}|}]$
Class2
$I_{(2,i)}$
$\mathbf{F}_{(M,i)} = [\mathbf{f}_{(M,i)}^1, \cdots, \mathbf{f}_{(M,i)}^{|\Omega_{I(M,i)}|}]$
Class$M$
$I_{(M,i)}$

Figure 6.2: The feature matrices $\mathbf{F}_Q$ for the query and $\mathbf{F}_{(i,i)}, \cdots, \mathbf{F}_{(M,i)}$ for all the labeled images are extracted.

in 2-D and 3-D to a nonparametric classification framework. As we alluded to before, the goal of visual object category classification is to place a given a query into one of say, M, pre-specified classes, each class containing L labeled visual objects, as shown in Fig. **6.2** (left). As we did in the object detection task in Chapter 3, we can compute LARK descriptors densely from the query and each labeled images. Subsequently, feature matrices $\mathbf{F}_Q$ for the query, and $\mathbf{F}_{(1,i)}, \cdots, \mathbf{F}_{(M,i)}$, for all the labeled images are constructed (see Fig. **6.2** (right).) Here, the task at hand is to decide which class (c) the features $\mathbf{F}_Q$ from a query image Q are most likely to have come from. More formally, the M-ary hypothesis test of interest is shown in Table **6.2**.

$\mathscr{H}_1$:	Q belongs to class 1	$\Leftrightarrow$	$\mathbf{F}_Q$ comes from class 1($\mathbf{F}_1$) ,
$\mathscr{H}_2$:	Q belongs to class 2	$\Leftrightarrow$	$\mathbf{F}_Q$ comes from class 2 ($\mathbf{F}_2$),
$\vdots$		$\vdots$	
$\mathscr{H}_M$:	Q belongs to class M	$\Leftrightarrow$	$\mathbf{F}_Q$ comes from class M ($\mathbf{F}_M$).

Table 6.2: M-ary hypotheses for image classification

Assuming that the prior probabilities $P(\mathcal{H}_c)$ are equal, then the maximum a posterior (MAP) decision rule boils down to the M-ary maximum likelihood (ML) decision rule as similarity done in Appendix A.

$$\widehat{\mathcal{H}_c} = \arg\max_c P(\mathcal{H}_c|\mathbf{F}_Q) = \arg\max_c p(\mathbf{F}_Q|\mathcal{H}_c). \tag{6.1}$$

We estimate the PDF $p(\mathbf{F}_Q|\mathcal{H}_c)$ using a kernel density estimation method, which results in an empirical Bayes approach where the estimate $\widehat{p}(\mathbf{F}_Q|\mathcal{H}_c)$ is defined as a weighted sum of kernels centered at the features $\mathbf{f}_c$ which belong to the hypothesis $\mathcal{H}_c$. More specifically,

$$\widehat{p}(\mathbf{F}_Q|\mathcal{H}_c) = \frac{\sum_{i=1}^{L}\sum_{j\in\Omega_{I(c,i)}} G^{(i,j)}(\mathbf{f}_Q^{\ell} - \mathbf{f}_{(c,i)}^{j}, \mathbf{x}_Q^{\ell} - \mathbf{x}_{(c,i)}^{j})}{\sum_{\ell\in\Omega_Q}\sum_{i=1}^{L}\sum_{j\in\Omega_{I(c,i)}} G^{(i,j)}(\mathbf{f}_Q^{\ell} - \mathbf{f}_{(c,i)}^{j}, \mathbf{x}_Q^{\ell} - \mathbf{x}_{(c,i)}^{j})}, \quad \ell\in\Omega_Q, \quad j\in\Omega_{I(c,i)} \tag{6.2}$$

where $G^{(i,j)}$ is a locally data adaptive kernel function, Ω_Q is the query image domain consisiting of $|\Omega_Q|$ pixels, $\Omega_{I(c,i)}$ is the i^{th} labeled image in class c, consisting of $|\Omega_{I(c,i)}|$ pixels; and $\mathbf{x}_Q^{\ell}, \mathbf{x}_{(c,i)}^{j}$ are column vectors denoting spatial coordinates of the corresponding features $\mathbf{f}_Q^{\ell}$ and $\mathbf{f}_{(c,i)}^{j}$.

To proceed forward, we can make the simplifying assumption that $\mathbf{f}_Q^{1}, \mathbf{f}_Q^{2}, \cdots, \mathbf{f}_Q^{|\Omega_Q|}$, thought of as a random variable, are essentially independent, and identically distributed, given the hypothesis $\mathcal{H}_c$. The decision rule can then be rewritten as:

$$\begin{aligned}\widehat{\mathcal{H}_c} &= \arg\max_c \log\widehat{p}(\mathbf{F}_Q|\mathcal{H}_c) = \arg\max_c \log\widehat{p}(\mathbf{f}_Q^{1}, \cdots, \mathbf{f}_Q^{|\Omega_Q|}|\mathcal{H}_c)\\ &= \arg\max_c \sum_{\ell=1}^{|\Omega_Q|} \log\widehat{p}(\mathbf{f}_Q^{\ell}|\mathcal{H}_c).\end{aligned} \tag{6.3}$$

The apparent consequence now is that we need to estimate each local individual probability density $\widehat{p}(\mathbf{f}_Q^{\ell}|\mathcal{H}_c)$ separately:

$$\widehat{p}(\mathbf{f}_Q^{\ell}|\mathcal{H}_c) = \frac{1}{\beta'}\sum_{i=1}^{L}\sum_{j\in\Omega_{I(c,i)}} G^{(i,j)}(\mathbf{f}_Q^{\ell} - \mathbf{f}_{(c,i)}^{j}, \mathbf{x}_Q^{\ell} - \mathbf{x}_{(c,i)}^{j}), \quad \ell = 1, \cdots, |\Omega_Q|, \tag{6.4}$$

where β' is a normalization factor. Given the apparent strength of the information contained in LARKs[3], we can consider using a single (spatially and photometrically nearest) neighbor for the approximation. Using a separable (“bilateral") kernel yields:

$$\widehat{p}(\mathbf{f}_Q^\ell|\mathcal{H}_c) \approx \exp\left(-\frac{1}{2\sigma_r^2}\text{dist}(\mathbf{f}_Q^\ell,\mathbf{f}^{N_c(\ell)})\right)\exp\left(-\frac{1}{2\sigma_s^2}\text{dist}(\mathbf{x}_Q^\ell,\mathbf{x}^{\mathbf{N_c}(\ell)})\right), \tag{6.5}$$

where σ_r, σ_s are parameters controlling the fall-off of weights in photometric and spatial domains, respectively. $N_c(\ell)$ is the nearest neighbor of $\mathbf{f}_Q^\ell$ in the class c, and $\text{dist}(\mathbf{f}_Q^\ell, \mathbf{f}^{N_c(\ell)}) = ||\frac{\mathbf{f}_Q^\ell}{||\mathbf{F}_Q||} - \frac{\mathbf{f}^{N_c(\ell)}}{||\mathbf{F}_{N_c}||}||^2$ (where $\mathbf{F}_{N_c} = [\mathbf{f}^{N_c(1)}, \cdots, \mathbf{f}^{N_c(|\Omega_Q|)}]$.) The decision rule then becomes

$$\begin{aligned}
\widehat{\mathcal{H}_c} &= \arg\max_c \sum_{\ell=1}^{|\Omega_Q|} \log \widehat{p}(\mathbf{f}_Q^\ell|\mathcal{H}_c) \\
&\Rightarrow \arg\max_c \sum_{\ell=1}^{|\Omega_Q|} -\frac{1}{2\sigma_r^2}\left(\frac{\|\mathbf{f}_Q^\ell\|^2}{\|\mathbf{F}_Q\|_F^2} + \frac{\|\mathbf{f}^{N_c(\ell)}\|^2}{\|\mathbf{F}_{N_c}\|_F^2} - \frac{2\rho(\mathbf{f}_Q^\ell, \mathbf{f}^{N_c(\ell)})\|\mathbf{f}_Q^\ell\|\|\mathbf{f}^{N_c(\ell)}\|}{\|\mathbf{F}_Q\|_F\|\mathbf{F}_c\|_F}\right) - \frac{1}{2\sigma_s^2}\|\mathbf{x}_Q^\ell - \mathbf{x}^{N_c(\ell)}\|^2, \\
&= \arg\max_c \frac{1}{\sigma_r^2}\underbrace{<\frac{\mathbf{F}_Q}{\|\mathbf{F}_Q\|_F}, \frac{\mathbf{F}_{N_c}}{\|\mathbf{F}_{N_c}\|_F}>_F}_{\text{Geometric nearness}} - \frac{1}{2\sigma_s^2}\underbrace{\|\mathbf{x}_Q^\ell - \mathbf{x}^{N_c(\ell)}\|^2}_{\text{Spatial nearness}}.
\end{aligned} \tag{6.6}$$

The first term above measure the “geometric" similarity of the query features to the nearest feature in each class, and the second term measures the spatial separation between the locations of these respective features. As such, this approach will provide a very natural and robust way of comparing the likeness of a query to appropriately and automatically selected elements of several classes of objects. The class which shows the largest overall similarity is the one to which the query will be matched. It is important to note that this approach also allows for the online updating of the labeled class examples. That is to say, if the query is assigned to a given class c^* with high empirical likelihood, then this example can be included as a new labeled instance of this class.

[3] Since LARKs lie on manifolds of relatively low co-dimension, and the statistical distribution of such features tends to be heavy-tailed, it is generally sufficient to use a very small number of labeled features in class c to get a reasonable estimate of the conditional density $\widehat{p}(\mathbf{f}_Q^\ell|\mathcal{H}_c)$.

As such , the class will now contain one more instance, which will then make it easier to identify subsequent queries as belonging to this class or not. This approach can be efficiently realized by using approximate-r-nearest-neighbors algorithm [191] and KD-tree implementation of [192], thus can be scaled to larger scale classification tasks as well.

