Doctoral Dissertation

A Study of Local Binary Pattern Features for Effective Feature Matching

NGUYEN, Thao Ngoc

Supervisor: **Professor Kazunori MIYATA**

School of Knowledge Science
Japan Advanced Institute of Science and Technology

September 2014

Abstract

"We are drowning in information but starved for knowledge." (John Naisbitt, Megatrends, 1982) is a famous quote that best describes our status in this technology era. That is, the data is produced at an incredible rate while we have little ability to analyze it. The numerical and text data has been well treated in computers, thanks to their statistical analysis power granted by the data mining scientists. However, processing visual data has remained challenging since computers cannot interpret the data in the same manner as human. Therefore, how to represent visual data in computers and let them process it semantically has become a vital question nowadays.

The dissertation approaches the above question from a computer vision aspect. It searches for effective means to reconstruct the image properties in computers and hence improves the quality of the visual analysis. In this way, the information in images are correctly transformed from the non-declarative form to the declarative one, allowing us to manipulate the data easily on computers and gain insights into complex problems. This is expected to bring human toward the discovery of valuable visual knowledge.

The Local binary pattern (LBP) is studied comprehensively to enhance the representation of image properties. It is a popular family of visual features in computer vision due to their high discriminative power and computationally simplicity. Specifically, the effectiveness of LBP features in three important tasks, including interest region description, pedestrian detection and background subtraction, is thoroughly examined. This demonstrates the robustness and generality of LBP in various visual tasks and thus well facilitating computers in interpreting different kinds of visual information.

The first contribution of this dissertation is the two novel LBP features for describing salient regions in images. Their underlying concepts are straightforward and computationally simple, thus they are suitable for many types of applications, especially those emphasizing the processing speed. The features are robust to different photometric and geometric image transformations, enabling us to achieve accurate correspondences between parts of images. These properties help the proposed interest region descriptors address two competing factors, matching accuracy and computational cost, at the same time, while this is still a critical issue for several modern descriptors using other features, such as gradient or pixel intensity. The success of novel LBP features motivates their

development in higher-level tasks such as pedestrian detection, background subtraction and panorama image stitching.

The second contribution is a robust pedestrian detector using the proposed LBP feature in the first contribution. The encoding of this LBP is revised to better characterize edges along diagonal and vertical directions, which are most visible and meaningful details in an upright pedestrian body, so that the feature well distinguishes pedestrians from other objects in the image. The proposed detector combines LBP with color channels and gradient histogram to represent the subjects in different aspects, namely texture, color and gradient changes in magnitude and orientation. The advanced learning framework of [34] is adopted to resolve the computational bottleneck in constructing the feature pyramid. In this way, the proposed detector effectively identifies subjects of various poses in different challenging environments while achieving a speed of 15.2 frame per second (fps). This encourages its implementation in practical systems like surveillance, car driver assistant and human-computer interaction.

The third contribution is a multi-layer background modeling framework to extract moving objects from a video sequence. This framework models the scene background by processing consecutive frames through two layers: the block-wise layer considers blocks of pixels while the pixel-wise layer manipulates each individual pixel. The proposed LBP feature in the first contribution is used to represent the texture in a block, thus the framework is more robust to illumination changes and shadows, which frequently occur in background subtraction. The multi-layer framework operates in a coarse-to-fine manner to better reduce the errors than the traditional Mixture of Gaussians approach. It supports the object analysis for surveillance, video segmentation and event detection.

The final contribution is the introduction of an effective surveillance system that automates the detection of pedestrians in the monitored area. The three proposed techniques have been integrated perfectly to produce a unified system. The proposed system first extracts foreground regions using the background modeling framework in the third contribution then finds pedestrians in these regions with the pedestrian detector in the second contribution. When multiple cameras are used, a panorama view is created with the help of LBP features in the first contribution. This demonstrates the generality of the proposed LBP in the sense that they can participate effectively in all phases of the surveillance process. This doctoral research has been successful in improving the visual perception of computers to a semantic level. It therefore contributes to the computer vision as well as the knowledge discovery in Knowledge science.

Declaration

This dissertation is my own work under the supervision of the professors in JAIST and University of Science, VNU-HCMC. It contains nothing that is the outcome of work done in collaboration with others, except where specified in the text. This dissertation is not substantially the same as any that I have submitted for a degree or diploma or other qualification at any other university. The content of the dissertation does not exceed the prescribed limit of 60,000 words.

NGUYEN, Thao Ngoc
September 23, 2014

Acknowledgements

First and foremost I want to express my gratitude and thanks to my doctoral supervisor, Professor Kazunori Miyata for his dedication and thoughtfulness. He has given me a lot of valuable experience and support during my four-year study. Despite the amount of work at his end, he has always followed my research progress carefully and made helpful comments. This dissertation would not have been completed without his diligent guidance and continuous encouragement.

I would like to thank my dissertation examiners, Professor Issei Fujishiro from Keio University, Professor Tsutomu Fujinami, Professor Taketoshi Yoshida, and Associate Professor Dam Hieu Chi from JAIST, for their interest in my work and for making their time available for me.

I am also grateful to Professor Le Hoai Bac, who has introduced me to computer science and offered me changes to do researches and study higher degrees.

Thanks to the FIVE-JAIST dual program between JAIST and University of Science, VNU-HCMC, for the financial support during my study.

Thanks to all members in the Miyata Laboratory for their friendliness and constructive comments during discussions. They have made my doctoral life less stressful and memorable.

The final thanks must go to my family for their absolute confidence in me and to my husband for being my best companion in this challenging journey.

Contents

List of Figures

List of Tables

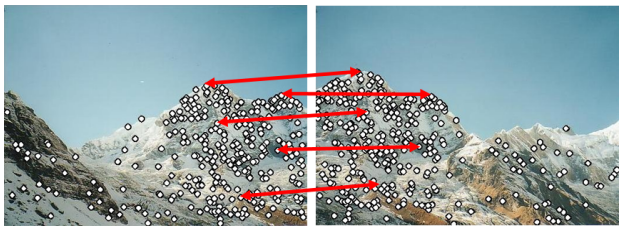
Chapter 1

Introduction

This chapter introduces the research context where the dissertation is motivated and the contribution of this doctoral work to development of Computer vision and Knowledge science. The first section describes the research motivation. The second and third sections define key concepts in the dissertation. The four section presents research objectives and contributions. Finally, the chapter is closed with the dissertation roadmap.

1.1 Overview

As human, we are granted an amazing visual system to perceive the information from the world around at apparent ease. Let us consider some examples in Fig.1.1. Given a pair of images, we can point out their similarities and dissimilarities easily, even when the details are transformed significantly under different conditions of illumination, viewpoint, or image resolution. We are also able to count the people in a photograph and interpret their activities in no time, though some of them have tiny shapes or suffer occlusion. Even a hard case to computer like tracking moving subjects (e.g. car or pedestrian) is not worth a challenge to human. Perceptual psychologists have spent several decades to reveal the mechanism of the human visual system but only achieves some of its principles [79, 85, 110]. Despite that ideal ability, the information processing power of human is usually limited, preventing us from solving intensive or complicated tasks, such as looking for the presence of a person in a collection of trillions images, solving high-order equations to find the geometric correlation between images. Such tasks have become very common when the visual data is growing more and more promptly in this technology era. An automated assistant to do these tasks on behalf of



(a) Matching details between two images

(b) Tracking moving targets

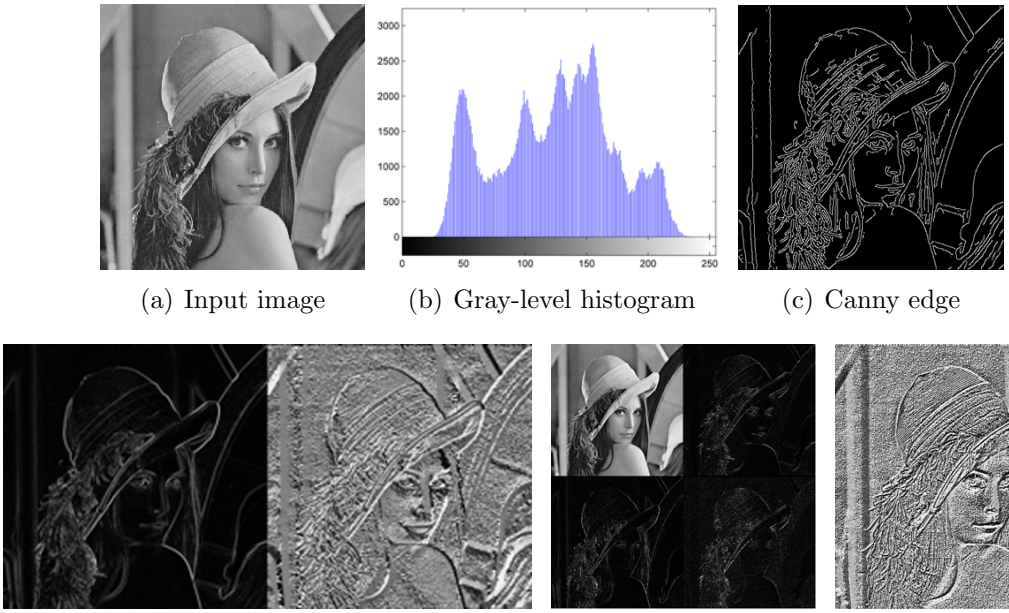
(c) Counting people

Figure 1.1: Humans can perform visual tasks with no difficulty, yet computers need special assistance from humans to achieve the same goals.

human is therefore an essential need.

We have seen how computers facilitate data mining scientists to discover knowledge from huge data or enable mathematicians to reach new levels of understanding by solving complex problems. It is obviously not due to their in-born intelligence but to the reasoning and inference abilities given by human. Computers enhance this simple intelligence several times with their computational power. Therefore, it is reasonable to expect a similar effect with visual knowledge once the visual perception in computer is enabled. Computers perceive the world in a completely different manner to that of humans, which causes several difficulties to understand and improve their perception. For example, we observe from Fig.1.1(a) blue sky and snowy mountain, whereas computers interpret the picture as an array of pixels with various values. However, they possess incredibly computational powers that are beyond most humans. For instance, to find images containing Barack Obama, Google Image Search returns 1,110 results within 1.06 seconds, or to recognize fingerprints, a modern system takes 10 milliseconds per match with an equal error rate $< 1\%$ [42]. Identifying the great potentiality of computers in accompanying humans to reveal the visual knowledge from the world, I conduct this dissertation with the desire to find an effective means to represent the visual information in computers and let them process it more similarly to human, i.e. enabling their visual perception.

Computer vision is an appropriate approach for my research topic since its mission is to feed the visual perception to computers, i.e. to describe the world that we see in images and reconstruct its properties, such as shape, illumination,

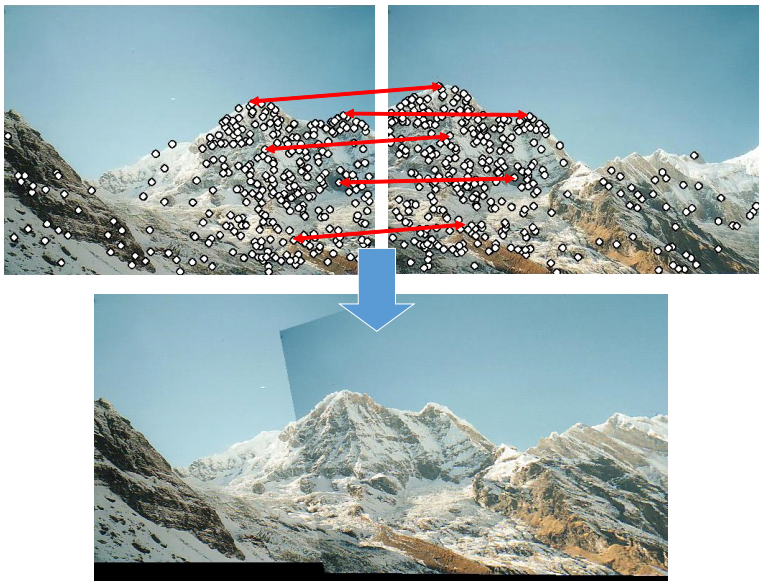


(a) Input image (b) Gray-level histogram (c) Canny edge

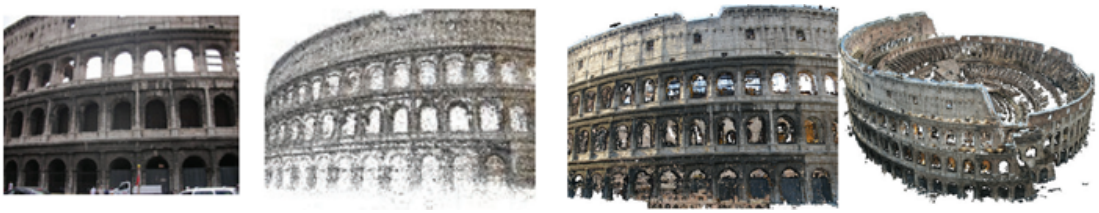
(d) Gradient magnitude and direction (Prewitt operator) (e) Haar wavelets (f) LBP textures

Figure 1.2: Common features for describing image details and structures.

and color distribution. Achievements from this research may enhance the quality of visual analysis to help computers interpret the information contained in images correctly. Based on that, semantic content or meaning from images can be extracted as an appropriate data representation for the visual analytics to get insights into sophisticated problems. Hence, the gap between information and knowledge will be narrowed. To fulfill this purpose, we can start with several features in computer vision, such as intensity histogram, edge, gradient, wavelets, etc. (cf. Fig.1.2). However, most of them are not meaningful enough for real-world textures or computationally too complex to meet the real-time requirement of many vision applications. Proposed in the year 2002, the *Local binary pattern* (LBP) (cf. Fig.1.2(f)) has been shown as a promising local texture that is very discriminative and computationally efficient. It is successfully applied in several problems such as face recognition [10, 13, 164], texture classification [46, 105], human action recognition [87], and medical image analysis [69, 99]. The LBP is quite new, compared to other visual features like edge and gradient, whose first publications date back to the 1980s or earlier. Its researches have been growing promptly and attracting lots of attention as well as critics from the computer vision community. Therefore, the studies of LBP feature in this dissertation greatly encourage its development in the future.



(a) Two images depicting a mountain in different views are stitched into a panorama image using SIFT features and RANSAC algorithm [21].



(b) 3D reconstructed model of Collosseum from 1,167 photos [8].

Figure 1.3: Applications of feature matching.

1.2 Feature matching and challenges

Feature matching is an important stage in several computer vision applications since it greatly influences the accuracy of higher-level tasks. Good examples exist in panorama stitching [21] (cf. Fig.1.3(a)), epipolar geometry estimation [86, 119], 3D model reconstruction [8, 117, 135] (cf. Fig.1.3(b)), as well as texture classification [67], object recognition [55, 81], and human action recognition [87].

The problem is defined as the comparison of two sets of feature descriptors obtained from two different image to provide detail correspondences between images. Features may be specific structures in the image, such as points, edges or objects, or the results of a general neighborhood operation, such as gradient magnitude and directions, wavelet signals, or local textures (cf. Fig.1.2). To manipulate them in computers, the corresponding descriptors are usually represented as boolean variables, numerical values, multi-dimensional vector or continuous

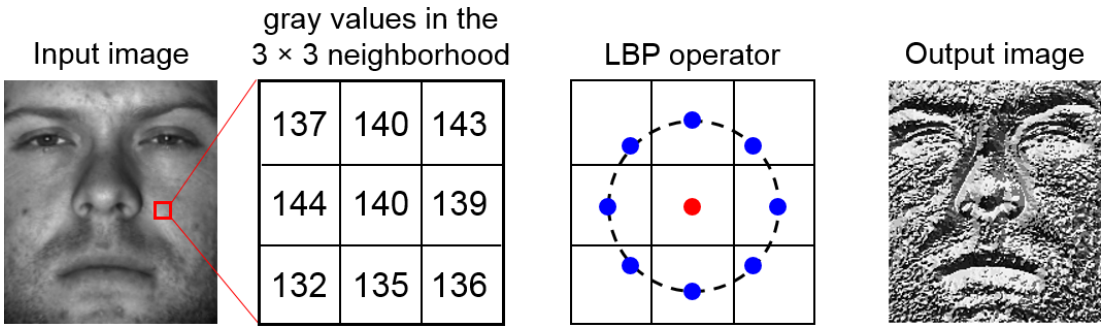


Figure 1.4: An example of computing LBP textures. A LBP operator is applied on every pixel of the image to capture information in the 3×3 neighborhood centered at the pixel, resulting in a value at the corresponding location in the output image.

functions. Therefore, the problem of feature matching reduces to a simple task of computing the similarity (or distance) between mathematical terms using some similar measure (e.g. Euclidean or Mahalanobis distance).

In practice, it is very challenging due to phenomena such as occlusions, non-rigid deformations, illumination changes, image rotation, poor textural content, etc. (cf. Fig.3.19 and Fig.4.9). Therefore, several requirements on the adopted feature are introduced to make the matching effective. Tuytelaars and Mikolajczyk [139] suggest six properties that an ideal local feature should have, which can be summarized to two following main points:

1. The feature descriptor should be *repeatable* and *precise* so that features detected on the scene part visible in both images should be found in both images and accurately localized with respect to scale and possibly shape.
2. The intensity patterns underlying features should show a lot of variation, such that features can be distinguished and matched, i.e. *distinctiveness*.

Most existing features cannot achieve all properties or at the same time pursuing quality and speed. Therefore, a feature that enables effective matching while being feasible enough for real-time applications remains elusive.

1.3 A brief review of Local binary pattern

1.3.1 Local binary pattern

The LBP [105] is a robust and efficient first-order texture operator for image analysis, which transforms a grayscale image into a registered map describing small-scale appearances of the original image. It describes textures in a $n \times n$

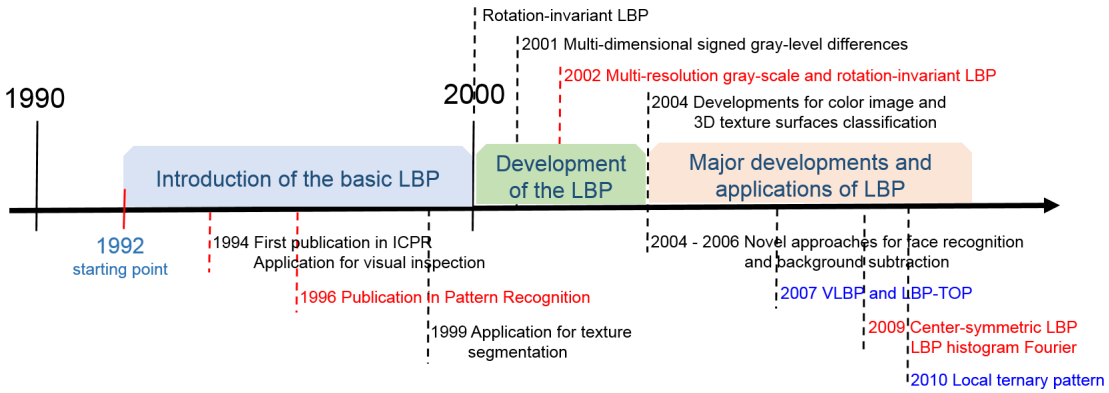


Figure 1.5: The history of LBP developments. The third and fourth phases are combined in the "Major developments and applications of LBP" component.

neighborhood around every pixel of an image by thresholding the gray values of neighbors with that of the center pixel and considering the result as a binary number (cf. Fig.1.4). It can be seen as a unifying approach to the traditionally divergent statistical and structural models of texture analysis [115].

The LBP has been shown to be highly discriminative, invariant to monotonic gray-level changes and computationally simple, thus it is a promising solution for several computer vision tasks, especially those requiring large-scale or real-time processing. Since its proposal in the year 2002, the LBP has soon become popular in the community in the sense that several studies of LBP have been made and its use in various applications has increased significantly.

1.3.2 The history of LBP

According to [115], the development of LBP has four main phases, which are illustrated in Fig.1.5 and described as follows:

Introducing the basic LBP operator. The LBP research started in the year 1992 with the idea that two-dimensional textures can be described by two complementary local measures, pattern and contrast, and thus by separating pattern from contrast, invariance to monotonic gray-scale changes can be obtained. It was first published in the International Conference on Pattern Recognition (ICPR) 1994 [102] and later extended into the Pattern Recognition [103]. The LBP and contrast operators were adopted in some applications like unsupervised texture segmentation [101] and visual inspection [113], obtaining clearly better results than that of the state-of-the-art at that time.

Developing the LBP operator from basic LBP and extensions. In the late

1990s, the use of multi-dimensional signed gray-level differences instead of absolute differences [106] and the rotation-invariant LBP [114] were proposed, which are key ideas for the multi-resolution gray-scale and rotation-invariant LBP nowadays [104, 105]. Henceforth, its use in various applications has increased rapidly and several LBP studies have been motivated. Major developments were the Center-symmetric LBP for interest region description [55], LBP histogram Fourier [11] for rotation-invariant texture description, and Local ternary patterns for face recognition [129]. LBP features for color image [83] and for 3D texture surfaces classification [112] were also studied.

Spatial LBP for face description. In the mid-2000s, Ahonen et al. proposed a novel approach that divides the face image into several regions from which LBP features are extracted and concatenated into a feature vector [9, 10]. It has inspired a large number of studies for further improvement and been adopted in various practical applications. The approach and its variant have been used for problems such as face recognition and authentication, face detection, facial expression recognition, gender classification and age estimation.

Spatio-temporal LBP for motion and activity analysis. Also in the mid-2000s, the first application of LBP in motion analysis was background modeling and moving object detection [53, 54]. Each pixel is modeled as a group of adaptive LBP histograms that are calculated over a circular region around the pixel. The approach is highly robust against illumination variations, the multimodality and the dynamism of background. In the year 2007, the spatio-temporal VLBP and LBP-TOP [164] were proposed, establishing a basis for problems such as facial expression recognition utilizing facial dynamics [164], face and gender recognition from video sequences [50], and recognition of actions and gait [63, 87].

1.4 Research objective and contributions

1.4.1 Research objective

This dissertation studies the effectiveness of LBP features in describing image properties in order to improve the quality of feature matching and hence better facilitating the visual analysis in computers. To fulfill this objective, the performance of LBP features are thoroughly examined on three major computer vision tasks: (1) interest region description, (2) pedestrian detection and (3) background subtraction.

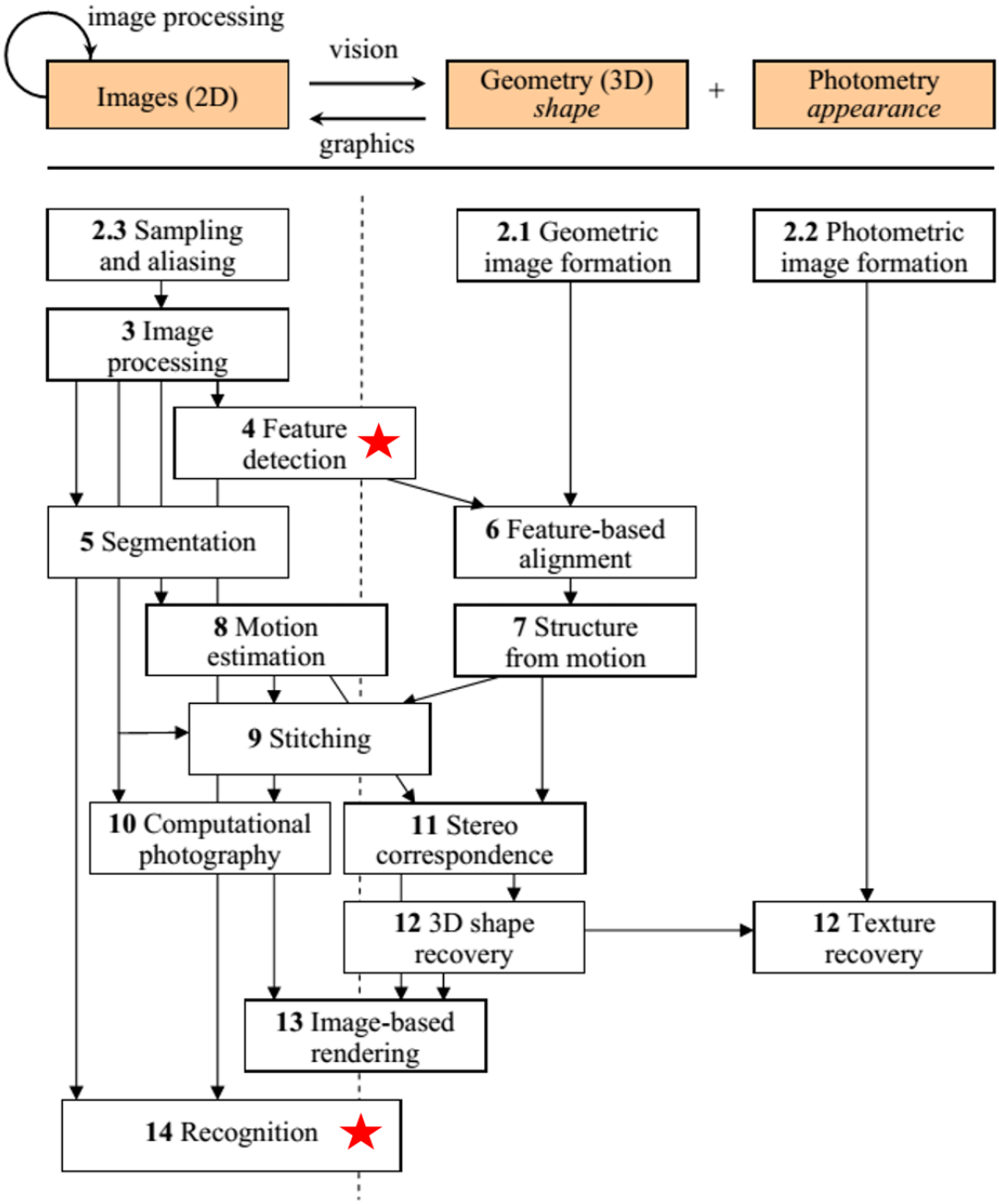


Figure 1.6: A taxonomy of the topics covered in computer vision [125]. The whole figure should be taken with a large grain of salt, as there are many additional subtle connections between topics not illustrated here. The red stars mark the general topics to which the three studies in the dissertation belong.

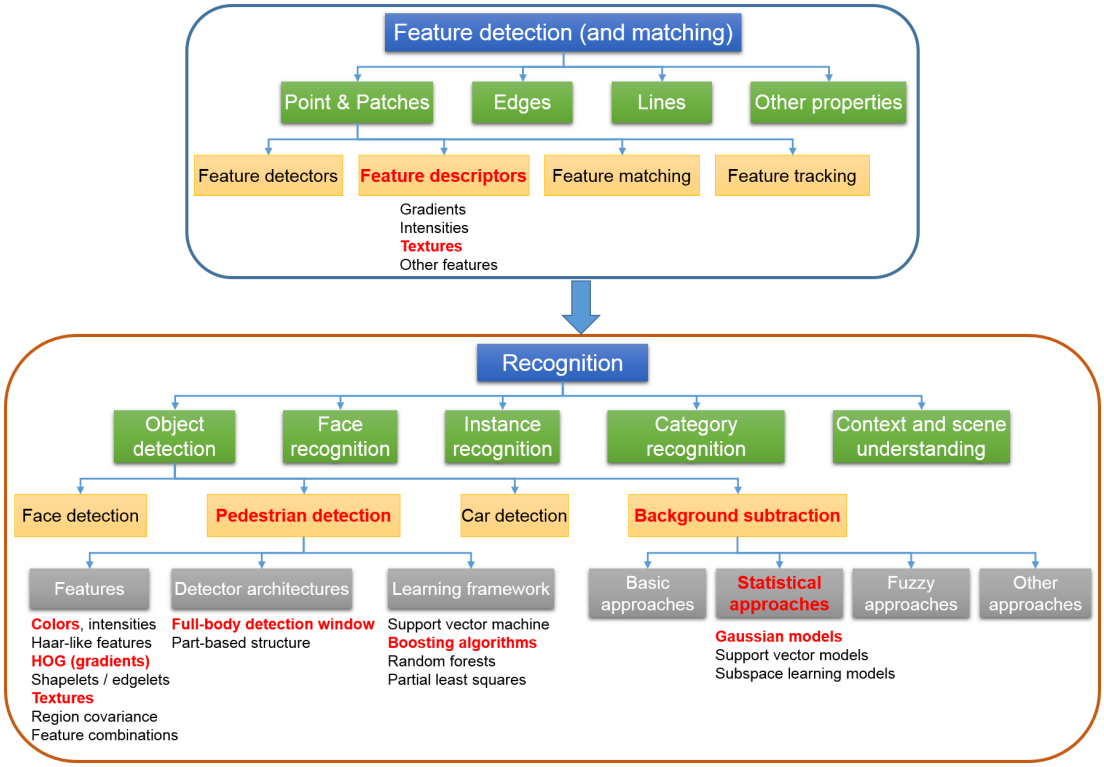


Figure 1.7: Topics in the 'Feature detection" and "Recognition" fields. Techniques related to the three studies are highlighted in red.

The above topics are selected due to their important roles in computer vision. Figure 1.6 shows a taxonomy of the topics covered in computer vision, in which the topic (1) belongs to the general topic "Feature detection" while the topics (2) and (3) are in "Recognition". Figure 1.7 zooms into the two corresponding general topics. The interest region description searches for robust features to describe the low-level image details. Features that are successful in this task are possible to be improved for high-level tasks. The background subtraction and pedestrian detection usually rely on the first task to select features for their methods. In addition, the three techniques can be joined to build a complete pedestrian surveillance system, which is very meaningful for both academic research and practical applications. By consequently studying the LBP feature in these topics, the dissertation gradually improves the visual perception of computers from the most basic level, i.e. comparing general image details, to more sophisticated ones, i.e. identifying moving objects and distinguishing pedestrians from other types of objects. This demonstrates the effort of my doctoral research to fill the gap of visual interpretation between computers and humans.

The first study aims for a meaningful representation of interest regions with

(a) Proposed method (IO-QLBP) (b) SIFT [81]

Figure 1.8: Two matching results provided by the proposed IO-QLBP and the baseline SIFT[81] descriptors. Red lines indicate correspondences that are correctly identified, whereas yellow lines describe correspondences that the descriptor fails to match.

the LBP feature so that details between images can be matched correctly. Therefore, it falls into the research of point and patches. This study enables the computers to compare low-level details in images, which serves as the preliminary step for many subsequent tasks, such as feature-based alignment, image stitching, stereo correspondences, etc. It also has a direct support to the "recognition" topic by providing means of sparse object representation. *The second study* targets the detection of pedestrians in static images, which is done by searching patches that potentially describe a person at every place of an image. Its applications are commonly found in our daily life like organizing personal photo albums or collecting images related to a specific person. From this study, the computers can improve their visual perception to a higher level, which is to identify objects rather than fragments only. *The third study* employs the temporal dimension in the video sequence to perform object detection. It detects moving objects by first building a model representing the background scene and then, for each incoming frame, considering pixels that yield great deviation from the model as parts of moving objects. Combining the background subtraction and pedestrian detection allows computers to handle semantic tasks like detecting the presence of human in the monitored area using surveillance camera.

1.4.1.1 Interest region description

This study searches for an effective feature descriptor to describe interest regions in a pair of images, so that region correspondences between two images can be established correctly afterwards (cf. Fig.1.1(a)). A good feature should be discriminative to identify regions depict the same scene part in both images and robust against image transformations. Two novel descriptors, *Intensity order quartet local binary pattern* (IO-QLBP) and *Multi-scale region perpendicular local*

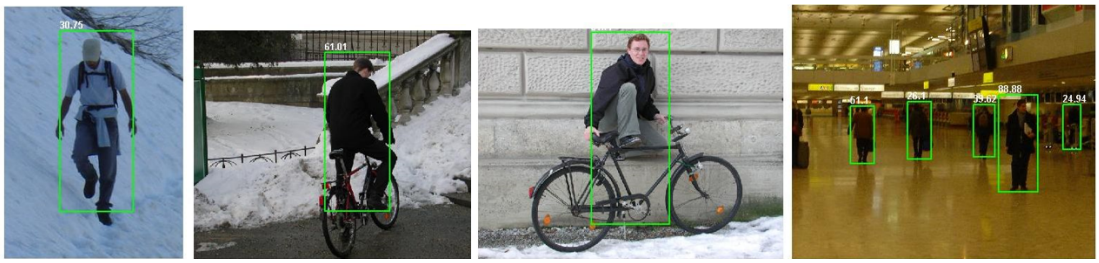


Figure 1.9: Some detection results on the INRIA benchmark provided by the proposed multi-cue pedestrian detector.

binary pattern (MSR-PLBP), are proposed. The IO-QLBP efficiently produces a compact feature from a set of four neighbors. Patterns are generated in a selective manner with a weighting scheme and an adaptive thresholding scheme that is locally set for every pair of pixels. Meanwhile, the MSR-PLBP compares each pixel in the image with its four neighbors distributed evenly on two fixed perpendicular axes originated at the pixel. It proposes the use of multiple pattern candidates, instead of conventionally assigning one patter per pixel, to address the drawback of LBP on near-uniform regions. The local image contrast is also incorporated to improve the adaptability to different illumination changes. Figure 1.8 shows an example comparing the IO-QLBP with the baseline [81] in matching images.

1.4.1.2 Pedestrian detection

This study aims to adopt the novel QLBP feature in the previous study to build a robust human detector so that pedestrians in static images can be well distinguished from backgrounds as well as from other objects. Challenges come from the wide variations of pedestrian appearances, which are due to articulated pose, clothing, illumination changes, and occlusions, thus making the problem far from being completely solved. The QLBP is generalized to an abstract representation that comprises all possible encodings from a neighborhood of four points. Since the upright human body has specific cues that distinguishes itself from other objects, an encoding that emphasizes these cues will be likely to boost the detection rate. The proposed detector jointly characterizes each detection window by a normalized gradient histogram, three color channels and three generalized QLBP images computed in each color channel. This allows describing the pedestrian appearance with complementary aspects, i.e. intensity changes in different orientations, color and texture, hence high performance is expected. Some detection examples are shown in Fig.1.9.

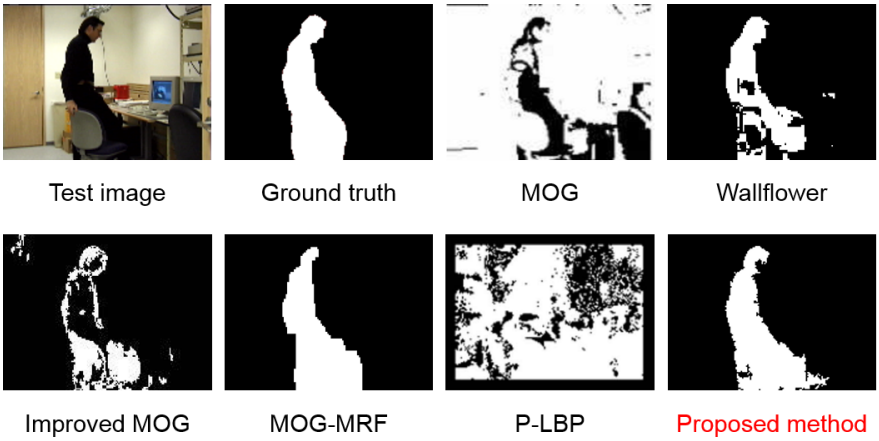


Figure 1.10: The performance of different background subtraction algorithms on the scenario of sudden illumination changes. The result of the proposed method is very similar to the ground truth.

1.4.1.3 Background subtraction

This study deals with the temporal information in video frames to extract moving objects (e.g. pedestrians, cars, etc.) from background. It is an initial yet essential step in several computer vision systems like motion tracking or event detection. Nevertheless, it is very challenging since practical scenes may contain complex dynamic effects, such like moving vegetation, rippling water, sudden illumination changes and occlusions. An efficient multi-layer background modeling framework is introduced to process the incoming frame in a coarse-to-fine manner with a block-wise layer and a pixel-wise layer. The former is constructed with the QLBP feature in the previous study, which is highly robust to illumination changes and computationally simple. Meanwhile, the latter characterizes individual pixels by distributions of colors. This allows us to classify stationary/non-stationary blocks effectively so that only blocks containing moving pixels are further analyzed in the latter layer. Both qualitative and quantitative evaluations show that the proposed framework is able to capture more global structures while maintaining the fineness of details. This advantage is hard to achieve with single approaches. Fig.1.10 shows an example of background subtraction with the proposed method.

1.4.1.4 A unified surveillance system from three proposed techniques

We integrate the proposed techniques in three studies, namely the interest region description, pedestrian detection and background subtraction, into an effective unified surveillance system. The proposed system uses the novel LBP features

in the first study to represent the visual objects. Moving objects are extracted from video frames of a CCTV camera using the background modeling method in the third study and then pedestrians are distinguished from other moving objects using the proposed pedestrian detector in the second study. In this way, a surveillance system that automates the detection of human is enabled, supporting the monitoring of the entering/leaving of visitors in public places (e.g. the school ground or office hall), or the prevention of potential intruders in restricted areas. Since each camera has a limited field of view, objects and activities at the border may not be described precisely. We therefore extend the proposed system to the multi-view mode by creating a panorama view from multiple cameras and performing the surveillance on this new common view. In this way, a considerable number of fragments in each view are connected, improving the quality of detection as well as trajectory tracking and action recognition. The PLBP feature is applied to the panorama image stitching to attain better feature alignment. Illustrations of the proposed surveillance system in single-view and multi-view modes are shown in Fig.6.4 and Fig.6.6.

1.4.2 Contribution to the Computer vision research

Based on the above, the academic significance of this doctoral dissertation is described as follows.

Novelty. This dissertation provides novel methods for effective feature matching that are specialized in interest region description, pedestrian detection, and background subtraction. These methods are all successful in boosting the performance to a higher level than that of many modern methods while maintaining acceptation computational speeds. Specifically, in the first study, two proposed region descriptors improve the accuracy about 3-15% while they are both several times faster than other descriptors. The multi-cue detector in the second study reduces the average miss rate to 15.4% with a frame rate of 15.2 fps. It is equivalent or better than many modern detectors while significantly outperforms the baseline HOG (46%, 0.239 fps). In the final study, the multi-layer background subtraction framework uses the novel LBP texture feature to halve the number of errors, compared to other competing algorithms, when encountering sudden illumination changes. It innovatively processes the information in a coarse-to-fine manner instead of concentrating on a single component. All of those results demonstrate the superiority of LBP in relation to other features, such as gradient, color channels, or wavelets.

Originality. The proposed methods provide several novel definitions of LBP features in terms of encoding, thresholding scheme as well as the strategy of generating pattern, hence considerably enhancing the robustness and efficiency of LBP. The proposed features include three key properties that completely distinguish themselves from existing methods. First, they require a small set of neighbors to depict the image properties while preserve the meaningfulness of patterns. In this way, the issue of high dimensionality is resolved and the applicability of LBP in tasks that need dense matching is improved. Second, instead of assigning one pattern per pixel, the novel scheme of pattern generation creates a set of pattern candidates, each of which has a confidence value, so that the lack of information in noisy regions are diminished. Third, prior knowledge of image environments, such as contrast, effects of illumination changes, etc. is incorporated. This makes the features more adaptive to environment changes in an automatic manner, i.e. less manual effort of tuning parameters is required.

Applicability. The dissertation provides a complete review of LBP features in different computer vision problems, providing a starting point for people who want to select features for their applications in object detection, texture classification, action recognition, etc. The proposed descriptors in image matching effectively support higher-level computer vision tasks such as panorama image stitching, 3D reconstruction and in-between view establishment in virtual/augmented reality. The multi-cue pedestrian detector may contribute to some person tracking systems, such as a security system that it recognizes a person as suspect or non-suspect or a driver assistance system that detect pedestrian and raises alerts to drivers. For these applications, the multi-layer background subtraction could serve as an initial step to help filtering non-interest regions (i.e. regions having static contents) and thus significantly reduces the detection errors. To demonstrate a unique application of the proposed LBP features in this dissertation, we combine three related techniques to build a simple pedestrian surveillance system that detects the presence of human in the monitored area. Quantitative and qualitative evaluations show that the system performs effectively and stably in real scenarios.

1.4.3 Contribution to the Knowledge science

This thesis contributes to the Knowledge science in the three following aspects:

Facilitating the visual analytic process to create knowledge: The key challenge for visual analytics is to derive semantic content or meaning from images in real

time (cf. Chapter 4 of [134]). This doctoral research enhances the quality of visual analysis to help computers understand the information contained in the image rather than the image itself only. Therefore, it helps us move toward an appropriate data representation for the visual analytics. Using visual analytics, we can create knowledge from analytical reasoning with the facilitation of interactive visual interfaces.

Modeling knowledge creation process: Identifying image details seems to be straightforward for human but involves sophisticated processes in the brain, which has not been fully discovered by psychology. This hinders researchers in their effort to bring computers close to human. This dissertation proposes several techniques to gradually improve the ability of computer in revealing the visual knowledge from huge amount of data. The proposed techniques are developed and verified by standard experiments and experts, thus ensuring the validity of the knowledge creation process.

Discovering knowledge and regularities for inference: Humans make decision based on their intuition and experience, yet they are hardly able to point out the differences or relations between two images clearly. Proposed features can be utilized to discover various types of image patterns and examine these patterns in a statistical manner. Therefore, they help researchers to attain novel visual knowledge and regularities for inference.

1.5 The dissertation roadmap

The other chapters of this dissertation are organized as follows. Chapter 2 introduces the Local binary pattern and its related works. The next three chapters presents the studies on interest region description, pedestrian detection and background subtraction, respectively. They shares a similar structure in which the proposed ideas are first described, followed by the comparative evaluations, and finally a brief discussion about the advantages and drawbacks of the proposed methods is given. Chapter 6 integrates the proposed techniques in three studies into a unified surveillance system, demonstrating the contribution of novel LBP features in enabling computers to handle the semantic vision task. Chapter 7 summarizes what the dissertation has achieved and draws plans for future work.

The novelty and originality have been stated in Sect.1.4.2 of this chapter. The three key properties that constitutes the originality are discussed more deeply when proposed methods are described in detail in each chapter. These chapters

also make clear how these methods address the issues of accuracy and speed at the same time and to what extent they improve the quality of feature matching. The general applicability of LBP is mentioned in Chapter 2 while specific applications of proposed methods are discussed in corresponding chapters. The unique application of all proposed techniques, i.e. the pedestrian surveillance system, is described in Chapter 6.

Chapter 2

Local Binary Pattern

This chapter introduces the Local binary pattern (LBP), which is a robust and efficient texture operator for image analysis. With the advantages of highly discriminative, invariant to monotonic gray-level changes and computationally simple, the LBP and its variants constitutes a large family of texture features that perform excellently in various computer vision applications.

The content of the chapter is organized as follows. Section 2.1 defines the LBP operator. Its advantages and drawbacks are discussed in Sect.2.2. Section 2.3 is reserved for the Center-symmetric local binary pattern, a widely popular LBP variant in the field of interest region description. Other variants are briefly summarized in Sect.2.4. Finally, Section 2.5 lists several applications of LBP.

2.1 Local binary pattern

The Local binary pattern (LBP) is a first-order image operator that transforms a grayscale image into a registered map describing small-scale appearances of the original image. Ojala et al. first introduced the LBP during the years 1994 - 1996 [102, 103], in which the operator assigns a label to every pixel of an image by thresholding a 3×3 neighborhood centered at the pixel and considering the result as a binary number. In the year 2002, several years after the original publications of LBP, they extended the operator into a more generic form allowing difference sizes of neighborhood and numbers of neighbors [105]. The term "local binary pattern" henceforth refers to the generic operator rather than its precursor.

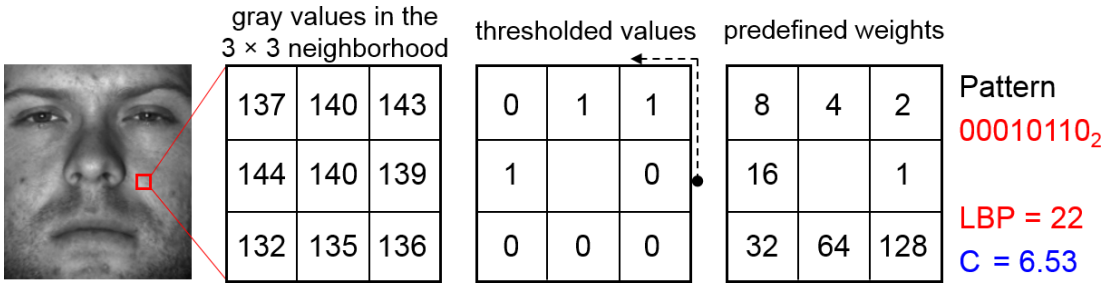


Figure 2.1: An example of basic LBP. The dotted arrow indicates the starting point and direction (i.e. anticlockwise) of the encoding. Thresholding results at every position are combined into a binary bit sequence.

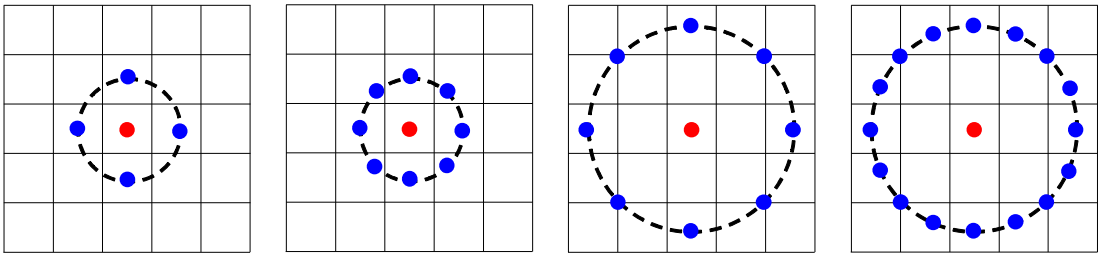
2.1.1 The basic LBP operator

The basic LBP operator [102, 103] was motivated by the idea that two-dimensional textures can be described by two complementary local measures, i.e. pattern and contrast, and by separating pattern information from contrast, invariance to monotonic gray-level changes can be obtained. It compares eight bordering pixels (neighbors) in a 3×3-neighborhood with the center pixel. A "0" bit is produced if the considered neighbor has a lower gray value than that of the center pixel, while a "1" bit is assigned in the opposite case. The LBP code is then computed by multiplying bits with predefined weights and summing the results, resulting in $2^8 = 256$ different possible codes. The contrast measure C is obtained by subtracting the average of gray values below the center value from the average of gray values above or equal to the center value. Figure 2.1 demonstrates the computation of basic LBP with a given 3×3-patch. The one-dimensional distribution of LBP codes or two-dimensional distribution of LBP and contrast are used for further analysis like classification or segmentation.

2.1.2 The generic LBP operator

In the year 2002, Ojala et al. revised the operator into a more generic form that puts no limitation to the neighborhood size or to the number of sampling points, whereas the basic version is bounded to eight neighbors in a 3×3-pixel region. The derivation of LBP presented below is summarized from [105].

Let I denote the monochrome image in consideration, $g_c = I(x, y)$ denote the gray value of the center pixel of a circular neighborhood of radius R ($R > 0$), and $g_p = I(x_p, y_p)$ $(p = 0, \ldots, P-1)$ correspond to the gray values of P neighbors



(a) $P = 4, R = 1$ (b) $P = 8, R = 1$ (c) $P = 8, R = 2$ (d) $P = 16, R = 2$

Figure 2.2: The circularly symmetric neighbor sets for different (P, R). Gray values of points that are not in the center of pixels are interpolated.

evenly spaced on the circle. The coordinate of a neighbor p is determined by

$$\begin{aligned} x_p &= x + Rcos(2\pi p/P) \\ y_p &= y - Rsin(2\pi p/P) \end{aligned} \tag{2.1}$$

If any neighbor does not fall exactly in the center of pixels, its gray value is estimated by bilinear interpolation. Figure 2.2 illustrates some circularly symmetric neighbor sets for different (P, R).

The texture T in a local neighborhood of a monochrome texture image is defined as the joint distribution of gray levels of P $(P > 1)$ image pixels:

$$T = t\left(g_c, g_0, \ldots, g_{P-1}\right) \tag{2.2}$$

The gray value of the center pixel g_c can be subtracted, without loss of information, from that of the circularly symmetric neighbors g_p $(p = 0, ., P - 1)$.

$$T = t\left(g_c, g_0 - g_c, g_1 - g_c, \ldots, g_{P-1} - g_c\right) \tag{2.3}$$

It is assumed that the center pixel is statistically independent of the differences, allowing the joint distribution to be factorized as follows.

$$T \approx t(g_c)t\left(g_0 - g_c, g_1 - g_c, \ldots, g_{P-1} - g_c\right) \tag{2.4}$$

Although the exact independence is not warranted in practice, we accept a small loss of information to achieve invariance with respect to gray-level shifts.

The distribution $t(g_c)$ in Eq.2.4 describes the overall luminance in the image, thus containing no helpful information for local texture pattern analysis. The joint different distribution, on the other hand, conveys many textural character-

istics [106], and hence:

$$T \approx t\left(g_0 - g_c, g_1 - g_c, \ldots, g_{P-1} - g_c\right) \tag{2.5}$$

The operator records the occurrences of various patterns in the neighborhood of every pixel on a P-dimensional histogram, therefore it is highly discriminative. The differences in all directions are zeroes in constant regions, whereas they are large for spots. On a slowly sloped edge, the operator records the highest difference in the gradient direction and zero values along the edge.

The signed differences $g_p - g_c$ are not affected by changes in mean luminance, thus the joint difference distribution is invariant against gray-level shifts. However, it is not warranted to be invariant against other changes like gray-level scaling. In order to alleviate the issue, only the signs of the differences are considered instead of their exact values:

$$T \approx t\left(s(g_0 - g_c), s(g_1 - g_c), \ldots, s(g_{P-1} - g_c)\right) \tag{2.6}$$

where

$$s(z) = \begin{cases} 1 & z \geq 0, \\ 0 & z < 0 \end{cases} \tag{2.7}$$

The generic $\mathrm{LBP}_{P,R}$ operator is derived from Eq.2.6 by assigning a binomial factor 2^p for each signed $s(g_p - g_c)$, resulting in a unique number that characterizes the spatial structure of the local image texture:

$$\mathrm{LBP}_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p \tag{2.8}$$

This equation indicates that each signed difference is interpreted as a binary bit. Hence, there are 2^P distinct LBP codes in total.

The term "local binary pattern" reflects the function of the operator, i.e. a local neighborhood is thresholded at the gray value of the center pixel into a binary pattern. The $\mathrm{LBP}_{8,1}$ (i.e. $P = 8$ and $R = 1$) is similar to the basic LBP except two points: 1) the pixels in the neighbor set are indexed to form a circular chain, and 2) the gray values of diagonal pixels are estimated by interpolation. These modifications are essential for developing LBP into a rotation invariant version (cf. Sect.2.1.4).

2.1.3 Achieving uniformity for LBP patterns

Ojala et al. [105] introduce a uniformity measure that describes the number of bitwise transitions in a pattern, i.e. from "0" to "1" or vice versa, when the bit sequence is considered circular.

$$U(LBP_{P,R}) = \begin{array}{l} |s(g_{P-1} - g_c) - s(g_0 - g_c)| + \\ \sum_{p=1}^{P-1} |s(g_p - g_c) - s(g_{p-1} - g_c)| \end{array} \tag{2.9}$$

Patterns having U values of at most 2 are called *uniform patterns*, which are denoted by LBP^{u2}. For example, the patterns 00000000_2 (0 transition), 11111111_2 (0 transition), 01110000_2 (2 transitions) and 11001111_2 (2 transitions) are uniform, while the patterns 11001001_2 (4 transitions) and 01010011_2 (6 transitions) are not. In the histogram computation, every uniform pattern is mapped into a separate bin while all non-uniform patterns are assigned to one "miscellaneous" bin, producing a $P(P-1)+3$-dimensional vector.

The advantage of uniform patterns lies in two folds. First, most of the LBP patterns in natural images are uniform. In the experiments with texture images [105], these patterns account for a bit less than 90% of all patterns when using the (8,1)-neighborhood and for around 70% with the (16,2)-neighborhood. Meanwhile, Ahonen et al. [10] observed from their experiments with facial images that the amounts of uniform patterns in the (8,1)-neighborhood and (8,2)-neighborhood are 90.6% and 85.2%, respectively. Second, using uniform patterns instead of all patterns gives better results in several applications. They are thus believed to be more robust, i.e. less prone to noise. In addition, the dimensionality of the histogram reduces significantly from 2^P to $P(P-1)+3$ and the distribution estimation could be done reliably with fewer samples. Therefore, the LBP is promising for recognition tasks involving a wide variety of textures.

2.1.4 Achieving rotation invariance for LBP patterns

The $\text{LBP}_{P,R}$ is unqualified to problems involving much rotation, such as image matching, because its circular neighborhood is greatly affected by rotations of the original image. The neighborhood itself translates into another location, while its neighbors correspondingly rotates about the origin into a different orientation. Since g_0 is defined to be the gray value of the neighbor lying to the right of the center pixel, that naturally results in a different $\text{LBP}_{P,R}$ code (cf. Fig.2.3).

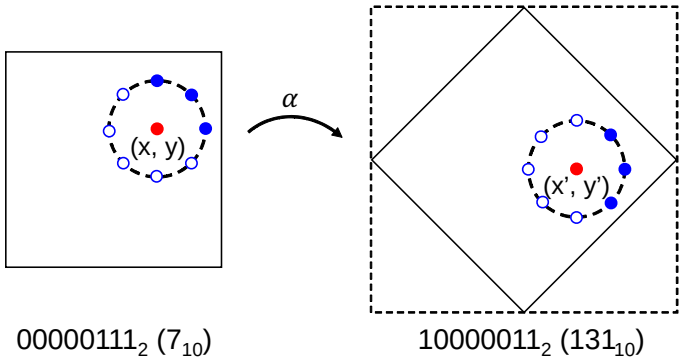


Figure 2.3: The effect of image rotation on points in a circular neighborhood.

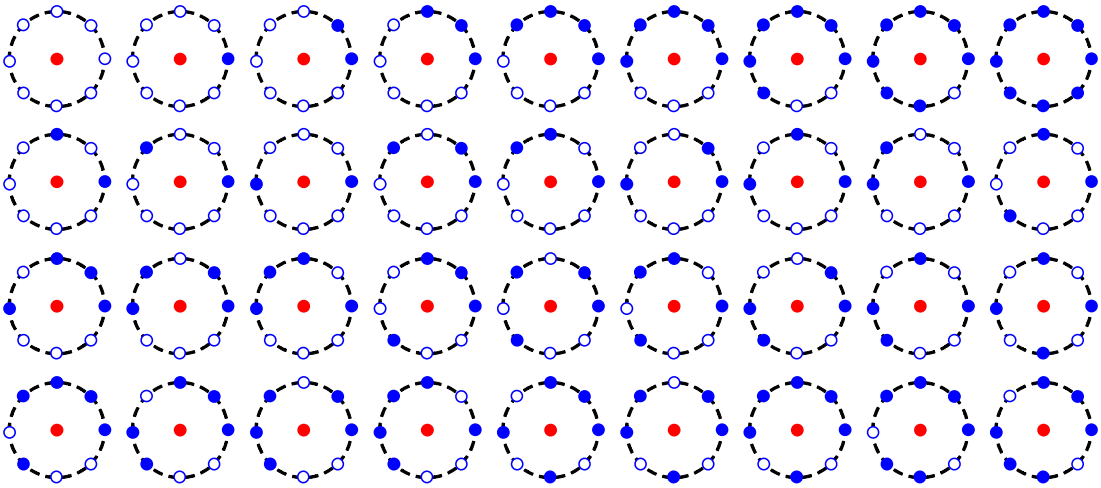


Figure 2.4: The 36 unique rotation invariant patterns that can occur in the $(8, R)$ neighborhood. Points corresponding to bit "1" are denoted by blue circles, while those corresponding to bit "0" by white circles. The first row shows patterns that are both uniform and rotation invariant.

To achieve rotation invariance, Ojala et al. [105] consider every bit sequence circular and rotates it into a minimum value as follows:

$$LBP^{ri}_{P,R} = \min_i ROR(LBP_{P,R}, i) \;\; i = 0, \ldots, P-1 \tag{2.10}$$

where $ROR(x, i)$ performs i circular bit shifts to the right on the bit sequence x. For instance, the bit sequences 10000010_2, 00001010_2, and 00101000_2 arise from different rotations of the same pattern, and it is possible to map all of them into the minimum value 00000101_2 by right shifting 7, 1 and 3 times, respectively. Figure 2.4 illustrates 36 unique $LBP^{ri}_{8,R}$ patterns, each of which illustrate some texture primitives. For example, in the first row, the first pattern detects bright spots, the eighth for dark spots and flat areas, and the fourth for edges.

The rotation invariant pattern can also be improved to be "uniform":

$$LBP_{P,R}^{riu2} = \begin{cases} \sum_{p=0}^{P-1} s(g_p - g_c) & if\ U(LBP_{P,R}) \leq 2 \\ P+1 & otherwise \end{cases} \tag{2.11}$$

The superscript *riu2* denotes rotation invariant patterns having U values of at most 2. There are exactly $P+1$ LBP^{riu2} patterns that can occur with a circularly symmetric neighbor set of P pixels (cf. the first row of Fig.2.4). Equation 2.11 assigns a unique label to each pattern according to its number of bit "1" and uses the "miscellaneous" label $P+1$ for other patterns.

The rotation invariance is not always warranted because of the rough quantization of the angular space, i.e. $\alpha = 360^o/P$. A straightforward solution is to increase the number of neighbors P, yet it should be done with certain considerations. The circular neighborhood of radius R can contain a limited number of pixels (e.g. nine for $R = 1$), setting an upper limit to the number of non-redundant neighbors. In addition, the set of possible codes grows exponentially with P, requiring appropriate means of storing and searching on a lookup table of 2^P entries.

2.2 Properties of local binary pattern

The Local binary pattern has several good properties that support its wide applications in computer vision.

- The $\text{LBP}_{P,R}$ is, by definition, invariant against any monotonic gray-level transformation, such as gray-level shift or gray-level scaling. That is, as long as the order of gray values in the image is preserved, the output of $\text{LBP}^{P,R}$ remains unchanged.
- The LBP is a nonparametric method that requires no assumptions about the underlying distributions. It does not have many parameters to be set, thus reducing the implementation burden and heading towards automation.
- The LBP has a highly discriminative power. It can tolerate a wide range of illumination changes that commonly occur in natural images. It also requires no specific normalization of the input image, except some trivial smoothing operations for eliminating random noises.
- The operator structure is usually intuitively and computationally simple, which is a useful property that encourages the use of LBP in various appli-

cations, especially those aiming for real-time performance.

- The LBP code is quantized by its nature. This allows image regions descriptors that collect LBP features on a square grid to perform bilinear interpolation only. Meanwhile, descriptors using gradients usually perform a heavy tri-linear interpolation. Therefore, the LBP has an advantage in terms of computational efficiency.

The drawback of LBP, as well as of all local descriptors using vector quantization, is that even a small change in the input would cause a change in the output. The LBP may suffer instability on noisy images or near-uniform regions, e.g. sky and plain wall, due to its thresholding scheme. Let us examine the function $s(g_p - g_c)$ in two cases: $(g_p = 29,\ g_c = 30)$ and $(g_p = 30,\ g_c = 30)$. It returns 0 in the first case and 1 for the second case. Several improvements have been proposed. The first approach replaces the term $s(g_p - g_c)$ with $s(g_p - g_c + a)$. The higher the value of a is, the larger changes in gray level are allowed without affecting the thresholding results [54]. The second approach introduces a scaling factor τ to the function s, i.e. $s(g_p - \tau g_c)$, allowing the operator invariant against gray-level transform by a scale factor [74]. Relatively small values of a or τ should be used to retain the discriminative power. The third approach extends the second one so that the threshold is locally adaptive to every pair of pixels [146]. These improvements generally enable more robustness in many applications like background subtraction and face recognition.

In addition, the use of $s(g_p - g_c)$ rather than $g_p - g_c$ causes performance degradation when dealing with image details whose values of $g_p - g_c$ are of the same sign yet different magnitudes, though it is the key point for invariance against monotonic illumination changes. This causes little affection to problems like texture classification and face recognition but great impact to interest region description and human detection. Therefore, the latter problems usually differentiate the contribution of patterns to the histogram by some means of weight [49, 60], rather than considering all patterns equivalent as done in the ordinary LBP.

2.3 Center-symmetric local binary pattern

The LBP has properties that favor its use in interest region description, such as the computational simplicity and the strong invariance against monotonic illumination changes. However, it has low robustness on flat image areas and its 2^P-dimensional histogram is too long for a region descriptor. To address

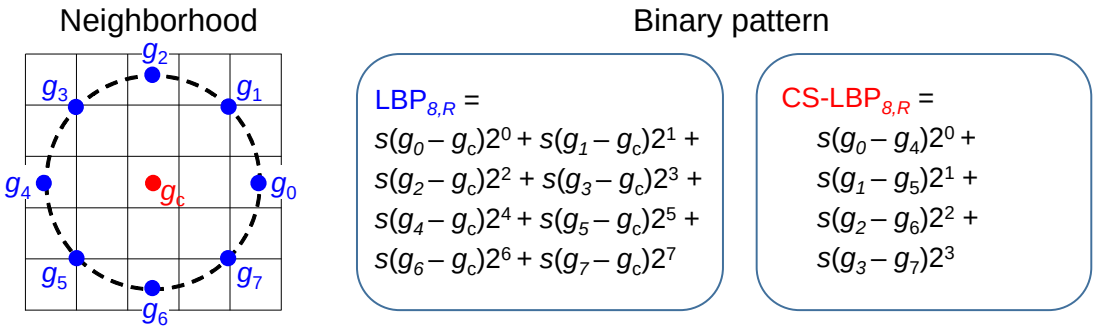


Figure 2.5: The neighborhood and the binary pattern of CS-LBP$_{8,R}$. The LBP$_{8,R}$ pattern is shown for comparison.

these issues, Heikkila et al. [55] introduce the Center-symmetric local binary pattern (CS-LBP), which aims for a more compact representation by modifying the scheme of comparing pixels in a neighborhood. That is, instead of comparing every neighbor with the center pixel, the CS-LBP considers center-symmetric pairs of neighbors as follows:

$$CS - LBP_{P,R}(x,y) = \sum_{p=0}^{(P/2)-1} s(g_p - g_{p+(P/2)})2^p,$$
$$s(z) = \begin{cases} 1 & z > T, \\ 0 & otherwise \end{cases} \tag{2.12}$$

where P corresponds to the number of neighboring pixels equally spaced on a circle of radius R centered at (x, y), g_p and $g_{(P/2-1)}$ correspond to the gray values of neighbors in center-symmetric pairs. Figure 2.5 illustrates the CS-LBP$_{8,R}$ operator structure. The coordinates of neighbors are determined and interpolated (if necessary) using the same approach described in Sec.2.1.

In comparison with the LBP, the CS-LBP halves the number of pixel pairs for the same number of neighbors P, thus significantly reducing the dimensionality of the histogram from 2^P to $2^{P/2}$ dimensions. For example, the LBP$_{8,R}$ produces 256 (2^8) possible codes, whereas the CS-LBP$_{8,R}$ generates only 16 (2^4) codes. In addition, the CS-LBP attains higher stability on flat image regions by thresholding the gray-level differences against a small non-zero T as proposed in [54]. The CS-LBP is, by definition, closely related to some gradient operators in the sense that it takes into account pairs of center-symmetric pixels in a neighborhood. Therefore, it is said to inherit good properties from both gradient and texture operators. Experiments on image matching and object recognition [55] have shown that the CS-LBP descriptor performs more robustly than descriptors

using either texture or gradient only.

Thanks to its high robustness and efficiency, the CS-LBP has a wide range of applications such as object recognition [55], background subtraction [156], pedestrian detection [155, 167], and human action recognition [87]. However, there is still room for improvement. The thresholding scheme adopts a small non-zero T to alleviate a great deal of sensitivity in flat image regions (cf. Eq.2.12), yet it cannot handle drastic cases like soft cast shadows. The cast shadow tends to make the covered region darker than other regions by a scale factor but preserves the texture information, and therefore a global threshold seems to be not appropriate. The proposed methods address this problem by improving the adaptability of the thresholding scheme to different local changes, thus gaining more promising results (cf. Sect.3.3 and Sect.3.4).

2.4 Other variants of local binary pattern

The success of LBP in various computer vision tasks has inspired a number of researches on LBP variants. These researches aim for more robustness and efficiency by addressing the limitations of LBP or modifying the operator according to the needs of specific applications. This section briefly introduces several variants that improve LBP on three primary aspects: preprocessing, thresholding and encoding, and neighborhood topology. Readers who are interested in more types of variants should refer to the survey in [115].

2.4.1 Preprocessing

Preprocessing the input image is useful because it provides a new medium allowing LBP features to be extracted more precisely. The Gabor filter has been widely used for this purpose since it complements the LBP perfectly. It encodes appearance information over a broad range of scales, whereas the LBP capture fine details. Zhang et al. [163] filters images with four Gabor filter of difference scales and orientations before extracting LBP features. Their method achieves high performance in face recognition but suffers high dimensionality.

Ji et al. [59] extracts threshold-restricted LBP features from high-frequency coefficients of pyramid Haar wavelet for text characterization. It preserves and uniforms inconsistent text-background contrasts while filtering gradual illumination variations. Kim et al. [64, 66] build a human detector by obtaining CS-LBP

features from three wavelet-transformed sub-images. The detector can attain good accuracy and boost the performance speed to near real-time.

Tan and Triggs [129] present a simple and efficient preprocessing chain for facial images, including gamma correction, difference of Gaussian filtering, masking (optional) and contrast equalization. It counters the effects of illumination variations, local shadowing and highlights, while preserving the essential appearance details, thus greatly contributing to face recognition.

Several studies use edge detection to enhance the gradient information before the LBP computation. Yao and Chen [158] propose the use of Local edge patterns (LEP) and color features for color texture retrieval. The LEP is derived in a LBP-like manner from a binary edge image created with Sobel filter and thresholding. For shape localization, Huang et al. [56] describe the local appearance of each facial points by computing LBP features on both original and Sobel-filtered gradient magnitude images. A similar idea is implemented in [165] for face image representation but further applied on Gabor real and imaginary features.

Several other methods are also promising to be applied before extracting LBP features. For instance, the curvelet transformed images in medical image analysis [69], multi-scale heat kernel matrices for face recognition [71], etc.

2.4.2 Neighborhood topology

The circular shape of neighborhood is vital for rotation invariance. However, problems like face recognition are keen on anisotropic structures, allowing neighborhoods to be defined in different shapes. Liao and Chung [73] combine their variant of elliptical neighborhood with a local gradient (contrast) measure, resulting in much improved results than that of the ordinary LBP. Nanni et al. [99] examine several neighborhood topologies (circle, ellipse, parabola, hyperbola and Archimedean spiral) and encodings in their research for medical image analysis. The operator using quinary encoding in an elliptic neighborhood is shown to have the best performance. Meanwhile, the neighborhood in [111] consists of two orthogonal lines lying along the horizontal and vertical directions, respectively. Binary codes are obtained separately for each direction and the magnitude characterizing details such as edges and corners is then computed.

Wolf et al. [152] explore dierent ways of using bit strings to encode the similarities between patches of pixels. For every pixel of the image, the Three-patch LBP (TPLBP) considers a $w \times w$ central patch and S neighboring patches distributed uniformly on a ring of radius r around the pixel. It forms pairs of

patches from those α-patches apart along the circle, and compares their values with that of the central patch. Meanwhile, the Four-patch (FPLBP) uses two rings centered on the pixel. Their methods share a similar idea with the Multi-block LBP [75], in which the ordinary comparison between single pixels is replaced with the comparison between average gray values of square pixels blocks.

2.4.3 Thresholding and encoding

***n*-ary encoding.** It is possible to change the binary encoding to n-ary ($n > 2$) for better discriminative power. The Local ternary patterns (LTP) [129] encodes the gray-level differences into three values (1, 0, or -1) to effectively deal with near-uniform regions in face recognition. It splits every pattern into positive and negative components, from which histograms are computed separately and then concatenated. The Scale invariant LTP (SILTP) [74] adopts a similar encoding for background subtraction, yet each comparison is represented by two bits (01, 00, or 10). Nanni et al. [98, 99] examine different encodings (binary, ternary and quinary), in which the binary and ternary operate similarly to the LBP and LTP, while the quinary uses five values (-2, -1, 0, 1, or 2) and two thresholds. They show that the Elongated quinary patterns (EQP) performs best on medical image analysis, whereas the Elongated ternary patterns (ELTP) is the leading in classifying pain states from facial expressions.

Ahonen et al. [12] propose the soft LBP histogram using two fuzzy membership functions, i.e. one pixel typically contributes to more than one bin. A similar idea is introduced in [57] for ultrasound texture characterization.

The above methods are, however, no longer strictly invariant to local monotonic grayscale changes and their histograms are much longer.

Encoding style. Hafiane et al. [51] compares all pixels in a neighborhood with their median value, while the mean value is used in [44, 61]. The method of Fu and Wei [45] works similarly to the CS-LBP but additionally compares the center pixel with the mean of neighborhood. Xu et al. [155] defines four horizontal and vertical symmetric pixel pairs and one pair of the center and the average of neighbors. These methods achieve good performance but produce long histograms.

In their study of object detection, Trefny and Matas [137] introduce the Transition LBP (tLBP), which compares pairs of adjacent neighbors in clockwise direction, demonstrating the relation between neighbors. They also propose the Direction coded LBP (dLBP) that operates similarly to the CS-LBP but includes

the center pixel. It describes each comparison by two bits: the first bit indicates whether the center pixel is an extrema and the other denotes whether the difference of border pixels due to the center grows or falls. Chan et al. [25] makes pairwise comparisons of adjacent neighbors similarly to the tLBP, yet the process is in anticlockwise direction and the center pixel is included.

Mu et al. [97] design the Semantic LBP (S-LBP) and Fourier LBP (F-LBP) for accurately detecting human in photo albums. The S-LBP defines several continuous 1 bits an arch, which is represented by its principle direction and length, and computes a 2D histogram of patterns having at most one arch. In the latter, real valued color distance between the k^{th} samples and central pixel are computed and transformed into frequency domain. Low-frequency coefficients are then used to capture salient local structures around current pixel.

Zhang et al. [161] propose the Local derivative pattern for robust face recognition, which derivatively extracts various n-order spatial patterns from (n-1)-order ones, whereas the LBP simply defines first-order relations between the center pixel and neighbors. The Local directional pattern [58], on the other hand, computes edge responses in eight directions for every pixel using some edge detector (e.g. Kirsch, Prewitt, or Sobel) and encodes them into an 8-bit binary code.

Thresholding scheme. Heikkila et al. [54] replace the term $s(g_p - g_c)$ with $s(g_p - g_c + a)$. The higher the value of a is, the larger changes in gray level are allowed without affecting the thresholding results. Nevertheless, a relatively small value (e.g. $a = 3$) should be used to retain the discriminative power. This approach has been adopted in several studies, such as [45, 55, 99, 129, 152]. Meanwhile, Liao et al. [74] incorporate a factor τ to the function s, i.e. $s(g_p - \tau g_c)$, which helps the operator invariant against gray-level transform by a scale factor. This scheme is extended by Wang et al. [146] so that it is adaptively set for each pair of pixels. Pixels of the image are grouped into edge and texture types, whose distributions are then used to estimate the threshold.

2.4.4 Other types of variants

Rotation invariance. Beside the LBP^{ri} (cf. Sect.2.1.4), we can obtain rotation invariance from several other means. Ahonen et al. [11] show that rotations of the input image cause cyclic shifts of values in the LBP^{u2} histogram, and therefore discrete Fourier transform can be applied to construct rotation invariant features. Guo et al. approach the problem from two different aspects. The first method [46]

incorporates the directional statistical information, i.e. the mean and standard deviation of local absolute differences. In addition, the least square estimation is used to adaptively minimize the local difference for more stable directional statistical features. Meanwhile, the second method [47] uses the variance of a local neighborhood as an adaptive weight to adjust the contribution of LBP code in histogram computation. It assigns high variance for high frequency regions because these regions contribute more to the discrimination of texture images.

Multiple color channels. The LBP was originally developed for monochrome images, yet it can be extended for color images. Anbarjafari [13] proposes a face recognition system that computes LBP features separately on three HSI channels and studies different decision-making techniques to combine decisions from each channel. Zhu et al. [169] compute their Orthogonal combination of LBP on each channel of a color space (six color spaces are examined) then concatenate all sub-histograms to get the final descriptor. The Opponent color LBP [83], on the other hand, jointly considers texture and color. It computes patterns on individual color channels, and for pairs of color channels, takes the center pixel from one channel and neighboring pixels from the other. R-G and G-R are rejected due to high redundancy. Thus, six histograms (R, G, B, R-G, R-B, and G-B) are selected, resulting in a descriptor six times longer than an ordinary one.

Temporal dimension. Zhao et al. [164] propose the Volume LBP (VLBP), which considers a neighborhood volume of three frames, i.e. the current, previous and following frames, producing a 2^{3P+2}-dimensional histogram. However, it needs a large number of neighbors to attain enough robustness. They also introduce the LBP-TOP to simplify the computation, which extracts LBP features from three orthogonal planes (XY, XT and YT), incorporating spatial information in XY plane and spatial temporal co-occurrence statistics in XT and YT planes. Mattivi and Shao [87] extends the number of slices on every axis of the LBP-TOP. On the XY dimension, beside the original plane centered in the middle, they add two planes at positions of 1/4 and 3/4. The same is done for XT and YT dimensions. They also suggest applying LBP-TOP on gradient and Gabor images of orthogonal slices. Chan et al. [25] adopts the use of three orthogonal planes to construct their operator for representing dynamic texture and appearance information of mouth-regions. Together with the linear discriminant analysis, their operator drastically improves the performance of lip-biometric trait.

2.5 Applications of local binary pattern

In the last two decades, an increasing popularity of LBP features can be observed in computer vision. Many studies have been introduced, either proposing novel methodology or bringing LBP to new applications. This section briefly describes some representative variants from different areas.

The LBP was first proposed as a simple, yet efficient, multi-resolution approach for gray-level and rotation invariant texture classification [105]. This problem aims to classify an unseen texture sample into one of predefined classes by using rules derived from the learning of texture samples with known classes. Developed from the strong base of texture analysis, the LBP and its variants [46, 47, 51, 83, 158] perform so well in many comparative studies with publicly available texture datasets that they are believed to have no rival in this field.

Automatic face analysis has been useful in several applications, e.g. biometric identification, visual surveillance, human-machine interaction, etc. The challenges come from large variations of face appearance due to changes in pose, expression, illumination and other factors like age and make-up. The LBP is highly discriminative and invariant to monotonic illumination changes, thus suiting perfectly to the problem. That explains for the success of LBP in most subfields, including face detection [44, 61], face recognition [13, 56, 71, 73, 75, 111, 129, 152, 161, 163, 165], and facial expression recognition [45, 58, 98, 164].

The aforementioned problems take a lot of time to train the system effectively with discriminative samples, while the recognition could be done in no time afterwards. The interest region description is, on the other hand, more likely to be unsupervised. Local features are thus required to be distinctive, robust against common image transformations (e.g. viewpoint changes, image blur, scale and rotation), computationally simple and compact. The CS-LBP and studies in [40, 49, 169] are among those achieving good matching results.

The LBP has also gained a great deal of attention from the field of object detection and recognition because it can effectively describe the texture of different objects, such as car, animals, people, etc. [55, 91, 128, 137, 169]. The success of LBP in face analysis has motivated several studies to extend the target from face to the whole body. Pedestrian detection [64, 66, 97, 147, 155, 167] and human action recognition [87] are active topics of LBP researches.

When dealing with non-static environments like video footages, it is vital to separate the targeted objects from background scene before performing a higher-

level process, such as tracking or recognition. However, dynamic activities in the background, including illumination changes, swaying vegetation, rippling water, and flickering monitors, cause the segmentation extremely difficult. Furthermore, the background modeling algorithm should operate in real-time. Due to their computational simplicity and discriminative power, several LBP variants have been good options for background subtraction system [53, 54, 74, 146, 156, 162]

There are also many other applications of LBP. For example, text detection [59], image retrieval [127], biomedical image analysis [69, 99], ultrasound texture characterization [57] and lip-based speaker authentication [25]. More details are available in the book written by Pietikäinen et al. [115].

Chapter 3

Description of Interest Regions with Local Binary Pattern

This chapter presents the study of LBP for effective description of interest regions. Innovative encoding strategies and thresholding schemes are introduced to build two novel LBP region descriptors: *Intensity order quartet local binary pattern* (IOQLBP) and *Multi-scale region perpendicular local binary pattern (MSR-PLBP)*. These descriptors effectively match co-occurrence regions in pair of images and recognize similar objects in large databases. Evaluations on standard benchmarks show that they have better performances than that of state-of-the-art descriptors.

The content of the chapter is organized as follows. Section 3.1 gives the problem definition. Related works are described in Sect.3.2. Section 3.3 and Section 3.4 present the two proposed methods. Evaluations on the image matching and object recognition tasks are described in Sect.3.6 and Sect.3.7, respectively. Finally, Section 3.8 closes the study.

3.1 Problem definition

Feature detection and matching plays an important role in many computer vision applications. It detects salient features in a pair of images and establishes a set of correspondences from features in the first image to those in the second image. These correspondences set up the basis for higher-level vision tasks. For instance, stitching images seamlessly into a panorama image [21], 3D model reconstruction [117, 135] (cf. Fig.1.3), epipolar geometry estimation [86, 119], as well as texture classification [67], object recognition [55, 81], and human action recognition [87].

A general pipeline of feature extraction includes three primary steps. First,

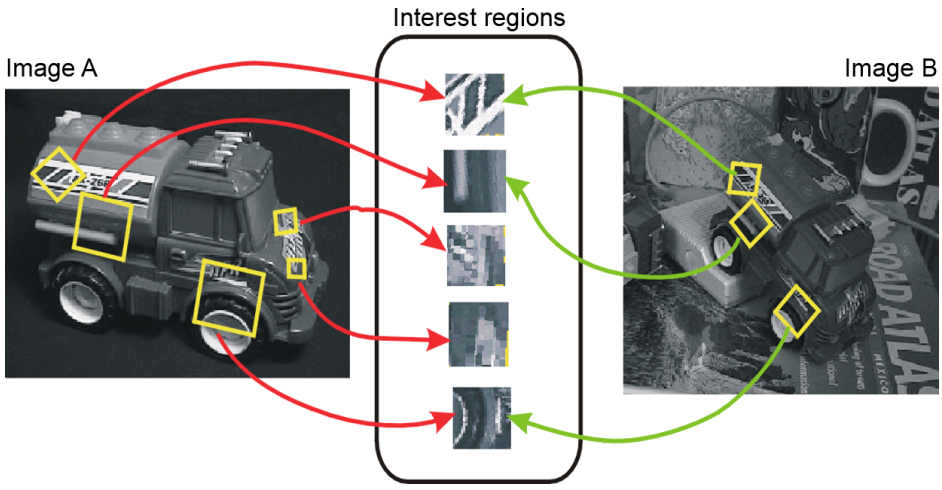


Figure 3.1: Interest regions are extracted from salient points in the two images. A pair of regions describing the same detail of the toy truck establishes a correspondence. The image is obtained from the lecture slide of Steve Seitz (http://courses.cs.washington.edu/courses/cse576/08sp/).

detecting *keypoint features* (or *interest points*) in the images, i.e. salient locations such as mountain peaks, building corners, doorways, or interestingly shaped patches. Second, describing these features by the appearance of patches of pixels surrounding the point locations, hence called *interest regions*. These regions are often defined to be scale-covariant or affine-covariant in order to handle basic image transformations (e.g. translation, rotation, scaling, etc.). Finally, computing a feature descriptor for every obtained interest region. The feature descriptor is usually represented in the form of histogram, i.e. an n-dimensional numerical vector. Correspondences from features in the first image to those in the second image can be established afterwards according to the distances between their vectors (e.g. the Euclidean or Manhattan distance). Figure 3.1 illustrates the extraction of interest regions from a pair of images.

The first two steps can be done with a region detector, which is selected from a number of approaches in the literature. Harris-Laplace/Hessian-Laplace [92, 93] and Laplacian of Gaussian [78] are scale-covariant, while Harris-Affine/Hessian-Affine [93], Intensity/Edge-based Regions (IBR/EBR) [138] and Maximally Stable Extremal Regions [86] are affine-covariant. The latter approaches are preferred because they cope with transformations other than scale and translation. A comprehensive survey of detectors is provided in [95].

The problem of interest region description covers the last step. It is completely not trivial in the sense that two images in a pair may significantly differ from each other due to geometric transformation like translation, rotation, and scale,

as well as photometric transformation like brightness or exposure changes (cf. Fig.3.19). A robust feature descriptor should produce similar descriptions of an interest region even when the viewing conditions vary, i.e. invariant to image transformations, or there are detection errors. In addition, it should be at the same time discriminative enough so that feature vectors of corresponding regions are closely related while far from those of other regions. How to build a descriptor satisfying these expectations is what this study aims for.

3.2 Related work

3.2.1 A review of related approaches

Researchers have explored different image properties, such as pixel intensity, edge, gradient and texture, to improve the performance of local descriptors. These features exhibit their advantages and drawbacks in different situations; therefore, a robust descriptor under all challenges is still an open problem. In this section, we briefly summarize recently published descriptors, while emphasizing those using gradient or texture since they are closely related to the proposed method.

Gradient-based descriptors have been long studied for region description because gradients can intuitively describe directional changes in an image. The Scale-invariant feature transform (SIFT) [81] is the most popular descriptor because of its good performance in various applications, such as object recognition [81], panorama image stitching [21] and 3D scene reconstruction [117]. It is a 3D histogram in which a 4×4 square grid and eight bins are used to quantize gradient location and orientation, respectively. The Gradient location-orientation histogram (GLOH) replaces the square grid in SIFT by a log-polar grid with three radial and eight angular bins. Experiments show that both SIFT and GLOH outperform conventional methods, such as shape context, steerable filters, spin images, differential invariants and moment invariants [94]. Ke and Sukthankar [62] apply Principal Components Analysis (PCA) to the normalized image gradient patch, thus making the PCA-SIFT descriptor more compact and distinctive than the original SIFT. Tola et al. [135] merge the ideas of SIFT and GLOH to propose DAISY, which is an efficient dense matching descriptor consisting of circular regions defined by Gaussians with increasing variance. Meanwhile, the Rotation-invariant feature transform (RIFT) [67] divides a region into concentric rings and computes a gradient orientation histogram within each ring. The

Multisupport region order based gradient histogram (MROGH) [40] combines intensity orders and gradients in multiple support regions to boost its performance significantly. These methods are robust to common image transformations, yet unstable under complex illumination changes.

Intensity order-based descriptors are preferred to gradient-based descriptors in handling the illumination challenge since the intensity order is invariant to monotonic illumination changes. SMD [48] selects pixels pairs from extremal regions and penalizes the order flips between pairs to gain robustness to localization error as well as to intensity noise. The Ordinal spatial intensity distribution (OSID) descriptor [131] captures the texture and structure information by binning in both the ordinal space and spatial space, and hence is invariant to monotonic brightness changes. The Local intensity order Pattern (LIOP) [148] uses the intensity order of all sampled neighboring points to exploit the local information effectively. The disadvantage of this approach is that the intensity order is sensitive to Gaussian noise, especially when nearby pixel values are close.

LBP-based descriptors are good alternatives thanks to the computational simplicity and strong tolerance against illumination changes of the LBP operator [105]. The LBP does not consider gray values directly but abstracts the relation between pairs of pixels to some degree, thus effectively avoiding the effects of noise. Despite their wide applications in face recognition [13, 129], image retrieval [127] and background subtraction [54, 74], LBP features had been used limitedly in region description because of high-dimensional histograms until the emergence of the Center-symmetric local binary pattern (CS-LBP) [55]. The CS-LBP descriptor adopts the squared grid of SIFT and replaces gradients by CS-LBP textures, thus effectively combines good properties of SIFT and LBP. Gupta et al. [49] concatenate two features, the Histogram of Relative Intensities (HRI) and Center-symmetric local ternary patterns (CS-LTP), to obtain a HRI-CSLTP descriptor that is robust to Gaussian noise. The Local ordinal contrast pattern (LOCP) [25] encodes patterns from pairwise ordinal information in adjacent circular neighborhoods and performs effectively in lip-based speaker authentication. The Multisupport region rotation and intensity monotonic invariant descriptor (MRRID) [40] employs a similar framework to that of MROGH [40] but replaces gradients by CS-LBP-based texture features.