6.2.3 Joint Classification and Restoration

Frequently, visual object recognition problems are approached under the assumptions that images and videos are “clean". While the trend towards exemplar-based methods is clearly established, the utility of such methods suffers when the images are corrupted or disturbed. Indeed, performance suffers, often catastrophically, in the presence of degraded images and videos. Most learning-based recognition systems heavily rely on low-level features extracted from an interest point detector. While popular interest point detectors such as Harris-affine detector [19], Hessian-affine detector [19], or Maximally Extreme Stable Region detector [193] are known to be robust in the presence of moderate amount of white Gaussian noise and blur, they are not designed for most other type of degradation. Essentially all existing interest point detectors fail to detect region of interest in the presence of severe degradation caused by various common conditions such as non-stationary blur, snow or rain, air turbulence, and etc., as shown in Fig. **6.3** and Fig. **6.4**. As a consequence, recognition systems become unreliable and useless in the presence of severe degradation of data, which occurs often in real-world applications. In order to deal with these problems, we might preprocess degraded data (denoising, deblurring, dehazing, or removal of snow) before attempting object recognition. However, preprocessing of degraded data without taking into

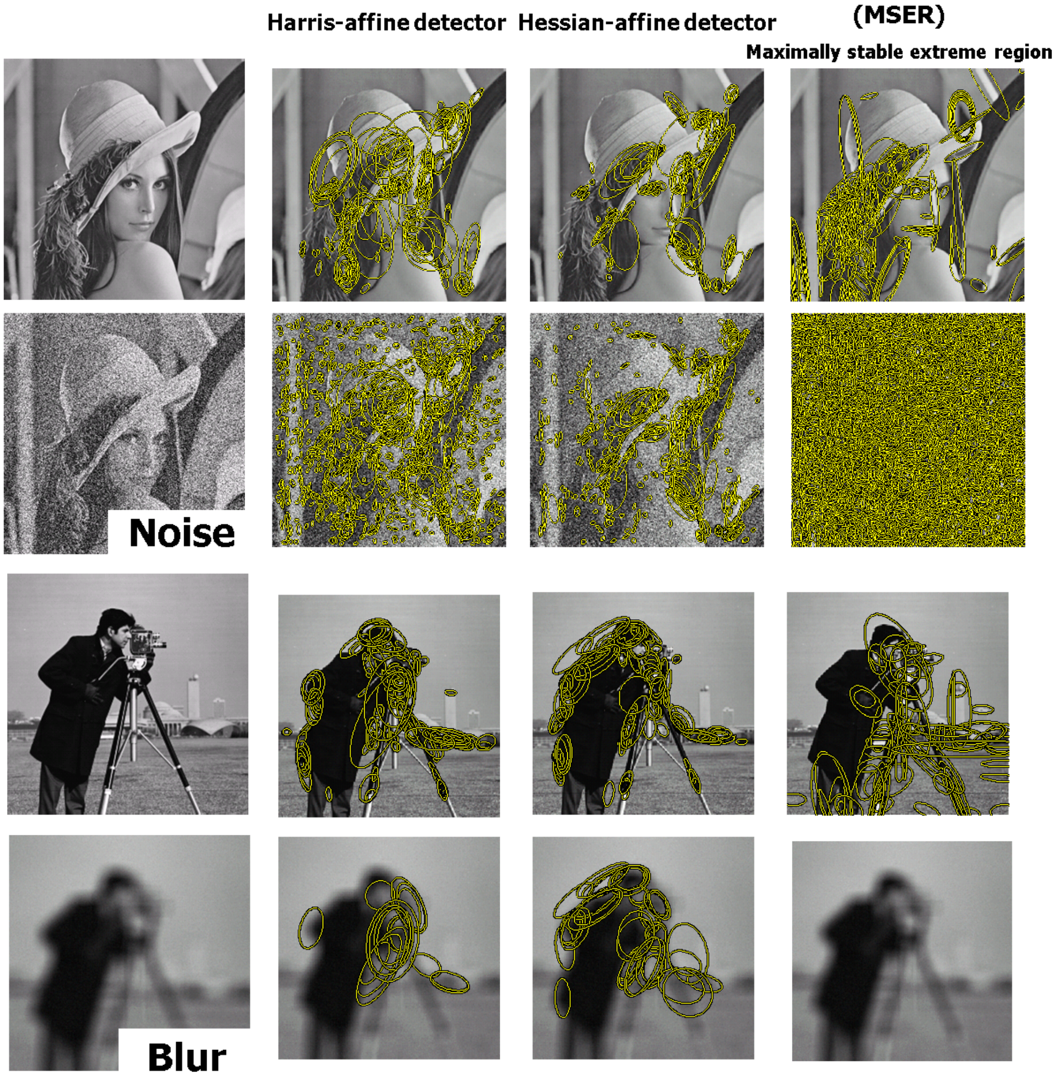


Figure 6.3: State of the arts interest point detectors such as Harris-affine detector, Hessian-affine detector, and MSER do not work well in the presence of stochastically severe degradation (WGN and blur). It is apparent that most existing object recognition systems based on these detectors can not perform well with the degraded test data unobserved before.

account what we try to recognize might further distort the process, and thus is at best

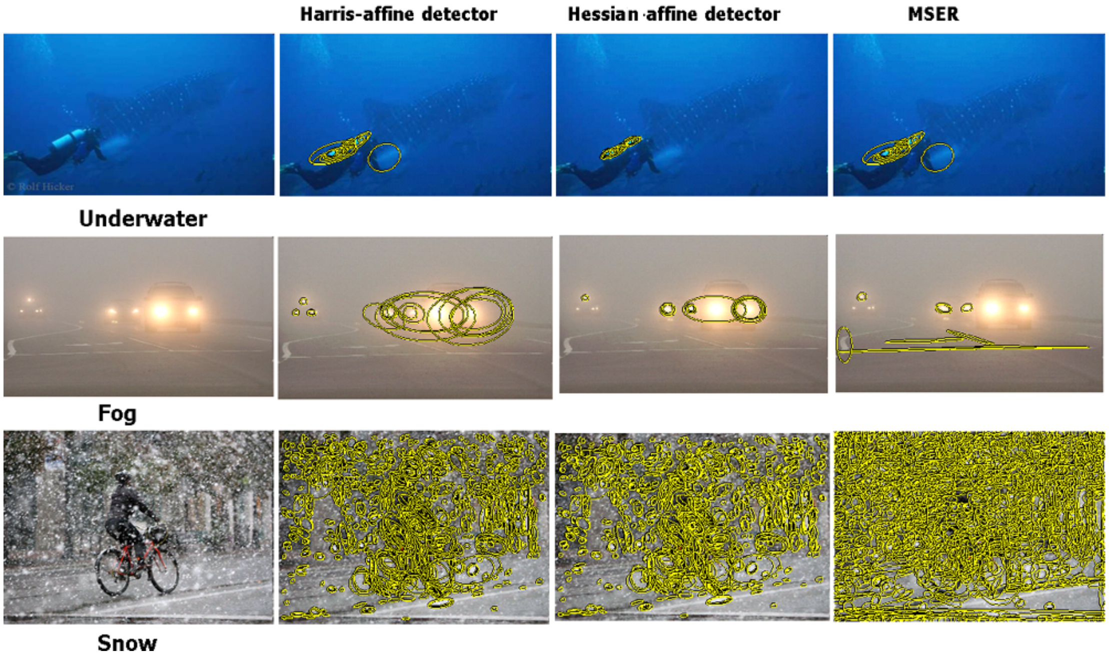


Figure 6.4: State of the arts interest point detectors such as Harris-affine detector, Hessian-affine detector, and MSER do not work well in the presence of systemically severe degradation (underwater, fog, or snow). It is apparent that most existing object recognition systems based on these detectors can not perform well with the degraded test data unobserved before.

suboptimal[4]. As a concrete example, the performance of face recognition deteriorates when the query faces are of lower resolution than face images in the database. The classical approaches to matching a low resolution face to a high resolution gallery (namely, upsampling the query, or downsampling the gallery images) have been shown to be quite ineffective in practice because of undesirable distortions [195].

Here, we propose a setting in which we can unify these problems and deal with restoration and recognition simultaneously. To begin, let us first consider the object *detection* problem, where a “clean" query image is given, but where the target image is degraded. Formally, given a query Q, and a noisy target image $\widetilde{T}$, we wish

[4] If we know that snow or rain result in degradation in advance, we can benefit from a method [194] that can detect snow or raindrop.

to not only identify objects within $\widetilde{T}$ that are similar to Q, but also estimate “clean" version of these objects, producing a (at least partially) restored target image $\widehat{T}$. So the task at hand is to both detect relevant regions in the given target, and to restore these relevant regions. Our nonparametric approach provides a way to do just that. An object function that can lead us to this solution can be formulated as follows:

$$\arg\max_{i,\mathbf{t}} \underbrace{\rho(\mathbf{F}_Q, \mathbf{F}_{T_i})}_{\text{Detection}} - \underbrace{\alpha \sum_{\ell=1}^{n} (\mathbf{q}^\ell - \mathbf{t}^\ell)^\top \mathbf{W}_Q^\ell (\mathbf{q}^\ell - \mathbf{t}^\ell)}_{\text{Resotration}}, \tag{6.7}$$

where α is a regularization parameter; $\mathbf{q}^\ell$ is the vector representation of patches from the query image, and $\mathbf{t}^\ell$ is a candidate patch from the target image $\widetilde{T}$, centered at $\mathbf{x}_\ell$. Finally, the matrix $\mathbf{W}_Q$ contains weights reflecting the level of similarity between $\mathbf{q}^\ell$ and $\mathbf{t}^\ell$ detected from the features. More simply put, the intuition behind the above formulation is this: When we detect a part of the target that matches the query (first term in the cost), we have identified a whole collection of corresponding patches in the regions from the query and the target. This correspondence can be used (the second term in the cost) to restore the noisy patches, using “similar" patches from *both* the query, and the target, in a fashion not dissimilar to the popular non-local means paradigm. In the standard patch-based processing approach, the similar patches provided by the matching query. Naturally, the degree of similarity will determine the relative weights given to these patches; and this is measured using the matrix $\mathbf{W}_Q$. The second term in the cost function above can use weights from different patch-based approaches for restoration such as NLM, the more effective nonparametric methods described in [46, 196, 61], or the guided filtering [124]. Motivated by [197], [198, 199] recently proposed iterative guided filtering framework and achieved state of the art guided restoration (e.g., flash/no-flash denoising/deblurring) results. We expect that the technique by [198, 199] can be useful for the second term in the cost function as well.