In addition, appropriate selections of image filters or learning strategies may improve the performance. Song et al. [123] extract 2D discrete cosine transform (DCT) features in the frequency domain and select a subset of the reordered

coefficients as their compact descriptor. To address the problem of illumination sensitivity, Zambanini et al. [160] use multi-scale and multi-oriented even Gabor filters while Shi et al. [122] introduce fuzzy reasoning to their descriptor.

Region division is an essential technique to improve the distinctiveness of a descriptor. The region is divided into several sub-regions, and features in each sub-region are then concatenated. Most previous division methods are based on the spatial location. For example, SIFT [81], CS-LBP [55], HRI-CSLTP [49] use a 4×4 Cartesian grid while GLOH [94] utilizes a log-polar grid. The spatial-based approach maintains rotation invariance by rotating the region following a dominant orientation. However, experiments on synthetic [150] and real [40] data show that estimating this orientation is error prone and time consuming. RIFT [67] uses concentric rings to avoid this issue, yet the ring-shaped division is less discriminative than grid-shaped division. Segmentation according to the intensity order has recently been proposed [40] and used in MROGH, MRRID [40] and LIOP [148]. This approach is theoretically rotation invariant, thus is able to overcome the above limitations. However, a minor drawback is that it is less spatially discriminative than the spatial-based approach.

Most aforementioned methods have been developed on a single support region. Although the support size greatly affects the matching accuracy [95], it is usually empirically determined for specific applications. In addition, a single region is not discriminative enough to distinguish incorrect matches from correct matches [40, 96]. Mortensen et al. [96] augment SIFT with a global context vector that adds curvilinear shape information from a much larger neighborhood and thus reduces the number of mismatches. Bin et al. [40] extract support regions of various sizes from a detection point and compute a feature histogram in each region. These descriptors perform effectively under common transformations, yet heavy computation arises as a side effect.

3.2.2 Brief descriptions of competing approaches

The following six descriptors are selected to participate in the comparative studies in Sect.3.6 and Sect.3.7.

The **SIFT** [81] computes a 3D histogram of gradient location and orientation, where location is quantized into a 4×4 square grid and gradient angle is quantized into eight orientations (cf. Fig.3.2).

The **DAISY** [135] computes gradient magnitude layers in different orientations and applies Gaussian convolution. To compute the histogram for a region,

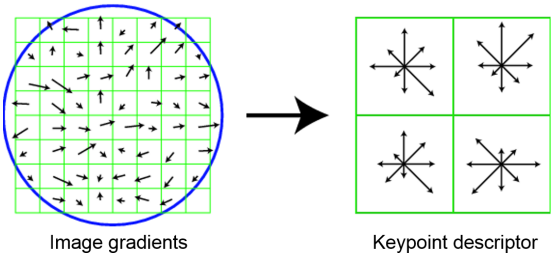


Figure 3.2: An example of SIFT with the 2×2 square grid. The image is obtained from the slide of David Lowe [81].

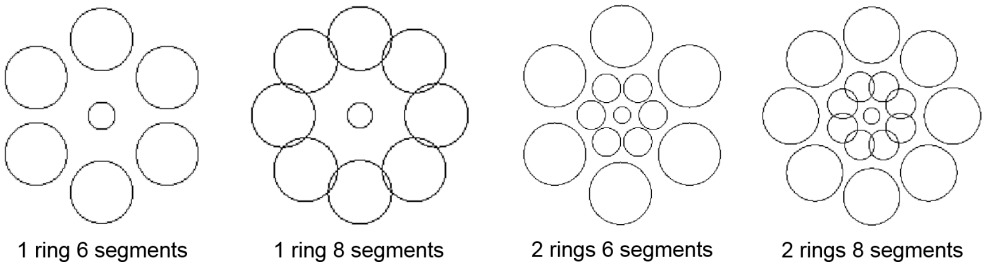


Figure 3.3: Different configurations of DAISY.
Different configurations of DAISY. The T1-8-2r8s is in the rightmost figure.

it reads corresponding values from the pre-computed convoluted layers. The T1-8-2r8s configuration [151] is selected for implementation (cf. Fig.3.3). It includes two rings, eight segments per ring, and quantizes the gradient angle into eight orientation bins.

The **HRI-CSLTP** [49] adopts the 4×4 grid in SIFT where each cell contains 16 bins of relative intensities, the j^{th} bin stores pixels of the cell that have intensities in the j^{th} interval of the intensity range, and 8 bins of CS-LTP patterns. The HRI-CSLTP finally concatenates two feature histograms.

The **LIOP** [148] divides a local patch into six ordinal bins using the overall intensity order. A LIOP pattern is defined as a permutation of gray values of four neighboring points of a pixel. It computes 24-dimensional pattern histograms for every ordinal bin and concatenates them to a unique feature vector.

The **MROGH** [40] divides each of four support regions into six ordinal bins in a similar manner to [148]. It then computes histograms of eight gradient orientations for every ordinal bin and combines them together.

The **MRRID** [40] has a similar framework to that of MROGH but replaces gradient orientations by 16 CS-LBP-like texture patterns. The descriptor is computed on four support regions and four ordinal bins.

3.2.3 Homography estimation

To determine the number of correspondences and correct matches for the evaluation protocol in Sect.3.6.1, a *homography* is required. Hartley and Zisserman [52] have given a definition of *homography* as follows.

Definition 1 A *projectivity* is an invertible mapping h from points in $\mathbb{P}^2$ (i.e. homogeneous 3-dimensional vectors) to points in $\mathbb{P}^2$ such that three points x_1, x_2 and x_3 lie on the same line if and only if $h(x_1)$, $h(x_2)$ and $h(x_3)$ do.

Projectivities form a group since the inverse of a projectivity is also a projectivity, and so is the composition of two projectivities. *Collineation*, *projective transformation* or *homography* are synonymous with *projectivity*.

Theorem 1 A mapping $h : \mathbb{P}^2 \mapsto \mathbb{P}^2$ is a projectivity if and only if there exists a non-singular 3×3 matrix H such that for any point in $\mathbb{P}^2$ represented by a vector x it is true that $h(x) = Hx$.

Any point in $\mathbb{P}^2$ is represented as a homogeneous 3-vector, x, and Hx is a linear mapping of homogeneous coordinates. The theorem is an equivalent algebraic definition of a projectivity. It asserts that any projectivity arises as such a linear transformation in homogeneous coordinates, and that conversely any such mapping is a projectivity. The proof is available in [52].

Accordingly, an alternative definition of a projective transformation is given:
Definition 2 A planar projective transformation is a linear transformation on homogeneous 3-vectors represented by a non-singular 3×3 matrix:

$$\begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{3.1}$$

or more briefly, $x' = Hx$.

The matrix H is a *homogeneous matrix*, i.e. multiplying it with an arbitrary non-zero scale factor does not alter the projective transformation. There are eight independent ratios amongst the nine elements of H, and it follows that a projective transformation has eight degrees of freedom.

Two images can be related by a homography if the following conditions are held [28] (cf. Fig.3.4):

- Intrinsic parameters of two cameras, namely O1 and O2, are identical.
- The scene consists of rigid geometry.
- The observed scene geometry is planar or the camera movement from O1

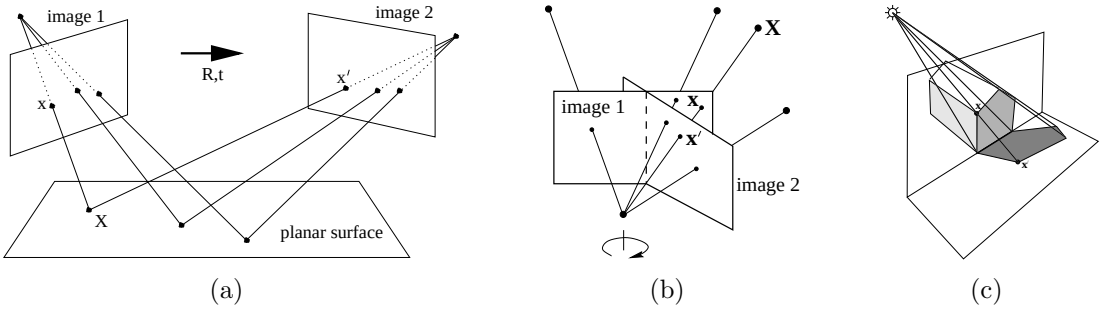


Figure 3.4: The projective transformation, $x' = Hx$, between (a) two images induced by a world plane, (b) two images with the same camera center (e.g. a camera rotating about its center or varying its focal length), and (c) the image of a plane (the end of the building) and the image of its shadow onto another plane (the ground plane). Figures are obtained from [52].

to O2 is purely rotational.

All benchmarks in Sect.3.6 are strictly set up to ensure of such a projectivity. Therefore, to establish the ground truths for pairs of images, it is required to have accurate corresponding matrices H. A common homography estimation includes two steps [94]. First, an approximation of the homography is computed using manually selected correspondences. It is used to warp the transformed image to be roughly aligned with the reference image. Second, a robust small baseline homography estimation algorithm computes an accurate residual homography between the reference image and the warped image, with automatically detected and matched interest points [52]. The composition of approximate and residual homographies results in an accurate homography between the images. Alternatively, a recently proposed method in [28] has been shown to increase the accuracy of homographies by using a differential evolution approach for the optimization of a new and feature-independent cost function.

3.2.4 The region size and its effect

Most affine-covariant detectors output elliptical interest regions of various sizes (except those of MSER and EBR that are required to be normalized into ellipses). Harris-Affine, Hessian-Affine and MSER tend to produce many small regions while the others yield larger regions.

According to [95], the region size is measured as the geometric average of the half-length of both axes of the ellipse, which corresponds to the radius of a circular region with the same area. There is a distinction between *distinguished*

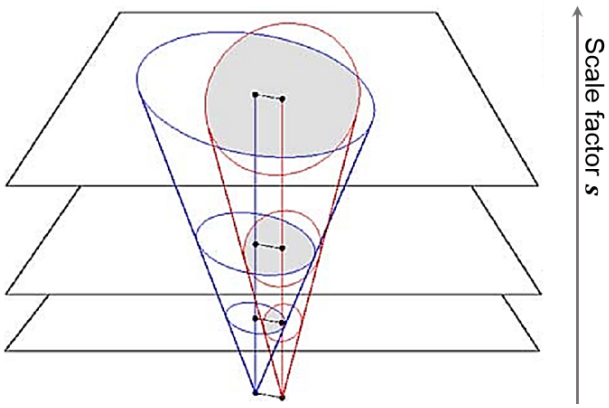


Figure 3.5: The overlap between two regions changes proportionally to the scaling factor s. The image is obtained from [95].

regions and *measurement regions*, in which the former refers to the set of pixels that have effectively contributed to the affine detector response while the latter can be any region obtained by an affine-covariant construction.

Larger regions typically have better chances of overlapping other regions and higher discriminative power as they contain more information, though there is a higher risk of being occluded or not covering a planar part of the scene. Simply rescaling the region, i.e. using a different measurement region, suffices to boost the overlap performance of a region detector [95]. This can be interpreted as follows. Given an elliptical distinguished region, the measurement region is also an ellipse centering on this region but with an arbitrary scale. From a geometrical point of view, varying the scaling defines a cone out of the image plane (with elliptical cross-section) with s being a distance on the cone axis (cf. Fig.3.5). There are such two cones in the reference image: one from the distinguished region of this image and the other from the mapped distinguished region of the transform image. The cones remain separate as the scaling goes to zero while the relative amount of overlap approaches unity when the scaling goes to infinity. In this study, the descriptors are usually computed on regions whose sizes are at least three times larger than that of the corresponding detected regions, i.e. $s \geq 3$. These regions are called *support region*. Some approaches like MROGH and MRRID use multiple support regions by varying the factor s.

3.3 The IO-QLBP descriptor

3.3.1 Method overview

Inspired by the effectiveness of CS-LBP [55] in interest region description, we propose the IO-QLBP descriptor to boost the matching accuracy and speed to

a comparable or higher level than that of modern descriptors. The following properties enable IO-QLBP to achieve good robustness.

1. The QLBP operator efficiently produces a 16-dimensional feature vector from four neighbors, whereas the CS-LBP$_{8,2}$ requires eight neighbors. It well preserves the property of simultaneously capturing gradient and texture while halving the interpolation cost.
2. An adaptive thresholding scheme is introduced to address the drawback of global constant threshold (cf. Sect.2.3). The new scheme locally sets a threshold value for every pair of neighbors during the encoding, enabling the operator to be not only more robust to noise but also invariant against gray-level scaling by a multiplying constant.
3. Each pattern is associated with a weight value denoting its gradient strength, which is then accumulated in the corresponding histogram bin. This weighting scheme improves the matching accuracy better than Gaussian and uniform schemes while requiring no additional complex operation.
4. The IO-QLBP descriptor extracts features from multiple support regions and aggregates them into intensity orders using the region division strategy of [40]. Accordingly, the descriptor is highly discriminative and rotation invariant without any dominant orientation estimation.

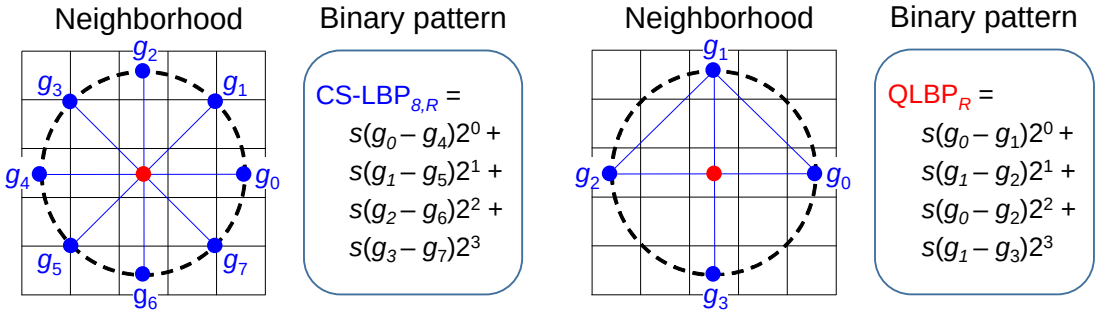
3.3.2 The QLBP texture operator

The Quartet local binary pattern (QLBP) defines for every pixel (x, y) in the image four neighbors that are evenly spaced on a circle of radius R centered at (x, y) then encodes the information within the neighborhood as follows:

$$\begin{aligned} QLBP_R(x,y) &= \sum_{p=0}^{1} s(g_p - g_{p+1})2^p + s(g_p - g_{p+2})2^{p+2} \\ s(g_a - g_b) &= \begin{cases} 1 & g_a - g_b > \tau \min(g_a, g_b), \\ 0 & otherwise \end{cases} \end{aligned} \tag{3.2}$$

where g_p corresponds to the gray value of the p^{th} neighbor and τ is a scaling factor. The term "quartet" indicates that the set of neighbors has four points. Figure 3.6 shows an example of QLBP, compared with the CS-LBP$_{8,R}$.

The scaling factor τ is empirically selected so that $t \in [0, 0.05]$. Although the performance may slightly change with different values of τ, this interval suggests a good starting point for parameter tuning. The higher the τ is, the larger



(a) CS-LBP$_{8,R}$ (b) QLBP$_R$

Figure 3.6: The neighborhood and the binary pattern of QLBP$_R$. The corresponding information of CS-LBP$_{8,R}$ is shown for comparison.

intensity changes are allowed without affecting the results. $\tau = 0$ leads to the original mode in [105], while $\tau > 0.05$ begins to decrease the performance. Prior works use a global constant ($T = 0.01$ [54] and $T \in [0, 0.02]$ [55]), or associate a similar $\tau = 0.05$ with the center pixel [74]. The QLBP, on the other hand, incorporates τ into pairs of neighbors, producing patterns that are more adaptive and tolerant to drastic transformations. Note that the τ in [74] is tuned for background subtraction, its range of value cannot be applied for interest region description because the natures of two problems are different. Measuring the amount of noise in the image and studying the behavior of gray-level transforms are promising approaches to find τ theoretically. However, they involve deep knowledge of image analysis and somewhat go beyond the main purpose of this study, thus they are left for future work.

Heikkila et al. [55] examine different numbers of neighbors P and radii R then show that the CS-LBP$_{8,2}$ performs best. That inspired us to seek for a more efficient structure while maintaining an acceptable accuracy. The proposed QLBP with three following properties has met our expectation.

First, the QLBP operator efficiently revives the gradients in different orientations by considering four pairs of pixels established from four neighbors. These pairs include two center-symmetric pairs depicting gradients along the horizontal and vertical directions and two non-center-symmetric pairs for diagonal directions. Meanwhile, the CS-LBP needs eight neighbors to fulfill the same goal. The QLBP is thus computationally simpler than the CS-LBP and the property of simultaneously capturing gradient and texture is well conserved.

Second, the adaptive thresholding scheme enables QLBP to be not only robust to noise but also invariant against gray-level scaling by a multiplying constant.

		No transformation				Noise				Gray-level scaling (s = 2)			
Pixel intensity		49 62 42 43 58 67 55 60 73				49 61 42 41 58 67 57 59 73				98 124 84 86 116 134 110 120 146			
Bit i^{th}		*3*	*2*	*1*	*0*	*3*	*2*	*1*	*0*	*3*	*2*	*1*	*0*
CS-LBP (T = 3)	T	3	3	3	3	3	3	3	3	3	3	3	3
	Binary pattern	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**1**
QLBP (τ = 0.05)	$\min(p_a, p_b)\tau$	3	2.15	2.15	3.1	2.95	2.05	2.05	3.05	6	4.3	4.3	6.2
	Binary pattern	**0**	**1**	**1**	**1**	**0**	**1**	**1**	**1**	**0**	**1**	**1**	**1**

Figure 3.7: The CS-LBP and QLBP operators under three transformations. Pixels or patterns affected by a transformation are circled and shown in red.

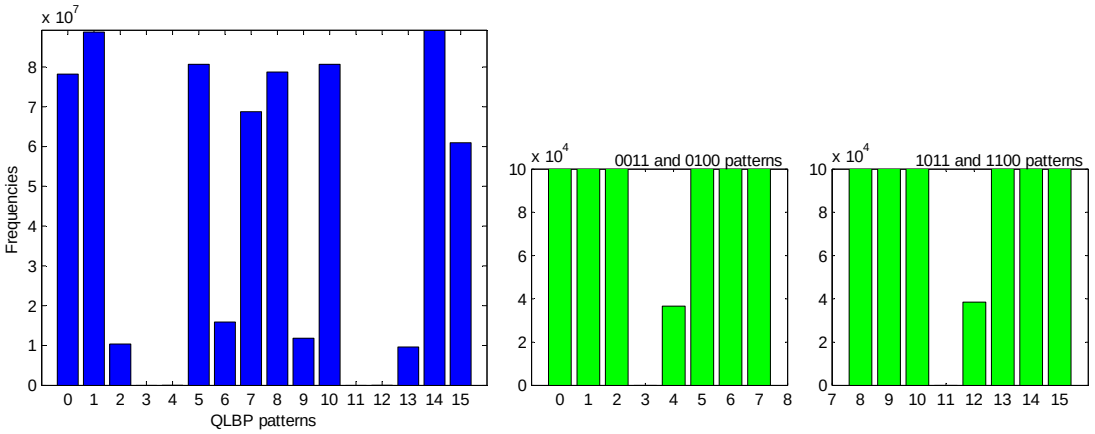


Figure 3.8: The statistics of QLBP patterns ($\tau \geq 0$). Low-frequency patterns are zoomed on the right, in which 0011_2 and 1011_2 never appear while 0100_2 and 1100_2 appear with frequencies of around 0.005%.

It locally sets a threshold value for every pair of neighbors during the encoding. Liao et al. [74] handle this transformation by a similar scaling factor, yet they associate it with the center pixel. Figure 3.7 shows that both CS-LBP and QLBP operators are robust to noise, yet ours is more stable under gray-level scaling.

Third, the dimensionality of the QLBP histogram is reduced from 16 dimensions to 12 dimensions. Due to the relation between pairs of neighbors, the patterns 0011_2, 1011_2, 0100_2 and 1100_2, appear with very low frequencies ($0 \sim 0.005\%$). They are thus replaced by nearest available patterns. To validate this assumption, we randomly select over a hundred of images from [3] (also used

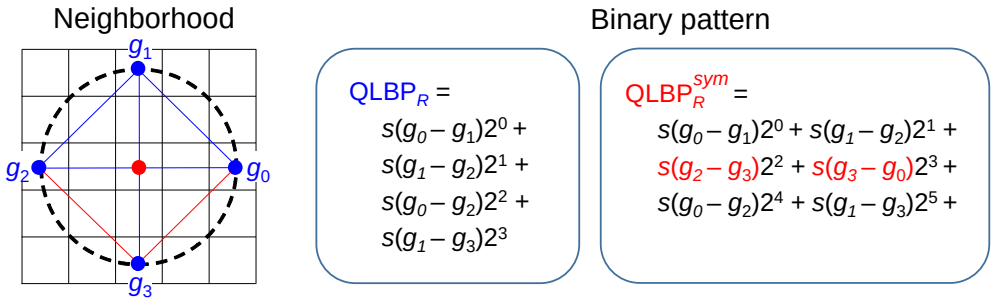


Figure 3.9: The neighborhood and the binary pattern of QLBP_R^{sym} in comparison with QLBP_R. The QLBP_R^{sym} additionally compares two pairs of neighbors, which are denoted by red lines in the neighborhood.

for parameter tuning in Sect.3.5) and compute the frequency of each pattern in these images (cf. Fig.3.8). It can also be proved by applying the transitivity rule on gray values g_0, g_1, and g_2 following Eq.3.2. In practice, this reduction can be implemented easily with a lookup table.

3.3.3 The symmetric QLBP texture operator

The good properties of the QLBP help us to build a region descriptor that is highly robust to challenging image transformations and on a par with state-of-the-art approaches. Nevertheless, there is still much room for improvement. We notice that the QLBP tends to favor the area around the second neighbor (i.e. g_1), thus encoding the information in a biased manner. Therefore, we revise the QLBP into a symmetric form called QLBP^{sym} (see Fig.3.9):

$$QLBP_R^{sym}(x,y) = \sum_{p=0}^{3} s(g_p - g_{(p+1) \bmod 4})2^p + \sum_{p'=0}^{1} s(g'_p - g_{p'+2})2^{p'+4}$$
$$s(g_a - g_b) = \begin{cases} 1 & g_a - g_b > \tau \min(g_a, g_b), \\ 0 & otherwise \end{cases} \tag{3.3}$$

The QLBP^{sym} originally has $2^6 = 64$ distinct patterns. Again, it is possible to reduce the set of patterns to 24 elements, as done similarly in the QLBP. The QLBP^{sym} makes two additional comparisons on the same neighbor set, thus encoding more information from neighborhood but only slightly increasing the computation cost. In this way, the QLBP^{sym} is expected to boost the matching accuracy better than that of the asymmetric variant.

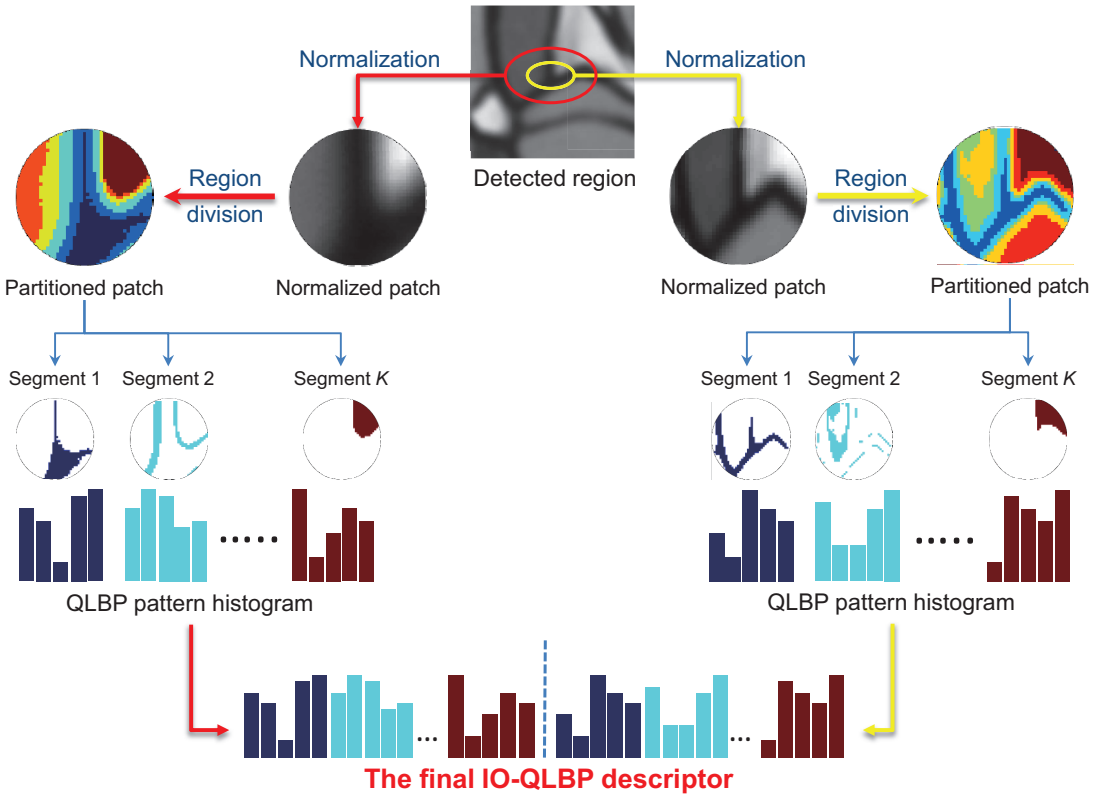


Figure 3.10: The construction pipeline of the IO-QLBP descriptor with two support regions.

3.3.4 The IO-QLBP descriptor pipeline

We first obtain interest regions of N different scales from an interest point with some affine-covariant detector. The regions are normalized to circular regions of the same radius before segmented according to their overall intensity orders. For each normalized region, we then compute QLBP patterns and pool them into corresponding segments. The IO-QLBP descriptor, whose name implies the cooperation between the intensity order-based region division and the QLBP texture, is finally built by concatenating histograms in every segment of N regions. The construction pipeline with $N = 2$ is illustrated in Fig.3.10. Note that the notation "QLBP" in this context refers to both QLBP and QLBP^{sym} operators. A note will be given if it is necessary to treat them separately.

3.3.4.1 Interest region detection

Interest regions are detected with an affine-covariant detector, such as Hessian-Affine or Harris-Affine. To achieve better performance, we vary the scaling factor to obtain multiple support regions of increasing scales that are centered on the

originally detected region. We simply follow [94] to normalize all support regions to circular patches of uniform diameter (41 pixels).

Smoothing is usually essential because the intensity order is sensitive to noise. We first use a Gaussian filter with sigma σ_a to smooth the input image then, after the normalization step, use another Gaussian filter with sigma σ_b to eliminate noises that may occur during the bilinear interpolation. The values of σ_a and σ_b should lie between 1 and 1.5 to maintain an adequate image quality.

3.3.4.2 Region division

As discussed in Sect.3.2, the region division strategy is able to improve the distinctiveness of a descriptor significantly. A typical approach divides a region into several sub-regions from which feature histograms are computed separately before concatenated into a global histogram. We adopt the intensity order-based strategy from [40] because it provides rotation invariance with less computation cost and errors than that of the grid-shaped division. It is also more stable than the ring-shaped division that greatly depends on the number of rings.

The approach of [40] partitions a normalized patch into K segments according to the intensity order of pixels within the region. Let $S = \{x_1, x_2, \ldots, x_n\}$ denote a patch containing n pixels and g_i the intensity (i.e. gray value) of x_i. It first obtains $\{f(1), f(2), \ldots, f(n)\}$, which is the permutation of the indices $\{1, 2, \ldots, n\}$ such that $g_{f(1)} \leq g_{f(2)} \leq \cdots \leq g_{f(n)}$. In other words, n pixels are sorted by their intensities in a non-descending order. K segments is then defined by locating $K + 1$ boundary points.

$$t_k = g_{f(s_k)} : t_0 \leq t_1 \leq \cdots \leq t_K, \quad s_k = \begin{cases} \lceil \frac{n}{K} k \rceil & k = 1, 2, \ldots, K \\ 1 & k = 0 \end{cases} \tag{3.4}$$

Finally, n pixels are pooled into K segments:

$$S_k = \{x_j \in S \,|\, t_k - 1 \leq g_j \leq t_k\}, \; k = 1, 2, \ldots, K \tag{3.5}$$

Figure 3.10 gives an example of intensity order-based division, in which individual segments within a patch are shown in different colors.

3.3.4.3 Feature construction

We compute QLBP histograms separately in K segments:

$$H_k = \left[h_{k_0}, h_{k_1}, \ldots, h_{k_{m-1}}\right] \tag{3.6}$$

with

$$h_{k_j} = |\{x \,|\, x \in S_k, \phi\,(QLBP(x)) = j\}| \tag{3.7}$$

where m is the number of distinct codes after eliminating low-frequency codes (i.e. m = 12 for QLBP and m = 24 for QLBP^{sym}), S_k is the k^{th} segment, $QLBP(x)$ denotes the binary code generated at a pixel x, and ϕ is defined to map a pattern to its corresponding histogram bin index according to a lookup table.

The IO-QLBP descriptor is, by definition, not able to distinguish textures having the same binary code but different magnitudes. Although this property is necessary for the invariance to illumination changes, it also causes the descriptor less robust to geometric transformations like viewpoint change or rotation. We alleviate the issue by associating a weight value w with each pattern $QLBP(x)$, which characterizes the total gradient magnitude around the pixel x. It is then accumulated in the histogram bin corresponding to the pattern.

$$w(QLBP(x)) = \sum_{p=0}^{1} |g_p - g_{p+1}| + |g_p - g_{p+2}| \tag{3.8}$$

Meanwhile, the weight value for QLBP^{sym} is:

$$w(QLBP^{sym}(x)) = \sum_{p=0}^{3} |g_p - g_{(p+1) \bmod 4}| + \sum_{p'=0}^{1} |g_{p'} - g_{p'+2}| \tag{3.9}$$

In simple words, the above weighting scheme collects gradient magnitudes from all pairs of neighbors during the encoding (cf. Eq.3.2 and Eq.3.3). We then improve the term h_{k_j} in Eq.3.6 as follows:

$$h_{k_j} = \sum_{x \in S_k,\, \phi(QLBP(x))=j} w(QLBP(x)) \tag{3.10}$$

Experimental results show that the proposed weighting scheme gives better performance than that of uniform and Gaussian weighting schemes (cf. Sect.3.5).

To keep the pattern rotational invariant, for a pixel x, the first neighbor is

located along the radial direction from the center of the normalized patch to x such that x lies between two points. Other three neighbors are anticlockwise sampled. Our method requires no interpolation because it is not grid-based and the QLBP is quantized by its nature. Therefore, the IO-QLBP descriptor achieves greater efficiency than the SIFT with tri-linear interpolation.

In a single support region, two non-corresponding points may be coincidentally similar to each other or two corresponding ones may be ignored due to localization error. This is because the single region is usually not meaningful enough to distinguish incorrect matches from correct ones. Multiple support regions capture the information in different scales to clear the ambiguity, thus significantly boosting the performance of a detector [95] or a descriptor [40]. We implement the above steps on N support regions and combine the resulting histograms together. The IO-QLBP descriptor is finally defined as follows:

$$IO - QLBP = \bigoplus_{\substack{i=1,\ldots,N \\ k=1,\ldots,K}} H_k^i \tag{3.11}$$

where N and K are the numbers of support regions and segments, respectively, and $\bigoplus$ denotes the concatenation of histograms. Experimental results (cf. Sect.3.6 and Sec.3.7) show that the IO-QLBP with two support regions can well approximate the MROGH using four regions, indicating the advantages of QLBP texture over gradient in terms of discriminative power and computation cost.

3.4 The MSR-PLBP descriptor

3.4.1 Method overview

The Multi-scale region perpendicular local binary pattern (MRS-PLBP) provides a simple yet effective mean of region description. It is able to handle most common image transformations while requiring lower computation cost than that of many modern methods. The following properties are key factors for its robustness.

1. The PLBP captures textures within the local neighborhood of a pixel by considering the relation between the pixel and its four neighbors, which are distributed evenly on two fixed perpendicular axes originated at the pixel. The brightness contrast of the region segment to which the pixel belongs is measured and incorporated to the thresholding scheme, providing the PLBP more robustness to illumination changes.

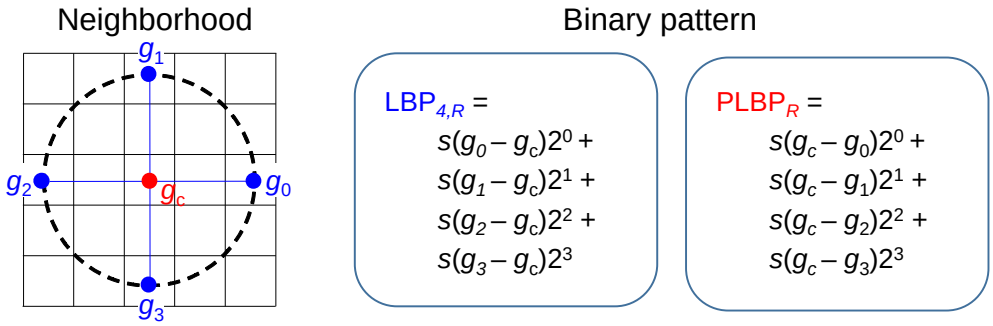


Figure 3.11: The neighborhood and the binary pattern of PLBP in comparison with $\text{LBP}_{4,R}$. The PLBP shares a very similar encoding to that of $\text{LBP}_{4,R}$, yet the pattern generation and the thresholding scheme clearly differentiate PLBP from $\text{LBP}_{4,R}$.

2. To diminish the risk of information loss in near-uniform regions, which is a drawback of the original LBP, a novel scheme of pattern generation is introduced. It produces for each pixel a set of pattern candidates. These candidates have distinct "strength", depending on the amount of gradient they capture, thus contributing to the histogram computation differently. This property clearly distinguishes PLBP from other LBP operators.
3. The Multi-scale region (MSR) division scheme defines multiple support regions from an interest point then sequentially performs ring-shaped and intensity order-based segmentations on each region to create a set of segments. Accordingly, the descriptor can attain highly discriminative power and be purely rotation invariant. Although each technique has been used independently, their combination is unique and more effective.
4. The MSR-PLBP descriptor extracts PLBP patterns from multiple support regions and pools them to corresponding segments. Histograms are then computed separately in every segment before concatenated into a single feature vector. Descriptors using a single support region (aka single-region) usually conflict with those using multiple support regions (aka multi-region) in terms of matching accuracy and computation cost. Meanwhile, the MSR-PLBP provides a flexible framework that enables users to acquire either a single-region or a multi-region descriptor by parameter tuning.

3.4.2 The PLBP texture operator

Let (x, y) denote the coordinate of the pixel being considered. The Perpendicular local binary pattern (PLBP) first establishes two perpendicular axes intersecting at (x, y) and a circle of radius R centered at (x, y). Intersections of each

Algorithm 1 The pattern generation scheme of the PLBP feature

Input: g_c - the gray value of the center pixel, $\{g_0, g_1, g_2, g_3\}$ - the gray values of four neighbors, and T - the threshold
Output: $E = \{e_1, e_2, ..., e_n\}$ - the set of candidates
$W = \{w_1, c_2, ..., w_n\}$ - the set of associated weights

```
1:  E = {0}, W = {0}                                   // initialize e_1 = 0, w_1 = 0
2:  for p = 0, ..., 3
3:    if |g_c − g_p| ≤ T                               // intensities are very close
4:      E_tmp = ∅, W_tmp = ∅
5:      for each candidate e ∈ E
6:        e_1 ← e + 2^p · 0                            // bit p^th = 0
7:        e_2 ← e + 2^p · 1                            // bit p^th = 1
8:        w_1, w_2 ← w
9:        add e_1, e_2 to E_tmp and w_1, w_2 to W_tmp
10:     end for
11:     E ← E_tmp, W ← W_tmp
12:   else                                             // dissimilar
13:     if g_c − g_p > T
14:       update all e ∈ E by e ← e + 2^p · 1
15:     else
16:       update all e ∈ E by e ← e + 2^p · 0
17:     end if
18:     update all weight w ∈ W by w ← w + 1
19:   end if
20: end for
```

axis and the circle define four neighbors of PLBP. The information within the neighborhood is then encoded as follows:

$$\mathrm{PLBP}_R(x,y) = \sum_{p=0}^{3} s(g_c - g_p)2^p, \quad s(z) = \begin{cases} 1 & z > T, \\ 0 & z < -T, \\ \varnothing & |z| \leq T \end{cases} \tag{3.12}$$

where R is the radius of the neighborhood, g_c and g_p correspond to the gray values of the center pixel at (x, y) and four neighbors respectively. The scalar T ($T > 0$) is estimated from the brightness contrast of the segment to which the center pixel belongs (cf. Sect.3.4.3). Figure 3.11 shows an example of PLBP in comparison with the $\mathrm{LBP}_{8,R}$. The symbol $\varnothing$ indicates that no thresholding decision is made for $s(z)$ because of ambiguity. The encoding therefore cannot process as usual and a special procedure, i.e. the Algorithm 1, is used instead.

The dissimilar pairs of pixels are usually more reliable than the similar pairs, because the Gaussian noise is likely to alter the order of pixels whose gray values are close to each other. Therefore, Equation 3.12 generates only patterns that have large gray-level differences. In this way, a number of noisy patterns are filtered, yet the discriminative power is affected, especially in near-uniform regions. Algorithm 1 addresses this issue by innovatively generating a set of PLBP pattern candidates for each pixel rather than conventionally assigning a single pattern. The case of $\varnothing$ is thus resolved in lines 4-11. To construct the descriptor, each candidate e_j at the location (x, y) is associated with a weight value:

$$w\left(e_j(x,y)\right) = \sum_{p=0}^{3} HS(|g_c - g_p|), \qquad h(z) = \begin{cases} 1 & z > T, \\ 0 & z \leq T \end{cases} \tag{3.13}$$

where T is the threshold in Eq.3.12 and $HS(z)$ is the Heaviside function at T. In simple words, the weight value characterizes the "strength" of a pattern by counting the number of gray-level differences that are large enough to be immune to noise. We will discuss in detail how to build a descriptor from these candidates and weights in the next subsection.

The PLBP and $\text{LBP}_{4,R}$ [105] share a similar method of comparing pixels, yet our operator has several properties that make it unique. It incorporates the brightness contrast of a region segment to the thresholding scheme, enabling the operator to attain more robustness to noise and to different illumination changes. Previous approaches usually define the threshold as a non-negative global constant [54, 55, 105, 129] or an adaptive term that varies proportionally to the center pixel [74]. They either produce a fix value or cover a small neighborhood around each pixel. The PLBP, on the other hand, balances the stability and adaptability by operating on a larger image region. In addition, it generates a set of pattern candidates for each pixel in the image. Each candidate is associated with a weight value defining its contribution to the histogram computation. This pattern generation scheme puts more emphasis on strong patterns (i.e. those having adequate gray-level differences) while utilizing weaker patterns in a selective manner. As a result, the PLBP performs effectively on near-uniform image regions.

3.4.3 The MSR-PLBP descriptor pipeline

The MSR scheme first sequentially performs ring-shaped and intensity order-based segmentations on every support region, resulting in a set of discrimina-

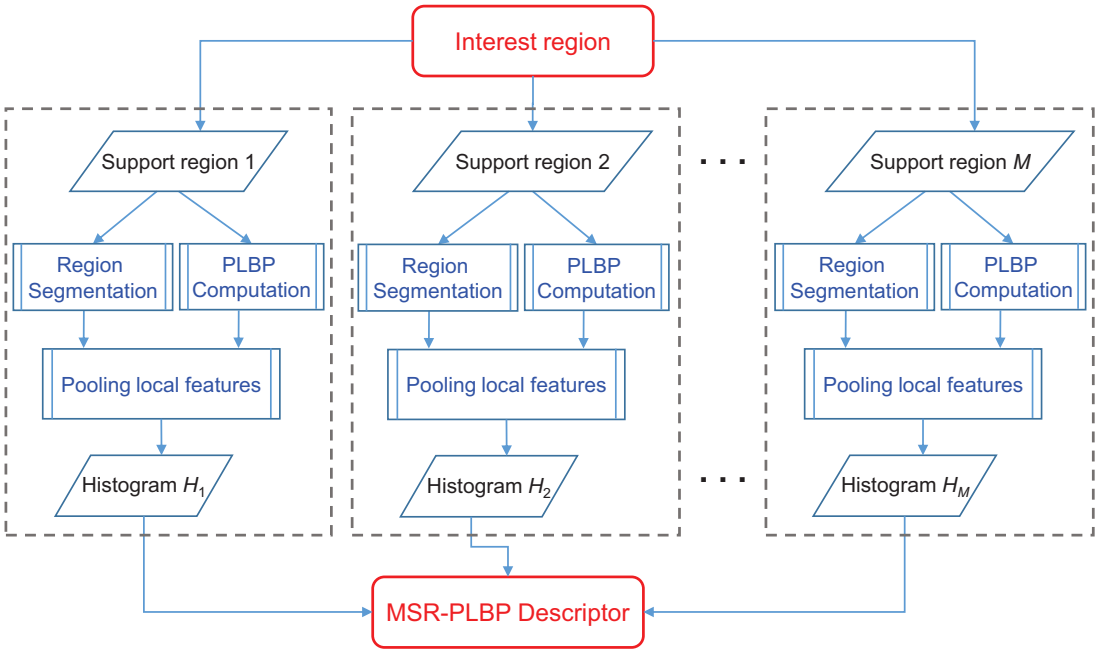


Figure 3.12: The construction pipeline of the MSR-QLBP descriptor. The Region Segmentation component sequentially performs ring-shaped and ordinal divisions on each of multiple support regions.

tive segments. PLBP patterns are computed for each pixel in a support region and pooled to corresponding segments. We then compute the PLBP histograms in every segment and concatenate them into a single feature vector, called the MSR-PLBP descriptor. These steps can be controlled easily by tuning some functional parameters, thus offering high flexibility to users. Figure 3.12 describes the overall construction pipeline.

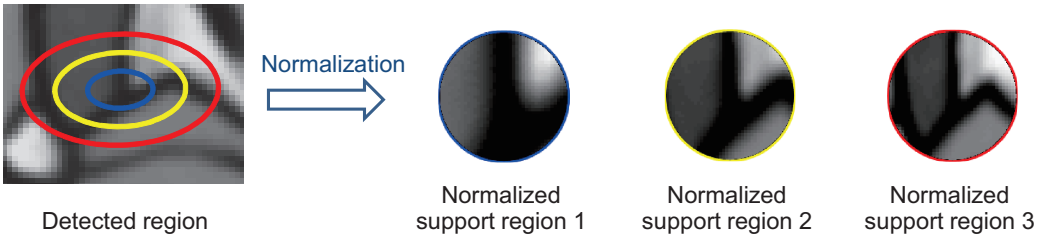
3.4.3.1 Interest region detection

We follow the same procedure described in Sect.3.3.4 for preprocessing.

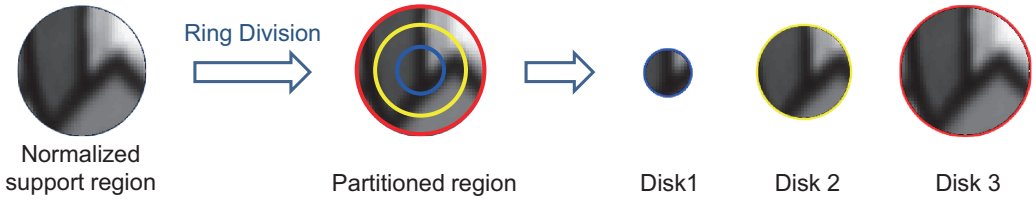
3.4.3.2 Region division

We introduce the Multi-scale region (MSR) division scheme that includes three steps: 1) define M nested support regions with equal increment of size centered at the interest point, 2) form N disks in each normalized support region, and 3) aggregate pixels in each disk into K segments according to their intensity order.

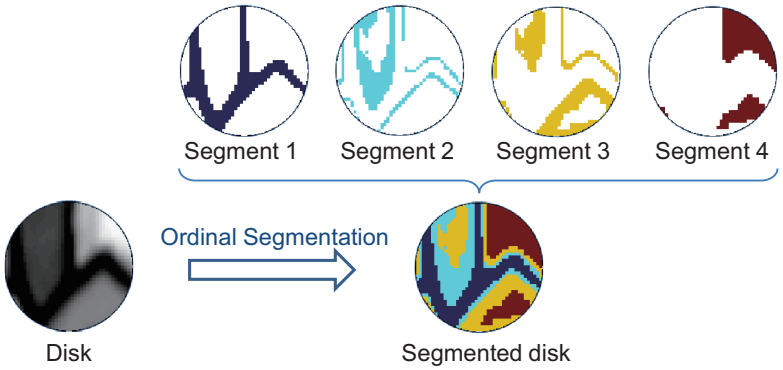
The first step adopts the concept of measurement region in [95] to define multiple nested support regions of different scales around an interest point (cf. Fig.3.13(a)). In the second step, we divide each normalized support region into N



(a) Three support regions ($M = 3$) are defined from an interest point. Their boundaries are shown in different colors.



(b) The normalized support region (e.g. the Normalized support region 3 in (a)) is spatially divided into a number of disks ($N = 3$). The disk boundaries are shown in different colors.



(c) Each disk (e.g. the Disk 3 in (b)) is further broken down into many segments according to the relative order of pixel intensities ($K = 4$).

Figure 3.13: The multi-scale region division scheme.

concentric rings of equal width and form N disks in a cumulative manner. The 1^{st} disk covers the innermost ring, the 2^{nd} disk includes the 1^{st} disk and the second innermost ring, and the expansion continues until the N^{th} disk occupies the whole region (see Fig.3.13(b)). The proposed scheme and RIFT [67] share the idea of concentric rings; however, the RIFT treats the rings separately while ours links them together. The final step involves sorting pixels within each disk by intensity and equally quantizing them into K segments based on their intensity order (see Fig.3.13(c)). It resembles the intensity order-based region division described in [40, 148] but the pixel aggregation is done on segments of a disk rather than of a support region. It is worth noting that the MROGH and MRRID [40] only

perform the first and third steps of our MSR scheme while we allow the support region to be further divided into several disks.

The MSR scheme provides several benefits to a descriptor. First, it supports rotation invariance without any dominant orientation estimation because all steps are theoretically invariant to rotation. Second, it compensates the lack of spatial information in intensity-based division methods by incorporating ring-shaped segmentation. Third, the use of intensity order enables more robustness to monotonic illumination changes. Finally, multiple support regions efficiently reduce incorrect matches thanks to the rich information collected from different scales. Although some techniques have been adopted individually in other methods, their cooperation as a single framework is unique and successfully improves the performance accuracy. In addition, the MSR scheme allows users to control the fineness of region division by tuning parameters, M, N and K, producing a descriptor that well reflects their preference between accuracy and computation cost.

3.4.3.3 Feature construction

We adopt Algorithm 1 to compute a set of PLBP pattern candidates for each pixel x in a normalized support region. To maintain the rotation invariance, the first neighbor of x is located along the radial direction from the region center to x so that x lies between the center and the neighbor, then three other neighbors are sampled anticlockwise according to the PLBP structure.

Because the Gaussian noise is likely to alter the order of pixels whose gray values are close to each other, some pattern candidates will be more reliable than the others if they contain large gray-level differences only. Therefore, each pattern is characterized by a weight that counts the number of pixel pairs satisfying the condition $|g_c - g_p| > T$ (see Eq.3.13). The threshold T is estimated independently for each disk based on the Michelson contrast [90]:

$$T = g\left(\frac{I_{max} - I_{min}}{I_{max} + I_{min}}\right) \tag{3.14}$$

where I_{min} and I_{max} represent the lowest and highest intensities respectively, and $g(z)$ is a Gaussian function. The contrast value ranges from 0 to 1. When the brightness contrast is low, we set T high to select $|g_c - g_p|$ that are truly unaffected by noise; otherwise, T is kept to be small.

We use the MSR division scheme to divide a normalized interest region into $M \times N \times K$ segments and compute the histograms separately in every segments.

Depending on which segment a pixel belongs to, its PLBP patterns will contribute to the corresponding histogram. The MSR-PLBP descriptor is finally constructed by concatenating histograms across segments:

$$\begin{aligned} MSR-PLBP &= \bigoplus_{\substack{i=1,\ldots,M,\ j=1,\ldots,N \\ k=1,\ldots,K}} H_k^{i,j} \\ H_k^{i,j} &= [h_{k_0}, h_{k_1}, \ldots, h_{k_{15}}] \\ h_{k_j} &= \sum_{x \in S_k,\, \phi(PLBP(x))=j} w(PLBP(x)) \end{aligned} \tag{3.15}$$

where $H_k^{i,j}$ is the histogram in the k^{th} segment of the j^{th} disk of the i^{th} support region, $\bigoplus$ denotes the concatenation of histograms, the function ϕ maps a PLBP pattern to its corresponding histogram bin index and S_k is the k^{th} segment. $H_k^{i,j}$ has $2^4 = 16$ entries, which are the number of distinct codes. Note that the term $PLBP(x)$ in this context refers to each pattern candidates of the pixel x. The MSR-PLBP requires no interpolation because it is not grid based and the PLBP feature is quantized by its nature. This offers a great advantage in computational efficiency compared to that of SIFT with heavy tri-linear interpolation.

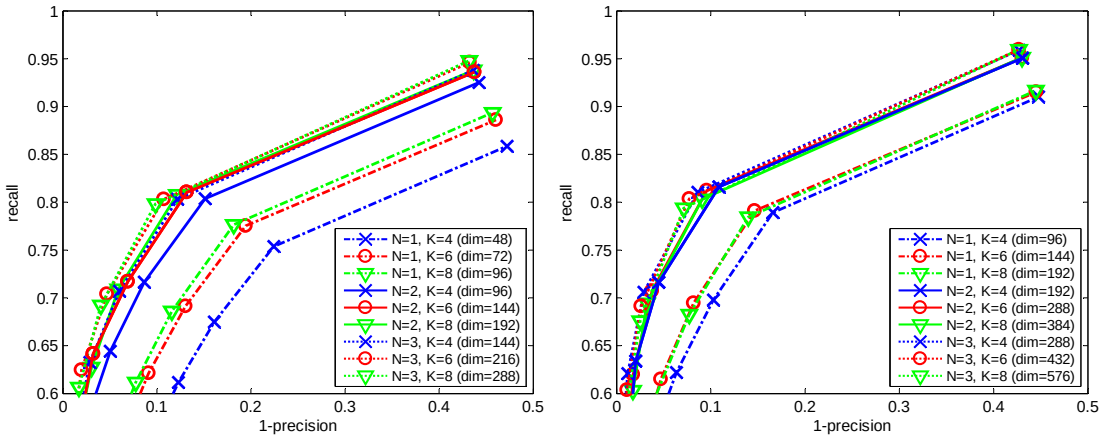
3.5 Parameter tuning for proposed descriptors

We examine the effects of different parameter settings to the performance of the proposed descriptors. Experiments are conducted on 140 image pairs downloaded from the web page of Mikolajczyk K. [3]. These images contain mainly zoom and rotation transformations. They are completely separate from the data for other evaluations in order to to avoid bias. The performance is depicted by the *average recall* versus *average 1-precision* curves (cf. Sect.3.6.1). For each descriptor, the parameter setting that yields good performance is selected as default setting. Changes (if available) will be noted specifically.

3.5.1 The IO-QLBP descriptor

The parameters of the IO-QLBP (or IO-QLBP^{sym}) includes: σ_a and σ_b - Gaussian smoothing values, R - the radius of the neighborhood, τ - the thresholding factor, K - the number of segments, and N - the number of support regions.

We first focus on the joint effect of the key parameters, N and K. According to Fig.3.14, IO-QLBP and IO-QLBP^{sym} perform better when N and/or K increases. $K = 6$ and $K = 8$ are comparable to each other and both much better than



(a) IO-QLBP (b) IO-QLBP^{sym}

Figure 3.14: The performances of IO-QLBP and IO-QLBP^{sym} under different parameter settings.

Figure 3.15: The performances of IO-QLBP and IO-QLBP^{sym} under different weighting schemes.

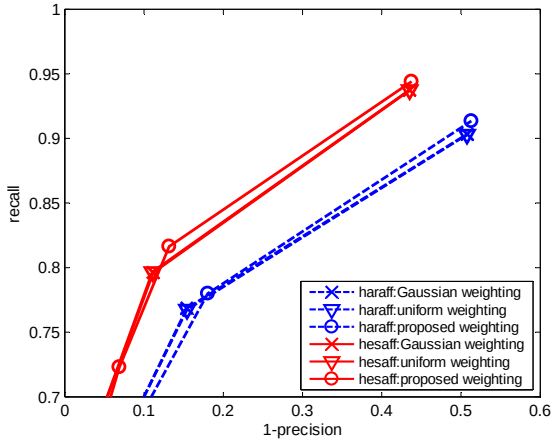


Table 3.1: The default parameter settings for IO-QLBP and IO-QLBP^{sym}

Parameter	Dimensions	No. patterns	σ_a	σ_b	K	N	R	τ
IO-QLBP	144	12	1.0	1.2	6	2	4	0.01
IO-QLBP^{sym}	192	24	1.0	1.2	4	2	4	0.01

$K = 4$. $N = 2$ outdistances $N = 1$, confirming the advantage of multiple support regions. However, the improvement becomes minor when N increases to 3 while the computational cost and dimensionality expand dramatically. To keep an optimal trade-off between accuracy and computation cost, we select ($N = 2$, $K = 6$) for IO-QLBP and ($N = 2$, $K = 4$) for IO-QLBP^{sym}. Since the IO-QLBP^{sym} is intrinsically stronger than the IO-QLBP, it does not require the region to be finely segmented. Meanwhile, other trivial parameters are empirically

Table 3.2: The default parameter settings for MSR-PLBP^s and MSR-PLBP^m.

Descriptor	Dimensions	MSR scheme			R	T		σ_a	σ_b
		M	N	K		a	c		
MSR-PLBP^s	192	1	2	6	5	0.02	0.1	1.0	1.2
MSR-PLBP^m	192	3	1	4					

selected. Table 3.1 shows default parameter settings for two descriptors.

The influences of different weighting schemes are also investigated. Figure 3.15 shows that, compared to Gaussian and uniform schemes, ours (cf. Eq.3.8 and Eq.3.9) is best suited for IO-QLBP descriptors.

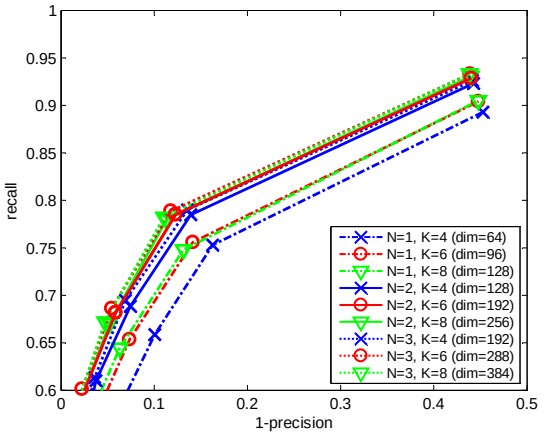
3.5.2 The MSR-PLBP descriptor

We focus on the radius R of the PLBP and three parameters, M, N and K, of the MSR scheme, while other parameters are empirically tuned. The analysis enables us to design the MSR-PLBP^s and MSR-PLBP^m, which are the single-region and multi-region variants, respectively. Their parameter settings are described in Table 3.2. These variants will be evaluated in Sect.3.6 and Sect.3.7 to verify the effectiveness of the proposed method.

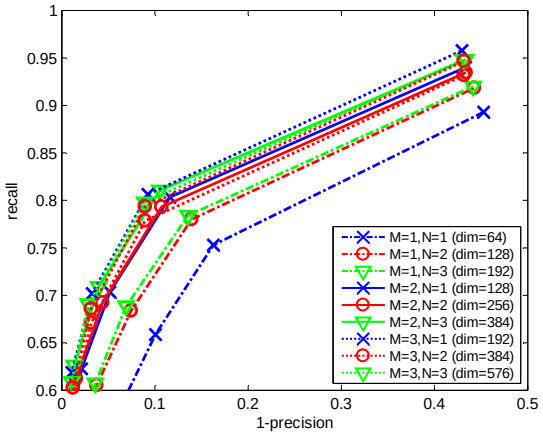
3.5.2.1 MSR-scheme parameters: M, N, and K

The joint effect of three parameters, including M - the number of support regions, N - the number of disks, and K - the number of ordinal segments, is examined. Figure 3.16(a) shows how K and N influence the performance when M is fixed to 1. The performance significantly improves when K increases from 4 to 6, but slightly declines when K increases from 6 to 8, which indicates the over-segmentation. Considerable improvements are observed with $N > 1$, confirming the effectiveness of ring-shaped division. However, because the normalized region is small, fragments may occur if N continues to increase. $N = 3$ yields a very minor improvement compared to that of $N = 2$. Therefore, for a single-region descriptor, $K = 6$ and $N = 2$ are recommended.

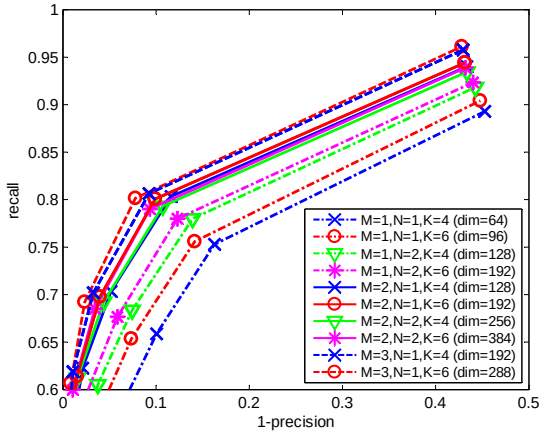
Figure 3.16(b) shows the joint effect of M and N when K is fixed. Similarly to [40], considerable improvements are observed when $M > 1$, which confirms the advantage of multiple support regions. Although N also boosts the performance, combining M and N does not give a double benefit. $(M = 2, N = 2)$ are very close to $(M = 3, N = 2)$, as well as $(M = 2, N = 3)$ to $(M = 3, N = 3)$. In addition,



(a) N vs. K ($M = 1$)



(b) M vs. N ($K = 4$)



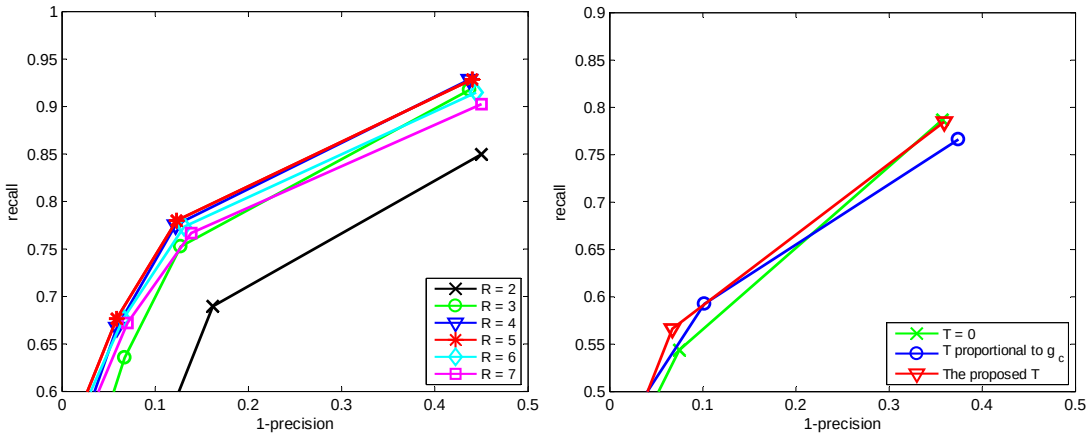
(c) Feasible (M, N, K) combinations

Figure 3.16: The performance of the MSR-PLBP descriptor with different (M, N, K) combinations.

if M is large, for example $M = 3$, increasing N tends to lower the performance. Therefore, $(M = 3, N = 1)$ is recommended for a multi-region descriptor since it yields the highest performance among different settings. Finally, we summarize some feasible settings in terms of performance and dimensionality so that users can decide which setting best fits their need (cf. Fig.3.16(c)). If an application prefers preciseness to compactness, one should choose $(M = 3, N = 1, K = 6)$. $(M = 1, N = 1)$ provides the most compact feature vector, hence is the fastest. Other settings balance the two factors to different extents.

3.5.2.2 The radius R of the PLBP

Different values of R were tested, as shown in Fig.3.17(a). The performance improves as R increases from 2 to 5; however, it declines after that because the



(a) The radius R (b) The threshold scheme

Figure 3.17: The performance of the MSR-PLBP descriptor with different R and thresholding schemes.

local neighborhood becomes too large, which loosens the relationship between pixels. In this study, we choose $R = 5$.

3.5.2.3 The Gaussian filters: σ_a and σ_b

The values of σ_a and σ_b should lie between 1 and 1.5 to avoid over-blurring. We simply followed [148] to choose $\sigma_a = 1.0$ and $\sigma_b = 1.2$.

3.5.2.4 The threshold T in Eq.3.14

The values $a = 0.02$ and $c = 0.1$ are empirically selected for the Gaussian function $g = ae^{(-x^2/(2c^2))}$, with the assumption that pixel intensities are normalized in the closed interval [0,1]. We evaluated the performance in three cases: $T = 0$ [105], T proportional to the center pixel g_c [74], and T computed from Eq.3.14. Figure 3.17(b) shows that the proposed T produces the best performance.

3.6 Evaluation on the image matching task

This section presents the performance analysis of the proposed descriptors, including the IO-QLBP (cf. Sect.3.3) and MSR-PLBP (cf. Sect.3.4), on the image matching task. Experiments are conducted on the Oxford benchmark and two datasets specialized in drastic illumination changes and viewpoint changes, and thus sufficing to verify the effectiveness of these descriptors to different types of image transformations.

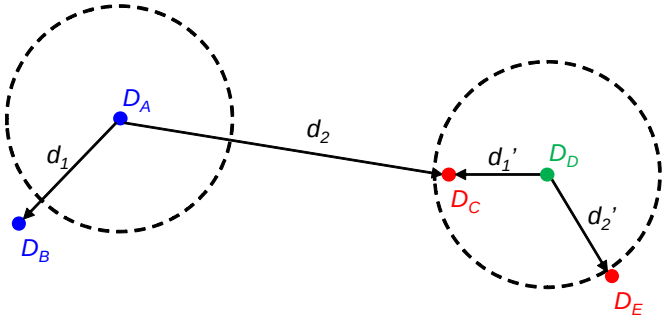


Figure 3.18: Three matching strategies. Descriptors depicting similar regions are denoted by the same color. Threshold-based matching: D_A fails to match D_B at a fix distance (dashed circles) while D_D incorrectly matches D_C and D_E. Nearest neighbor matching: D_A correctly matches D_B but D_D incorrectly matches D_C. Nearest neighbor distance ratio matching: the small d_1/d_2 correctly matches D_A and D_B and the large d_1'/d_2' correctly rejects matches for D_D. This example is obtained from [125].

3.6.1 Evaluation protocol

The evaluation criterion is adopted from [94], which is based on the number of correct matches and false matches on a pair of images. Each region A from the reference image is matched with a region B from the transformed (test) image by comparing their corresponding descriptors following one of the below matching strategies (cf. Fig.3.18).

1. Threshold-based matching: two regions are matched if the distance between their descriptors is below a threshold. A descriptor can have several matches and several of them may be correct.
2. Nearest neighbor matching: A and B are matched if the descriptor D_B is the nearest neighbor to D_A and if the distance between them is below a threshold. A descriptor has only one match.
3. Nearest neighbor distance ratio (NNDR) matching: it is similar to the second strategy except that the thresholding is applied to the distance ratio between the first and the second nearest neighbor. A and B are matched if

$$NNDR = \frac{d_1}{d_2} = \frac{\|D_A - D_B\|}{\|D_A - D_C\|} < t \tag{3.16}$$

   where D_A, D_B, and D_C are the descriptors of the target, first and second nearest neighbor, respectively, and t is a threshold.

We select the NNDR matching strategy because it additionally penalizes descriptors that have many similar matches.

The result is presented in terms of *recall* versus *1-precision*:

$$recall = \frac{\#\, correct\, matches}{\#\, correspondences} \tag{3.17}$$

$$1 - precision = \frac{\#\, false\, matches}{\#\, correct\, matches + \#\, false\, matches} \tag{3.18}$$

where $\#\, correspondences$ stands for the ground truth number of matched regions. The curves are obtained by varying the distance threshold. A good descriptor should have a recall approaching 1 at any precision.

The number of correct matches and correspondences are determined by the overlap error ϵ [95], which measures how well the regions correspond under a transformation, i.e. a homography in this case.

$$\epsilon = \frac{1 - (A \cap H^T B H)}{A \cup H^T B H} \tag{3.19}$$

where H is the homography matrix between the images (cf. Sect.3.2.3). We assume a correct match if $\epsilon < 0.5$.

Interest regions are given by the Harris-Affine (haraff) or Hessian-Affine (hesaff) detector then normalized into circular shapes of a fixed diameter (41 pixels).

3.6.2 List of evaluated descriptors

We evaluate the effectiveness of the two proposed approaches through four descriptors: IO-QLBP, IO-QLBP^{sym}, MSR-PLBP^{s} and MSR-PLBP^{m}. The first two descriptors use QLBP and QLBP^{sym} operators, respectively. The MSR-PLBP^{s} demonstrates the MSR-PLBP approach with a single support region while the MSR-PLBP^{m} for multiple support regions.

Six descriptors that are closely related to the proposed approaches are selected for comparison: SIFT [81], DAISY [135], HRI-CSLTP [49], LIOP [148], MROGH and MRRID [40]. SIFT, DAISY, and MROGH are gradient based, while the HRI-CSLTP, LIOP and MRRID are intensity based. MRROGH and MRRID use multiple support regions. Their details are summarized in Sect.3.2.

Evaluating all descriptors from a single perspective may be biased since multi-region methods are based on larger amounts of information than that of single-region methods. Therefore, we split the set of evaluated descriptors into two parts. The multi-region group includes MROGH, MRRID, IO-QLBP, IO-QLBP^{sym} and

Table 3.3: The dimensions and average construction time of evaluated descriptors.

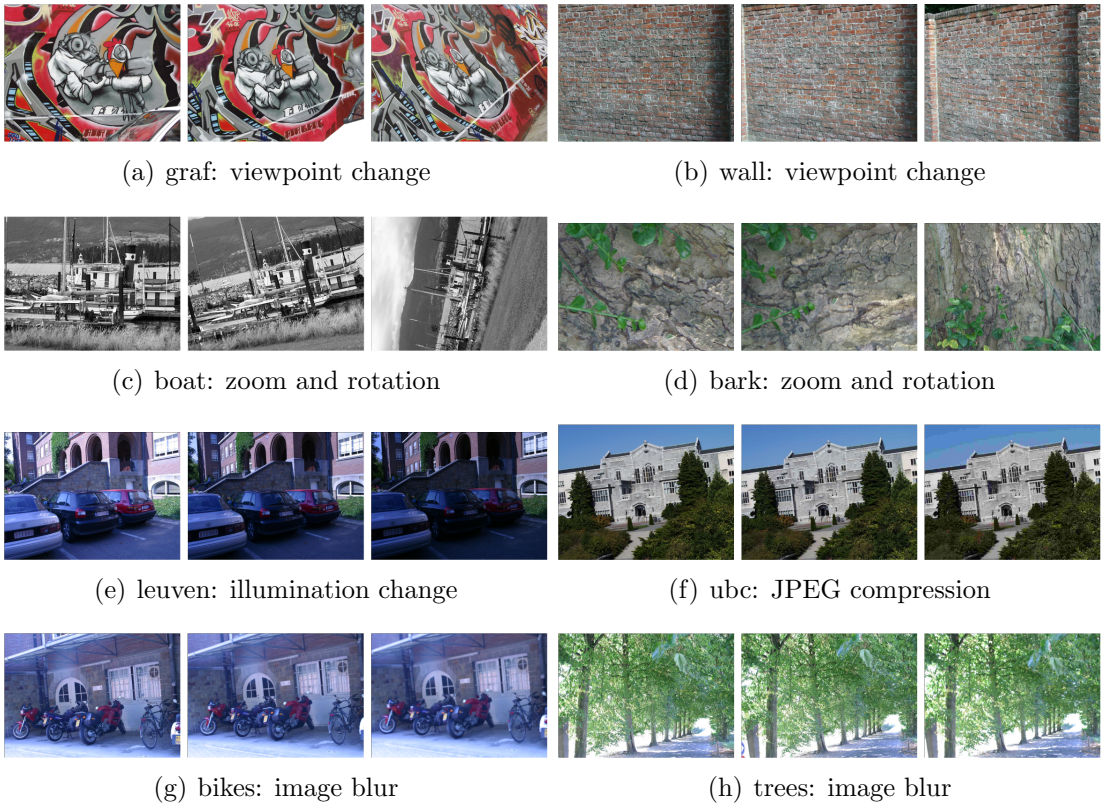
	Method	Dimensions	Construction time (millisecs)
Single-region group	**MSR-PLBP^s**	192	**1.03**
	LIOP	144	1.75
	SIFT	128	2.13
	DAISY	136	2.52
Multi-region group	**IO-QLBP**	144	**1.80**
	IO-QLBP^{sym}	192	**1.88**
	MSR-PLBP^m	192	**2.73**
	HRI-CSLTP	384	3.16
	MROGH	192	4.60
	MRRID	256	9.42

MSR-PLBP^m, while the rest descriptors belong to the single-region group. Binaries files are downloaded from [2, 4, 5] and run on an Intel Core i7 950 CPU 2.8 GHz PC. Table 3.3 shows the dimensions and average construction time of all descriptors. Our descriptors are the fastest in each group.

3.6.3 Matching results on the Oxford benchmark

The Oxford benchmark [94] includes images with six different geometric and photometric transformations, i.e. rotation, scale change, viewpoint change, image blur, JPEG compression and illumination change (cf. Fig.3.19). In the first four transformations, the structured scene and textured scene are designed, allowing the analyses of image transformations and scene types to be done separately. The former contains homogeneous regions with distinctive edge boundaries (e.g. graffiti and buildings) while the latter depicts repeated textures of various forms. There are eight image sets, each of which has six images demonstrating different levels of transformations. The images resolution is about 800×640. This benchmark has been widely used in many comparative studies [40, 49, 55, 94, 148].

We report the performances on the $1^{st} - 2^{nd}$ and $1^{st} - 4^{th}$ image pairs, which demonstrate small and large image transformations, respectively. Results on the $1^{st} - 4^{th}$ pairs are presented in Fig.3.20 to Fig.3.22, while those of the $1^{st} - 2^{nd}$ pairs are shown in Fig.A.2 to Fig.A.1 - Appendix A. The descriptors generally achieve high performance in easy $1^{st} - 2^{nd}$ cases, while greatly diverging from one another in challenging $1^{st} - 4^{th}$ cases. In addition, the selection of detector has little effect on the ranking order, which is similar to the observation in [94].



(a) graf: viewpoint change (b) wall: viewpoint change

(c) boat: zoom and rotation (d) bark: zoom and rotation

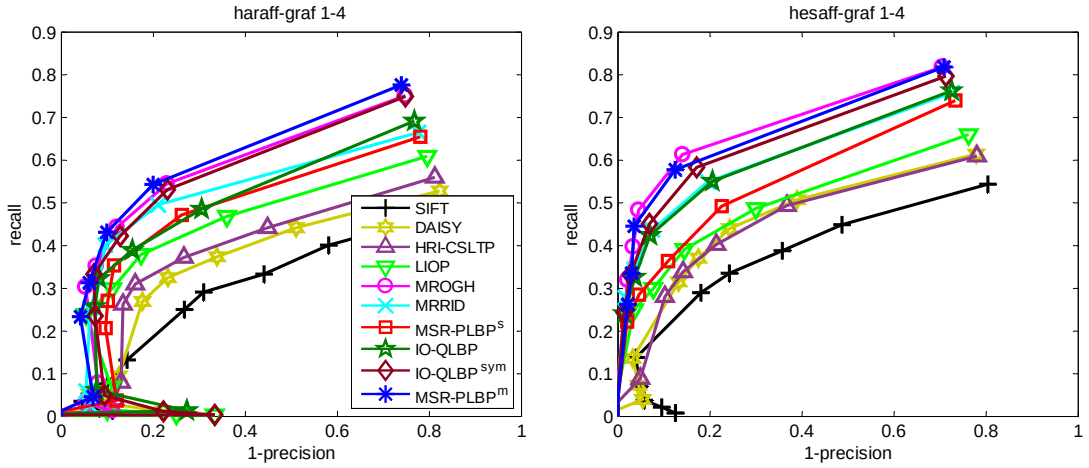
(e) leuven: illumination change (f) ubc: JPEG compression

(g) bikes: image blur (h) trees: image blur

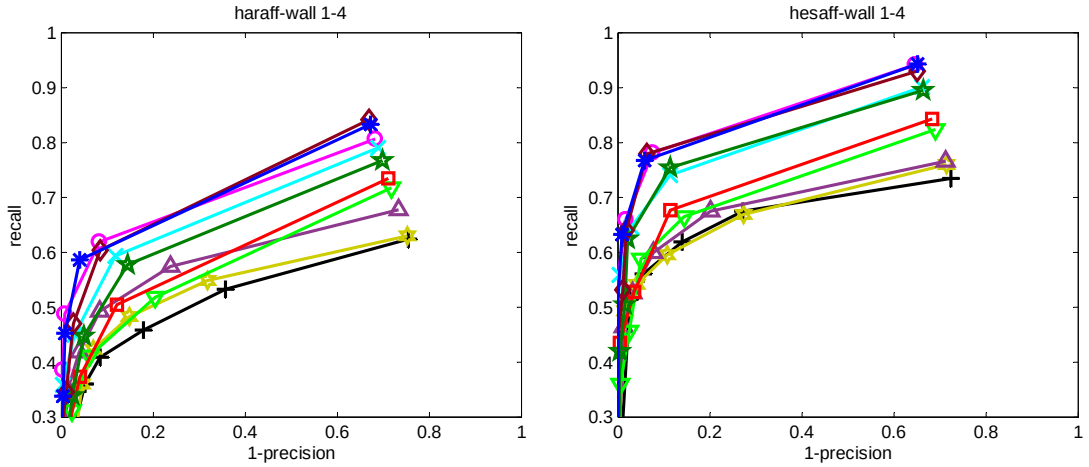
Figure 3.19: Sample images from the Oxford benchmark [94]. In each set, the first (reference image), second and fourth images are shown from left to right.

In the single-region group, the MSR-PLBP^s is leading in 28/32 cases. Its highest difference to the next place is 13% in *trees*. The LIOP follows our descriptor closely, with a difference of around 0.5-6%, but experiences several deteriorations in *boat 1-4* and *trees*. This is because the combination of ring-shaped and intensity order-based divisions is more discriminative than each individual division. The performance of all descriptors declines in *trees* because of the heavy blur; however, the MSR-PLBP^s still keeps a noticeably separation from other methods. This supports the superiority of PLBP over the gradient and intensity permutation in near-uniform image regions.

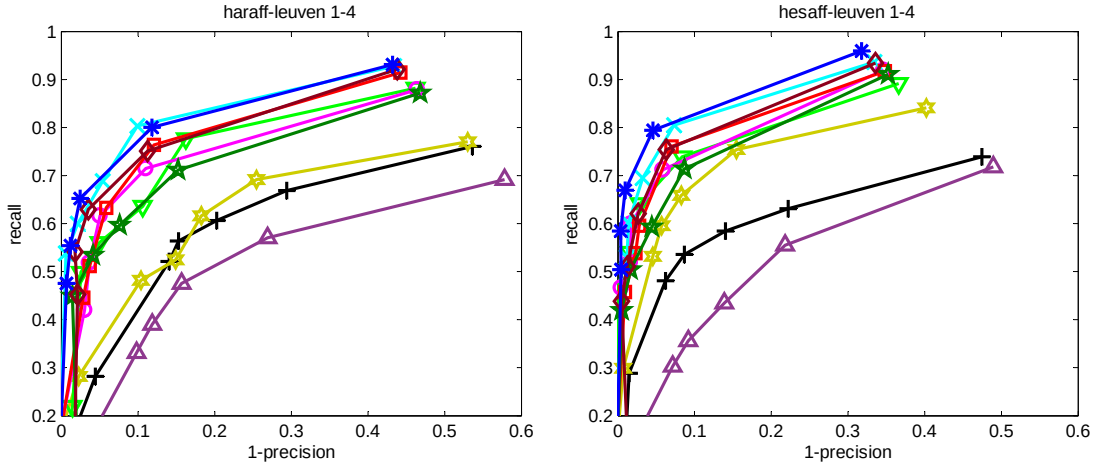
The MSR-PLBP^m, MROGH and MRRID are leading in most challenges thanks to multiple support regions. The differences of the MSR-PLBP^m to the closest and second closest competitors are around 0-2% and 1-5% respectively. Due to their core features, the MROGH performs well in geometric transformations, while the MRRID in photometric transformations. The MSR-PLBP^m is comparable with the MROGH and MRRID in both types of transformations (except *bark*),hence achieving better stability. The advantage of the MSR-PLBP^m



(a) graf: viewpoint change

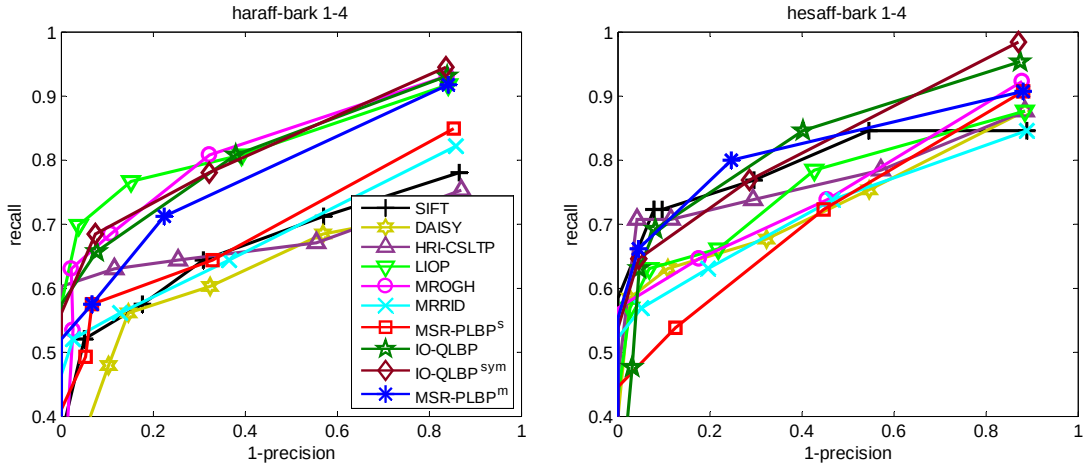


(b) wall: viewpoint change

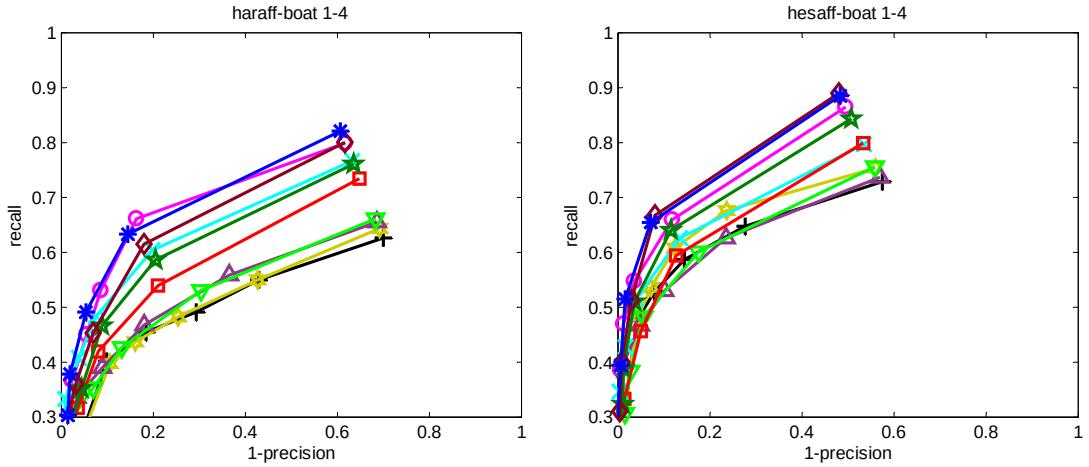


(c) leuven: illumination change

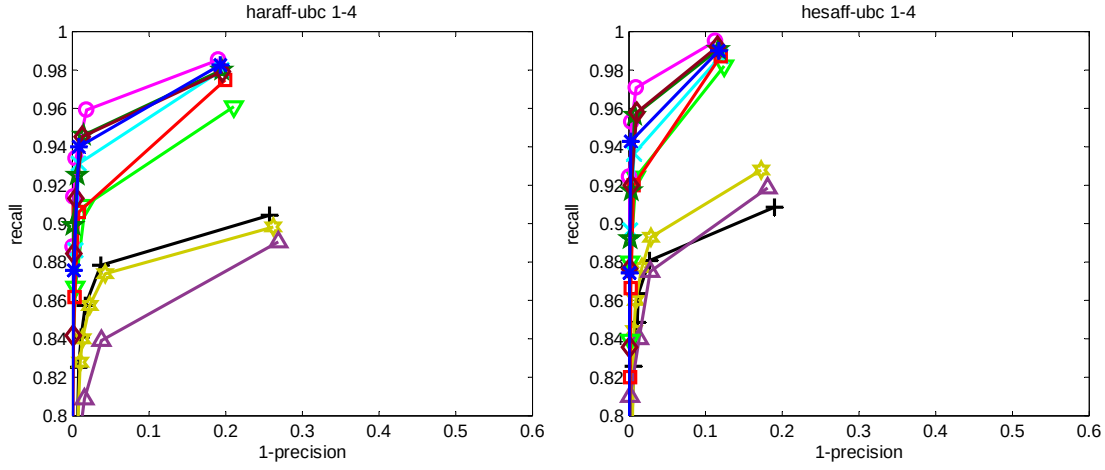
Figure 3.20: The performance of evaluated descriptors on the Oxford benchmark (Part 1/3). The scales are different through figures for better clarifying the plots.



(a) bark: zoom and rotation

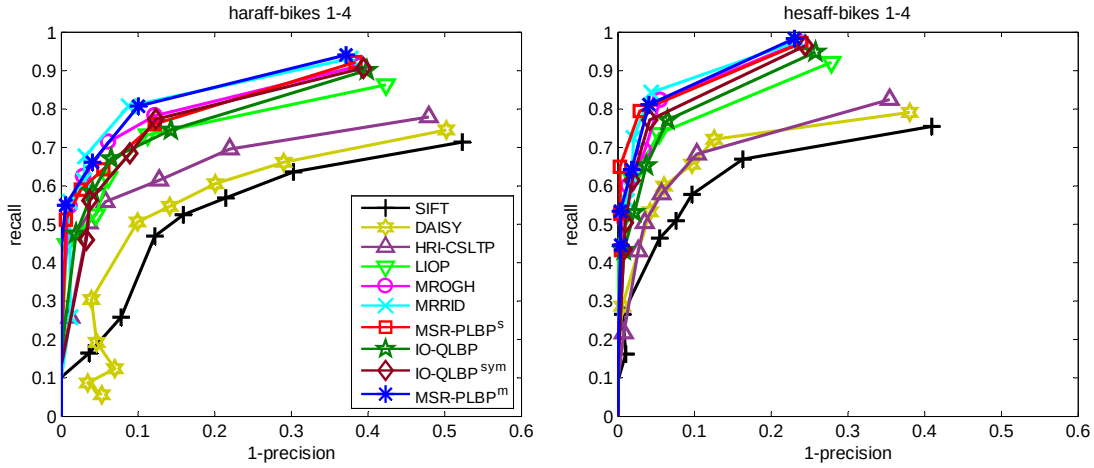


(b) boat: zoom and rotation

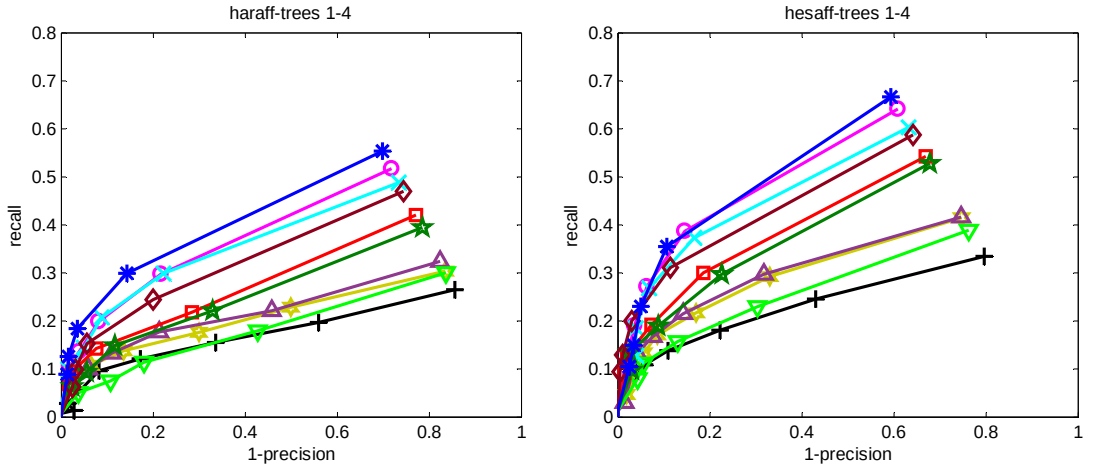


(c) ubc: JPEG compression

Figure 3.21: The performance of evaluated descriptors on the Oxford benchmark (Part 2/3). The scales are different through figures for better clarifying the plots.