Next let us consider the more common scenario, where the query image $\widetilde{Q}$ is degraded but labeled images in the database are “clean". In a manner similar to what we proposed above, we can consider the following objective function:

$$\arg\max_{c,\mathbf{q}} \underbrace{\frac{1}{\sigma_r^2}\rho(\mathbf{F}_Q,\mathbf{F}_{N_c}) - \frac{1}{2\sigma_s^2}\sum_{\ell=1}^{|\Omega_Q|} ||\mathbf{x}_Q^{\ell} - \mathbf{x}^{N_c(\ell)}||^2}_{\text{Classification}} - \alpha \underbrace{\sum_{\ell=1}^{n} (\mathbf{q}^{\ell} - \mathbf{t}^{N_c(\ell)})^{\top}\mathbf{W}^{N_c(\ell)}(\mathbf{q}^{\ell} - \mathbf{t}^{N_c(\ell)})}_{\text{Restoration}} \quad (6.8)$$

Analogous to the earlier formulation, the first term in the above objective function classifies the given query into a particular class objects, while the second term uses all the similar patches now available in this class to effect restoration of the noisy query which can also use [2, 46, 198] or other related patch-based approaches. Naturally, the next reasonable intellectual step would be to consider the joint recognition and restoration problems when both the query and the target are degraded. This is certainly within the purview of the line of work we propose, though presumably its solution is best sought in light of the above open problems.

Appendix A

Justification by Naive Bayes Framework

In this section, we show that the naive-Bayes approach in a multiple hypothesis testing framework leads to the Matrix Cosine Similarity-based decision rule. It is worth noting that this idea is partly motivated by [41] and [118] who derived optimal Bayes decision rule based on Euclidean distance and the whitened cosine similarity respectively for the image classification task.

As described in Chapter 3, the target T is divided into a set of overlapping patches and a class is assigned to each patch. Our task at hand is to figure out which class (i) the features from Q are most likely to have come from. Since we do not know the class-conditional pdf ($p(\overline{\mathbf{F}}_Q|class)$) of the normalized features extracted from Q, we set out to estimate it using a kernel density estimation method [87]. Once we have these estimates, we will show that the maximum likelihood (ML) decision rule boils down to computing and thresholding Matrix Cosine Similarity, which can be efficiently implemented using a nearest neighbor formulation.

By associating each patch (T_i) of the target image with a hypothesis, we now have the case where we wish to discriminate between M hypotheses ($\mathscr{H}_0, \cdots, \mathscr{H}_{M-1}$) as

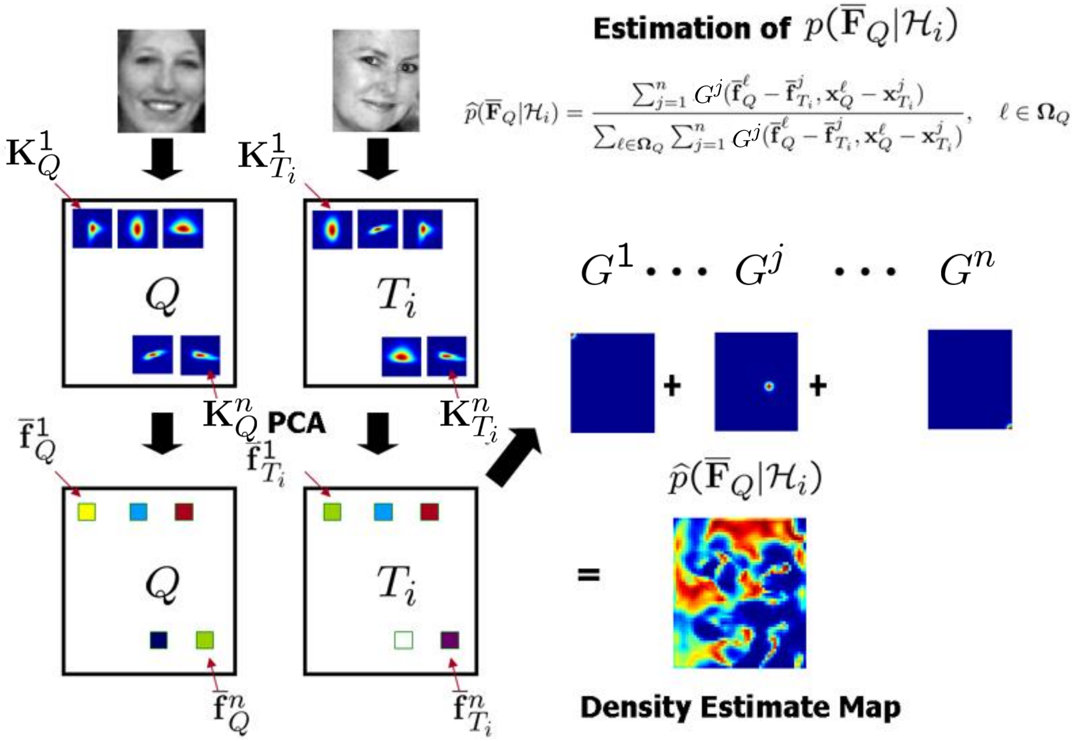

Estimation of $p(\overline{\mathbf{F}}_Q|\mathcal{H}_i)$
$\widehat{p}(\overline{\mathbf{F}}_Q|\mathcal{H}_i) = \frac{\sum_{j=1}^{n} G^j(\overline{\mathbf{f}}_Q^\ell - \overline{\mathbf{f}}_{T_i}^j, \mathbf{x}_Q^\ell - \mathbf{x}_{T_i}^j)}{\sum_{\ell \in \mathbf{\Omega}_Q} \sum_{j=1}^{n} G^j(\overline{\mathbf{f}}_Q^\ell - \overline{\mathbf{f}}_{T_i}^j, \mathbf{x}_Q^\ell - \mathbf{x}_{T_i}^j)}, \quad \ell \in \mathbf{\Omega}_Q$
$\mathbf{K}_Q^1$
$\mathbf{K}_{T_i}^1$
Q
T_i
$\mathbf{K}_Q^n$
PCA
$\mathbf{K}_{T_i}^n$
$\overline{\mathbf{f}}_Q^1$
$\overline{\mathbf{f}}_{T_i}^1$
$G^1 \cdots G^j \cdots G^n$
+
+
$\widehat{p}(\overline{\mathbf{F}}_Q|\mathcal{H}_i)$
=
Q
T_i
$\overline{\mathbf{f}}_Q^n$
$\overline{\mathbf{f}}_{T_i}^n$
Density Estimate Map

Figure A.1: The estimated conditional density $\widehat{p}(\overline{\mathbf{F}}_Q|\mathscr{H}_i)$ is a sum of kernels (weight functions) centered at the features $\overline{\mathbf{f}}_{T_i}$ in T_i which belongs to the hypothesis $\mathscr{H}_i$. In the Density Estimate Map, red value means a high conditional probability density $\widehat{p}(\overline{\mathbf{f}}_Q|\mathscr{H}_i)$ while blue value represents a low conditional probability density $\widehat{p}(\overline{\mathbf{f}}_Q|\mathscr{H}_i)$.

follows:

$$\begin{aligned}
\mathscr{H}_0&: Q \text{ is similar to } T_0 &&\Leftrightarrow \overline{\mathbf{F}}_Q \text{ comes from class } 0\ (\overline{\mathbf{F}}_{T_0})\ , \\
\mathscr{H}_1&: Q \text{ is similar to } T_1 &&\Leftrightarrow \overline{\mathbf{F}}_Q \text{ comes from class } 1\ (\overline{\mathbf{F}}_{T_1}), \\
&\vdots && \vdots \\
\mathscr{H}_{M-1}&: Q \text{ is similar to } T_{M-1} &&\Leftrightarrow \overline{\mathbf{F}}_Q \text{ comes from class } M-1\ (\overline{\mathbf{F}}_{T_{M-1}}).
\end{aligned}$$

Table A.1: M-array hypotheses for detection problem.

The task at hand is to find the most likely hypothesis (or a correct class) given the query

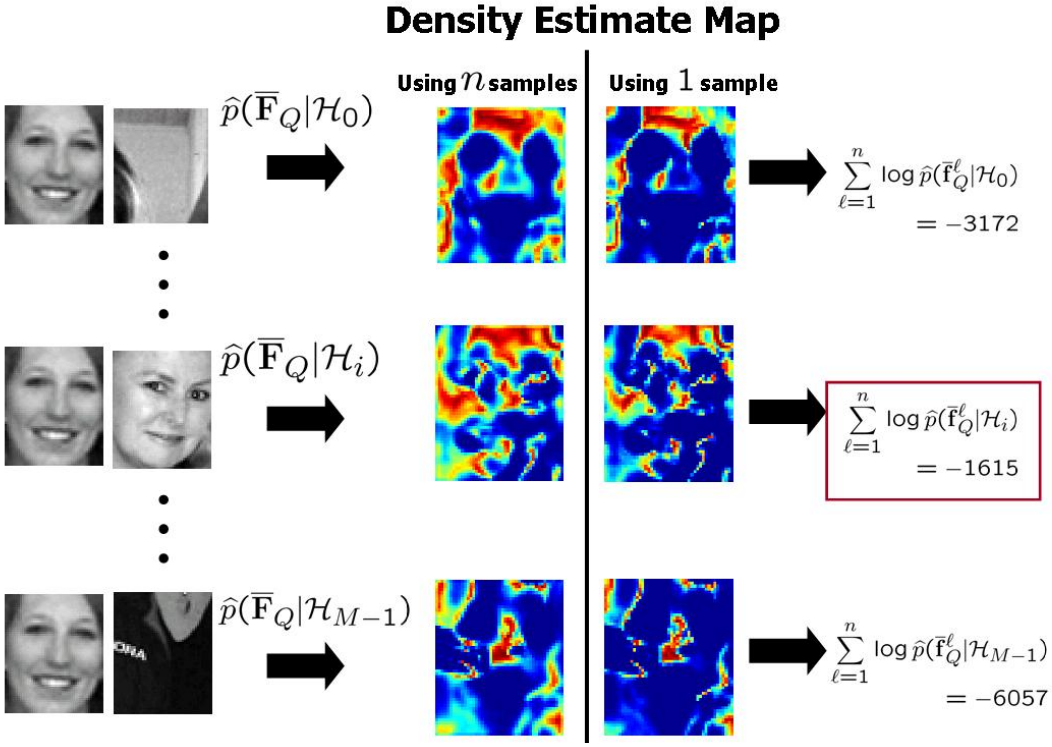

Density Estimate Map
Using n samples
Using 1 sample
$\widehat{p}(\overline{\mathbf{F}}_Q|\mathcal{H}_0)$
$\sum_{\ell=1}^{n} \log \widehat{p}(\overline{\mathbf{f}}_Q^\ell|\mathcal{H}_0)$
$= -3172$
$\widehat{p}(\overline{\mathbf{F}}_Q|\mathcal{H}_i)$
$\sum_{\ell=1}^{n} \log \widehat{p}(\overline{\mathbf{f}}_Q^\ell|\mathcal{H}_i)$
$= -1615$
$\widehat{p}(\overline{\mathbf{F}}_Q|\mathcal{H}_{M-1})$
$\sum_{\ell=1}^{n} \log \widehat{p}(\overline{\mathbf{f}}_Q^\ell|\mathcal{H}_{M-1})$
$= -6057$