(a) bikes: image blur



(b) trees: image blur

Figure 3.22: The performance of evaluated descriptors on the Oxford benchmark (Part 3/3). The scales are different through figures for better clarifying the plots.

is two-fold: 1) fewer support regions is required, thus faster, and 2) the PLBP improves good properties of the LBP [105] to be more discriminative and robust to monotonic illumination changes. It is worse than MROGH or MRRID in a few cases, such as *hesaff-bikes 1-4* and *hesaff-ubc*, but the differences are minor. The IO-QLBP^{sym} is comparable to the MROGH in most cases, except a loss of 5.4% in *trees*. The *trees* contains textures that are very similar to each other due to heavy blur, and thus descriptors using more support regions have an obvious advantage. Note that the MSR-PLBP^{m} performs better than the MROGH in this case despite its smaller number of regions. The IO-QLBP^{sym} outperforms the MRRID in geometric transformations, yet it is less robust to photometric transformations. This is because the IO-QLBP^{sym} does not consider supplementary information

(a) corridor (b) desktop

(c) desktop-square (d) desktop-square root

Figure 3.23: Pairs of reference - test images from the Illumination dataset [148].

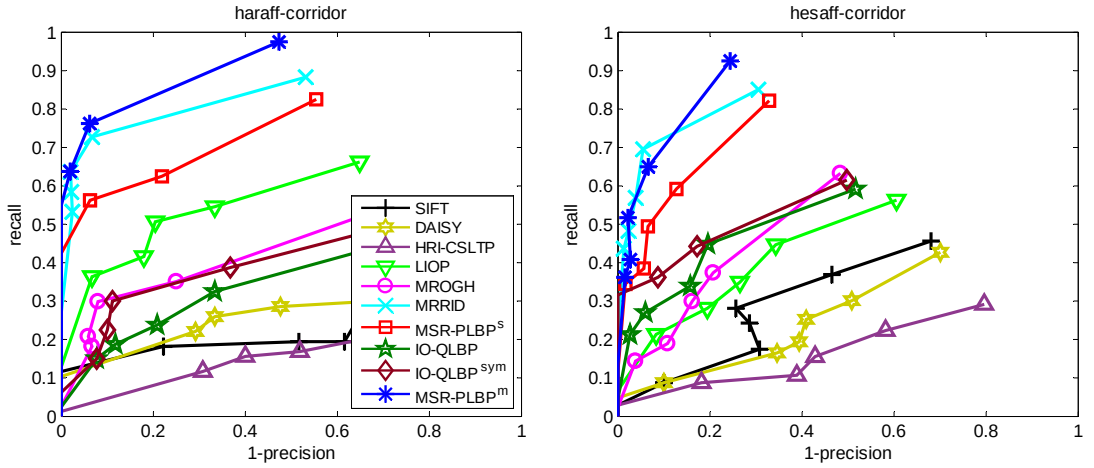
of contrast as in the MSR-PLBP^m. The IO-QLBP performs worst among multi-region descriptors but still overcomes the single-region methods.

In *bark*, the complex surface of the bark makes textures seem to be randomly arranged and thus the matching is very difficult. The IO-QLBP^{sym} and IO-QLBP surprisingly occupy the first places and the MSR-PLBP^m maintain the fifth place, after the MROGH and LIOP. Meanwhile, the ranking of other descriptors vary significantly. Therefore, it is hard to draw any strong conclusion from this challenge. In general, the proposed methods have shown to be efficient and stable in more challenges than that of other descriptors. In addition, they achieve high recalls from the outset, thus are promising for problems like image retrieval.

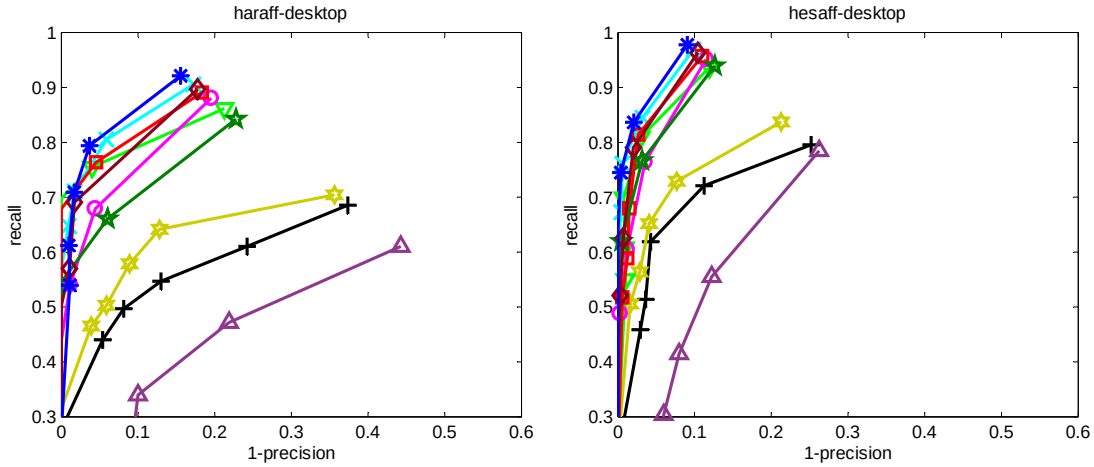
3.6.4 Matching results on the Illumination dataset

The Illumination dataset [148] is created primarily for examining the effectiveness of descriptors under drastic illumination changes. It contains two sets of images, *corridor* and *desktop*. Nonlinear transformations, including square root and square operations, were performed on the second image of *desktop* to synthesize images with monotonic intensity changes. All images are in the resolution of 1504×1000. Figure 3.23 shows pairs of images used in the evaluation.

The performances of evaluated descriptors are illustrated in Fig.3.24 and Fig.3.25. The MSR-PLBP^m performs best in all cases. The MRRID follows the MSR-PLBP^m closely with a difference of less than 3%. The performance of MSR-PLBP^s is lower than that of MRRID, around 0 - 3%) in *desktop* cases and 10% in *corridor*, thus it ranks third. However, it is still much better than



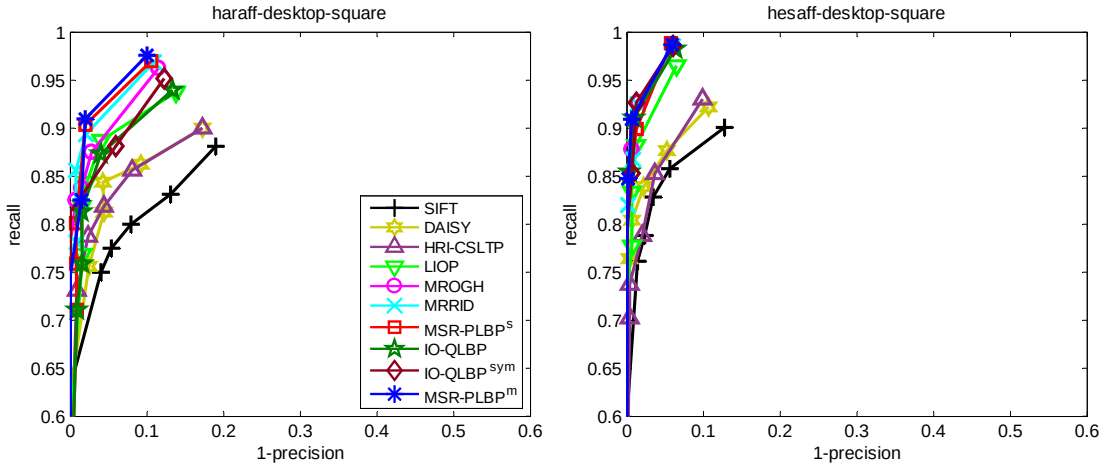
(a) corridor



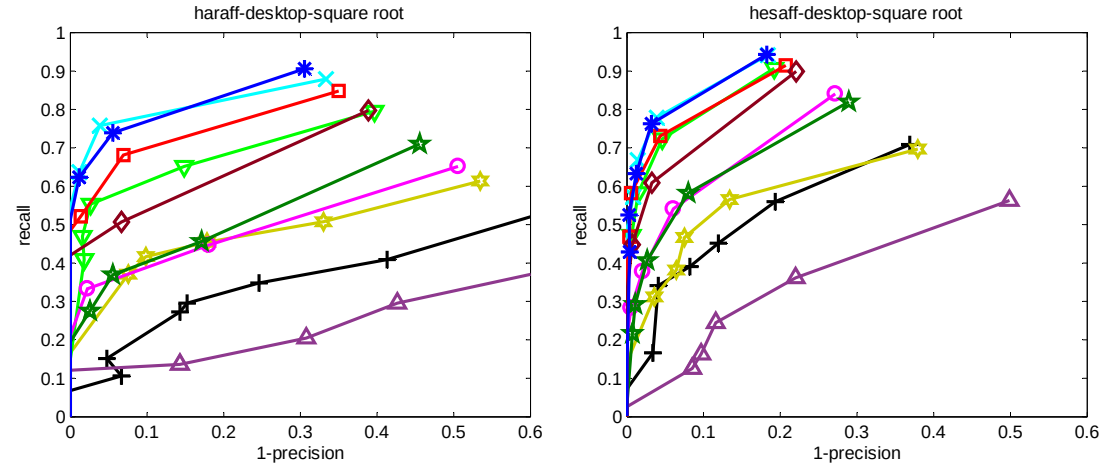
(b) desktop

Figure 3.24: The performance of evaluated descriptors on the Illumination dataset (Part 1/2). The scales are different through figures for better clarifying the plots.

the MROGH and LIOP. The IO-QLBP^{sym} performs comparably to the MROGH in most cases and its performance in *hesaff-desktop-square root* is even higher than that of MROGH 10%. The IO-QLBP ranks right after MROGH while performs much better than the SIFT, DAISY, and HRI-CSLTP. Since this dataset demonstrates different types of illumination changes, it best suits intensity order-based and texture-based descriptors, whereas gradient-based descriptors have few opportunities to exhibit their effectiveness. We indeed observed severe declines of MROGH in 4/8 cases. The LIOP achieve a moderate performance in *corridor*, which may indicate that its intensity order-based structure cannot adapt to drastic illumination changes well. The HRI-CSLTP fails in 6/8 cases because its CS-LTP feature uses only four neighbors and a fixed threshold, thus losing a



(a) desktop (square)



(b) desktop (square root)

Figure 3.25: The performance of evaluated descriptors on the Illumination dataset (Part 2/2). The scales are different through figures for better clarifying the plots.

large amount of information. In conclusion, the MSR-PLBP approach is capable of handling complex illumination changes in the sense that their variants perform efficiently and are leading in their groups. The IO-QLBP approach is not as good as the MSR-PLBP since its encodings characterize gradient changes rather than illumination changes. Nevertheless, the IO-QLBP^{sym} is better than the MROGH in terms of both matching accuracy and speed.

3.6.5 Matching results on the Viewpoint change dataset

We further evaluate the descriptors on the Viewpoint change dataset collected by [28]. The dataset consists five subsets, each of which includes six images

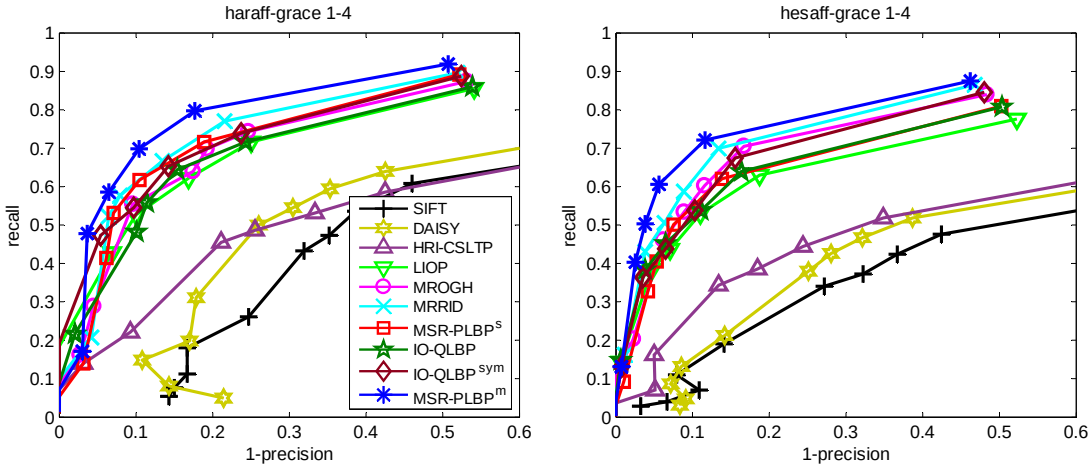
(a) grace: viewpoint change

(b) there: strong viewpoint change

(c) posters: viewpoint change + texture repetition

(d) underground: viewpoint change + pattern repetition

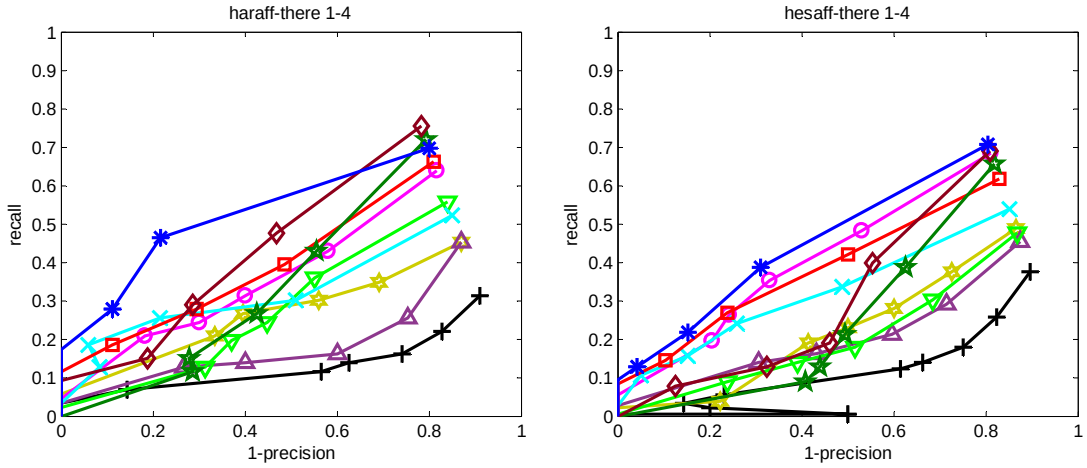
Figure 3.26: Sample images from the Viewpoint change dataset [28]. In each sequence, the first (reference image), second and fourth images are shown from left to right.



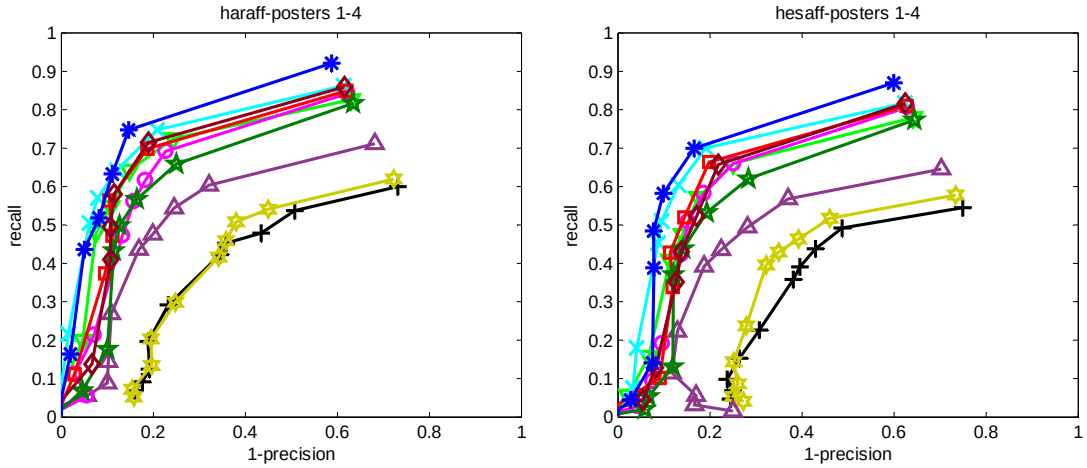
(a) grace

Figure 3.27: The performance of evaluated descriptors on the Viewpoint change dataset (Part 1/2). The scales are different through figures for better clarifying the plots.

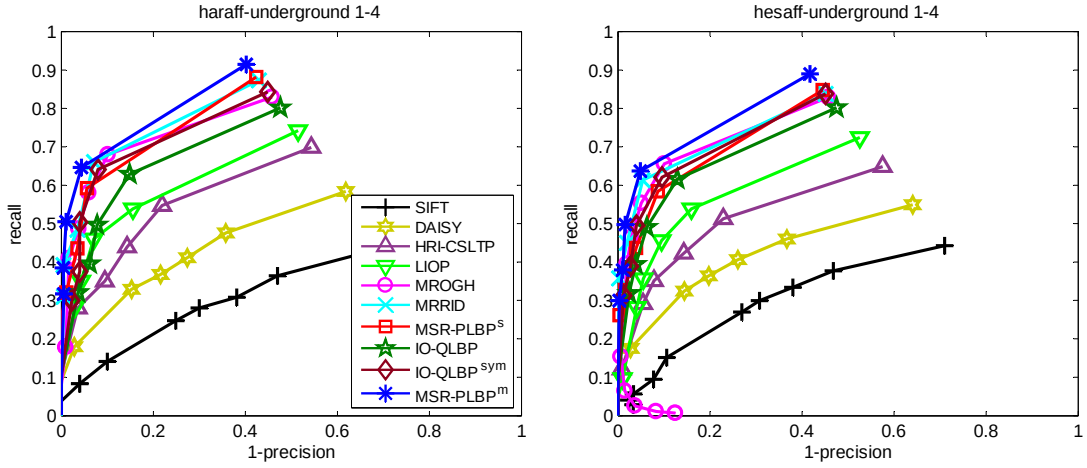
with different levels of difficulties. This resembles the structure of the Oxford benchmark [94]. All images are in the resolution 1536×1024, demonstrating viewpoint change, pattern repetitions or texture repetitions. We select four over five subsets for this experiment. Figure 3.26 shows some image samples from the dataset.



(a) there



(b) posters



(c) underground

Figure 3.28: The performance of evaluated descriptors on the Viewpoint change dataset (Part 1/2). The scales are different through figures for better clarifying the plots.

Similarly to the experiment in Sect.3.6.3, the results on the $1^{st} - 2^{nd}$ and $1^{st} - 4^{th}$ image pairs, are reported in Fig.3.27-3.28 and Fig.A.4-A.5 - Appendix A. The MSR-PLBP^m is comparable to or better than MROGH and MRRID in all cases, with differences of around 0-5%. The performances of MRRID and MROGH suddenly decrease in *haraff-there 1-4*, in which their largest losses to the MSR-PLBP^m at the *1-precision* of 20% are over 15%. The IO-QLBP^{sym} performs comparably to the MROGH in 14/16 cases, yet suffering a severe deterioration in *there-1-4*. We note that the MSR-PLBP^s performs better than the IO-QLBP several times, though it is single-region based. The advantage of the MSR-PLBP^s may lie in its pattern generation scheme that addresses the issue of noisy regions and the MSR scheme that improves the distinctiveness of a region.

3.7 Evaluation on the object recognition task

This section presents the performance analysis of the proposed descriptors on the object recognition task. Experiments are conducted on 266 images, 4000 images and 10,200 images of the 53 Objects and the Recognition Benchmark databases, demonstrating the behaviors of descriptors to different data scales. Our approaches not only perform effectively and consistently on these databases but also possess a great advantage in terms of speed compared to other methods.

3.7.1 Evaluation protocol

The performance of each descriptor is evaluated according to its recognition rate on a database. The recognition of objects goes as follows. First, in the preprocessing phase, we extract interest regions for all images using the Hessian-Affine detector, normalize them into circular shapes of a fixed diameter (41 pixels), and compute feature vectors on the normalized regions. Then, during the online recognition phase, each image in the database in turn plays the role of a query, its similarity scores to the remaining images are estimated and M images having highest scores are returned. Finally, the overall recognition rate is defined as:

$$Recognition\,rate = \frac{1}{N}\sum_{i=1}^{N}\frac{\#\,correctly\,returned\,images\,for\,query\,i^{th}}{\#\,returned\,images} \tag{3.20}$$

where N is the total number of images in the database.

In order to compute the similarity score between two images, A and B, we

adopt the Sørensen-Dice coefficient:

$$sim(A, B) = \frac{2|F^A \cap F^B|}{|F^A| + |F^B|} \tag{3.21}$$

where $F^A = \{f_1^A, f_2^A, ..., f_m^A\}$ is the set of features computed from m interest regions in the image A and $F^B = \{f_1^B, f_2^B, ..., f_n^B\}$ is for n interest regions in the image B ($m < n$). The denominator is the total number of features in two images. The numerator is the number of matched pairs (f_i^A, f_j^B) with the assumption that a f_i^A has only one best match f_j^B and their Euclidean distance $d(f_i^A, f_j^B) < T$. Because the feature spaces are different from each other, we simply tune T separately for each descriptor to achieve the best result, as done similarly in [40]. In this way, images showing the same of object as in the query will have a higher similarity score than those showing different objects. This similarity measure purely relies on the distinctiveness of a descriptor, thus providing a preliminary evaluation of descriptors on object recognition in advance of their integration into a more sophisticated system.

3.7.2 List of evaluated descriptors

Similarly to the previous study, there are ten descriptors participating the evaluation, including our four descriptors, i.e. IO-QLBP, IO-QLBP^{sym}, MSR-PLBP^s and MSR-PLBP^m, and six closely-related descriptors, i.e. SIFT [81], DAISY [135], LIOP [148], HRI-CSLTP [49], MROGH and MRRID [40].

3.7.3 Recognition results on the 53 Objects database

The 53 Objects database includes 265 images of 53 objects, in which each object is represented by five images taken under different viewpoints. The image resolution is 320 × 240. The database is maintained by the Computer Vision Laboratory, ETH Zurich, and available online at [1]. Figure 3.29 shows some sample images from the database.

Since each five images represent one object, when an image is used as query, the four remaining images from its group should ideally be at the top of the query result. Therefore, we return four top ranked images for each query ($M = 4$) and compute the recognition rate. The average number of detected regions per image is 130. Figure 3.30 shows the recognition rates of ten descriptors.

In the multi-region group, the MSR-PLBP^m and IO-QLBP^{sym} perform best.

Figure 3.29: Some representative images from the 53 Objects database. Each row shows a group of five images for one object.

The MROGH ranks third, whose differences to the top descriptors are around 3 % and 6 %, respectively. The IO-QLBP does not perform as well as the top descriptors, yet it is much better than the MRRID. In the single-region group, the MSR-PLBP^s is comparable to DAISY and HRI-CSLTP and better than SIFT and LIOP. Since most transformations in the database are viewpoint changes, descriptors designed primarily for illumination changes like MRRID and LIOP cannot work well. An example of recognition result is shown in Fig. 3.31.

3.7.4 Recognition results on the Recognition Benchmark

The Recognition Benchmark was first introduced in [100] and is available online at [7]. This database contains 2,550 groups of four images known to be taken of the same objects but under different conditions, resulting in a total of 10,200 images. The image resolution is 640 × 480. Image transformations in the database are mainly viewpoint changes, yet they are more drastic than that of the 53 Objects database. Illumination changes and image blur are also included. Therefore, this database is more challenging in terms of both transformation and scale. Figure 3.32 shows some sample images from the database.

For evaluation, we compute on average 390 interest regions per image and return three top ranked images for each query ($M = 3$). We conduct two experiments on a subset of 4,000 images and the full database of 10,200 images, respectively, thus the performances of descriptors on databases of medium and

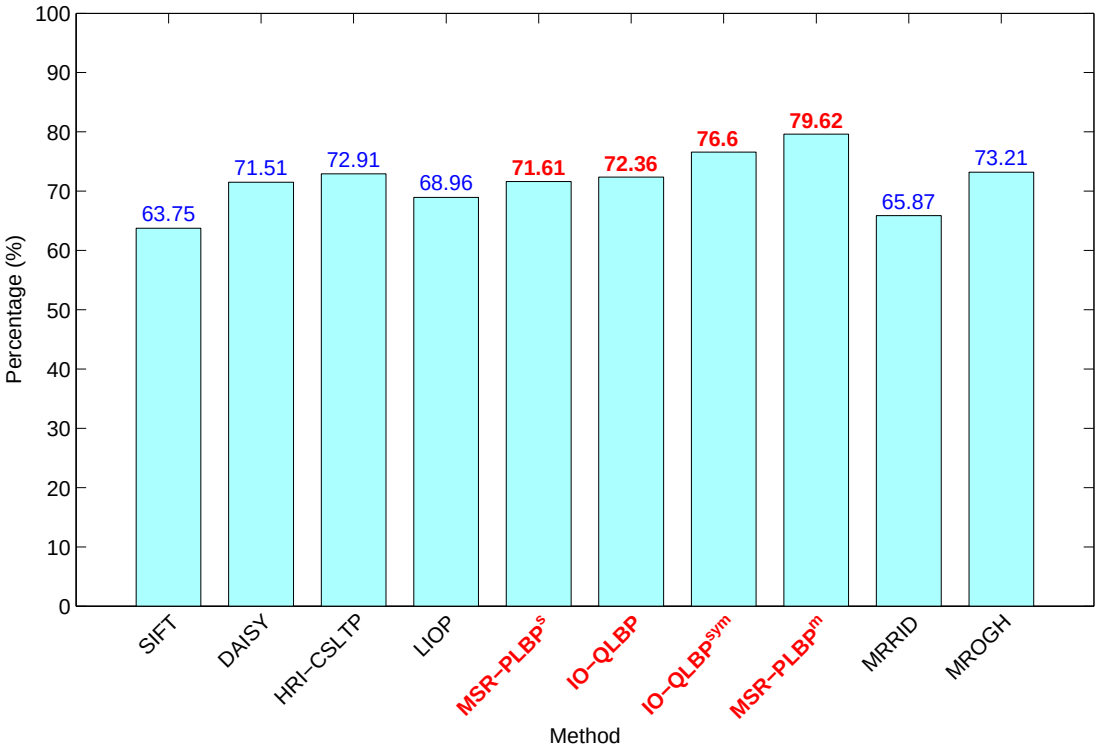


Figure 3.30: Recognition rates on the 53 Objects database. The names and percentage values of the proposed methods are shown in red.

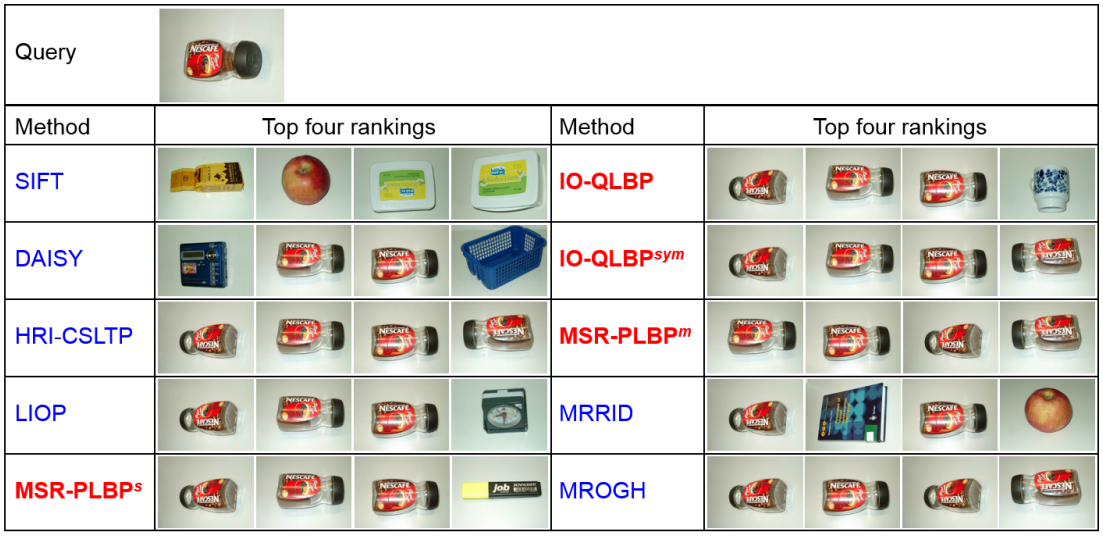


Figure 3.31: An example of recognition result on the 53 Objects database. The names of the proposed methods are shown in red.

large scales are examined. Figure 3.33 shows the recognition rates of evaluated descriptors in two test scenarios.

In the multi-region group, we again observe that the MSR-PLBP^m and IO-

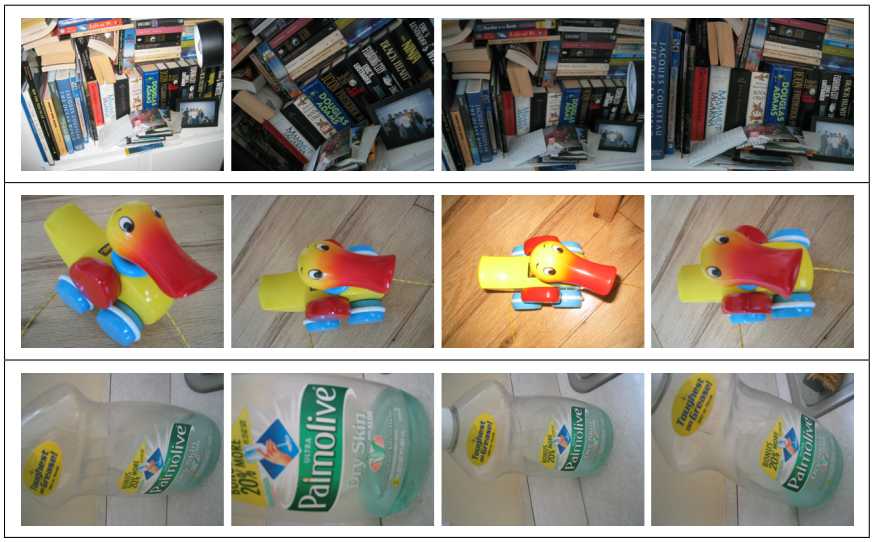


Figure 3.32: Some representative images from the Recognition Benchmark database. Each row shows a group of four images for one object.

QLBP^{sym} are the top ranking descriptors and are followed closely by MROGH and IO-QLBP. The MRRID performs worst and its differences to the next-to-last descriptor are around 10 %. In the single-region group, the MSR-PLBP^{s} slightly loses to the HRI-CSLTP (0.5 - 1.6 %) while outperforming the rest descriptors. Since illumination changes account for a small portion of image transformations in the database, the performances of LIOP and MRRID are not improved significantly. An example of recognition result is shown in Fig. 3.34.

3.8 Discussion and Conclusion

The proposed approaches have been evaluated thoroughly on three image matching datasets, which demonstrate different common image transformations, especially the illumination change and viewpoint change. The MSR-PLBP^{m} is generally most effective among evaluated descriptors. It is robust against both photometric and geometric transformations. Meanwhile, the IO-QLBP^{sym} only exhibits its strength in the geometric transformation. The IO-QLBP performs comparably to the MROGH, yet it cannot compete against the MRRID in the illumination or repetition cases. The MSR-PLBP^{s} is superior to other single-region descriptors and sometimes even to the multi-region IO-QLBP and MORGH. This is thanks to the discriminative power attained from the pattern generation and MSR schemes. With these achievements, the proposed approaches are promising for any applications of interest region description.

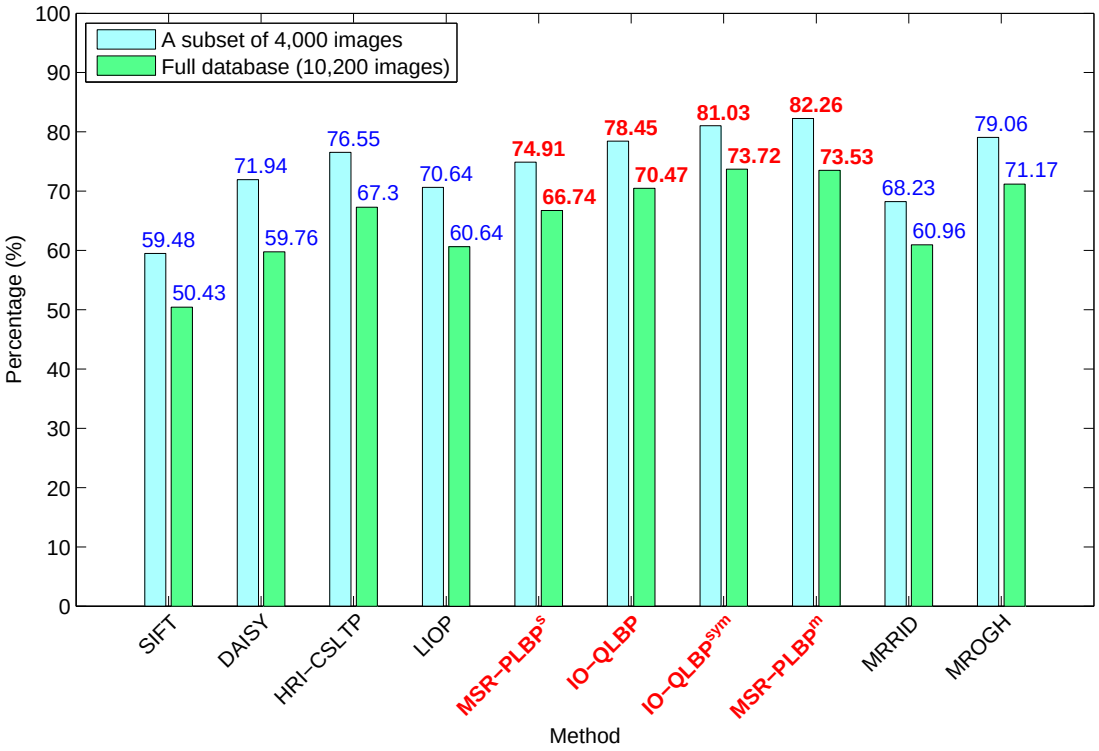


Figure 3.33: Recognition rates on the Recongition Benchmark database. The names and percentage values of the proposed methods are shown in red.

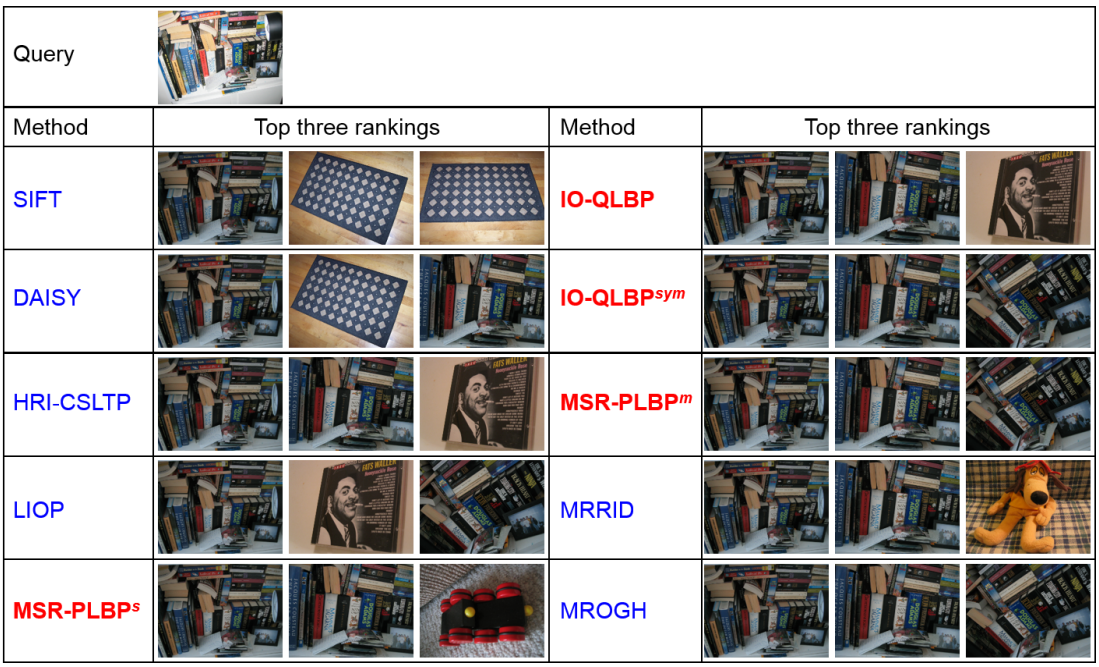


Figure 3.34: An example of recognition result on the Recognition Benchmark. The names of the proposed methods are shown in red.

In addition, our approaches achieve very good performance and stability in the object recognition task at different data scales. The MSR-PLBP^m and IO-QLBP^{sym} are the best among all descriptors. The IO-QLBP is comparable to the MROGH and much better than the MRRID. Meanwhile, the MSR-PLBP^s is superior to most single-region methods, except the HRI-CSLTP. Although the MRRID uses multiple regions and CS-LBP-based feature, it performs badly in this task. This may be due to the ignorance of gradient magnitudes in its encoding. The property is particularly useful for scenes under drastic illumination changes, in which the gradients are too noisy to be examined, yet it is not suitable for other transformations. The proposed approaches, on the other hand, preserve the gradient magnitudes under the form of weight values. The HRI-CSLTP achieves surprisingly high performance by combining the HRI and CS-LTP features. However, its high-dimensional vectors make the matching process extremely time-consuming. It takes twice as long as other descriptors to implement the nearest neighbor matching using Euclidean distance. Our approaches maintain an optimal trade-off between the recognition rate and speed, and hence are more promising solutions.

The MSR-PLBP approach has several limitations. First, although the number of support regions and the number of disks individually boost the performance, combining them does not yield a double effect. Some regions are not large enough and hence suffer over-segmentation when applying the MSR scheme. For future work, we will improve the MSR scheme so that the segmentation operates only on regions of adequate sizes. Second, the PLBP histogram has 16 dimensions, which are double the dimensions of the gradient orientation histogram [40, 81, 94]. This is also the problem of other texture-based descriptors, such as the CS-LBP [55], HRI-CSLTP [49], and MRRID [40]. The issue is more severe when multiple support regions and region divisions are adopted. Techniques such as DCT [123], Gabor filters [160], or PCA [62] will be integrated to make the PLBP histogram concise. Finally, the parameters in Eq.3.14 are empirically selected based on the data of a limited size. A deep analysis of image contrast or gray-level transform could be helpful to tune these parameters more theoretically. Meanwhile, the IO-QLBP approach fails in *corridor* and *there-1-4*, indicating its drawback in handling very drastic transformations. It also suffers the curse of dimensionality and has some free-choice parameters (e.g. the scaling factor τ in Eq.3.2). Therefore, it could share the achievements of future work with the MSR-PLBP.

Chapter 4

Pedestrian Detection with Local Binary Pattern

This chapter presents the study of LBP in pedestrian detection. The QLBP in the previous study is revised to a generic form that better characterizes the shapes of pedestrians. It is then integrated with the Aggregated channel features (ACF), an advanced learning framework in [34], to build the ACF-QLBP pedestrian detector. The novel detector performs robustly against a wide variation of human poses while maintaining an acceptable frame rate.

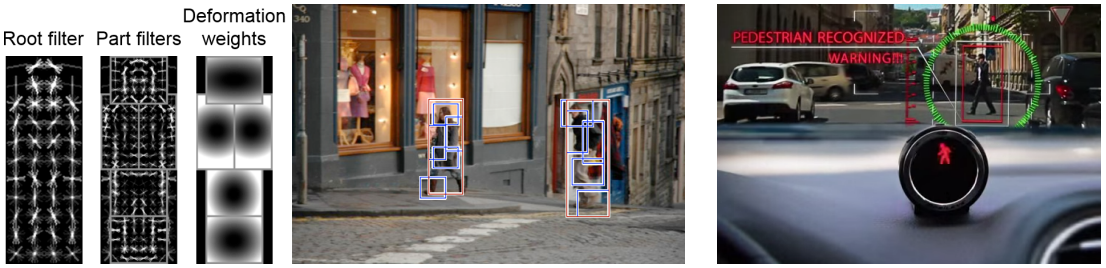
The content of the chapter is organized as follows. Section 4.1 gives the problem definition. Related works are described in Sect.4.2. Section 4.3 presents the proposed pedestrian detector. Performance evaluations are given in Sect.5.4. Finally, Section 4.5 closes the study.

4.1 Problem definition

Pedestrian detection is an important research field of computer vision that positively affects our human life. It is commonly applied in the advanced driver assistance system (ADAS) to help drivers to better notice pedestrians crossing the road. Let us consider some statistics of traffic crashes to see the necessity of a robust ADAS. In the year 2012, the number of pedestrian fatalities accounts for 14% of traffic fatalities in US (4,743 over 33,561) [1], whereas the corresponding number in Japan is 37% (1,634 over 4,411) [2]. The integration of pedestrian

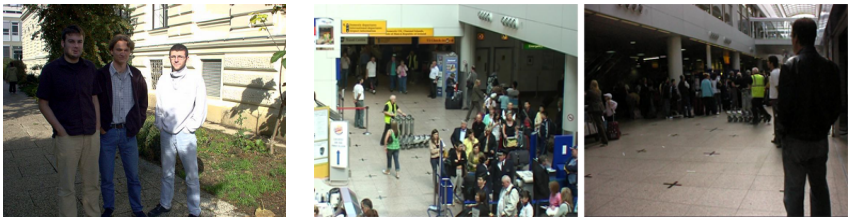
[1]NHTSA: http://www-nrd.nhtsa.dot.gov/Pubs/811888.pdf

[2]Metropolitan Police Department: http://www.keishicho.metro.tokyo.jp/kotu/roadplan/2rin_jiko.htm



(a) Deformable part models [41] (b) Mobileye C2-270

Figure 4.1: Examples of academic and industrial researches on pedestrian detection. (a) The model in [41] describes each object by a coarse root filter and several higher resolution part filters. (b) A commercial ADAS from Mobileye company.



(a) Illumination change (b) Viewpoint change

(c) Noise (d) Motion blur (e) Occlusion

Figure 4.2: Examples of extrinsic factors that affect the quality of pedestran detection.

detection techniques into the surveillance also contributes to the reduction of crimes in public areas. In the year 2009, an analysis conducted on surveyed areas from the United Kingdom to U.S. cities, such as Cincinnati and New York, have shown that using CCTV cameras generally decreases 16% in crime in parking lots, public transportation areas and other public settings [149]. This research also finds applications in robotics, entertainment, content-based indexing and retrieval, or care systems for the elderly and disabled. With several potential social benefits, pedestrian detection has greatly attracted the academic and industry communities, making itself a very active field (cf. Fig.4.1).

Pedestrian detection is defined as the problem of detecting and localizing pedestrians in static images and video footages. "Pedestrian", by definition,

Figure 4.3: Variations due to non-rigid deformations and intra-class variability. From left to right: Shape and size, occlusion of body parts, different clothing, blend of subject and background, and complex interaction.

means a person traveling on foot, especially in an area where vehicles go (cf. Cambridge and Merriam-Webster dictionaries). The subjects are thus usually restricted to people having upright fully visible poses. The constraint is sometimes relaxed to include those traveling using roller skates, skateboards, and scooters, as well as wheelchairs, motorbikes and bicycles, though their poses do not entirely meet the condition. This is because the ADAS is usually installed in cars, for which these types of pedestrians are frequent causes of dangers. The problem is a special case of a more general research domain, called object detection or object categorization, in which face, car and pedestrian are most widely researched. Nowadays, face detection has been well solved in the sense that its detection and recognition rates are usually over 90%, whereas car and pedestrian detection still suffers several limitations.

Detecting pedestrians in real scenes is extremely challenging due to the wide range of possible pedestrian appearances. Variations may arise from extrinsic factors, such as viewpoint change, illumination change, and occlusion. Practical detection systems are typically low-cost devices, hence images are likely to contain noise and motion blur (cf. Fig.4.2). Besides, the pedestrians themselves produce many variations due to non-rigid deformations and intra-class variability of shapes and other visual properties. For example, the body height and size, occlusion of body parts, different clothing, blend of subject and background, and complex interactions between people in the environments (cf. Fig.4.3). Applications working on video footages further deal with the object motion, camera shifting or dynamic background. The problem is even tougher when a real-time system is required, because accuracy and speed are competing factors. In recent years, researches in pedestrian detection have grown significantly, yet they are still very far from a complete solution.

4.2 Related work

4.2.1 A review of related approaches

There are three primary sources of influences on the performance of a pedestrian detector, including the feature(s) describing objects of interest, the detector architecture, and the learning technique(s).

Feature. A detector may adopt one feature or more to describe objects, such as Haar-like feature [141], Histogram of oriented gradients (HOG) [31], shapelets [118], region covariance [109, 140], etc. The HOG [31] is most preferred due to the large performance gain it provides. The underlying concept is to represent an object as dense grids of gradient histogram and normalize feature vectors at the block level. It is robust to illumination changes and able to capture both appearance and spatial information. Its nature perfectly meets the needs of object detection that no individual feature can compete. Although the original HOG [31] is computationally heavy, the issue has been soon addressed by improvements from [41, 84, 170]. Feature combinations, e.g. (HOG, edgelet, covariance) [154], (color channels, gradient magnitude and orientation) [36], etc., have been proposed to boost the performance beyond that of the HOG, yet they require advanced architectures to manage the workload as well as the compatibility between features. The LBP [105] is a robust texture operator, which is highly discriminative, invariant to monotonic gray-scale changes and computationally simple. Although the original LBP is not appropriate for dense matching because of the sensitivity in flat regions and high-dimensional histograms, extended variants successfully overcome the issues while retaining basic properties. The Semantic-LBP and Fourier-LBP are introduced to detect human in photo albums by redefining LBP according to semantic interpretation and Fourier transform, respectively [97]. Pyramids of CS-LBP/CS-LTP achieve comparable or better performance than that of HOG [167]. The LBP is also combined with other methods to produce robust feature combinations that outperform HOG in terms of both accuracy and speed, such as LbpHog [147], Modified CS-LBP with Haar-like features [155] and pyramid CS-LTP with Pyramid HOG (PHOG) [167].

Architecture. The sliding window-based and part-based methods are typical detector architectures, defining how the detector extracts features from images. The former slides a rectangular window along two dimensions of the image at different scales and extracts features for each window. A classifier is then trained to predict whether an unseen window contains a person or not. Detectors follow-

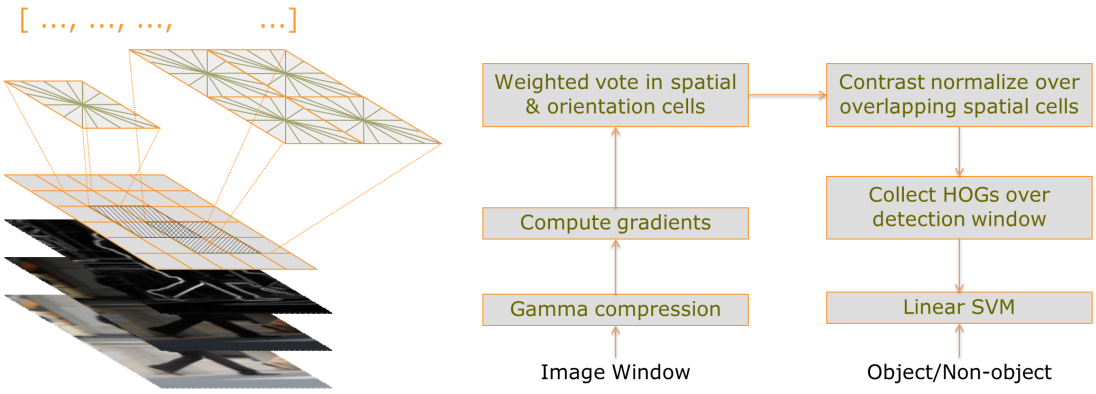


Figure 4.4: The pipeline of the HOG pedestrian detector [31]. The image is obtained from the slide of Navneet Dalal [31].

ing this approach include [16, 31, 34, 36, 118, 141, 147]. Meanwhile, the latter, which is adopted in [15, 41, 116], models an object as parts with a deformable configuration, thus better coping with occlusions and large appearance variations, though the number of parts are often limited for computational feasibility.

Learning algorithm. The support vector machines (SVM) and boosting algorithms are most popular thanks to their solid mathematic foundations and good performance in challenging detection tasks. They have both merits and demerits, thus requiring careful consideration for specific settings. The SVM is high speed and optimum-guaranteed, yet not suitable for large-scale feature training. On the other hand, the boosting automatically does feature selection and rejects redundant candidates in the early stage. Detectors of [31, 41, 84, 143, 147] adopt SVM, while those in [34, 36, 37, 118, 141] implement boosting. Other algorithms, such as random forest [130] and partial least squares [121], are also used with lower rates.

4.2.2 Brief descriptions of competing approaches

The following detectors are selected to participate in the comparative study in Sect.4.4.4. They are sorted according to their publication year for easy reading.

The **VJ** [141] uses AdaBoost to train a chain of progressively more complex region rejection rules based on Haar-like wavelets and space-time differences.

The **HOG** [31] divides an image into $m \times m$-pixel "cells", for each cell a local histogram of gradient directions is computed. The combined histogram entries form the representation. Contrast normalization for better invariance to illumination is done by accumulating a measure of local histogram energy on the

$n \times n$-cell "blocks". The normalized descriptor blocks are referred as Histogram of oriented gradient (HOG) descriptors. It then collects HOGs over the detection window and trains the combined feature vector with a linear SVM. The HOG detector is considered a milestone in object detection due to its great effectiveness at that time. Figure 4.4 shows the pipeline of HOG.

The **Shapelet** [118] uses AdaBoost to automatically learns a set of informative mid-level features ("shapelets"), which are computed from gradient responses. It then adopts the AdaBoost again to build the final classifier from these shapelets.

The **HikSvm** [84] improves the learning methodology by proposing an approximation to the histogram intersection kernel and incorporating it with the SVM. It allows substantial speedups, thus enabling the use of a non-linear SVM in sliding-window detection.

The **ChnFtrs** [37] extends the VJ [141] to compute Haar-like feature over multiple channels of the visual data, including LUV color channels, grayscale, gradient magnitude, and gradient histograms, providing a simple and uniform framework for integrating multiple types of features.

The **HogLbp** [147] combines the HOG and LBP texture and uses the linear SVM to train a global detector for whole scanning windows and part detectors for local regions. In this way, the partial occlusion is effectively handled. This detector is closely related to our approaches in term of LBP feature.

The **Pls** [121] augments widely used edge-based features with co-occurrence matrices and color information to represent the pedestrian. The Partial least squares (PLS) analysis is then employed to efficiently project the data onto a much lower dimensional subspace.

The **FPDW** [36] extends the approach in [37] to fast multi-scale detection. Dollar et al. have demonstrated how features computed at a single scale can be used to approximate feature at nearby scales, thus resolving the computational bottleneck that remains in several modern detectors.

The **LatSVM-V2** [41] involves enriching the model in [31] using a star-structured part-based model defined by a root filter plus a set of part filters and deformation models. It models unknown part positions as latent variables in a SVM framework (cf. Fig.4.1(a)).

The **MultiFtr+CSS** [143] introduces a new feature based on self-similarity of low-level features, termed CSS, to the combination of Haar-like features, shapelets, shape context and HOG feature. It significantly improves the classification performance for both static images and image sequence.

The **Crosstalk** [35] provides an optimized implementation of [36] and couples cascade evaluations at nearby locations, i.e. enabling the communication between neighboring detectors, hence the computational cost is greatly reduced.

The **VeryFast** [16] reverts the FPDW [36] to avoid resizing the input image at multiple scales and uses the "stixels world", a recently introduced fast depth information method, to quickly access to the geometric information from stereo. It attains high quality pedestrian detection at 135 fps.

The **Roerei** [17] is based on the ChnFtrs [37]. It includes properly designs of feature pooling, feature selection, preprocessing, and training algorithms.

The **SketchTokens** [76] introduces a local edge-based mid-level feature, called sketch tokens. Patches of human generated contours are clustered to form sketch token classes and a random forest classifier is used for classification.

The **ACF** [34] uses the same channel features and boosted classifiers as in [37]. The key difference is that the ACF uses pixel lookups in aggregated channels as features while sums over rectangular channel regions are used in [37].

4.3 The ACF-QLBP pedestrian detector

4.3.1 Method overview

The success of the QLBP feature in interest region description (cf. Chapter 3) has been an inspiration for its extensions into pedestrian detection. Specifically, we integrate the QLBP and two other features into the Aggregated channel features (ACF) [34] to form a multi-cue pedestrian detector, called ACF-QLBP. The proposed method performs robustly against several challenging pose changes and environmental conditions thanks to the following properties.

1. The QLBP encodes the information around every pixel in a detection window by comparing the relation between gray values of four neighbors, thus efficiently revealing the gray-value changes in horizontal, vertical and diagonal directions. In addition, an adaptive thresholding scheme is introduced to cope with noises and local illumination changes, giving a better discriminative power than that of the CS-LBP and LBP.
2. The generalized QLBP is revised from the definition in Sect.3.3 to abstractly represent all possible encodings that can be formed by taking pairs from four neighbors. Some encodings performs better than the others because they well characterize the essential cues of human body. Edges along diagonal

and vertical directions are experimentally shown to be most meaningful because they are more visible in upright and symmetric poses.

3. The QLBP is combined with three color channels and a gradient histogram to build a multi-cue detector, thus depicting a pedestrian in different aspects: texture, color, and gradient changes in magnitude and orientation. We adopt the ACF framework [34] for training and testing since it efficiently resolves the computational bottleneck of many modern detectors with an advanced pyramid construction technique. In this way, our detector speeds up to 15.2 frames per second (fps). Although this performance is not real time (i.e. $>$ 24 fps), it is definitely eligible for most pedestrian surveillance systems, which typically operate at 5-10 fps.

The ACF-QLBP detector successfully reduces the average miss rate to a lower degree than its precursor, i.e. the original ACF [34] (15% vs 17%), yet the speed drops down two times (31.9 fps vs 15.2 fps). Despite its superiority over several other modern detectors, especially the baseline HOG (46%, 0.239 fps), the proposed method does not clearly outperform the ACF. Several future works have been planned to address this issue. The proposal of ACF-QLBP has an important contribution to the research of LBP in pedestrian detection. Since only a few LBP-based detectors can attain comparable effectiveness to that of modern detectors [33], it confirms the advantages of LBP and encourages more studies on this feature.

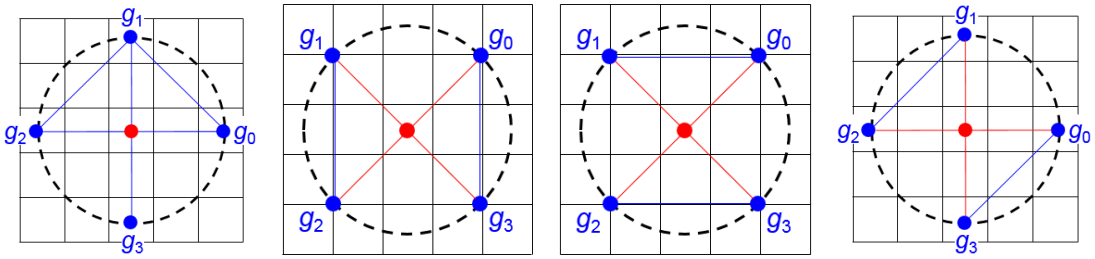
4.3.2 The generalized QLBP texture operator

Let $\mathcal{G}$ denote the set of gray values of four neighbors sampled evenly on a circle of radius R, $\mathcal{C} = \{(g^a, g^b) \mid g^a, g^b \in \mathcal{G}_4\}$ represent the ordered set of four predefined 2-tuples, each of which contains gray values of two neighbors, and T be a threshold. The generic form of QLBP is defined as follows:

$$\begin{aligned} QLBP_R &= \sum_{i=0}^{3} s(g_i^a - g_i^b)2^i \\ s(z) &= \begin{cases} 1 & z > T, \\ 0 & otherwise \end{cases} \end{aligned} \tag{4.1}$$

Each $\mathcal{C}_j$ is an encoding strategy, aka *configuration*. We experimentally examine several $\mathcal{C}_j$ to select the most appropriate configuration for pedestrian detection.

1. $\mathcal{C}_1 = \{(g_0, g_1), (g_1, g_2), \underline{(g_0, g_2)}, \underline{(g_1, g_3)}\}$, the sampling starts at the position of 0^o, i.e. on the horizontal axis, then goes anticlockwise.



(a) $\mathcal{C}_1$ (b) $\mathcal{C}_2$ (c) $\mathcal{C}_3$ (d) $\mathcal{C}_4$

Figure 4.5: Different configurations of the generalized QLBP.

2. $\mathcal{C}_2 = \{(g_0, g_3), (g_1, g_2), \underline{(g_0, g_2)}, \underline{(g_1, g_3)}\}$, the sampling starts at the position of $+45^o$ in the anticlockwise direction.
3. $\mathcal{C}_3 = \{(g_0, g_1), (g_2, g_3), \underline{(g_0, g_2)}, \underline{(g_1, g_3)}\}$, the same sampling as in $\mathcal{C}_2$.
4. $\mathcal{C}_4 = \{(g_0, g_3), (g_1, g_2), \underline{(g_0, g_2)}, \underline{(g_1, g_3)}\}$, the same sampling as in $\mathcal{C}_1$.

where emphasized pairs are underlined (cf. Fig.4.5). Because pedestrian poses are mostly upright and symmetric, edges along diagonal directions may be more discriminative than that of horizontal and vertical directions. That resembles the observation in [64]. We adopt the QLBP operator in Chapter 3 to be $\mathcal{C}_1$, and design three additional configurations for comparison. The $\mathcal{C}_2$ and $\mathcal{C}_3$ emphasizes diagonal edges by associating their corresponding tuples with larger values of 2^i, whereas the $\mathcal{C}_1$ and $\mathcal{C}_4$ highlights horizontal and vertical edges.

We are willing to accept a possible small loss of information when reducing the number of neighbors and design an adaptive thresholding scheme, instead of using a constant T, for compensation. The four T_k below are introduced to adaptively threshold the input of $s(z)$ in Eq.4.1:

1. The gray value of the center pixel g_c

$$T_1 = g_c \tau \tag{4.2}$$

2. The smaller value of (g^a, g^b)

$$T_2 = min(g^a, g^b)\tau \tag{4.3}$$

3. The average of gray values of the center pixel and four neighbors

$$T_3 = \left(g_c + \sum_{i=0}^{3} g_i \right) \tau / 5 \tag{4.4}$$

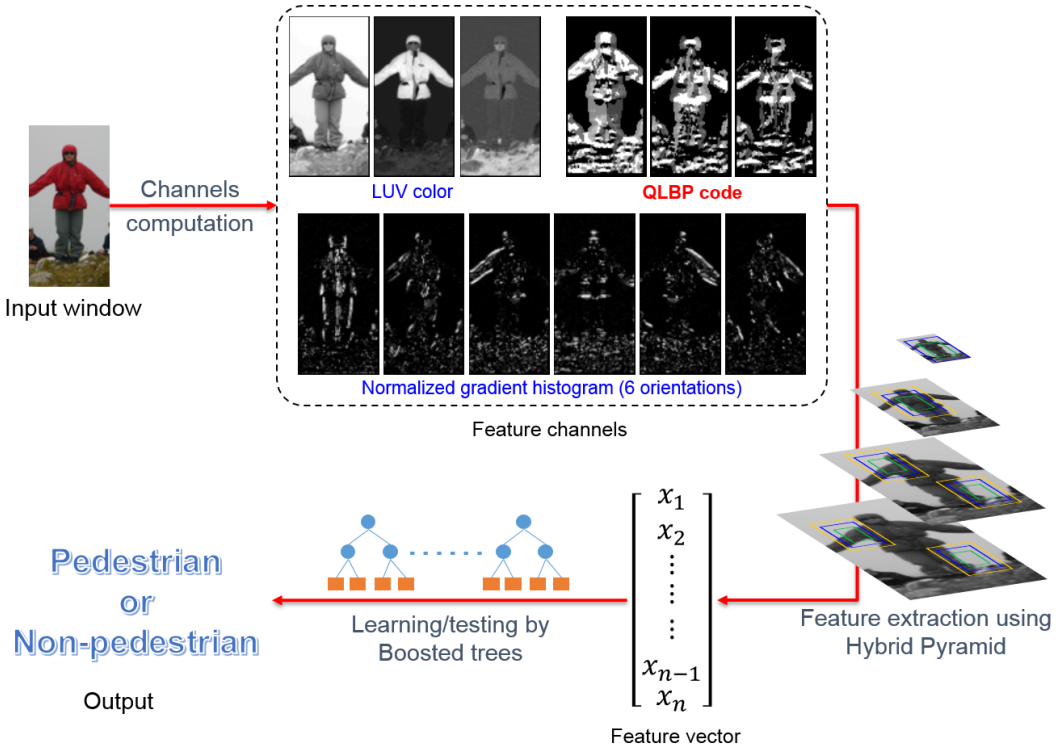


Figure 4.6: The pipeline of the ACF-QLBP pedestrian detector. Feature channels are managed by the ACF learning framework [34].

4. The median of gray values of the center pixel and four neighbors

$$T_4 = median(g_c, g_0, ..., g_3)\tau \tag{4.5}$$

where τ is an empirically selected constant. Experimental results in Sect.4.4 show that the above schemes facilitate the QLBP operator better than a small offset. In addition, the QLBP with an adaptive threshold can achieve higher performance than that of its precursors LBP and CS-LBP.

4.3.3 The detector pipeline

We introduce the use of QLBP to pedestrian detection by integrating this feature into a multi-cue detector. Together with three LUV color channels and a 6-orientation gradient histogram, it constitutes a robust feature combination that describes the pedestrian in different aspects, thus effectively boosting the detection accuracy. Figure 4.6 shows the pipeline of the proposed detector. We adopt the Aggregated channel features (ACF) [34] for learning the detector. This framework speeds up the training/testing using an advanced pyramid construc-

tion technique. In addition, it organizes features into modules, allowing the evaluation to be conducted on the same background. Since this study focuses on the proposal of effective feature rather than the detector architecture, we simple integrate the QLBP into the ACF framework and leave other details unchanged.

Input. Because the multi-cue detector deals with color channels, color input images are required. It is not a challenging assumption in the sense that most datasets nowadays are color supported. Similarly to [34], every image is smoothed with a [1 2 1]/4 filter for noise reduction.

Feature channels[3]. The detector adopts twelve channels of three different features, including six channels for a normalized gradient histogram of six orientations, three color channels (L, U, V) and three corresponding QLBP channels computed on each color layer. The ACF method [34] and ours share the two first features, yet for the last feature, we compute QLBP patterns instead of gradient magnitudes. This allows us to exploit objects in more aspects than that of [34]. We empirically select the CIE-LUV from a large variety of available color spaces (e.g. HSV, YIQ, YUV, sRGB, CIE-LAB, etc.), because it well approximates the human vision and achieves good performance in several detection tasks [32, 37].

Feature Pyramid. The feature pyramid is constructed by the Fast feature pyramids technique [34], which computes feature channels at one scale per octave and does extrapolation at intermediate scales. The coarsely-sampled pyramid is simpler yet sufficient to well approximate the finely-sampled pyramid, thus considerably speeding up the framework without noticeable loss in accuracy.

Boosting algorithm. To represent a 128×64 detection window, the ACF performs channel pixel lookup to divide each feature channel into 4×4 blocks and sum the pixels in each block, resulting in a vector of $128 \times 64 \times 12/16 = 6144$ dimensions. These vectors are then trained/tested using Adaboost [43] with 2048 depth-two decision trees as weak classifiers.

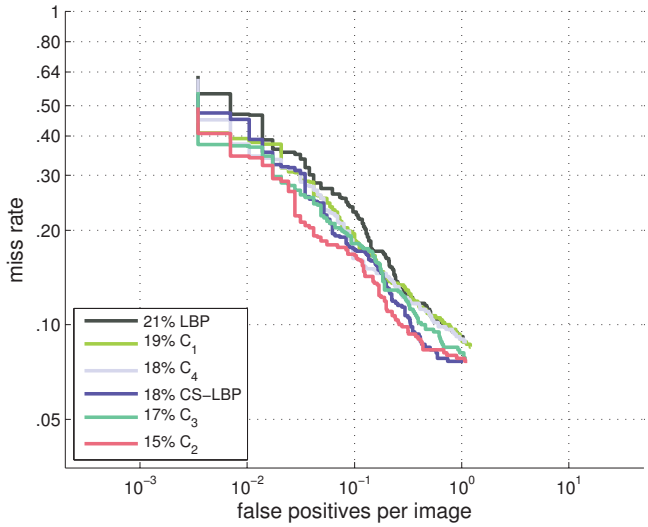
4.4 Performance evaluations

4.4.1 Evaluation protocol

The performance of a pedestrian detection algorithm is evaluated by the Detection Error Tradeoff (DET) curve, which features the *miss rate* versus *FPPI (False*

[3] A channel is a registered map of an input image so that each output pixel is computed from corresponding patches of input pixels [37].

Figure 4.7: The performance of QLBP configurations, in comparison with that of LBP and CS-LBP.



Method	LBP [105]	CS-LBP [55]	Our $\mathcal{C}_2$
Training time	543	401	350
Frame rate	4.79	7.02	15.2

Table 4.1: The training duration (in seconds) and frame rates (in fps) of LBP, CS-LBP and $\mathcal{C}_2$.

Positives Per Image). A lower curve demonstrates the better algorithm. The *miss rate* is defined as follows:

$$\begin{aligned} missrate &= 1 - detectionrate \\ &= \frac{\#FalseNegatives}{\#TruePositives + \#FalseNegatives} \end{aligned} \tag{4.6}$$

The evaluation methodology is fundamentally based on [31] with a difference that FPPI values are used for the x-axis instead of FPPW (False Positives Per Window) values. The per-window evaluation tends to isolate the performance of classifier from that of the whole detection system, whereas the per-image evaluation provides a more general view of the system [35, 167]. We use the overall miss rate between 10^2 and 10^0 FPPI as a reference point for all subsequent evaluations.

4.4.2 Evaluation of QLBP configurations

We evaluate four QLBP configurations, $\mathcal{C}_j$, (cf. Fig.4.5) to determine the most appropriate setting for pedestrian detection. It can be seen from Fig.4.7 that $\mathcal{C}_2$ performs best, followed by $\mathcal{C}_3$ and $\mathcal{C}_4$, while $\mathcal{C}_1$ performs worst. This ranking order indicates that configurations designed specifically for pedestrians, i.e. $\mathcal{C}_2$ and $\mathcal{C}_3$,

are able to achieve better performance than that of arbitrary configurations. They emphasize diagonal edges, which are likely to be more discriminative in upright human poses than horizontal and vertical edges, thus attaining lower miss rates. In addition, vertical edges are more visible than horizontal ones in this context, which may explain for the superiority of $\mathcal{C}_2$ to $\mathcal{C}_3$. It is worth noting that $\mathcal{C}_1$ does not perform as well as others, though the IO-QLBP descriptor is shown to be comparable to or better than most modern descriptors in interest region description (cf. Chapter 3). Since visual problems are by nature different from each other, simply applying a method from one field on another field without sufficient considerations usually produces unexpected results.

The four $\mathcal{C}_j$ are also compared with the LBP and CS-LBP. The LBP has the highest miss rate, which is 6% higher than that of our best configuration, $\mathcal{C}_2$. The CS-LBP performs comparably to $\mathcal{C}_3$ and $\mathcal{C}_4$, but loses to $\mathcal{C}_2$. The differences of miss rate are around 1% and 3% for the equivalence and inferior cases, respectively. The LBP uses eight neighbors to create a 256-dimensional histogram; the CS-LBP uses eight neighbors to generate a shorter representation of 16 dimensions, whereas ours only requires four neighbors to attain a similar 16-dimensional histogram. We may infer from observations that the fine quantization of patterns and dense neighbor sampling are not suitable for a detection task, whose subjects are often contaminated by wide variations of appearances and complex backgrounds. The training durations and frame rates of $\mathcal{C}_2$, LBP and CS-LBP are shown in Table 4.1 (other $\mathcal{C}_j$ do not differ much from $\mathcal{C}_2$, hence are not necessarily listed). The proposed operator halves the number of neighbors in the CS-LBP and emphasizes primary edges, therefore operating more efficiently. Despite the differences of miss rate between $\mathcal{C}_j$, they are all comparable to modern detectors mentioned in Fig. 4.10.

4.4.3 Evaluation of thresholding schemes

We evaluate four thresholding schemes introduced in Sect.4.3.2 in terms of their contributions to the robustness of a QLBP operator. The parameter τ is empirically selected for all schemes such that it lies in the range of [0, 0.01] for satisfactory results. According to Fig.4.8, T_2 performs best, followed by T_3 and T_1, and T_4 performs worst. T_2 compares the difference z in Eq.4.1 with the smaller value of (g^a, g^b), i.e. threshold values are set adaptively for every pair of neighbors, hence successfully handling challenges such as local illumination and sudden gray-value changes caused by noise. The miss rate of T_2 is 3-4% lower

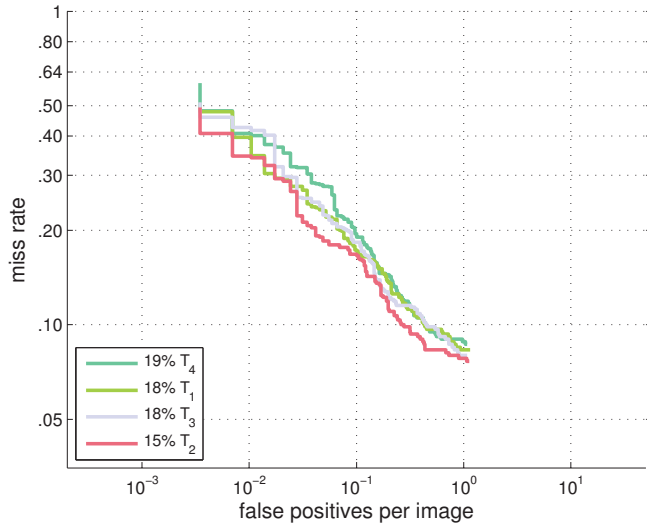


Figure 4.8: The performance of four thresholding schemes.

than that of other schemes, while T_1, T_3, and T_4 performs comparably to each other. Schemes other than T_2 define threshold values for pixels rather than for neighboring pairs and perform less effectively than T_2. In addition, T_4 is computationally heavier than other T_k because it requires sorting to compute the median value. In conclusion, we choose T_2 as the associated thresholding scheme for the QLBP operator.

4.4.4 Evaluation of ACF-QLBP and competing detectors

The comparative study is conducted on the INRIA benchmark [31], which includes 902 with-person images (positives), 1774 human annotations, and 1671 person-free images (negatives). For training, 1208 annotations, together with their left-right reflections, are selected, resulting in 2416 positives samples. 12,180 negative samples are created by randomly sampling patches from 1218 person-free images. Similarly, the rest 566 annotations with their reflections are used for testing. The dataset covers pedestrians in a wide range of poses (mostly upright), appearances, clothing, illumination, and background, as well as in some partial occlusions. Therefore, it is widely used for evaluating pedestrian detection algorithms. Some sample images from the dataset are shown in Fig.4.9.

We compare our multi-cue detector with 15 competitors, including ACF [34], ChnFtrs [37], Crosstalk [35], FPDW [36], HikSvm [84], HOG [31], HogLbp [147], LatSVM-V2 [41], MultiFtr+CSS [143], Pls [121], Roerei [17], Shapelet [118], SketchTokens [76], VeryFast [16] and VJ [141]. Brief descriptions of these methods are presented in Sect.4.2.2.

Figure 4.9: Sample images from the INRIA benchmark [31]. Challenges include partial occlusions and wide variations of pose, appearance, clothing, illumination, and background.

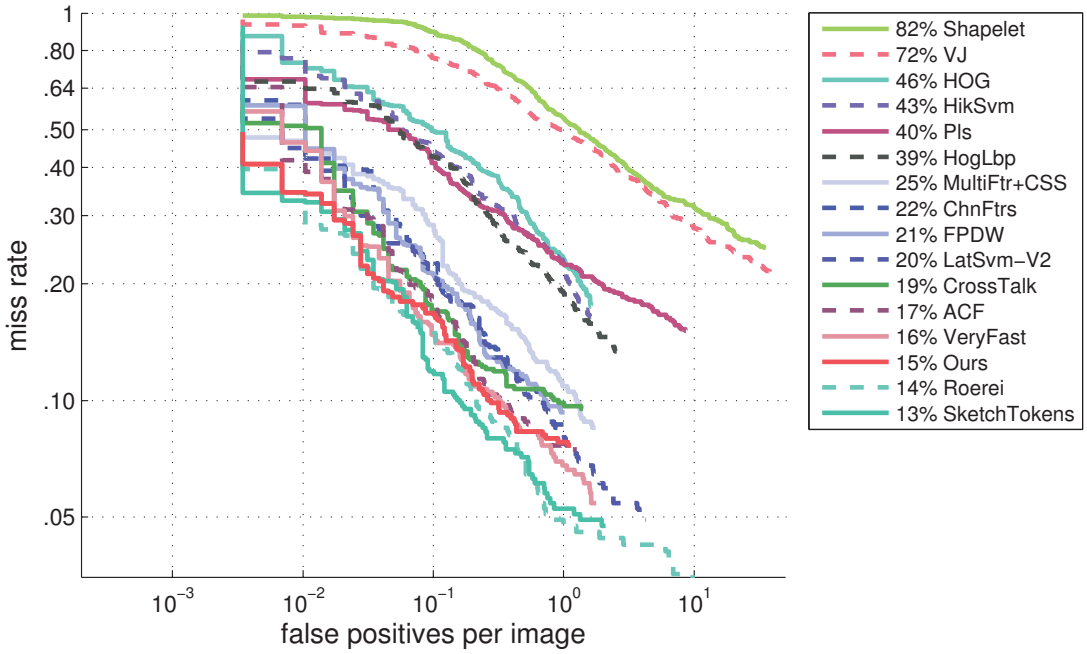


Figure 4.10: The performance of the ACF-QLBP and competing detectors on the INRIA benchmark. The data is obtained from the Caltech Pedestrian Detection Benchmark website [33] and are discussed in [34, 35].

The ACF-QLBP achieves a frame rate of 15.2 fps with an Intel Core i7 950 2.8GHz desktop, 640×480 image resolution, and non-optimized MATLAB implementation of QLBP. The configuration $\mathcal{C}_2$ and the thresholding scheme T_2 ($\tau = 0.005$) are selected. Table 4.2 presents the frame rates of all competitors. Our detector is slower than the ACF, Crosstalk and VeryFast, while much faster than the others. This is thanks to the selection of the advanced ACF framework.

It can be observed from Fig.4.10 that the ACF-QLBP has a low miss rate (around 15%) and appears in the top five most accurate methods. The ACF [34] and ours share the use of LUV color channels and normalized gradient histogram. These features are combined with a gradient magnitude channel in the ACF,

Algorithm	Frame rate	Algorithm	Frame rate
Pls [121]	0.018	ChnFtrs [37]	1.2
MultiFtr+CSS [143]	0.027	FPDW [36]	6.5
Shapelet [118]	0.051	**Ours**	**15.2**
HogLbp [147]	0.062	ACF [34]	31.9
HikSvm [84]	0.185	Crosstalk [35]	45.4
HOG [31]	0.239	VeryFast [16]	50
VJ [141]	0.447		
LatSVM-V2 [41]	0.629		

Table 4.2: The frame rates of the ACF-QLBP and competing detectors (in fps). The data is obtained from [34, 35]. The Roerei and SketchTokens do not appear in this table because their frame rates are not available.

while they are with three QLBP channels in ours. The ACF only exploits the information of color and gradient but ours further capture textures from QLBP channels. Therefore, we successfully reduce 2% of miss rate, demonstrating the advantage of new feature combination.

The proposed detector, however, performs worse than Roerei [17] and StretchTokens [76] with differences of about 1% and 2%, respectively. The Roerei improves the ChnFtrs architecture [37] and adopts an advanced normalization scheme, as well as several subtle modifications, whereas we simply integrate the QLBP into the ACF framework and leave other details unchanged. The StretchTokens combines sketch tokens with ten feature channels in ChnFtrs. Sketches are mid-level features, which are more discriminative than the low-level texture features, yet the sketch extraction is completely not straightforward.

The HogLbp uses the traditional LBP features and produces a higher miss rate than ours (24% vs 15%). This is because the QLBP is superior to the LBP, which has been already confirmed in the first evaluation (cf. Sect.4.4.2). The success also lies in two factors: a robust feature combination and a good detector architecture. First, the proposed detector captures different characteristics of a pedestrian, including color, texture and gradient changes in magnitude and orientation. These features complement each other and well facilitate the detector. Second, the adopted framework [34] possesses an advanced learning mechanism that allows the training to be accurate and fast. This mechanism also serves several robust detectors like the ACF, Crosstalk, and FPDW. The HogLbp, on the other hand, simply augments HOG and LBP features then trains them with a linear SVM. In addition, the proposed ACF-QLBP greatly outperforms the

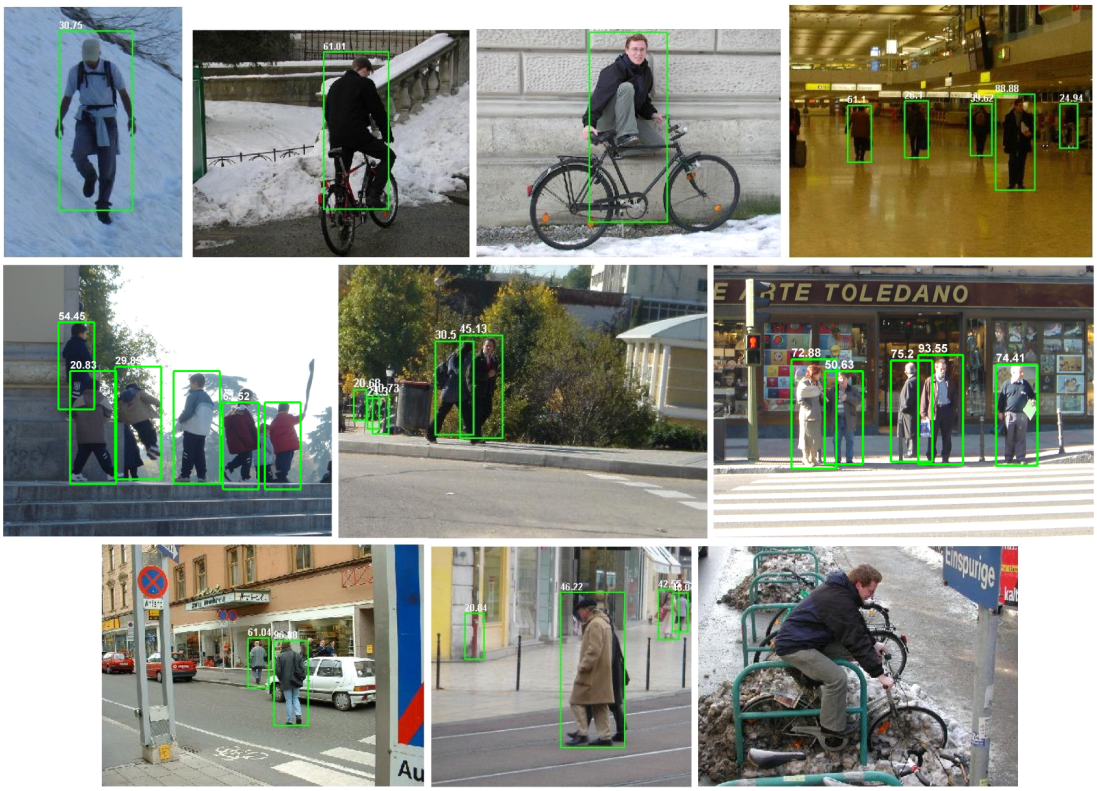


Figure 4.11: Some detection results on the INRIA benchmark. The first and second rows show successful cases, while the third row exhibits failure cases.

baselines HOG and VJ, indicating the advances of modern architectures and feature combinations.

Figure 4.11 shows some detection results using the proposed ACF-QLBP detector. This detector is able to detect pedestrians in different poses (e.g. turning back or bending the body on the bicycle), challenging backgrounds and lighting conditions. In addition, it is also able to detect pedestrians at very small size, as shown in the middle image of the second row. Nevertheless, it still suffers several failures, which are presented in the last row. It cannot discover a pedestrian under large occlusion, e.g. over a half of the body is occluded by a car (left figure) or by another pedestrian (middle image). Also in the middle figure, we observe a false detection on the gutter. The detector sometimes gets confused with objects that have upright poses, rectangular or cylindrical shapes (i.e. similar to the human body) and/or similar color tones with the clothing. The right figure shows a failure due to the heavy non-rigid deformation. Since our detector is sliding window based, it cannot collect enough features in these cases. A part-based architecture [41] or aids from multi-pedestrian detection [108] may resolve the problem.

4.5 Discussion and Conclusion

This study examines the effectiveness of LBP features in pedestrian detection with the aim of discovering an appropriate encoding and broadening the research of LBP for this problem. The proposed ACF-QLBP detector simultaneously considers three different types of features, including the generalized QLBP, color channel, and gradient histogram, to attain high robustness. The generalized QLBP approximates the CS-LBP [55] by reducing the number of neighbors and enhancing the adaptability of the thresholding scheme, thus it is computationally simpler yet more robust to noise. The comparative evaluation of the LBP, CS-LBP, and QLBP, shows the superiority of QLBP in terms of both detection accuracy and speed (cf. Sec.4.4.2). In the evaluation with competing detectors, the proposed detector can reduce the average miss rate to 15.4%, which is comparable to many modern detectors and much better than the baseline HOG. A frame rate of 15.2 fps is achieved, which means our detector is faster than 10/13 competing detectors.

A limitation is that the frame rate is still far from real-time (i.e. $>$ 24 fps). Adopting the same framework, the ACF [34] uses 10 channels (5120 candidate features) for each window while the ACF-QLBP needs 12 channels (6144 features). Ours is three times slower than ACF (15.2 fps vs. 31.9 fps), yet achieves a lower miss rate (15% vs. 17%). Although only a few detectors can attain high frame rates, this is still an important problem to resolve. In addition, since our detector still performs worse than Roerei [17] and StretchTokens [76], improving the detection accuracy is another mission. Enhancing the QLBP encoding or computing QLBP features on another stable medium rather than color channels are our initial ideas. Another minor limitation is that the τ in Eq.4.2-4.5 needs to be empirically selected, hence novice users may meet some difficulties even when a feasible value range is introduced. For future work, we will study the behavior of image transformation to discover a more independent thresholding scheme.

Chapter 5

Background Subtraction with Local Binary Pattern

This chapter introduces a novel background modeling framework that models the background scene with QLBP features and color. For each of the consecutive video frames, the framework first considers the feature distributions in blocks of pixels then examines the color distributions in each individual pixel. The combination of block-wise and pixel-wise approaches enables the framework to capture both global and local structures effectively, thus enabling high robustness against various environment conditions.

The content of the chapter is organized as follows. Section 5.1 gives the problem definition. Related works are described in Sect.5.2. Section 5.3 presents proposed algorithm. Performance evaluations are given in Sect.5.4. Finally, Section 5.5 closes the study.