Figure A.2: The estimated conditional probability densities $\widehat{p}(\overline{\mathbf{F}}_Q|\mathscr{H}_i)$ using n samples and 1 sample are shown in the middle and the scores on right side means $\sum_{\ell=1}^{n} \log \widehat{p}(\overline{\mathbf{f}}_Q^\ell|\mathscr{H}_i)$. The higher this score is, the more likely $\overline{\mathbf{F}}_Q$ comes from class i ($\overline{\mathbf{F}}_{T_i}$).

image Q. It is a well known fact [131, 200] that maximizing a posteriori probability $P(\mathscr{H}_i|\overline{\mathbf{F}}_Q)$ minimizes Bayes risk (or the average classification error.) Assuming that the prior probabilities $P(\mathscr{H}_i)$ are equal, then the maximum a posterior (MAP) decision rule boils down to the M-ary maximum likelihood (ML) decision rule.

$$\widehat{\mathscr{H}_i} = \arg\max_i P(\mathscr{H}_i|\overline{\mathbf{F}}_Q) = \arg\max_i p(\overline{\mathbf{F}}_Q|\mathscr{H}_i). \tag{A.1}$$

Since we do not know the conditional probability density function $p(\overline{\mathbf{F}}_Q|\mathscr{H}_i)$ of features $\overline{\mathbf{F}}_Q$ given the features $\overline{\mathbf{F}}_{T_i}$ of the target patch T_i, we need to estimate it using a kernel density estimation method, which results in the naive or empirical Bayes approach.

A.0.4 Locally Data-adaptive Kernel Density Estimation

The Parzen density estimator is a simple and generally accurate non-parametric density estimation method [87]. However, if the true conditional density that we want to model is close to a “non-linear" lower dimensional manifold embedded in the higher dimensional feature space, Parzen density estimator with an isotropic kernel is not the most appropriate method [90, 89, 88]. As explained earlier, the features $\overline{\mathbf{F}}_Q, \overline{\mathbf{F}}_{T_i}$ tend to generically come from long-tailed distributions, and as such, there are generally no tight clusters in the feature space. When we estimate a probability density at a particular point, for instance $\bar{\mathbf{f}}_Q^\ell$, the isotropic kernel centered on that point will spread its density mass equally along all the feature space directions, thus giving too much emphasis to irrelevant regions of space and too little along the manifold. Earlier studies [90, 89, 88] also pointed out this problem. This motivates us to use *a locally data-adaptive version of the kernel density estimator.*

The estimated conditional density $\widehat{p}(\overline{\mathbf{F}}_Q|\mathscr{H}_i)$ is defined as a sum of kernels (weight functions) centered at the features $\bar{\mathbf{f}}_{T_i}$ in T_i which belong to the hypothesis $\mathscr{H}_i$. More specifically,

$$\widehat{p}(\overline{\mathbf{F}}_Q|\mathscr{H}_i) = \frac{\sum_{j=1}^{n} G^j(\bar{\mathbf{f}}_Q^\ell - \bar{\mathbf{f}}_{T_i}^j, \mathbf{x}_Q^\ell - \mathbf{x}_{T_i}^j)}{\sum_{\ell \in \Omega_Q} \sum_{j=1}^{n} G^j(\bar{\mathbf{f}}_Q^\ell - \bar{\mathbf{f}}_{T_i}^j, \mathbf{x}_Q^\ell - \mathbf{x}_{T_i}^j)}, \quad \ell \in \Omega_Q, \tag{A.2}$$

where G^j is a locally data adaptive kernel function, Ω_Q is the query image domain consisting of $|\Omega_Q|$ pixels and $\mathbf{x}_Q^\ell, \mathbf{x}_{T_i}^j$ are column vectors denoting spatial coordinates of corresponding features $\bar{\mathbf{f}}_Q^\ell$ and $\bar{\mathbf{f}}_{T_i}^j$. A simple and intuitive choice of the G^j is to consider two terms for penalizing the spatial distance between the point of interest and its neighbors, and the radiometric “distance" between the corresponding features

$\bar{\mathbf{f}}_Q^\ell$ and $\bar{\mathbf{f}}_{T_i}^j$. More specifically, the kernel function is defined as follows:

$$\begin{aligned} G^j &= G_r^j(\bar{\mathbf{f}}_Q^\ell - \bar{\mathbf{f}}_{T_i}^j) K_s^j(\mathbf{x}_Q^\ell - \mathbf{x}_{T_i}^j), \\ &= \exp\left(-\frac{\text{dist}(\bar{\mathbf{f}}_Q^\ell, \bar{\mathbf{f}}_{T_i}^j)}{2\sigma_r^2}\right) \exp\left(-\frac{\|\mathbf{x}_Q^\ell - \mathbf{x}_{T_i}^j\|^2}{2\sigma_s^2}\right), \quad \ell \in \Omega_Q, \end{aligned} \tag{A.3}$$

where we define $\text{dist}(\bar{\mathbf{f}}_Q^\ell, \bar{\mathbf{f}}_{T_i}^j) = \| \frac{\mathbf{f}_Q^\ell}{\|\mathbf{F}_Q\|_F} - \frac{\mathbf{f}_{T_i}^j}{\|\mathbf{F}_{T_i}\|_F} \|^2$, and σ_r, σ_s are parameters controlling the fall-off of weights in radiometric and spatial domains.

Inserting equation (A.3) into equation (A.2), the estimated conditional density $\widehat{p}(\overline{\mathbf{F}}_Q|\mathscr{H}_i)$ becomes

$$\widehat{p}(\overline{\mathbf{F}}_Q|\mathscr{H}_i) = \frac{1}{\beta}\sum_{j=1}^{n} \exp\left(-\frac{\text{dist}(\bar{\mathbf{f}}_Q^\ell, \bar{\mathbf{f}}_{T_i}^j)}{2\sigma_r^2} - \frac{\|\mathbf{x}_Q^\ell - \mathbf{x}_{T_i}^j\|^2}{2\sigma_s^2}\right), \tag{A.4}$$

where $\beta = \sum_{\ell \in \Omega_Q} \sum_{j=1}^{n} G^j(\bar{\mathbf{f}}_Q^\ell - \bar{\mathbf{f}}_{T_i}^j, \mathbf{x}_Q^\ell - \mathbf{x}_{T_i}^j)$ is a normalization factor. Fig.**A.1** depicts how the conditional density function $\widehat{p}(\overline{\mathbf{F}}_Q|\mathscr{H}_i)$ is estimated, given Q and T_i.

In principle, all n features should be employed to obtain an accurate density estimation. However, this is too computationally time-consuming. Hence, as we describe next, we use an efficient approximation of this locally data-adaptive kernel density estimator.

A.0.5 Approximation of Locally Data-adaptive Kernel Density Estimate

Assuming that $\bar{\mathbf{f}}_Q^1, \bar{\mathbf{f}}_Q^2, \cdots, \bar{\mathbf{f}}_Q^n$ are i.i.d. given hypothesis $\mathscr{H}_i$, the ML decision rule can be rewritten by taking the log probability of the ML decision rule (A.1) as:

$$\begin{aligned} \widehat{\mathscr{H}_i} &= \arg\max_i \log \widehat{p}(\overline{\mathbf{F}}_Q|\mathscr{H}_i) = \arg\max_i \log \widehat{p}(\bar{\mathbf{f}}_Q^1, \cdots, \bar{\mathbf{f}}_Q^n|\mathscr{H}_i) \\ &= \arg\max_i \sum_{\ell=1}^{n} \log \widehat{p}(\bar{\mathbf{f}}_Q^\ell|\mathscr{H}_i). \end{aligned} \tag{A.5}$$