5.1 Problem definition

In several systems like video surveillance, motion capture and multimedia, it is essential to localize the humans in the scene so that their trajectories or activities can be recognized afterward with some tracking algorithm. A common approach is to detect the locations of humans in every single frame. The pedestrian detection techniques in Chapter 4 are instantiations of this approach. Although modern detectors have achieved very high detection rates, several unexpected errors still occur in complex contexts. The background subtraction in this chapter utilizes the temporal information from multiple frames of a video sequence to reduce such errors. It is commonly used when the image in question is a part of

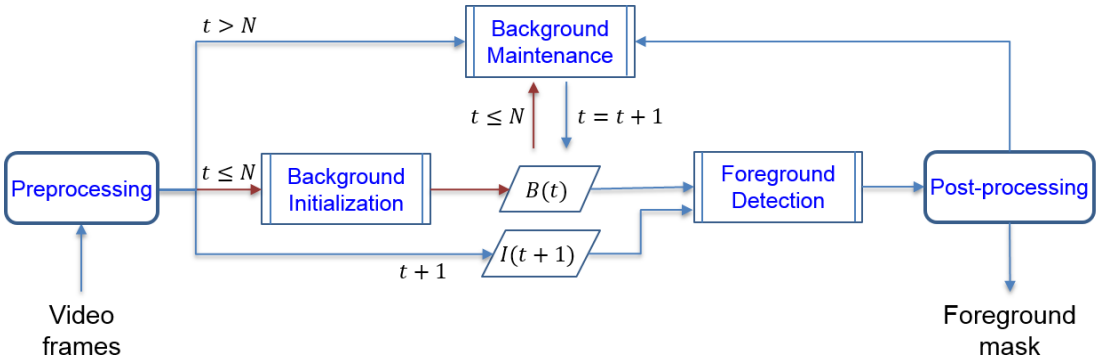


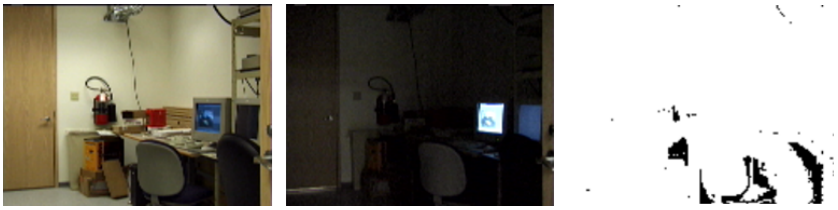
Figure 5.1: A typical pipeline of background subtraction. The term B denotes the background image while I denotes the current image.

a video stream.

The background subtraction (BGS) aims to build a representation of the scene, called the background model, and then finds deviations from the model for each incoming frame. Any significant change in an image region, compared to the background model, indicates a moving (foreground) object [159]. These moving objects are low-level visual cues that serve some higher-level object analysis processes, such as motion tracking, human gait classification, video segmentation and event detection. Because this process depends mainly on the feature differences between frames, it is worth noting that the input must be provided by a stationary camera so that the viewing position and angle are unchanged during the operation.

A typical background subtraction procedure consists of four components: background modeling, background initialization, background maintenance and foreground detection [19, 27]. The background modeling denotes the selection of an appropriate model to represent the background. The other components operate on the selected model following the pipeline in Fig.5.1. The background initialization first learns the model from the information of N first frames. These training frames should contain the background only, though it is not always the case. The foreground detection is triggered afterwards to classify a pixel as a background or foreground pixel. These pixels constitute a foreground mask, which is then applied on the current frame to obtain the moving objects. The background maintenance component controls the adaption of the background model over time following the changes occurring in the scene. Preprocessing and post-processing are optional steps, in which the former involves simple image processing tasks that change the raw input into a suitable format while the latter uses domain knowledge and computationally-intensive vision algorithms to eliminate pixels

(a) Gradual illumination change. The first three images presents an indoor scene with increasing illumination. The last image shows the background image created by the MOG [124]



(b) Sudden illumination change. The first image presents an indoor scene with light-on. The light suddenly goes out, resulting in the second image. The last image shows the background image created by the MOG [124]

Figure 5.2: The effects of illumination change to background subtraction. The input and background images are provided in [136] and [20], respectively. Black pixels belong to the background while white pixels to the foreground.

(a) Wavering tree branches. (b) Water rippling.

Figure 5.3: The effects of dynamic background to background subtraction. The input images are provided in [70] while the background images are in [20]. Black pixels belong to the background while white pixels to the foreground.

that do not correspond to the actual moving objects. The background modeling is most important since it determines how well the background subtraction adapts to the critical situations. Therefore, a large number of studies focus on improving this step and the term "background subtraction" and "background modeling" are usually used interchangeably.

The general requirements for a background modeling algorithm include spatial accuracy, temporal coherency, sensitivity and robustness. That is, the algorithm should be able to detect minor changes to obtain accurate object contours and it should performs consistently over time under varying conditions. Such an ideal algorithm is hard to be achieved in practice due to several challenges. The

arrival/removal/re-location of background objects like waving trees and rippling water makes the background usually dynamic. The background model is commonly initialized inadequately because of the presence of foreground objects in the training images. These objects occlude several areas of the background, hindering the modeling from learning an accurate background view. In addition, the foreground object may have similar texture as that of the background or become motionless for a long time, making the distinction between the real background and foreground very difficult. The illumination change is another source of false positives. It may alters the pixel intensity progressively (gradual changes) or abruptly (sudden changes), and therefore the model incorrectly recognizes these changes as indications of moving objects. Other challenges come from the poor image quality, camera jitter or camera automatic adjustments. The illumination change and dynamic background are by nature very critical and they also occur commonly in practical scenes, thus the background subtraction research identifies them as major issues [19] (cf. Fig.5.2 and Fig.5.3). More illustrates of the aforementioned challenges are shown in Sect.5.4.3.

5.2 Related work

5.2.1 Common trends in background modeling

There are a number of background modeling algorithms in the literature, which are discussed in several comprehensive surveys [20, 29, 39]. We briefly summarize common research trends and highlight some representatives in each trend.

- Basic background modeling: a single background image is modeled using the average [68], median [88], or mode of a histogram [166] over time.
- Statistical background modeling: the background is modeled as one or many distributions using a single Gaussian [153], a mixture of Gaussians [124], or kernel density estimation [38]. The pixel in question is compared with the distribution(s) so that it is classified as foreground or background pixel.
- Fuzzy background modeling: the fuzzy concept is introduced to account for the uncertainty, such as fuzzy running average [14] or Type-2 fuzzy mixture of Gaussians [14].
- Background clustering: the algorithms following this approach use the K-mean algorithm [23] or a codebook [65] to represent each pixel in the frame temporally by clusters. Incoming pixels are matched against the correspond-

ing cluster group and are classified depending on whether the matched cluster is considered part of the background.

- Neural network background modeling: a neural network is trained on N clean frames to classify each pixel as background or foreground, thus the background is modeled by weights of the neural network [30, 82].
- Wavelet background modeling: the coefficients of discrete wavelet transform (DWT) is employed to defined the background in the temporal domain [18].
- Background estimation: these methods adopt a Weiner filter [136], Kalman filter [89], or a Tchebychev filter [26] to estimate the background. Any pixel of the current image that deviates significantly from its estimated value is considered foreground pixels.

The statistical background modeling is most frequently adopted due to its robustness against critical situations and many developments have been introduced in this topic [20]. These approaches typically fall into three categories: Gaussian models, support vector models and subspace learning models. We describe each category in turn while giving more details on the Gaussian models because they are closely related to the proposed multi-layer framework.

Gaussian models: the basic way is to model the history of intensity change at each pixel by a single Gaussian. Wren et al. [153] use a single 3D Gaussian $N(\mu(x,y),\sigma(x,y))$ to describe the distribution of YUV color at each pixel $I(x,y)$ of a stationary background. The mean μ and the variance σ are learned from several consecutive frames. Every pixel (x,y) of an incoming frame is compared to the $N(\mu(x,y),\sigma(x,y))$. It is labeled as foreground pixel if its color greatly deviates from the distribution. However, this unimodal model cannot handle dynamic backgrounds where there are repetitive object motions (e.g. waving trees or water rippling), shadows or reflectance. Stauffer and Grimson [124] addresses the problem with a mixture of Gaussians (MOG). They match the pixel in consideration with every Gaussian of the background MOG. If a match is found, the mean and variance of the corresponding Gaussian is updated. Otherwise, a new Gaussian is initialized using the current pixel and then joined to the MOG. The drawback is that the number of Gaussians greatly depends on specific applications. Elgammal et al. [38] uses non-parametric kernel density estimation (KDE) to compute the probabilities at each pixel. Each pixel is matched not only based on the corresponding pixel in the background model but also to nearby pixels, thus incorporating the region-based information instead of using color only. This method is capable of handling camera jitter or small movements in

the background, yet it is quite computationally heavy.

Support vector models: It is based on the principle of support vector to model the background for different applications. Lin et al. [77] use a support vector machine for outdoor scenes. Wang et al. [145] adopt the support vector regression to handle illumination change in traffic surveillance. Tavakkoli et al. [132] apply support vector data descriptor to deal with dynamic backgrounds.

Subspace learning models: Learning with the PCA is employed in [107] to represent the background by a mean image and a projection matrix comprising the first p significant eigenvectors. The difference between the input image and its reconstruction is then computed to detect the foreground. Some improvements resort to the use of independent component analysis (ICA), which is a variant of PCA in which components are assumed to be mutually statistically independent instead of merely uncorrelated, to remove the rotational invariance of PCA [157]. Meanwhile, the Incremental non-negative matrix factorization (INMF) [22] is proposed to reduce the dimension, and the Incremental rank-(R_1, R_2, R_3) tensor [72] takes into account the spatial information.

5.2.2 Mixture of Gaussians

Stauffer and Grimson [124] model the recent history of the color features at each pixel in the image by a mixture of K Gaussians (MOG), demonstrating the multimodality of the background. This algorithm is based on two basic assumptions: 1) the time series of observations at a given pixel is independent of observations at other pixels, and 2) it can be modeled using a mixture of K Gaussians.

Each pixel is characterized by a three-dimensional vector x that contains the RGB color intensities at the pixel. The probability that x is observed at time t is given by:

$$\begin{aligned} P(x_t) &= \sum_{i=1}^{K} \omega_{i,t}\ \eta\left(x_t, \mu_{i,t}, \Sigma_{i,t}\right) \\ &= \sum_{i=1}^{K} \omega_{i,t}\ \frac{1}{(2\pi)^{n/2}|\Sigma_{i,t}|^{1/2}} e^{-\frac{1}{2}(x_t-\mu_{i,t})^{\intercal}\Sigma_{i,t}^{-1}(x_t-\mu_{i,t})} \end{aligned} \tag{5.1}$$

where $\omega_{i,t}$, $\mu_{i,t}$ and $\Sigma_{i,t}$ is the weight, mean and covariance matrix of the i^{th} Gaussian at time t. K is the maximum number of Gaussians for a pixel.

For computational reasons, the RGB color channels are assumed to be inde-

pendent and have the same variance. Hence, the covariance matrix is written as:

$$\Sigma_{i,t} = \sigma_{i,t}^2 I \tag{5.2}$$

where σ is the variance and I is an identity matrix.

The parameters of the MOG model include the number of Gaussians K, the weight $\omega_{i,t}$ associated to the i^{th} Gaussian at time t, the mean $\mu_{i,t}$, and the covariance matrix $\Sigma_{i,t}$ (or $\sigma_{i,t}^2$). They are initialized and updated as follows.

Parameter initialization. K is determined according to the available memory and computational power. It is usually set from 3 to 5 as proposed in [124]. The $\omega_{i,t}$ is assigned a low prior weight and the $\sigma_{i,t}^2$ is with a large initial variance, indicating the Gaussians are not observed frequently and not stable at the beginning.

Parameter update. After initialization, the first foreground detection can be made and then the parameters are updated. To determine the number of Gaussians to represent the background, K distributions are sorted in descending order following the criterion ω_i/σ_i and only the first B distributions are chosen:

$$B = \arg\min_b \left(\sum_{i=1}^{b} \omega_i > T \right) \tag{5.3}$$

where T determines the minimum fraction of the recent data contributing to the background model. This criterion indicates that the background should be represented by distributions that have high weights and weak variances, i.e. they are more present than other distributions and their values are practically constant.

When a new frame comes at time t, a match test is made for each pixel. If the Mahalanobis distance $\sqrt{(x_t - \mu_{k,t-1})^\intercal \Sigma_{k,t}^{-1} (x_t - \mu_{k,t-1})} < \delta\sigma_{k,t-1}$ ($\delta = 2.5$), the observation x_t matches the k^{th} Gaussian. If this matched Gaussian is identified as a background distribution, the corresponding pixel is classified as background. Otherwise, the pixel is a foreground pixel. The parameters of the k^{th} Gaussian are updated:

$$\mu_{k,t} = (1-\alpha)\mu_{k,t-1} + \alpha x_t \tag{5.4}$$

$$\sigma_{k,t}^2 = (1-\alpha)\sigma_{k,t-1}^2 + \alpha(x_t - \mu_{k,t})^T (x_t - \mu_{k,t}) \tag{5.5}$$

The weights of all Gaussians are updated to

$$\omega_{i,t} = (1-\alpha)\omega_{i,t-1} + \alpha M_{i,t}, \quad M_{i,t} = \begin{cases} 1 & \text{if } i = k \\ 0 & \text{else} \end{cases} \tag{5.6}$$

where α is the learning rate that controls the speed of adaption and k is the index of the matched Gaussian.

If no match is found with any of the K Gaussians, the corresponding pixel is a foreground pixel. The least probable distribution k is replaced with a new one whose parameters are:

$$\mu_{k,t} = x_t \tag{5.7}$$

$$\omega_{k,t} = \text{low prior weight} \tag{5.8}$$

$$\sigma_{k,t} = \text{large initial variance} \tag{5.9}$$

In practice, a low prior weight is equal to the learning rate α while the initial variance is set according to the range of values of a specific feature (e.g. $4 < \sigma_{r,g,b} < 15$ [120]).

The above maintenance procedure is applied for every incoming frame during the background subtraction.

5.2.3 Brief descriptions of competing approaches

The following algorithms are selected to participate in the evaluation in Sect.5.4.4.

The **MOG** [124] is the baseline in the statistical approach. It models each pixel with a mixture of Gaussians and uses an online approximation to update the model. The Gaussians with top values of ω_i/σ_i are selected to represent the background model. Each pixel x is classified based on whether the Gaussian that best matches the data at x is considered part of the background.

The **Wallflower** [136] is a three-component system. The pixel-level component performs Wiener filtering to make probabilistic predictions of the expected background. The region-level component fills in homogeneous regions of foreground objects. Finally, the frame-level component detects sudden global changes.

The **Improved MOG** [144] minutely modifies the implementation of the traditional MOG [124] in several aspects, such as dealing with shadow removal, background update and background subtraction. This greatly improves the per-

formance of MOG.

The **LBP-P** [54] models each pixel as a group of adaptive LBP histograms [105], which are calculated over a circular region around the pixel. The background update is done with a similar procedure to that of the MOG [124], yet it uses the histogram intersection to compute the proximity between data distributions and two learning rates to control the speeds of adaptation and weighting separately.

The **MOG-MRF** [120] adopts and enhances the MOG [124] with several improvements from [136, 144]. They model the smoothness of the foreground and background with a Markov random field, thus attaining more complete foreground objects, whereas most statistical techniques treats the pixels in the image independently and disregards the fundamental concept of smoothness.

5.3 The ML-QLBP framework

5.3.1 Method overview

Inspired by the good performance of the QLBP features in interest region description (cf. Chapter 3), we introduce the Multi-layer QLBP (ML-QLBP) background modeling framework, which integrates the QLBP^{sym} and RGB color intensities to model the background scene in a coarse-to-fine manner. The proposed approach performs robustly against challenging conditions, such as sudden illumination change and dynamic background, and keeps pace with several baseline and state-of-the-art methods. This success lies in the two following properties.

1. The background is modeled with two processing layers of the framework. The block-wise layer manages the distributions of local textures at blocks of pixels, while the pixel-wise layer deals with the distributions of normalized RGB color intensities at every pixel. During the subtraction process, the pixel blocks in the incoming frame are first matched with the texture distributions at corresponding locations and cell in the blocks are then classified into background/foreground. A background cell contains background pixels only, whereas a foreground cell includes both types of pixels. Therefore, pixels in the mixed cells are further matched with color distributions to find their categories. By jointly processing the frame at block-wise and pixel-wise levels, global structures can be utilized to avoid a great amount of false positives due to the instability of color, while the fineness of details

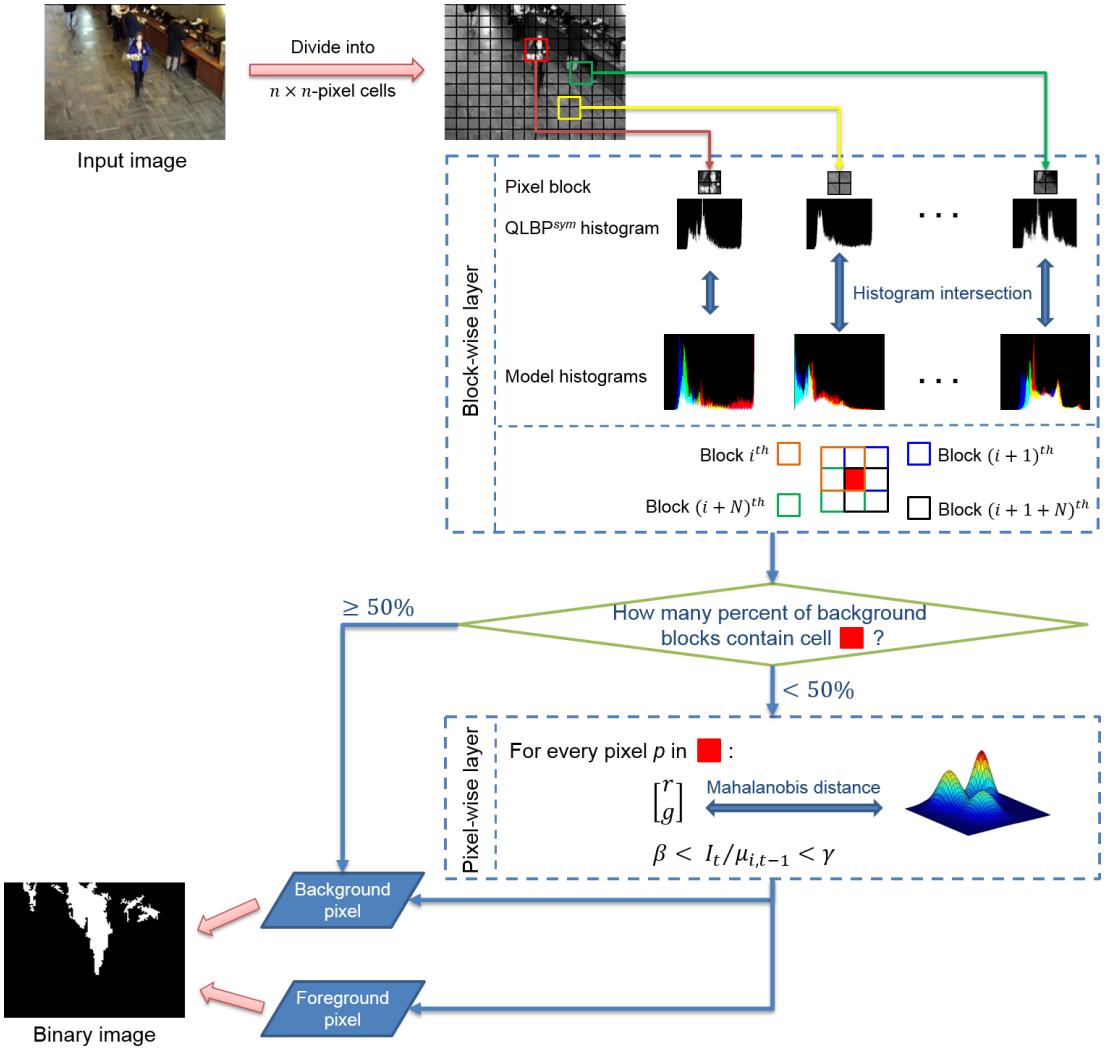


Figure 5.4: The proposed multi-layer background subtraction framework. The update is done at every blocks with the block-wise layer and at every pixel of the frame with the pixel-wise layer. The classification of a pixel into background/foreground depends on the type of the cell to which it belongs.

are maintained thanks to the manipulation on individual pixels.

2. We adopt the QLBP^{sym} feature to describe the local textures in a pixel block. It effectively captures the intensity changes in different orientations around the considered pixel and uses an adaptive thresholding scheme to cope with noises and local illumination changes. Therefore, it attains a better discriminative power compared to that of other LBP features. In addition, because the feature encoding only operates on four neighboring points, the operator is very computationally efficient and suitable for real-time applications like background subtraction.

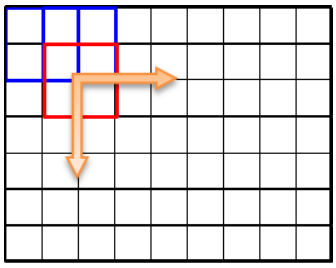


Figure 5.5: Creating partially overlapping blocks of pixels from 2×2 cells using a sliding window.

5.3.2 Framework construction

The proposed multi-layer framework models the background scene by observing the behaviors of QLBP^{sym} textures at blocks of pixels and colors at individual pixels. The block-wise layer maintains a mixture of QLBP^{sym} histograms at each block and updates them in a similar manner to that of [54]. Meanwhile, the pixel-wise layer deals with the normalized RGB intensities following the multivariate MOG [124]. In this way, both global and local structures in the background can be well preserved. Figure 5.4 describes the pipeline of our framework.

5.3.2.1 The block-wise layer with QLBP^{sym} features

The incoming frame is first divided into cells of $n \times n$ pixels and then a sliding window of 2×2 cells slides along two dimensions of the frame (one cell per move) to create overlapping blocks (cf. Fig.5.5). The use of partially overlapping blocks enables more accurate extraction of objects than in the case of non-overlapping.

In the block-wise layer, the textures at each block location are characterized by K average QLBP^{sym} histograms. K is selected by users, which is usually from 3 to 5 [124]. To compute the $QLBP^{sym}$ histogram, we follow the proposed procedure for the IO-QLBP descriptors in Chapter 3. Intensities of pixels within the block are sorted in non-descending order and the block is partitioned into S segments according to the overall intensity order ($S = 2$ in our experiments). $QLBP^{sym}$ histogram are computed and concatenated across segments. The model histogram i^{th} has an associated weight ω_i so that the sum of weights is equal to 1, i.e. $\Sigma_{i=1}^{K}\omega_i = 1$. The larger weight indicates a more present texture distribution.

Similar to the MOG [124], we sort the model histograms in decreasing order

according to ω_i and select the first B histograms to represent the background:

$$B = \arg\min_b \left(\sum_{i=1}^{b} \omega_i > T_B \right) \tag{5.10}$$

where T_B determines the minimum fraction of data contributing to the background model.

Let h_t denote the QLBP^{sym} histogram computed at the given block in an incoming frame $f(t)$. h_t is compared with the current K model histograms at the corresponding location using the histogram intersection:

$$\cap(h_t, \mu_{i,t-1}) = \sum_{j=1}^{M} \min(h_t, \mu_{i_j,t-1}) \tag{5.11}$$

where $\mu_{i,t-1}$ is the i^{th} model histogram and M is the number of histogram bins. The histogram intersection measures the common part of two histograms with simple operations, thus it is suitable for high dimensional features like LBP.

When h_t matches the k^{th} model histogram, i.e. $\cap(h_t, \mu_{k,t-1}) \geq T_H$ where T_H is a user-defined threshold, if the model histogram is part of the background, the pixel block is identified as background. Otherwise, it is a foreground block (a mixture of background and foreground pixels). The parameters of the k^{th} model are updated as follows:

$$\mu_{k,t} = \alpha_b h_t + (1 - \alpha_b)\mu_{k,t-1} \tag{5.12}$$

where α_b is the learning rate controlling the adaptation speed of the background model ($\alpha_b \in [0, 1]$). The bigger the learning rate, the faster the adaptation is.

The weights of all model histograms are also updated:

$$\omega_{i,t} = (1 - \alpha)\omega_{i,t-1} + \alpha M_{i,t} + \beta, \quad M_{i,t} = \begin{cases} 1 & \text{if } i = k \\ 0 & \text{else} \end{cases} \tag{5.13}$$

where α_w controls the speed of updating weights, β is a small adjusted constant [171] and k is the index of the matched Gaussian.

If no match is found with any of the K model histogram, the pixel block is classified as foreground. The least probable model histogram k^{th} is replaced with

a new one:

$$\mu_{k,t} = h_t \tag{5.14}$$

$$\omega_{k,t} = \text{low prior weight} \tag{5.15}$$

5.3.2.2 The pixel-wise layer with normalized RGB color

In this layer, we characterize each pixel by its normalized RGB color intensities. The normalized RGB color space is used rather than the original RGB because of its advantages to minor illumination changes like shadows [38, 120, 144]. The normalized chromaticity coordinates can be written as:

$$\begin{bmatrix} r \\ g \\ b \end{bmatrix} = \frac{1}{R+G+B} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{5.16}$$

We follow [120] to define the new color space (r, g, I), where $I = (R+G+B)$. All channels are scaled to [0, 255]. In this color space, shadows or highlights are expected to alter the intensities only and therefore, to avoid them, I should follow the criterion $\beta < \frac{I_t}{I_{t-1}} < \gamma$, where β and γ are empirically set to 0.6 and 1.5, respectively

The behaviors of pixels through frames are modeled using the conventional MOG approach [124], in which a mixture of K (r, g) color distributions are maintained at each pixel location. In the context of multimodal, the above criterion of I becomes $\beta < \frac{I_t}{\mu_{i,t-1}} < \gamma$ where $\mu_{i,t-1}$ is the mean of the i^{th} Gaussian. Therefore, a match should satisfy both criteria, including the Mahalanobis distance is below a threshold and I is in the valid range.

Since the normalized color (r or g) is very noisy when the intensity is low, we adopt the strategy in [144] that uses two color spaces alternatively according to the values of I. That is, when $I < I_{td}$, the (R, G, I) color space is used, while the (r, g, I) color space is for $I \geq I_{td}$, where I_{td} is a user-defined threshold.

5.3.2.3 The pipeline of the multi-layer framework

The background subtraction pipeline can be described as follows (cf. Fig.5.4):

- Input: The RGB-formatted color image is first converted into the grayscale image and then divided into multiple overlapping blocks of 2×2 cells, each cells has $n \times n$ pixels.

- The block-wise layer calculates the QLBP histogram for each block and updates the corresponding background models following the mechanism presented in Sect.5.3.2.1. If a block has $\cap(h_t, \mu_{i,t-1}) < T_H$ with all K model histograms, it is identified as a foreground block. Otherwise, it is a background block.
- Cells in the blocks are further classified into background/foreground using a voting strategy. If 50% of the blocks (or more) to which the cell belongs are background, the cell is identified as background.
- The pixel-wise layer updates the (r, g) color distributions of all pixels in the image following the mechanism presented in Sect.5.3.2.2. Pixels in the background cell contribute directly to the background region in the output while those in the foreground cell are further matched with corresponding Gaussian color models to determine their categories.
- Output: the process ends when every pixel in the input image is classified and an output binary image is created with black background pixels and white foreground pixels.

5.4 Performance evaluation

5.4.1 Evalution protocol

In order to provide a comparative evaluation with other competing methods, we evaluate the results with the two following quantities:

1. False negatives (FN): the number of foreground pixels incorrectly detected as background.
2. False positives (FP): the number of background pixels incorrectly detected as foreground.

5.4.2 Parameter selection

The frame is divided into cells of 10×10 pixels and blocks of 2×2 cells are established so that they overlap one another one cell in each dimension.

In the block-wise layer, each block is characterized by at most $K = 5$ model histograms. We empirically select the learning rate $\alpha_b = 0.01$ and the weight updating rate $\alpha_w = 0.01$. The incoming histogram is considered to match a model histogram if their histogram intersection is equal or above $T_H = 0.8$. B

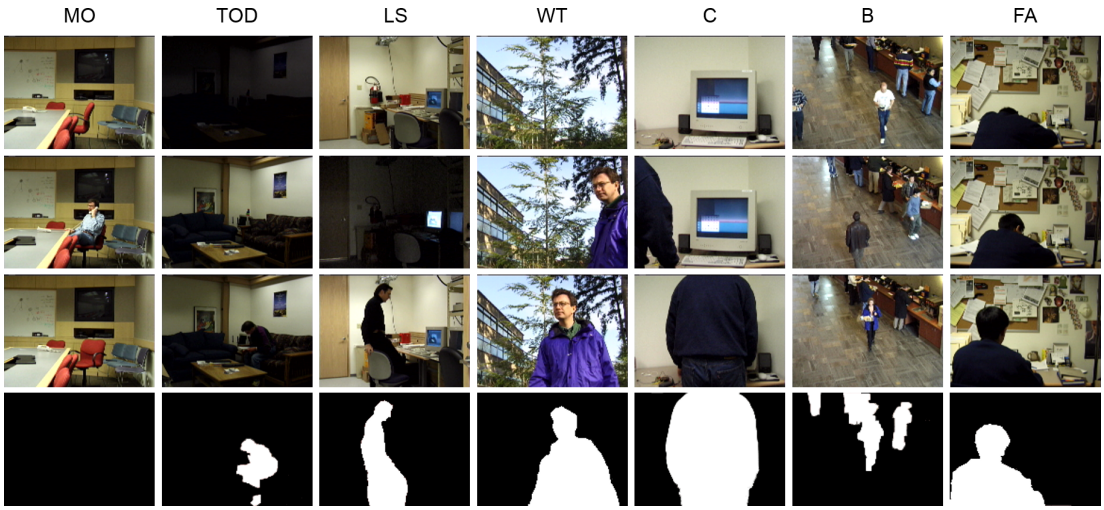


Figure 5.6: Sample images from the Wallflower dataset. Each scenario corresponds to a column in the figure. From top to bottom: two frames of the sequence, the test image, and the ground truth.

first model histograms that satisfy Eq.5.10 with $T_B = 0.7$ are selected to represent the background model. The QLBP^{sym} operator has a radius of four pixels. Each cell is partitioned into three segments according to the intensity order, from which QLBP^{sym} histograms are computed and concatenated.

In the pixel-wise layer, each pixel is characterized by at most $K = 5$ Gaussian distributions of normalized RGB color intensities. The learning rate α is 0.001. The variance $\sigma_{r,g}$ is between 2 and 15. To handle sudden illumination changes, we adopted the strategy in [136] to boost the learning rate to 0.1 when the number of foreground pixels are over 70% the total number of pixels in the frame and then gradually reducing it back to the original values after some frames.

5.4.3 The Wallflower dataset

We evaluate the performance of our proposed framework using the Wallflower benchmark [136]. The dataset consists of seven image sequences, in which each sequence presents a different type of difficulty that a practical background modeling system may encounter. Therefore, it has been widely used for assessing background subtraction methods. The sequences in Wallflower benchmark are: Moved Object (MO), Time of Day (TOD), Light Switch (LS), Waving Trees (WT), Camouflage (C), Bootstrap (B) and Foreground Aperture (FA) (cf.Fig.5.6). Each image sequence is stored at a resolution of 160 × 120 pixels and contains one manually segmented ground truth image for evaluation.

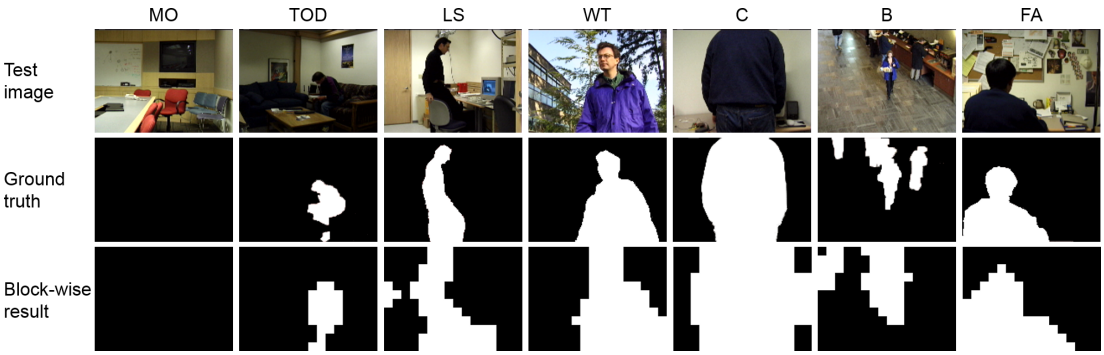


Figure 5.7: The performance of the block-wise layer on the Wallflower benchmark. First row: test frames of each image sequences. Second row: the hand-segmented ground truth. Third row: the block-wise segmentation result.

Table 5.1: The performance of the block-wise layer. The FN is the number of blocks wrongly labeled as background and the FP is the number of blocks wrongly labeled as foreground.

Sequence	**FN**	**FP**	**Total Error**
Moved Object (MO)	0	0	
Time Of Day (TOD)	1	10	183
Light Switch (LS)	0	34	
Waving Trees (WT)	0	29	(13.61%)
Camouflage (C)	8	29	
Bootstrap (B)	0	46	
Foreground Aperture (FA)	0	26	

5.4.4 Evaluation results

We first evaluate the performance of the standalone block-wise layer. Each image has 192 cells of 10×10 pixels. The layer attains an average error rate of 13.61% over seven sequences (cf. Table 5.1), thus it is robust enough the facilitate the subsequent pixel-wise layer to reduce the errors.

The proposed ML-QLBP framework is compared with several baseline and state-of-the-art background modeling algorithms, including the Mixture of Gaussians (MOG) [124], Wallflower [136], MOG with modifications (Improved MOG) [144], MOG with Markov Random Field for post-processing (MOG-MRF) [120], pixel-wise LBP histogram based approach (LBP-P) [54]. The methods in [120, 124, 144] share with the ML-QLBP the use of MOG, yet they only operate on the pixel level while ours considers both the block and pixel levels. The LBP-P computes histograms of traditional LBP features for each pixel, whereas the ML-QLBP collects QLBP features on blocks of pixels.

Figure 5.8: The qualitative evaluation on the Wallflower benchmark.

From Fig. 5.8, it can be observed that the proposed ML-QLBP and MOG-MRF are the two best methods in the sense that their results are most similar to the ground truth. The *Moved Object* sequence is simple and therefore all methods attain perfect results. Our framework performs better than the MOG-MRF in the *Time Of Day*, *Waving Trees* and *Camouflage*. This is thanks to the effectiveness of the block-wise layer so that background regions are identified accurately and thus prevents the appearance of false positives in these regions. Meanwhile, other methods encounter several false positives and negatives. In the *Foreground Aperture* case, all methods produce objects with holes inside except the MOG-MRF. It is due to the advantage of Markov Random Field (MRF) for post-processing. The MRF exploits the history of pixel activities to estimate the likelihood of background membership, thus gaining better results than that of conventional connected component algorithms that depend solely on the neighborhood in a single image. Although our framework uses only a simple contour-based method for post-processing, the error difference compared

to MOG-MRF is not significant (84 pixels). Therefore, it could be inferred that our framework is superior in adaption. This ability also contributes a good result in highly dynamic background such as *Bootstrap*, in which the proposed ML-QLBP is comparable to the MOG-MRF with a difference of 24 pixels. The *Light Switch* is the most challenging task for most of existing methods. To handle this case, a global change detection mechanism is necessary. Our method implements a similar mechanism to that of [136] to handle sudden illumination, whereas the traditional MOG and LBP-P do not use global detection and hence fail in this case. The ML-QLBP achieves a comparable result to that of MOG-MRG with a very minor difference of 7 pixels.

The quantitative comparison is presented in Table 5.2. Since the number of false positives and false negatives for each sequence was not provided in [54], the LBP-P is not included in the table. Table 5.2 shows FN and FP over seven sequences, total error - the sum of FN and FP (TE), and total error without Light Switch (TE*). Because the Light Switch causes a large number of errors that may distort the overall result, a total error without Light Switch is necessary to evaluate the performance on other challenges. Our method gives the lowest TE and TE* among five methods.

We used a standard PC with an Intel Core i7 950 CPU 2.8 GHz PC and achieved the frame rate of 150 fps was achieved. This makes the framework suitable for applications that require real-time processing. The frame rate is high enough to compensate for worse conditions (e.g. lower computer configuration, higher image resolution, etc.) so that the smooth motion is preserved.

5.5 Discussion and conclusion

In this study, we propose an efficient multi-layer background subtraction framework that combines pixel-wise and block-wise approaches. This framework is robust to various challenging scenarios thanks to the use of a simple yet highly discriminative LBP feature and the coarse-to-fine manner that helps to eliminate a great amount of false positives, which are commonly found in standalone approaches. It has been shown to be comparable to state-of-the-art methods, and therefore it is a promising solution for video processing applications like surveillance or video segmentation.

The limitation is that errors in the block-wise layer may greatly affect the performance of the pixel-wise layer. A false negative cell creates several of false

Table 5.2: The quantitative evaluation on the Wallflower benchmark.

		Sequence								
Algorithm	**Errors**	**MO**	**TOD**	**LS**	**WT**	**C**	**B**	**FA**	**TE**	**TE***
MOG	FN	0	1088	1633	1323	398	1874	2442	27053	11251
	FP	0	20	14169	341	3098	217	530		
Wallflower	FN	0	961	947	877	229	2025	320	11478	10156
	FP	0	25	375	1999	2706	365	649		
Improved MOG	FN	0	597	1481	44	106	1176	1274	6581	4431
	FP	0	358	669	288	413	134	41		
MOG-MRF	FN	0	47	204	**15**	16	**1060**	**34**	3808	3058
	FP	0	402	546	**311**	467	**102**	**604**		
Proposed method	FN	0	**378**	**252**	**40**	**76**	1049	215	**3593**	**2850**
	FP	0	**16**	**491**	**164**	**268**	137	507		

negative pixels while a false positive region is a burden to the pixel-wise MOG. Understanding the interaction between two layers and the feedback mechanism is our future work. In addition, achievements from the research of LBP in interest region description may improve the quality of the block-wise layer.

Chapter 6

Pedestrian Surveillance with Proposed Techniques

This chapter introduces a surveillance system to detect the presence of pedestrians in the monitored area. It is a unified framework based on the integration of the three proposed techniques in previous chapters, namely interest region description, background subtraction and pedestrian detection. It helps the administrators to improve their observation and sets up a solid foundation for the development of more sophisticated surveillance systems. The system works effectively on practical scenarios of different environmental conditions, encouraging its development into a real application in the future.

The content of the chapter is organized as follows. Section 6.1 defines the pedestrian surveillance task. The unified framework is described in Sect.6.2 and then evaluated in Sect.6.3. Finally, the last section gives some discussion points and plans for future work.

6.1 Pedestrian surveillance

Pedestrian surveillance is the monitoring of pedestrians, usually in terms of their presence and behaviors, for the purpose of managing the activities in certain places and protecting pedestrians from crimes and dangers. The surveillance systems are commonly found in public areas such as schools, shopping malls or subway stations, where a great number of people gather and hence there are also a lot of potential threats to the life safety and security. Figure 6.1 describes a surveillance system that uses eight cameras to observe the entering/leaving of visitors in a university campus. The surveillance task could be more complex,

Figure 6.1: A surveillance system monitors the presence of visitors in a university campus. It uses eight cameras installed at different location (the leftmost figure) to capture information at multiple viewpoints (View 1-8). Images are obtained from the PETS 2009 dataset [6].

(a) Grand Central Terminal

(b) Collective dynamics patterns

Figure 6.2: A surveillance system can further discover interesting collective dynamics of pedestrians to help the manager to get insight into the activities in the station [168].

such as collecting the motion paths of pedestrians and analyzing the collective dynamic patterns to identify the traffic flows in the monitored area (cf. Fig.6.2) [168].

The surveillance is a long-term and intensive task because it is usually carried out in a 24/7 manner, i.e. without interruption regardless of time or day. In addition, most systems nowadays still require great manual labor to manipulate multiple cameras, perhaps up to twenty or thirty devices. Therefore, there is a high possibility of missing important events due to the working overload. Modern surveillance systems are heading towards automation by utilizing the advanced of computer vision technologies so that the monitoring could be done more accurately and less depends on the human effort, which is limited and unstable.

An automated system typically follows the pipeline in Fig.6.3. It consecutively receives video frames transmitted from a CCTV camera and processes them in three phases, which demonstrate the increasing levels of automation. It first detects and localizes pedestrians appearing in the monitored scene. Instances of the same subject are then matched through frames to track the trajectories of

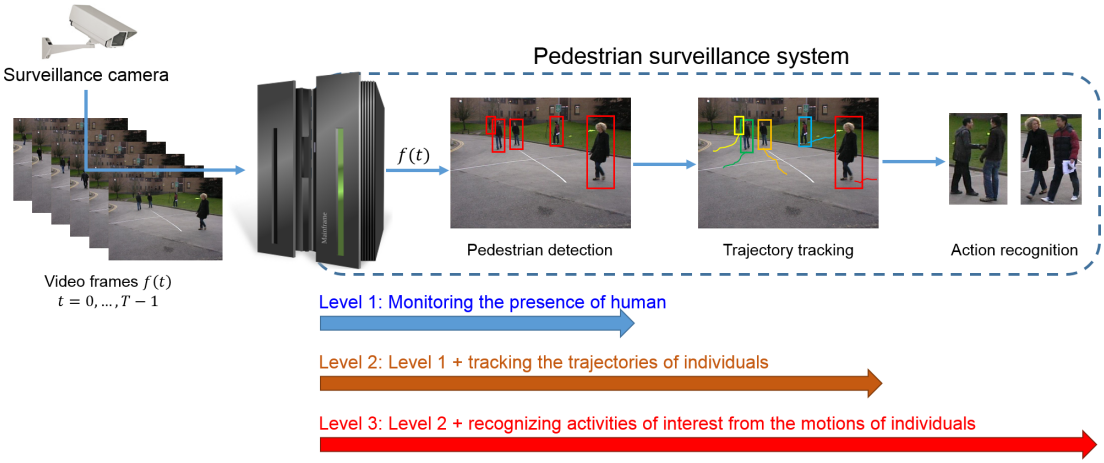


Figure 6.3: The pipeline of a surveillance system typically consists of three phases, demonstrating increasing levels of automation.

different pedestrians. Finally, activities of interest such as abnormal running or crowd formation are recognized based on the temporal information of trajectory and object shape deformation. The pipeline can be extended to multiple input sources by using many cameras, as illustrated in Fig.6.1. This can improve the surveillance quality significantly but needs advanced hardware technologies and algorithms to manage the process. In practice, a surveillance system not necessarily follows all phases but varies according to the level of monitoring. For systems that monitor the presence of human to investigate the population or prohibit intrusion, correctly detecting the movements is essential. Marketing analysts may further demand the trajectories of customers in a department store to understand the shopping habits. The system in public places like parking lots and subway stations are most complex since they have to perform all phases to identify criminal suspects or dangerous events.