What we do next is to estimate each local individual probability density $\widehat{p}(\overline{\mathbf{f}}_Q^{\ell}|\mathscr{H}_i)$ separately:

$$\widehat{p}(\overline{\mathbf{f}}_Q^{\ell}|\mathscr{H}_i) = \frac{1}{\beta'} \sum_{j=1}^{n} G^j(\overline{\mathbf{f}}_Q^{\ell} - \overline{\mathbf{f}}_{T_i}^{j}, \mathbf{x}_Q^{\ell} - \mathbf{x}_{T_i}^{j}), \quad \ell = 1, \cdots, n, \tag{A.6}$$

where $\beta' = \sum_{\ell=1}^{n} \sum_{j=1}^{n} G^j(\overline{\mathbf{f}}_Q^{\ell} - \overline{\mathbf{f}}_{T_i}^{j}, \mathbf{x}_Q^{\ell} - \mathbf{x}_{T_i}^{j})$ is a normalization factor. As nicely motivated in [41] and discussed in Chapter 3, since the distribution of the features on the low-dimensional manifold tends to follow a power-law (i.e., long-tail or heavy-tail), it should be sufficient to use just a few features in T_i to get a reasonable estimate of the conditional density $\widehat{p}(\overline{\mathbf{f}}_Q^{\ell}|\mathscr{H}_i)$. Therefore, we consider using a single (spatially nearest) neighbor for the approximation, which yields:

$$\begin{aligned}
\widehat{p}(\overline{\mathbf{f}}_Q^{\ell}|\mathscr{H}_i) &\approx \exp\left(-\frac{1}{2\sigma_r^2}\text{dist}(\overline{\mathbf{f}}_Q^{\ell}, \overline{\mathbf{f}}_{T_i}^{\ell})\right), \quad \ell = 1, \cdots, n, \\
&= \exp\left(\frac{-(\frac{\|\mathbf{f}_Q^{\ell}\|^2}{\|\mathbf{F}_Q\|_F^2} + \frac{\|\mathbf{f}_{T_i}^{\ell}\|^2}{\|\mathbf{F}_{T_i}\|_F^2} - \frac{2\rho(\mathbf{f}_Q^{\ell}, \mathbf{f}_{T_i}^{\ell})\|\mathbf{f}_Q^{\ell}\|\|\mathbf{f}_{T_i}^{\ell}\|}{\|\mathbf{F}_Q\|_F\|\mathbf{F}_{T_i}\|_F})}{2\sigma_r^2}\right).
\end{aligned} \tag{A.7}$$

The approximate version of density estimator using one sample is compared to $\widehat{p}(\overline{\mathbf{F}}_Q|\mathscr{H}_i)$ estimated using all n samples in Fig. **A.2**. Qualitatively, we observe that the resulting estimates are quite similar. More precisely, consistent with [41], we have verified that the use of the approximation takes little away from the performance of the overall algorithm, which is discussed in Section 3.2.1. Since $\log \widehat{p}(\overline{\mathbf{f}}_Q^{\ell}|\mathscr{H}_i)$ is approximately proportional to $-(\frac{\|\mathbf{f}_Q^\ell\|^2}{\|\mathbf{F}_Q\|_F^2}+\frac{\|\mathbf{f}_{T_i}^\ell\|^2}{\|\mathbf{F}_{T_i}\|_F^2}-2\rho(\mathbf{f}_Q^\ell,\mathbf{f}_{T_i}^\ell)\frac{\|\mathbf{f}_Q^\ell\|\|\mathbf{f}_{T_i}^\ell\|}{\|\mathbf{F}_Q\|_F\|\mathbf{F}_{T_i}\|_F})$, the ML decision rule becomes

$$\begin{aligned}
\widehat{\mathscr{H}_i} &= \arg\max_i \sum_{\ell=1}^{n} \log \widehat{p}(\bar{\mathbf{f}}_Q^\ell|\mathscr{H}_i) \\
&\Rightarrow \arg\max_i \sum_{\ell=1}^{n} -\left(\frac{\|\mathbf{f}_Q^\ell\|^2}{\|\mathbf{F}_Q\|_F^2}+\frac{\|\mathbf{f}_{T_i}^\ell\|^2}{\|\mathbf{F}_{T_i}\|_F^2}-\frac{2\rho(\mathbf{f}_Q^\ell,\mathbf{f}_{T_i}^\ell)\|\mathbf{f}_Q^\ell\|\|\mathbf{f}_{T_i}^\ell\|}{\|\mathbf{F}_Q\|_F\|\mathbf{F}_{T_i}\|_F}\right), \\
&= \arg\max_i (-2+2\sum_{\ell=1}^{n} \frac{{\mathbf{f}_Q^\ell}^T \mathbf{f}_{T_i}^\ell}{\|\mathbf{F}_Q\|_F\|\mathbf{F}_{T_i}\|_F}), \\
&= \arg\max_i \sum_{\ell=1}^{n} \frac{{\mathbf{f}_Q^\ell}^T \mathbf{f}_{T_i}^\ell}{\|\mathbf{F}_Q\|_F\|\mathbf{F}_{T_i}\|_F} = \arg\max_i < \frac{\mathbf{F}_Q}{\|\mathbf{F}_Q\|_F}, \frac{\mathbf{F}_{T_i}}{\|\mathbf{F}_{T_i}\|_F} >_F . \qquad \text{(A.8)}
\end{aligned}$$

We can clearly see that the ML decision rule in Equation (A.8) boils down to the computation of the Matrix Cosine Similarity, due to the relationship $< \frac{\mathbf{F}_Q}{\|\mathbf{F}_Q\|_F}, \frac{\mathbf{F}_{T_i}}{\|\mathbf{F}_{T_i}\|_F} >_F \approx \frac{2+\sum_{\ell=1}^{n} \log \widehat{p}(\bar{\mathbf{f}}_Q^\ell|\mathscr{H}_i)}{2}$. While the assumptions leading to the above conclusions may seem somewhat restrictive, in practice they appear to hold true, and they do provide a framework in which the proposed detection algorithm can be considered optimal in the naive Bayes sense.

Appendix B

Controlling The False Discovery Rate

We associate each voxel ($f(\rho_i)$) of the RM with a null hypothesis up to M hypotheses ($\mathscr{H}_0, \cdots, \mathscr{H}_{M-1}$) as:

$\mathscr{H}_0$:	T_0 is not similar to the given query Q	$\Leftrightarrow$	$f(\rho_0) < \tau$,
$\mathscr{H}_1$:	T_1 is not similar to the given query Q	$\Leftrightarrow$	$f(\rho_1) < \tau$,
$\vdots$	$\vdots$		$\vdots$
$\mathscr{H}_{M-1}$:	T_{M-1} is not similar to the given query Q	$\Leftrightarrow$	$f(\rho_{M-1}) < \tau$.

Table B.1: M-ary null hypotheses.

where τ is a threshold for detection. Suppose that there are m_0 true null hypotheses among the M test hypotheses. Let R denote the number of hypotheses rejected. This observable random variable R can be decomposed as $V + S$, where V is the number of *incorrectly* rejected null hypotheses and S is the number of *correctly* rejected null hypotheses. The proportion of errors committed by falsely rejecting null hypotheses can be viewed through $\frac{V}{R}$. Let U be the unobservable random quotient,

$$U = \begin{cases} \frac{V}{R} & \text{if} \quad R > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{B.1}$$

The false discovery rate (FDR) is defined as $\mathbf{E}(U)$, the expected error rate. The Benjamini-Hochberg procedure proposed in [132] controls the FDR at a desired level α, while maximizing $\mathbf{E}(R)$. Let $\{p_0, p_1, \cdots, p_{M-1}\}$ denote the p-values corresponding to the test statistics $\{f(\rho_0), f(\rho_1), \cdots, f(\rho_{M-1})\}$ and $p_{(0)} \leq p_{(1)} \leq \cdots \leq p_{(M-1)}$ denote the ordered p-values corresponding to the hypotheses $\{\mathscr{H}_{(0)}, \mathscr{H}_{(1)}, \cdots, \mathscr{H}_{(M-1)}\}$. By definition, $p_i = 1 - P_{\mathscr{H}_i}$ where $P_{\mathscr{H}_i}$ is the cumulative distribution function of resemblance volume under the null hypothesis $\mathscr{H}_i$. The FDR-controlling procedure is easily implemented. For the M pixels being tested, the general procedure is as follows:

1. Select a desired FDR bound α between 0 and 1. This is the maximum FDR that we are willing to tolerate on average.

2. Order the p values from the smallest to largest:

   $p_{(0)} \leq p_{(1)} \leq \cdots \leq p_{(M-1)}$

   Let $f(\rho_{(i)})$ be the voxel corresponding to $p_{(i)}$.

3. Let γ be the largest i for which

   $p_{(i)} \leq \frac{i}{M}\alpha$.

4. Identify the threshold τ corresponding to $p_{(\gamma)}$ and declare that the pixels of RM which is above τ contain similar actions to the given query Q.

Bibliography

[1] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," *IEEE International Conference on Computer Vision (ICCV)*, 1998.

[2] A. Buades, B. Coll, and J. M. Morel, "Nonlocal image and movie denoising," *International Journal of Computer Vision (IJCV)*, vol. 76, no. 2, pp. 123–139, 2008.

[3] N. Dalal and B. Triggs, "Histogram of oriented gradietns for human detection," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[4] D. Gao, V. Mahadevan, and N. Vasconcelos, "On the plausibility of the discriminant center-surround hypothesis for visual saliency," *Journal of Vision*, vol. 8, no. 7, pp. 13,1–18, 2008.

[5] X. Hou and L. Zhang, "Saliency detection: A spectral residual approach," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[6] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 29, no. 12, pp. 2247–2253, December 2007.

[7] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," *IEEE Conference on Pattern Recognition (ICPR)*, June 2004.

[8] N. Bruce and J. Tsotsos, "Saliency based on information maximization," *In Advances in Neural Information Processing Systems (NIPS)*, vol. 18, pp. 155–162, 2006.

[9] A. Treisman and G. Gelade, "A feature-integration theory of attention," *Cognitive Psychology*, vol. 12, no. 1, pp. 97–136, 1980.

[10] L. Itti and P. Baldi, "A principled approach to detecting surprising events in video," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 631–637, 2005.

[11] E. Shechtman and M. Irani, "Matching local self-similarities across images and videos," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[12] F. Devernay, "A non-maxima suppression method for edge detection with sub-pixel accuracy," *Technical report, INRIA*, no. RR-2724, 1995.

[13] S. Agarwal, A. Awan, and D. Roth, "Learning to detect objects in images via a sparse, part-based representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 26, no. 11, pp. 1475–1490, November 2004.

[14] B. Wu and R. Nevatia, "Simultaneous object detection and segmentation by boosting local shape feature based classifier," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2007.

[15] A. Kappor and J. Winn, "Located hidden random fields: Learning discriminative parts for object detection," *In Proc. European Conference Computer Vision (ECCV)*, vol. 3954, pp. 302–315, May 2006.

[16] J. Mutch and D. G. Lowe, "Multiclass object recognition with sparse, localized features," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[17] H. A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 20, no. 1, pp. 23–36, 1998.

[18] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision (IJCV)*, vol. 20, pp. 91–100, 2004. [Online]. Available: citeseer.ist.psu.edu/lowe04distinctive.html

[19] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool, "A comparison of affine region detectors," *International Journal of Computer Vision (IJCV)*, vol. 65, no. 1, pp. 43–72, 2005.

[20] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 24, no. 4, pp. 509–522, April 2002.

[21] E. Shechtman and M. Irani, "Space-time behavior-based correlation -or- how to tell if two underlying motion fields are similar without computing them?" *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 29, no. 11, pp. 2045–2056, November 2007.

[22] H. Ning, T. Han, D. Walther, M. Liu, and T. Huang, "Hierarchical space-time model enabling efficient search for human actions," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 6, pp. 808–820, 2009.

[23] I. Laptev and P. Perez, "Retrieving actions in movies," *IEEE International Conference on Computer Vision (ICCV)*, 2007.

[24] S. Hinterstoisser, V. Lepetit, S. Ilic, P. Fua, and N. Navab, "Dominant orientation templates for real-time detection of texture-less objects," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[25] G. B. Huang, M. Ramesh, T. Berg, and E. L. Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," in *University of massachusetts, Amherst, Technical Report 07-49*, 2007.

[26] L. Wolf, T. Hassner, and Y. Taigman, "Similarity scores based on background samples," in *Asian Conference on Computer Vision (ACCV)*, 2009.

[27] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman, "Learning object categories from Google's image search," *IEEE Internatioanl Conference on Computer Vision (ICCV)*, 2005.

[28] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[29] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision (IJCV)*, vol. 57, no. 2, pp. 137–154, 2004.

[30] J. Sivic and A. Zisserman, "Video Google: a text retrieval approach to object matching in videos," *IEEE International Conference on Computer Vision (ICCV)*, 2003.

[31] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[32] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Workshop on Generative Model-based Vision*, 2004.

[33] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 32, no. 9, pp. 1–20, 2010.

[34] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset,"

California Institute of Technology, Tech. Rep. 7694, 2007. [Online]. Available: http://authors.library.caltech.edu/7694

[35] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The PASCAL Visual Object Classes Challenge 2009 (VOC2009) Results.”

[36] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *IEEE Conference on Computer Vision & Pattern Recognition*, 2009.

[37] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld, “Learning realistic human actions from movies,” in *IEEE Conference on Computer Vision & Pattern Recognition*, 2008.

[38] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, “LabelMe: a database and web-based tool for image annotation,” *International Journal of Computer Vision (IJCV)*, vol. 77, no. 1-3, pp. 157–173, 2008.

[39] A. Torralba, R. Fergus, and W. Freeman, “80 million tiny images: a large dataset for non-parametric object and scene recognition,” *IEEE Transcations on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 30, no. 11, pp. 1958–1970, 2008.

[40] J. Hays and A. Efros, “Scene completion using millions of photographs,” *ACM SIGGRAPH*, vol. 26, no. 3, 2007.

[41] O. Boiman, E. Shechtman, and M. Irani, “In defense of nearest-neighbor based image classification,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[42] H. Zhang, A. Berg, M. Maire, and J. Malik, “SVM-KNN: Discriminative nearest neighbor classification for visual category recognition,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[43] K. Grauman and T. Darrell, "The pyramid match kernel: Efficient learning with sets of features," *Journal of Machine Learning Research*, vol. 8, pp. 725–760, 2007.

[44] C. Yeo, P. Ahammad, K. Ramchandran, and S. S. Satry, "High-speed action recognition and localization in compressed domain videos," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, pp. 1006–1015, Aug 2008.

[45] H. J. Seo and P. Milanfar, "Generic human action detection from a single example," *IEEE International Conference on Computer Vision(ICCV)*, Sep 2009.

[46] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," *IEEE Transactions on Image Processing (TIP)*, vol. 16, no. 2, pp. 349–366, February 2007.

[47] H. J. Seo and P. Milanfar, "Nonparametric bottom-up saliency detection by self-resemblance," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1st International Workshop on Visual Scene Understanding (ViSU 09)*, Apr 2009.

[48] ——, "Static and space-time visual saliency detection by self-resemblance," *The Journal of Vision,*, vol. 9(12), no. 15, pp. 1–27, 2009. [Online]. Available: http://journalofvision.org/9/12/15/,doi:10.1167/9.12.15

[49] ——, "Training-free, generic object detection using locally adaptive regression kernels," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 1688–1704, September 2010.

[50] ——, "Generic human action recognition from a single example," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 867–882, May 2009.

[51] ——, "Using local regression kernels for statistical object detection," *IEEE International Conference on Image Processing (ICIP)*, October 2008.

[52] H. J. Seo, G. Bradski, and P. Milanfar, "Scalable LARK descriptors for real-time object detection and recognition," *Under review*, 2011.

[53] H. J. Seo and P. Milanfar, "Nonparametric face verification using a novel representation," *To appear in IEEE Trans. on Information Forensics and Security*, 2011.

[54] ——, "A non-parametric approach to automatric change detection in MRI images of the brain," *IEEE International Symposium on Biomedical Imaging (ISBI): From Nano to Macro*, 2009.

[55] R. Kimmel, *Numerical Geometry of Images*. Springer, 2003.

[56] G. Peyr*é* and L. Cohen, "Geodesic methods for shape and surface processing," *In Advances in Computational Vision and Medical Image Processing: Methods and Applications (springer)*, vol. 13, pp. 29–56, 2008.

[57] K. Kavukcuoglu, M. Ranzato, R. Fergus, and Y. LeCun, "Learning invariant features through topographic filter maps," *IEEE Conference on Computer Vision and Computer Vision (CVPR)*, 2009.

[58] D. Pelli, "Close encounters: An artist shows that size affects shape," *Science*, vol. 285, pp. 884–886, 1999.

[59] J. G. Ravin and P. M. Odell, "Pixels and painting," *Archives of Ophthamology*, vol. 126, no. 8, pp. 1148–1151, 2008.

[60] L. Harmon and B. Julesz, "Masking in visual recognition: effects of two-dimensional filtered noise," *Science*, vol. 180, no. 91, pp. 1194–1197, 1973.

[61] H. Takeda, P. Milanfar, M. Protter, and M. Elad, "Super-resolution without explicit subpixel motion estimation," *IEEE Transactions on Image Processing (TIP)*, vol. 18, no. 9, pp. 1958–1975, September 2009.

[62] S. Paris and F. Durand, "A fast approximation of the bilateral filter using a signal processing approach," *European Conference on Computer Vision (ECCV)*, 2006.

[63] Y. Ke and R. Sukthankar, “PCA-SIFT: A more distinctive representation for local image descriptors,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[64] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, “Learning realistic human actions from movies,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[65] A. Klaser, M. Marszalek, and C. Schmid, “A spatio-temporal descriptor based on 3d-gradients,” *British Machine Vision Conference (BMVC)*, 2008.

[66] L. Itti and P. Baldi, “Bayesian surprise attracts human attention,” *In Advances in Neural Information Processing Systems (NIPS)*, vol. 18, pp. 1–8, 2006.

[67] L. Itti and C. Koch, “Computational modeling of visual attention,” *Nature Rev. Neurosci*, vol. 2, no. 3, pp. 194–203, 2001.

[68] A. Yarbus, “Eye movements and vision,” *New York: Plenum*, 1967.

[69] M. Chun and J. M. Wolfe, “Visual attention,” *Blackwell Handbook of Perception, Blackwell Publishers Ltd.*, pp. 272–310, 2001.

[70] D. Gao and N. Vasconcelos, “Integrated learning of saliency, complex features, and object detectors from cluttered scenes,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[71] L. Itti, C. Koch, and E. Niebur, “A model of saliency-based visual attention for rapid scene analysis,” *IEEE Transcations on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 20, pp. 1254–1259, 1998.

[72] L. Zhang, A. Deshpande, and X. Chen, “Denoising vs. deblurring: Hdr imaging techniques using moving cameras,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[73] L. Zhang, M. Tong, and G. Cottrell, "SUNDAy: Saliency using natural statistics for dynamic analysis of scenes," *In Proceedings of the 31st Annual Cognitive Science Conference, Amsterdam, Netherlands. Mahwah: Lawrence Erlbaum*, 2009.

[74] C. Kanan, M. Tong, L. Zhang, and G. Cottrell, "SUN: Top-down saliency using natural statistics," *Visual Cognition*, vol. 17, no. 6, pp. 979–1003, 2009.

[75] S. Marat, T. Phuoc, L. Granjon, N. Guyader, D. Pellerin, and A. Guerin-Dugue, "Modelling spatio-temporal saliency to predict gaze direction for short videos," *International Journal of Computer Vision (IJCV)*, vol. 82, no. 3, pp. 231–243, 2009.

[76] X. Hou and L. Zhang, "Dynamic visual attention: Searching for coding length increments," *In Advances in Neural Information Processing Systems (NIPS)*, vol. 21, pp. 681–688, 2008.

[77] S. Marat, M. Guironnet, and D. Pellerin, "Video summarization using a visual attentional model," *EUSIPCO, EURASIP*, pp. 1784–1788, 2007.

[78] Q. Ma and L. Zhang, "Saliency-based image quality assessment criterion," *Advanced Intelligent Computing Theories and Applications. With Aspects of Theoretical and Methodological Issues (LNCS)*, vol. 5226, pp. 1124–1133, 2008.

[79] A. Niassi, O. LeMeur, P. Lecallet, and D. barba, "Does where you gaze on an image affect your perception of quality? Applying visual attention to image quality metric," *IEEE International Conference on Image Processing (ICIP)*, 2007.

[80] D. Gao and N. Vasconcelos, "Discriminant saliency for visual recognition from cluttered scenes," *In Advances in Neural Information Processing Systems (NIPS)*, vol. 17, pp. 481–488, 2004.

[81] V. Mahadevan and N. Vasconcelos, "Background subtraction in highly dynamic scenes," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[82] A. Oliva, A. Torralba, M. Castelhano, and J. Henderson, “Top-down control of visual attention in object detection,” *IEEE International Conference on Image Processing (ICIP)*, 2003.

[83] W. Kienzle, F. Wichmann, B. Scholkopf, and M. Franz, “A nonparametric approach to bottom-up visual saliency,” *In Advances in Neural Information Processing Systems (NIPS)*, vol. 19, pp. 689–696, 2007.

[84] C. Guo, Q. Ma, and L. Zhang, “Spatio-temporal saliency detection using phase spectrum of quaternion Fourier transform,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[85] M. Bregonzio, S. Gong, and T. Xiang, “Recognising actions as clouds of space-time interest points,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[86] E. R. H. R. Tavakoli and J. Heikkila, “Fast and efficient saliency detection using sparse sampling and kernel density estimation,” *Lecture Notes in Computer Science*, vol. 6688, pp. 666–675, 2011.

[87] B. Silverman, *Density estimation for statistics and data analysis*. Monographs on Statistics and Applied Probability 26, New York: Chapman & Hall, 1986.

[88] Y. Bengio, H. Larochelle, and P. Vincent, “Non-local manifold Parzen windows,” *In Advances in Neural Information Processing Systems (NIPS)*, vol. 18, pp. 115–122, 2005.

[89] T. Brox, B. Rosenhahn, and H.-P. S. D. Cremers, “Nonparametric density estimation with adaptive anisotropic kernels for human motion tracking,” *2nd. Workshop on Human Motion, Springer-Verlag Berlin Heidelberg (LNCS)*, vol. 4814, pp. 152–165, 2007.

[90] P. Vincent and Y. Bengio, "Manifold Parzen windows," *In Advances in Neural Information Processing Systems (NIPS)*, vol. 15, pp. 825–832, 2003.

[91] Y. Fu and T. S. Huang, "Image classification using correlation tensor analysis," *IEEE Transactions on Image Processing (TIP)*, vol. 17, no. 2, pp. 226–234, 2008.

[92] Y. Fu, S. Yan, and T. S. Huang, "Correlation metric for generalized feature extraction," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 30, no. 12, pp. 2229–2235, 2008.

[93] Y. Ma, S. Lao, E. Takikawa, and M. Kawade, "Discriminant analysis in correlation similarity measure space," *IEEE International Conference on Machine Learning (ICML)*, 2007.

[94] O. Meur, P. L. Callet, and D. Barba, "Predicting visual fixations on video based on low-level visual features," *Vision Research*, vol. 47, pp. 2483–2498, 2007.

[95] K. vande Sande, T. Gevers, and C. Snoek, "Evaluation of color descriptors for object and scene recognition," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[96] T. Judd, F. Durand, and A. Torralba, "Fixations on low-resolution images," *Journal of Vision*, vol. 11, no. 4, pp. 1–20, 2011.

[97] J. Wolfe, "Guided search 2.0: A revised model of guided search," *Psychonomic bulletic* & *Reveiw*, vol. 1, pp. 202–238, 1994.

[98] S. Kullback, "Information theory and statistics," *Dover Publications*, 1968.

[99] M. Shahram, "Statistical and information-theoretic analysis of resolution in imaging and array processing," *Ph.D thesis, University of California, Santa Cruz*, 2005.

[100] J. Ponce, M. Hebert, C. Schmid, and A. Zisserman, *Toward Category-Level Object Recognition*. Lecture Notes in Computer Science, 2007.

[101] C. H. Lampert, M. B. Blaschko, and T. Hofmann, "Beyond sliding windows: Object localization by efficient subwindow search," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[102] H. Masnadi-Shirazi and N. Vasconcelos, "High detection-rate cascades for real-time object detection," *IEEE International Conference on Computer Vision (ICCV)*, 2007.

[103] T. Darrell and A. Pentland, "Classifying hand gestures with a view-based distributed representation," *In Advances in Neural Information Processing Systems*, vol. 6, pp. 945–952, 1993.

[104] T. Starner and A. Pentland, "Visual recognition of American sign language using hidden Markov model," *International Workshop on Automatic Face and Gesture Recognition*, 1995.

[105] C. Carlsson and J. Sullivan, "Action recognition by shape matching to key frame," *Workshop on Models Versus Examplars in Computer Vision*, 2001.

[106] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 23, no. 3, pp. 1257–1265, March 2001.

[107] S. Ali and M. Shah, "Human action recognition in videos using kinematic features and multiple instance learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 33, no. 2, pp. 288–303, Feb 2010.

[108] J. Niebles and L. Fei-Fei, "A hierarchical models of shape and appearance for human action classification," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2007.

[109] Z. Laptev and T. Lindeberg, "Space-time interest points," *IEEE International Conference on Computer Vision (ICCV)*, October 2003.

[110] A. Oikonomopoulous, I. Patras, and M. Pantic, “Spationtemporal saliency for human action recognition,” *IEEE International Conference on Multimedia and Expo (ICME)*, 2005.

[111] P. Scovanner, S. Ali, and M. Shah, “A 3-dimensional SIFT descriptor and its application to action recognition,” *ACM Multimedia*, 2007.

[112] Y. Ke, R. Sukthankar, and M. Hebert, “Efficient visual event detection using volumetric features,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[113] T. Kim and R. Cipolla, “Canonical correlation analysis of video volume tensors for action categorization and detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 8, pp. 1415–1428, 2009.

[114] P. Turaga, R. Chellappa, V. Subrahmanian, and O. Udrea, “Machine recognition of human activities: A survey,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 11, pp. 1473–1488, November 2008.

[115] J. Niebles, H. Wang, and L. Fei-Fei, “Unsupervised learning of human action categories usig spatial-temporal words,” *International Journal of Computer Vision (IJCV)*, vol. 79, no. 3, pp. 299–318, March 2008.

[116] L. Q. L. Yuan, J. Sun and H.-Y. Shum, “Image deblurring with blurred/noisy image pairs,” *ACM Transactions on Graphics (SIGGRAPH)*, 2007.

[117] Y. Ke, R. Sukthankar, and M. Hebert, “Event detection in crowded videos,” *IEEE International Conference on Computer Vision (ICCV)*, 2007.

[118] C. Liu, “The Bayes decision rule induced similarity measures,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 29, no. 6, pp. 1086–1090, 2007.

[119] J. W. Schneider and P. Borlund, "Matrix comparison, part 1: Motivation and important issues for measuring the resemblance between proximity measures or ordination results," *Journal of the American Society for Information Science and Technology*, vol. 58, no. 11, pp. 1586–1595, 2007.

[120] J. Rodgers and W. Nicewander, "Thirteen ways to look at the correlation coefficient," *The American Statistician*, vol. 42, no. 1, pp. 59–66, 1988.

[121] T. Tuytelaar and C. Schmid, "Vector quantizing feature space with a regular lattice," *IEEE International Conference on Computer Vision (ICCV)*, October 2007.

[122] F. Jurie and B. Triggs, "Creating efficient codebooks for visual recognition," *IEEE International Conference on Computer Vision (ICCV)*, 2005.

[123] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[124] X. He, S. Yan, Y.Hu, P. Niyogi, and H.-J. Zhang, "Face recognition using Laplacianfaces," *IEEE Transacions on Pattern Analaysis and Machine Intelligence (PAMI)*, vol. 27, no. 3, pp. 328–340, 2005.

[125] P. Ahlgren, B. Jarneving, and R. Rousseau, "Requirements for a cocitation similarity measure, with special reference to Pearson's correlation coefficient," *Journal of the American Society for Information Science and Technology*, vol. 54, no. 6, pp. 550–560, 2003.

[126] Y. Escoufier, "Operator related to a data matrix: a survey," *Compstat 2006 - Proceedings in Computational Statistics, 17th Symposium Held in Rome*, pp. 285–297, 2006.

[127] M. Tatsuoka, *Multivariate Analysis*. Macmillan, 1988.

[128] R. J. Rummel, *Applied Factor Analysis*. Evanston, Ill.: Northwestern University Press, 1970.

[129] P. Horst, *Matrix Algebra for Social Scientists*. New York: Holt, Rinehart, and Winston, 1963.

[130] T. Calinski, M. Krzysko, and W. Wolynski, "A comparison of some tests for determining the number of nonzero canonical correlations," *Communication in Statistics, Simulation and Computation*, vol. 35, pp. 727–749, 2006.

[131] R. Duda, P. Hart, and D. Stork, *Pattern Classification, 2nd Edition*. New York: John Wiley and Sons Inc, 2000.

[132] Y. Benjamini and Y. Hochberg, "Controling the false discovery rate: A practical and powerful approach to multiple testing," *Journal of the Royal Statistical Society. Series B*, vol. 57, no. 1, pp. 289–300, 1995.

[133] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayer, "Attribute and simile and classifiers for face verification," in *IEEE International Conference on Computer Vision (ICCV)*, 2009.

[134] D. M. Chen, S. S. Tsai, V. Chandrasekhar, G. Takacs, J. Singh, and B. Girod, "Tree histogram coding for mobile image matching," *IEEE Data Compression Conference (DCC)*, March 2009.

[135] V. Chandrasekhar, G. Takacs, S. Tsai, D. Chen, R. Grzeszczuk, and B. Girod, "CHoG: Compressed histogram of gradients: A low bit-rate feature descriptor," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[136] N. Petkov and E. Subramanian, "Motion detection, noise reduction, texture suppression and contour enhancement by spatiotemporal gabor filters with surround inhibition," *Biological Cybernetics*, vol. 97, pp. 423–439, 2007.

[137] I. N. Juenjo, E. Dexter, I. Laptev, and P. PerezPerez, "View-independent action recognition from temporal self-similarities," *IEEE Transations on Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp. 172–185, Jan 2010.

[138] J. Liu, S. Ali, and M. Shah, "Recognizing human actions using multiple features," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2008.

[139] H. Jhuang, T. Serre, L. Wolf, and T. Poggio, "A biologically inspired system for action recogntion," *IEEE International Conference on Computer Vision(ICCV)*, October 2007.

[140] D. Batra, T. Chen, and R. Sukthankar, "Space-time shapelets for action recogntion," *IEEE Workshop on Motion and video Computing (WMVC)*, January 2008.

[141] Z. Zhang, Y. Hu, S. Chan, and L.-T. Chia, "Motion context: A new representation for human action recognition," *European Conference on Computer Vision (ECCV)*, 2008.

[142] A. Fathi and G. Mori, "Action recognition by learning mid-level motion features," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[143] M. Sun, G. Bradski, B. Xu, and S. Savarese, "Depth-encoded Hough voting for joint object detection and shape recovery," *European Conference on Computer Vision (ECCV)*, 2010.

[144] K. Schindler and L. V. Gool, "Action snippets: How many frames does human action recognition require," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[145] J. Liu and M. Shah, "Learning human actions via information maximization," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[146] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," *In proceeding of Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS)*, October 2005.

[147] A. Wong and J. Orchard, "A nonlocal-means approach to examplar-based inpainting," *IEEE International Conference on Image Processing*, 2008.

[148] K. Rapantzikos, Y. Avrithis, and S. Kollias, “Dense saliency-based spationtemporal feature points for action recognition,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[149] A. Bandopadhay and J. Fu, “Searching parameter spaces with noisy linear constraints,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1988.

[150] G. Medioni, I. Cohen, F. Bremond, S. Hongeng, and R. Nevatia, “Event detection and analysis from video streams,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 23, no. 8, pp. 873–890, August 2001.

[151] T. Veit, F. Cao, and P. Bouthemy, “Probabilistic parameter-free motion detection,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2004.

[152] M. Marszalek, I. Laptev, and C. Schmid, “Actions in context,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[153] S. K. Divvala, D. Hoiem, J. H. Hays, A. Efros, and M. Hebert, “An empirical study of context in object detection,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[154] A. Oliva and A. Torralba, “The role of context in object recognition,” *Trends in Cognitive Science*, November 2007.

[155] D. Gavrila and V. Philomin, “Real-time object detection for smart vehicles,” *IEEE International Conference on Computer Vision (ICCV)*, 1999.

[156] V. Ferrari, F. Jurie, and C. Schmid, “From images to shape models for object detection,” *International Journal of Computer Vision (IJCV)*, vol. 87, no. 3, pp. 284–303, 2009.

[157] R.J.Radke, S. Andra, O. Al-Lofahi, and B. Roysam, "Image change detection algorithms: A systematic survey," *IEEE Transactions on Image Processing (TIP)*, vol. 14, no. 3, pp. 294–307, March 2005.

[158] M. Bosc, F. Heitz, J. Armspach, I. Namer, D. Gounot, and L. Pumbach, "Automatic change detection in multimodal serial MRI: applicaton to Multiple Sclerosis lesion evolution," *NeuroImage*, vol. 20, pp. 643–656, July 2003.

[159] J. W. Patriarche and B. J. Erickson, "Part 1. automated change detection and characterization in serial MR studies of brain tumor patients," *Journal of Digital Imaging*, vol. 20, no. 3, pp. 203–222, September 2007.

[160] F. Rousseau, S. Faisan, F. Heitz, J. Armspach, Y.Chevalier, F.Blanc, J. Seze, and L. Rumbach, "An a contario approach for change detection in 3D multimodal images: Application to Multiple Sclerosis in MRI," *IEEE Engineering in Medicine and Biology Society (EMBS)*, pp. 2069–2072, August 2007.

[161] T. Pecot and C. Kervrann, "Patch-based markov models for change detection in image sequence analysis," *The International Workshop on Local and Non-Local Approximation in Image Processing*, August 2008.

[162] S. Shen, A. Szameitat, and A. Sterr, "Detection of infarct lesions from single MRI modality using inconsistency between voxel intensity and spatial location-a 3D automatic approach," *IEEE Transactions on Information Technology in Biomedicine*, vol. 12, no. 4, pp. 532–540, 2008.

[163] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *IEEE Conference on Computer Vision and Computer Vision (CVPR)*, 1991.

[164] G. B. Huang, V. Jain, and E. Leonard-Miller, "Unsupervised joint alignment of complex images," in *IEEE International Conference on Computer Vision (ICCV)*, 2007.

[165] L. Wolf, T. Hassner, and Y. Taigman, "Descriptor based methods in the wild," in *Faces in Real-Life Image Workshop in European Conference on Computer Vision (ECCV)*, 2008.

[166] C. Sanderson and B. C. Lovell, "Multi-region probabilistic histograms for robust and scalable identity inference," in *International Conference on Biometric (ICB)*, 2009.

[167] N. Pinto, J. J. Dicarlo, and D. D. Cox, "How far can you get with a modern face recognition test set using only simple features?" in *IEEE Conference on Computer Vision and Computer Vision (CVPR)*, 2009.

[168] M. Guillaumin, J. Verbeek, and C. Schmid, "Is that you? Metric learning approaches for face identification," in *IEEE International Conference on Computer Vision (ICCV)*, 2009.

[169] Y. Taigman, L. Wolf, and T. Hassner, "Multiple one-shots for utilizing class lable information," in *British Machine Vision Conference (BMVC)*, 2009.

[170] Z. Cao, Q. Yin, X. Tang, and J. Sun, "Face recognition with learning-based descriptor," in *IEEE Conference on Computer Vision and Computer Vision (CVPR)*, 2010.

[171] M. Yang, D. Kriegman, and N. Ahuja, "Detecting faces in images: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 24, no. 1, pp. 34–58, 2002.

[172] P. Phillips, P. Flynn, T. Scruggs, K. Bowyer, C. Jin, K. Hoffman, J. Marques, M. Jaesik, and W. Worek, "Overview of the face recognition grand challenge," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[173] L. Wolf, T. Hassner, and Y. Taigman, "The one-shot similarity kernel," in *IEEE International Conference on Computer Vision (ICCV)*, 2009.

[174] G. Hua and A. Akbarzadeh, “A robust elastic and partial matching metric for face recognition,” in *IEEE International Conference on Computer Vision (ICCV)*, 2009.

[175] T. Ojala, M. Pietikainen, and T. Maenpaa, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 24, no. 7, pp. 971–987, 2002.

[176] L. Fei-Fei, R. Fergus, and P. Perona, “One-shot learning of object categories,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 28, no. 4, pp. 594–611, 2006.

[177] J. Sherman and W. J. Morrison, “Adjustment of an inverse matrix corresponding to changes in the elements of a given column or a given row of the original matrix,” *Annals of Mathematical Statistics (Abstract)*, vol. 20, p. 621, 1949.

[178] H. Wang, S. Z. Li, and Y. Wang, “Generalized quotient image,” *IEEE Conference on Computer Vision and Computer Vision (CVPR)*, 2004.

[179] Y. Freund, S. Dasgupta, M. Kabra, and N. Verma, “Learning the structure of manifolds using random projections,” *In Advances in Neural Information Processing Systems (NIPS)*, 2007.

[180] J. R. del Solar, R. Verscheae, and M. Correa, “Recognition of faces in unconstrained enviornments: A comparative study,” in *EURASIP Journal on Advances in Signal Processing (Recent Advances in Biometric Systems: A Signal Processing Perspective), Vol. 2009, Article ID 184617, 19 pages*.

[181] C. E. Connor, “A new viewpoint on faces,” *Science*, vol. 330, pp. 764–765, 2010.

[182] Y. Liu, K. L. Schmidt, J. F. Cohn, and S. Mitra, “Facial asymmetry quantification

for expression invariant human identification." *Computer Vision and Image Understanding*, vol. 91, pp. 138–159, 2003.

[183] B. Kulis and K. Grauman, "Kernelized locality-sensitive hashing for scalable image search," *IEEE International Conference on Computer Vision (ICCV)*, 2009.

[184] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2009.

[185] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," in *Technical Report. California Institute of Technology*, 2007.

[186] J. C. van Gemert, J. Geusebroek, C. Veenman, and A. W. M. Smeulders, "Kernel codebooks for scene categorization," in *ECCV*, 2008.

[187] J. Wang, J. Yang, K. Yu, F. Lv, and Y. Gong, "Locality-constrained linear coding for image classification," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[188] A. Berg, J. Deng, and F.-F. Li, "IMAGENET large scale visual recognition challenge 2010 (ilsvrc2010)," http://www.image-net.org/challenges/LSVRC/2010/index.

[189] X. Sun, H. Kashima, T. Matsuzaki, and N. Ueda, "Averaged stochastic gradient descent with feedback: An accurate, robust, and fast training method," in *IEEE International Conference on Data Mining (ICDM)*, 2010.

[190] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, June 2010.

[191] D. Mount and S. Arya, "ANN: A library for approximate nearest neighbor searching," in *In CGC 2nd Annual Workshop on Comp. Geometry*, 1007.

[192] S. Arya and H.-Y. A. Fu, "Expected-case complexity of approximate nearest neighbor searching," in *In Symposium on Discrete Algorithms*, 2000.

[193] J. Matas, O. Chum, M. Urvan, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal region," *British Machine Vision Conference(BMVC)*, 2002.

[194] J. Bossu, N. Hautiére, and J.-P. Tarel, "Rain or snow detection in image sequences through use of a histogram of orientation of streaks," *International Journal of Comput Vision*, vol. 93, pp. 348–367, 2011.

[195] P. Hennings-Yeomans, S. Baker, and B. Kumar, "Simultaneous super-resolution and feature extraction for recognition of low-resolution faces," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[196] H. Takeda, S. Farsiu, and P. Milanfar, "Deblurring using regularized locally-adaptive kernel regression," *IEEE Transactions on Image Processing (TIP)*, vol. 17, no. 4, pp. 550–563, April 2008.

[197] P. Milanfar, "A tour of modern image processing," *Invited feature article in review IEEE Signal Processing Magazine*, 2011. [Online]. Available: http://users.soe.ucsc.edu/~milanfar/publications/journal/ModernTour_final.pdf

[198] H. Seo and P. Milanfar, "Robust flash denoising/deblurring by iterative guided filtering," *Submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*.

[199] ——, "Computational photography using a pair of flash/no-flash images by iterative guided filtering," in *Submitted to IEEE International Conference on Computer Vision (ICCV)*, 2011.

[200] S. Kay, *Fundamentals of Statistical Signal Processing, Volume1, Estimation Theory*. Prentice Hall, 1993.