From the viewpoint of computer vision, the surveillance task is completely not trivial because it involves several sub-fields to build different parts of the system. These research topics may include the background modeling and pedestrian detection for the detection phase, the camera topology estimation, re-identification and trajectory grouping for the tracking phase and finally the machine learning for action recognition. These techniques all require deep knowledge of vision properties to select the most appropriate features. In addition, many difficulties come from external factors, such as complex background scenes, crowded population, wide variations of pedestrian appearance and behaviors.

This dissertation introduces a pedestrian surveillance system that implements

the first level of automation (cf. Fig.6.3) using the three proposed techniques in previous chapters. Moving objects are first detected using the proposed background modeling framework in Chapter 5 and then classified into pedestrians/non-pedestrians using the pedestrian detector in Chapter 4. The QLBP and QLBP^{sym} in Chapter 3 play the key role in these techniques, allowing the surveillance to be accurate with an acceptable frame rate of 15.1 fps. Since most CCTV cameras have limited field-of-views, using a single camera cannot recognize objects and activities at the border. We therefore introduce an extension of the proposed system with multiple cameras. The system creates a panorama image from frames of different cameras and maps the background subtraction results on this image. The PLBP feature in Chapter 3 is used to enhance the feature alignment during image stitching. In this way, the shapes of individuals are revealed more completely and their motions are joined seamlessly, boosting the effectiveness of subsequent processes, i.e. trajectory tracking and action recognition.

6.2 The proposed unified framework

We introduce a surveillance system that automates the detection and localization of pedestrians in the monitored scene. This system is capable of observing the entering/leaving of visitors in public places, e.g. school ground or office hall, or detecting potential intruders in prohibited areas. Therefore, it is suitable for users who want to balance the workload between machines and administrators and those who have limited budgets. Two versions of the proposed system for a single camera and multiple cameras are described in the next following subsections, introducing alternative solutions to different practical demands.

6.2.1 Single-view surveillance system

The proposed system receives input video frames consecutively from a stationary CCTV camera. It performs ideally on the D1 Resolution (704 x 480 pixels), which is available in most cameras even those with low prices. At the time t, pixels in the frame $f(t)$ are distinguished into foreground/background pixels using the background modeling algorithm in Chapter 5. Connected foreground pixels are grouped into blobs $\{m_i\}_{i=1}^{N}$, which demonstrate image regions with temporal changes. These blobs mostly include pedestrians, yet they sometimes contain birds, cars and background objects (e.g. water fountain or waving tree

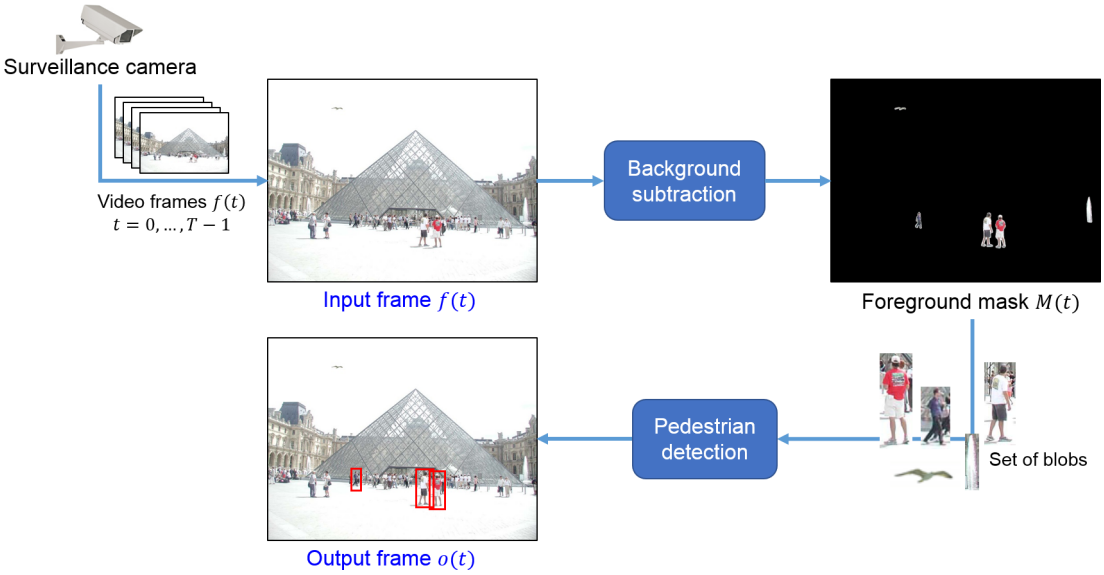


Figure 6.4: The pipeline of the proposed surveillance system with a single camera.

branches) since they also greatly alter pixel values. The system is not interested in movements of non-pedestrian objects, thus corresponding blobs are filtered using the ACF-QLBP pedestrian detector in Chapter 4. Finally, the pedestrians are localized by rectangular bounding boxes in the output frame $o(t)$. Figure 6.4 shows the overall pipeline of the proposed system. Since the effectiveness of each individual technique has been already verified through standard evaluations in previous chapters, it is reasonable to expect high performance from this integration. Experiments in Sect.6.3 show that our system can perform robustly on practical scenarios of different environmental conditions with the speed of 15.1 fps. This frame rate ensures smooth motion for the human vision and the feasibility for most surveillance applications (except those in casinos or banks that require values of more than 30 fps).

6.2.2 Multi-view surveillance system

In wide places such as school ground or building hall, a single camera is insufficient because of the finite sensor field-of-view. A common technical solution is to install multiple cameras at different locations to capture fragments of the scene. However, the administrator may meet several difficulties to track a target traveling from one view to another view since the human vision needs time to adapt to new scene structures. The panorama surveillance camera resolves the problem with 180^o or 360^o images. This new technology is promising but quite

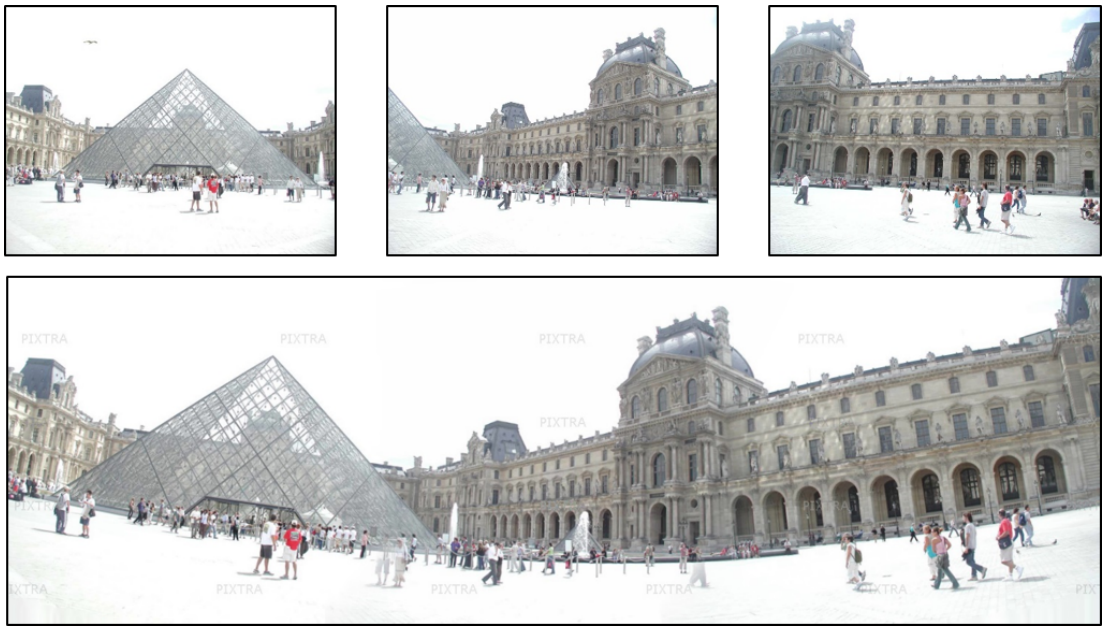


Figure 6.5: An example of creating a panorama image (the bottom row) from three different views (the top row). Images are obtained from the Pixtra PanoStitcher software (http://www.pixtra.com/).

expensive. The current price of a panorama camera is usually at least four times as high as that of a simple camera.

We utilize the advances of computer vision research to fuse the ideas of multiple cameras and panorama image into a more feasible solution. Specifically, the proposed system extends its working environment from a single camera to multiple cameras, stitches a panorama image from all views and performs the detection on the new common view. It is able to deal with standard cameras, thus less technology dependent and possibly more economical. To capture a panorama scene, the cameras are intentionally arranged such that two adjacent devices share at least 15-30% view overlap. An example of panorama image is shown in Fig.6.5. We use some modern image stitching method (e.g. [21]) to composite the panorama scene from frames of multiple views. The feature alignment is enhanced with the PLBP feature in Chapter 3, which has been shown to be faster and more robust than most local features, such as SIFT, DAISY or MORGH. The proposed system does background subtraction separately on each view then maps the resulting moving blobs onto the same panorama image. Finally, the pedestrian detector identifies and localizes pedestrians with bounding boxes. Since then, the trajectory tracking and action recognition can be done on the panorama view. Figure 6.6 outlines the pipeline of our multi-view system.

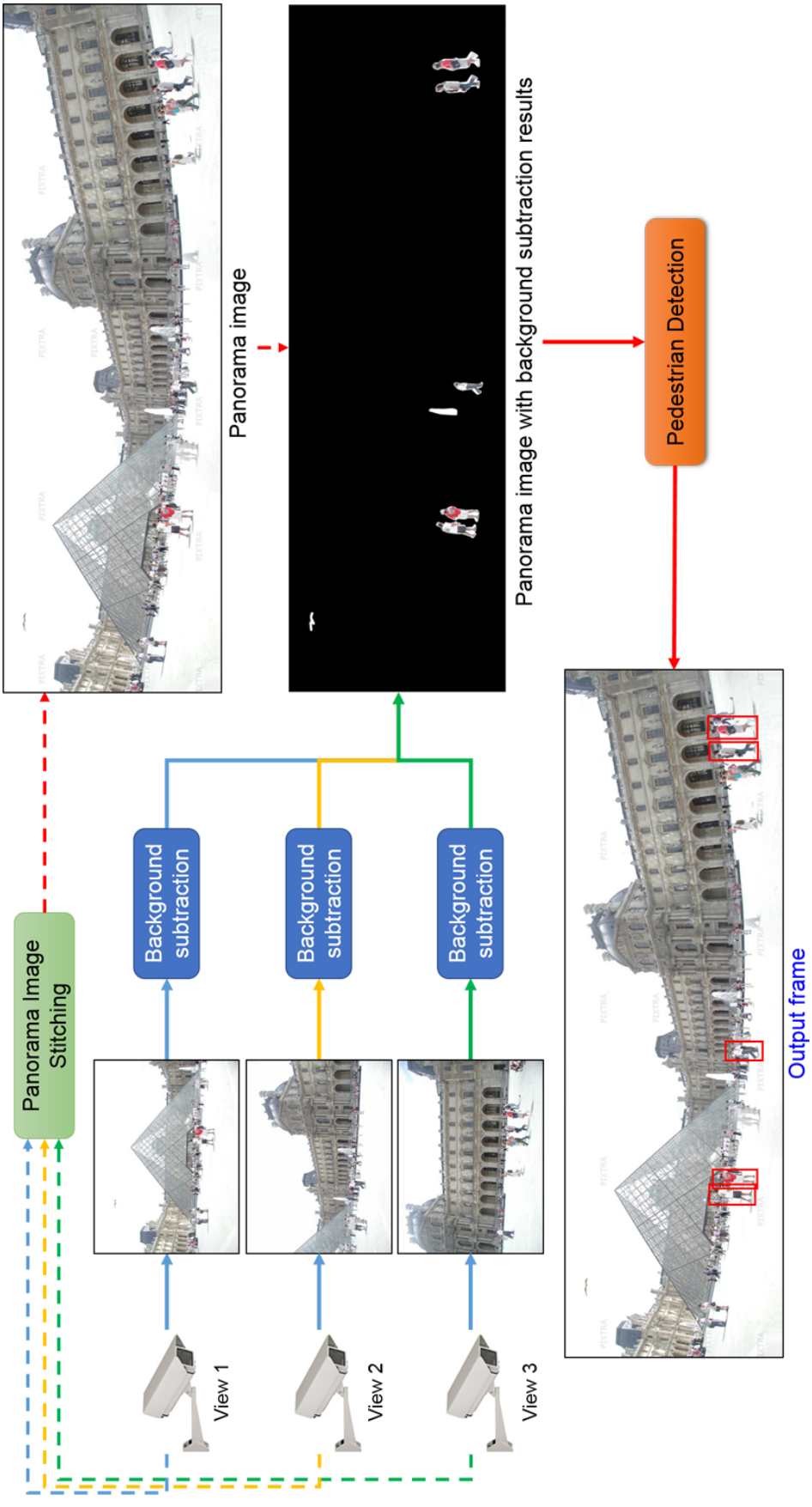
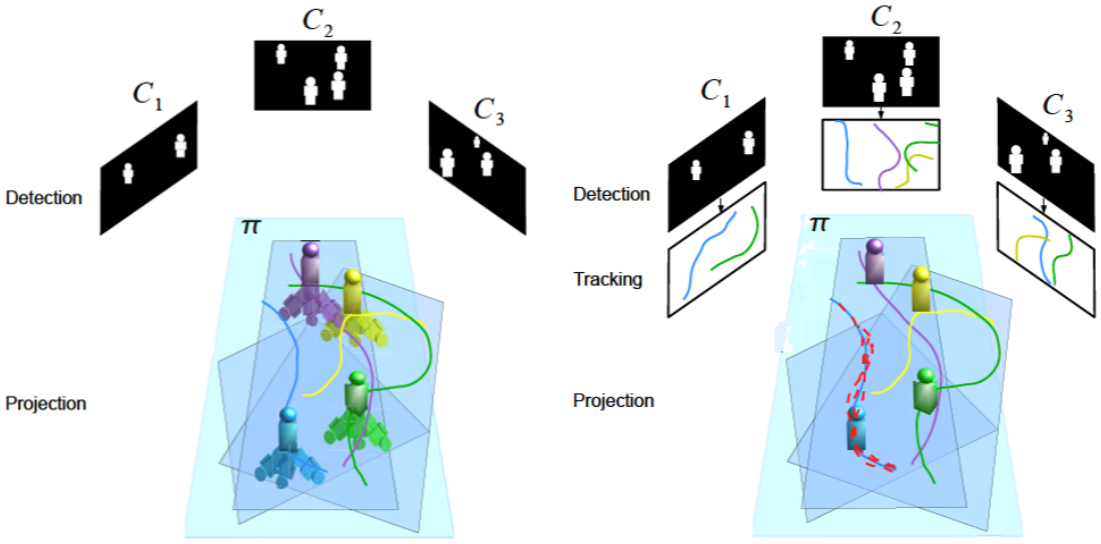


Figure 6.6: The proposed system is extended to multiple cameras (e.g. three cameras). It projects the background subtraction results from individual views onto the panorama image prior to the detection. Trajectory tracking and action recognition are also done on this panorama view.



(a) Fuse-first (b) Track-first

Figure 6.7: The fuse-first and track-first mechanisms. Images are obtained from [126].

It is worth noting that typical image stitching methods are too heavy to be invoked at every frame. However, we can address the problem easily with the following suggestions. First, since the panorama image mainly contains the background details (foreground regions are mapped later from separate views), it is reasonable to update it partially or with a longer interval. Second, several robust stitching methods are available. They improve the core algorithm or adopt GPU [80, 142] to boost the processing speed up to 30 fps or more. Finally, the position and viewing angle of a stationary surveillance camera are fixed, allowing us to do some pre-calculation of the scene structure. These modifications greatly reduce the computation cost for stitching.

The proposed multi-view system resembles the fusion-fist mechanism in the multi-view tracking literature, i.e. the detection information is projected onto a common view prior to the tracking. It is the reversal of the track-first mechanism, which performs everything on each view then projects and links the resulting information on other views [126]. Figure 6.7 contrasts the two mechanisms. The fusion-fist approach is computationally simpler, though a higher data transfer load is required. Most methods in this category build a hypothesis common view by mapping all views to a top view or constructing a ground plane occupancy map. This involves sophisticated mappings and the hypothesis view is obscure to most users. Our solution benefits from the modern vision techniques to operate directly on the panorama view. There are some attempts to do the tracking

Sequence	Camera model	Resolution	Frame rate
S1.L1.001	Axis 223M	768×576	~ 7
S1.L1.005	Sony DCR-PC1000E 3xCMOS	720×576	~ 7
S1.L1.008	Canon MV-1 1xCCD w	720×576	~ 7

Table 6.1: The details of selected sequences. The second column shows the camera models used to film the videos.

on panorama images, yet they focus on using panorama cameras [24, 133]. The proposed system, on the other hand, less depends on the hardware technologies.

6.3 Performance evaluation

6.3.1 Evaluation data

The evaluation is conducted on three video sequences of the Performance Evaluation of Tracking and Surveillance Contest (PETS) 2009 [6]. The videos are recorded in the campus of the University of Reading (UK) at different viewpoints and time stamps in the day, demonstrating practical scenarios of surveillance. The PETS 2009 dataset supports several highly diversified and challenging sequences, for both single and multiple cameras, thus it is widely used for evaluating pedestrian detection and tracking algorithms.

Each sequence includes 795 frames, some of which are shown in conjunction with the detection results in Fig.6.8-6.10. More details are described in Table 6.1. We build the background models using the training data provided in the dataset.

6.3.2 Evaluation results

We evaluate the performance of the proposed single-view system while leaving the evaluation of the multi-view system for future work. This is because the multi-view system extends the working environment from a single camera to the multiple cameras, yet it preserves all related techniques, namely the QLBP features, the ML-QLBP background modeling algorithm and the ACF-QLBP pedestrian detector. In addition, different from other multi-view fuse-first methods that depend on the quality of the projection onto a complex hypothesis view, the panorama image only alters the scene structure moderately and hence has little effect on the detection and tracking steps. Therefore, the effectiveness in the single-view case is sufficient to draw a preliminary conclusion for the multi-view case.

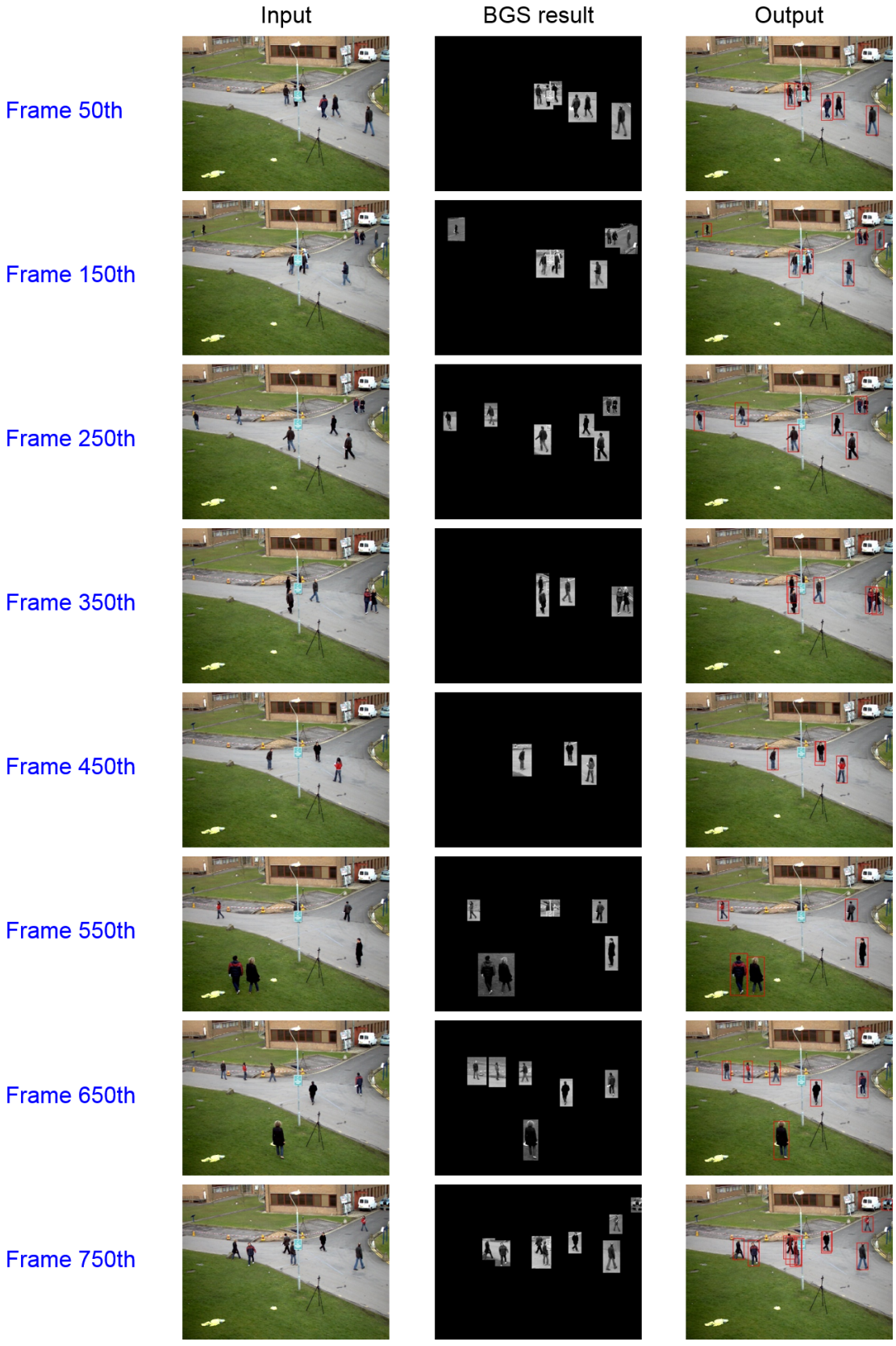


Figure 6.8: The detection result on the S2.L1.001 sequence. Detected pedestrians are marked by red bounding boxes.

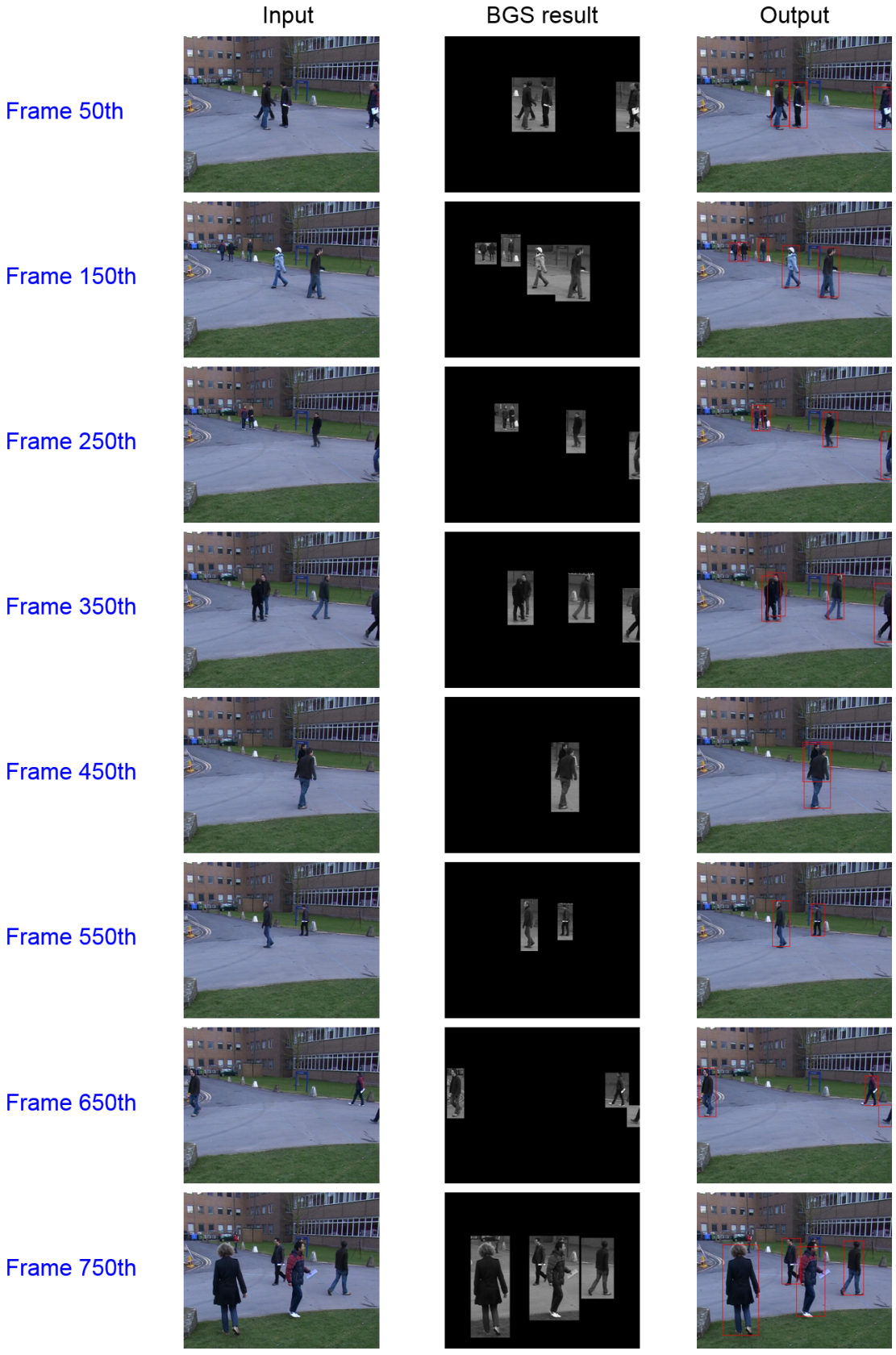
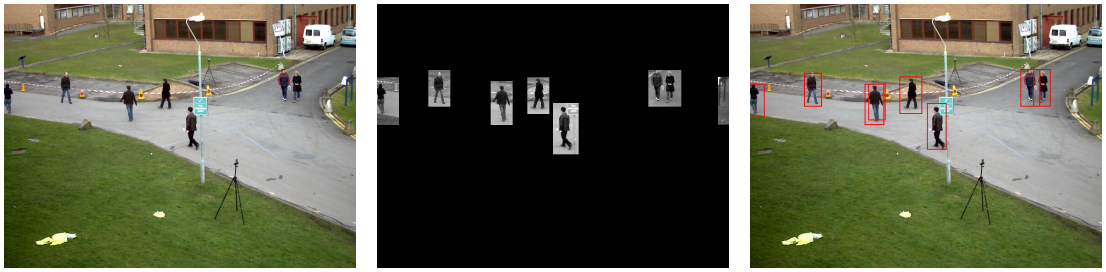


Figure 6.9: The detection result on the S2.L1.005 sequence. Detected pedestrians are marked by red bounding boxes.

Figure 6.10: The detection result on the S2.L1.008 sequence. Detected pedestrians are marked by red bounding boxes.



(a) Input frame 283th (b) BGS result (c) Output frame

Figure 6.11: An example of false positive in the background subtraction phase. The rightmost blob in subfigure (b) is a false positive. It is rejected in the pedestrian detection phase. The background model gradually reduces the error until the frame 288th.

Figure 6.8-6.10 present the detection results on the three PETS 2009 video sequences. For each sequence, we select eight frames at equal intervals to demonstrate the surveillance progress and the intermediate background subtraction results are also presented for comparison.

The background subtraction module performs well on all sequences despite the large difference of lighting conditions between sequences. This is thanks to the robustness of the QLBP feature against illumination changes, making the system more feasible to the wide variations of the outdoor environment. This phase aims to extract moving blobs for the pedestrian detection, thus accurate object contours are unnecessary. We therefore modify the background modeling algorithm so that the subtraction is done coarsely on the block-wise level to obtain the bounding boxes of the moving blobs. This allows the process to attain a frame rate of $\sim 50 fps$ in this phase. False positives and false negatives sometimes appear, yet they are soon alleviated in subsequent frames by the adaptive mechanism of MOGs. The false positives have little effect on the final results because the pedestrian detector will recognize that there is no pedestrian in these regions and hence they will be rejected (cf. Fig.6.11). Meanwhile, the false negatives are more severe because the pedestrian detector only performs on the moving blobs. However, they rarely appear because the background model is customized to be highly sensitive to the changes of feature in the blocks.

The ACF-QLBP pedestrian detection operates on the moving blobs provides by the previous phase, instead of on the whole frame. Because the foreground usually accounts for smaller portion of the image than that of the background, this mechanism can save a considerable amount of computation cost, allowing

the process to run at $\sim$ 22.6 fps, which is much faster than the full-image mode (15.2 fps). As described in Chapter 4, our detector is trained on the INRIA benchmark [31]. The benchmark shares no scene and subject with the PETS dataset, except the ratio between the subject and the image size. Nevertheless, the ACF-QLBP successfully identifies pedestrians of various sizes and poses in all sequences, proving its robustness and generality. Although it encounters some false detections and incomplete localization (too small or too large bounding boxes) due to occlusion or feature instability, these errors are negligible. We consider the scene context to alleviate the false negatives (miss detections) in a heuristic manner. In a video sequence, the pedestrian should move smoothly and continuously until s/he arrives the exit. Therefore, when the detector finds no pedestrian in a moving blob B, the system will search for with-pedestrian bounding boxes in the moving blobs that overlap B in previous frames. If there is such a bounding box, there is a high possibility that a pedestrian appears in B and thus a new bounding box is created with a certain shift to demonstrate the movement of the pedestrian. In this way, we incorporate the temporal information to improve the performance of the single-frame detector.

In order to evaluate the overall performance of the system, we present all locations of pedestrians through frames onto a single map and compare them with the corresponding manual labeled ground truth. Let $[x, y, w, h]$ denote the bounding box covering a pedestrian, where (x, y) is the coordinate of the top-left corner, w and h are the width and height, respectively. The location of the pedestrian is represented by a dot at $(x + w/2, y)$. As seen from Fig.6.12-6.14, the dot maps generated by our data and the ground truth data are very similar to each other. This demonstrates the effectiveness and stability of the proposed system. Note that the dots between two maps cannot match exactly at the same location because the positions and sizes of the bounding boxes vary slightly. In addition, the ground truth is strictly labeled, i.e. occluded people are also counted, while our detector intrinsically cannot handle large occlusions (cf. Chapter 4). However, these differences do not alter the general correlation between two maps and therefore it is still possible to verify the correctness of the system.

The proposed system is recommended to run on a standard PC or more. Table 6.2 presents the elapsed time and frame rate for each phase of the process. Our system attains an average frame rate of 15.1 fps on an Intel Core i7 950 CPU 2.8 GHz PC, which satisfies the required speed for common surveillance applications.

Table 6.2: The elapsed time (seconds) and frame rate (fps) for each phase of the surveillance process (tested on a standard PC).

Sequence	Background subtraction		Pedestrian detection		Average FPS
	Time	FPS	Time	FPS	
S2.L1.001	17.61	45.13	36.16	21.98	14.78
S2.L1.005	17.28	46.00	35.30	22.52	15.12
S2.L1.008	17.03	46.68	34.77	22.86	15.35

Table 6.3: The elapsed time (seconds) and frame rate (fps) for each phase of the surveillance process (tested on a laptop).

Sequence	Background subtraction		Pedestrian detection		Average FPS
	Time	FPS	Time	FPS	
S2.L1.001	27.17	29.26	50.24	15.82	10.27
S2.L1.005	28.04	28.36	49.40	16.09	10.27
S2.L1.008	27.97	28.42	48.62	16.35	10.40

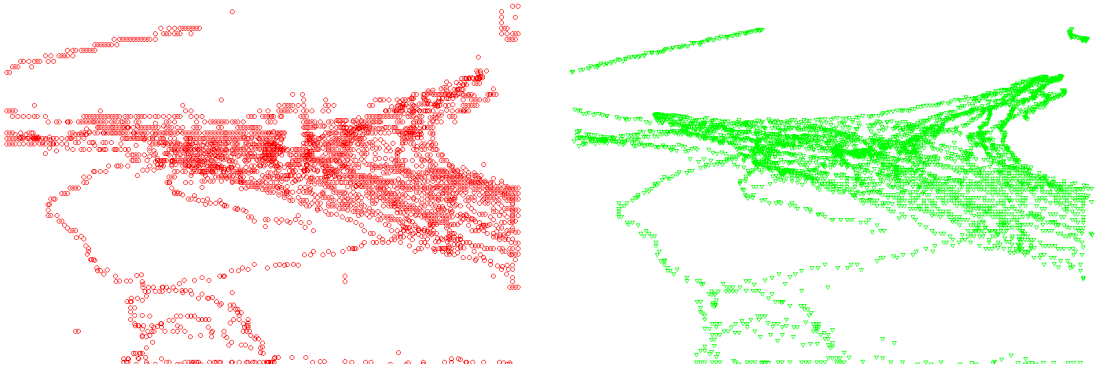
This encourages us to extend the proposed system to multi-view processing as well as developing it into a practical application in the future. We also run the system on a Intel Core i5-2430M CPU 2.4 GHz laptop (cf. Table 6.3). Although the speed is not promising as in the case of a PC, yet it is still in the feasible range of surveillance, i.e. from 5 to $\sim$ 30 fps.

6.4 Discussion and conclusion

We introduce an effective pedestrian surveillance system, which is unified from three proposed techniques in interest region description, background subtraction and pedestrian detection. It automates the detection of human, which is an essential step in the surveillance process. Therefore, it is suitable for applications like monitoring the entering/leaving of visitors in public places or detecting intruders in prohibited areas. The evaluation on standard video sequences have shown its good abilities. This success lies in the robustness of novel LBP features in every phase. In addition, the system attains an average frame rate of 16 fps on a standard PC, which is sufficient for the human vision and thus promising for common surveillance demands (usually in 5 $\sim$ 30 fps).

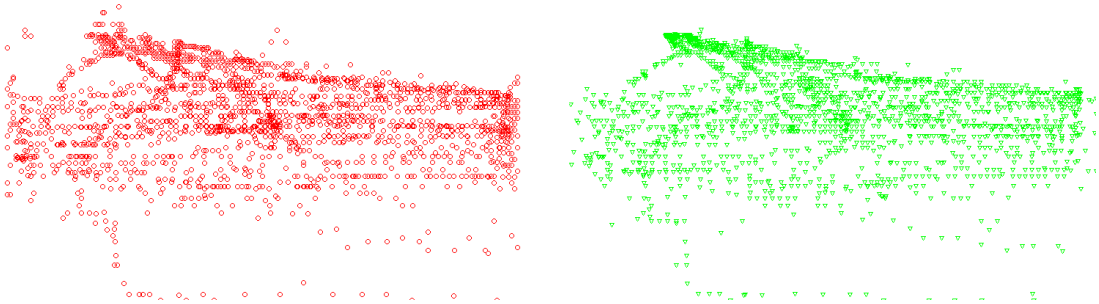
We note that the bottleneck of our system is the pedestrian detection, though changing the detection mechanism on the whole frame to on specific moving blobs

have improved the frame rate significantly (from 15.2 fps to 22.6 fps). In previous chapters, we have presented the future works to improve related techniques. Their achievements will contribute to the enhancement of this system. In addition, some code optimization will be able to speed up the process.



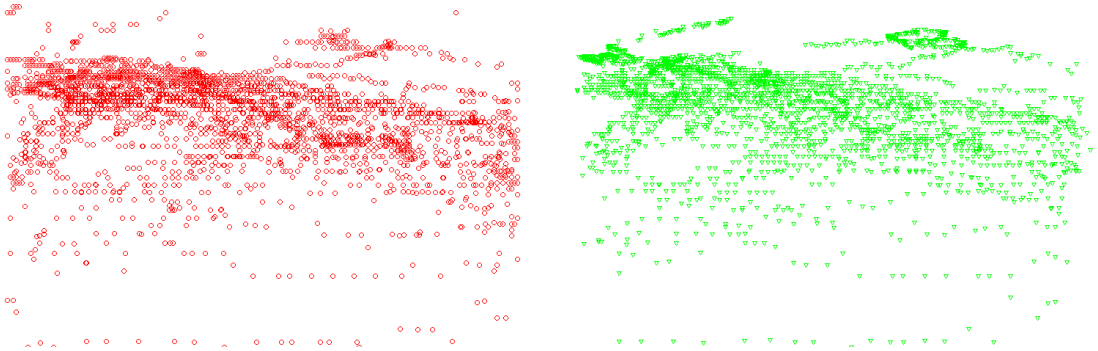
(a) Our data (b) Ground truth data

Figure 6.12: The dot maps represent all locations of pedestrians in the S2.L1.001 sequence, which are generated by our system (a) and the ground truth (b).



(a) Our data (b) Ground truth data

Figure 6.13: The dot maps represent all locations of pedestrians in the S2.L1.005 sequence, which are generated by our system (a) and the ground truth (b).



(a) Our data (b) Ground truth data

Figure 6.14: The dot maps represent all locations of pedestrians in the S2.L1.008 sequence, which are generated by our system (a) and the ground truth (b).

Chapter 7

Conclusion

This chapter presents the conclusions for the studies and analysis reported in the dissertation and describes future work. The first section summarizes achievements from the studies. The second section points out limitations that need to be addressed in future work. The final section discusses some prospects of this research in the future.

7.1 Summary

This dissertation has systematically studied the LBP features to attain an effective means of feature matching. The abilities of interpreting and distinguishing image properties are the key factors that enable computers to perform high-level tasks more similarly to human, such as identifying objects or finding the similarity and dissimilarity between objects. Therefore, the achievements in this dissertation could contribute to the development of visual perception for computers.

To fulfill its goal, the dissertation has analyzed the advantages and drawbacks of the LBP features in three major computer vision tasks, including the interest region description, pedestrian detection and background subtraction.

Chapter 3 introduces the IO-QLBP and MSR-PLBP descriptors for interest region description. They are able to match details between different images under challenging geometric and photometric transformations. The problems of matching accuracy and computational cost are simultaneously addressed though they are competing factors. Meanwhile, other descriptors usually encounter either of these issues. This success is constituted from two key factors. First, the computational simplicity enables them to be implemented easily for certain applications. Second, they are robust to several challenging image transformations

and noise. This results in their excellent performance in image matching and object recognition.

Chapter 4 presents the ACF-QLBP, a robust pedestrian detection built from the generalized QLBP and the advanced learning framework of [34]. The QLBP texture well characterizes edges along diagonal and vertical directions, which are identified as the most visible and essential cues of the human body. It is combined with three color channels and a gradient histogram to describe the target from different aspects, namely texture, color, and gradient changes in magnitude and orientation. In this way, the proposed detector attains high robustness against the wide variations of human poses while maintaining an acceptable frame rate.

Chapter 5 describes the ML-QLBP background modeling method. The block-wise layer examines the distributions of QLBP^{sym} textures in blocks of pixels to classify cells in the blocks into background/foreground while the pixel-wise layer further analyzes the obtained foreground cells with distributions of colors to identify true moving pixels. Its robustness and efficiency come from the combination of the block-wise and pixel-wise approaches into a unified framework. In addition, the QLBP^{sym} effectively captures global structures while strongly resisting to the effects of illumination change. These properties allow the system to reduce the errors better than approaches using pixel-wise or block-wise subtraction only. It is also comparable to several modern background modeling approaches at that time.

Chapter 6 integrates the three proposed techniques into a unified pedestrian surveillance system. It consecutively processes video frames transmitted from a stationary CCTV camera through two phases. First, the frame is compared with the background model to extracting moving blobs, which may contain pedestrians, birds, cars, and non-stationary background objects also. Second, the pedestrian detector classifies whether there are pedestrians in the moving blobs and localizes them by rectangular bounding boxes in the output frame. The system has shown good performance through evaluations on challenging scenarios, encouraging its development into a practice application in the future.

The effectiveness of the proposed techniques is comprehensively evaluated through several comparative studies with competing baseline and state-of-the-art approaches in each field. They are all shown to be on a par with their competitors in one or many aspects, such as matching accuracy, computational cost, or the balance of these two factors. This makes the novel LBP features prominent among a large variation of features in computer vision.

7.2 Limitations and future work

Despite of significant achievements, this dissertation still encounters several limitations and hence leaving space for further improvements. The following content discusses each limitation as well as possible solutions to overcome the issue.

Generality. The dissertation has provided two effective LBP features for three different major computer vision tasks. However, they need careful considerations to adjust their parameters and hence they cannot be applied without profound understanding of the nature of each problem. Texture analysis is the fundamental knowledge that may help us to create more general and innovative LBP encodings. In addition, the QLBP and PLBP should be extended for P neighbors instead of fixing them with four neighbors. Although the current settings are sufficient to compete with modern approaches, enhancing their generality may enable many interesting properties and thus open more opportunities.

High dimensionality. It is an intrinsic drawback of most LBP features because robustness and compactness are competing factors. Approaches using gradients usually provide compact feature vectors yet they are computationally heavy due to the computation of gradient orientation and magnitude. The LBP also records the magnitude of gray-level changes but ignores the quantization of orientations, thus it needs more elements in the feature vector to compensate the information loss. In the study of interest region description, gradient-based descriptors use eight orientations while our descriptors need 16 distinct binary codes to achieve the same level of robustness. In the pedestrian detection problem, a sliding window is characterize by features of 5120 dimensions in the ACF [34] and 6144 dimensions in the proposed ACF-QLBP, resulting a significant drop of frame rate in our method. Enhancing the encoding, computing LBP features on a more stable medium rather than gray values (e.g. Gabor images or DCT), or applying PCA to reduce the dimensions are promising suggestions.

Parameter tuning. The proposed approaches have some parameters that need tuning. For example, the scale factor τ in the QLBP (Eq.3.2) and the threshold T that is estimated from a Gaussian function of brightness contrast in the PLBP (Eq.3.14). Up to now, they are usually selected according to experiments or empirically selected following some subjective observations. The lack of a mathematic foundation causes difficulties in verifying their effectiveness in general cases. A deep analysis of image contrast or gray-level transform could be helpful to fill this gap.

7.3 Future prospects

To this end, the dissertation has provided good solutions to partially resolve the problem of representing visual data and enabling the visual perception in computers, yet there are still several issues worthy of note in the future.

The computer vision by nature should have close connections with cognitive sciences and behavioral sciences because it relates the human visual perception with the computational power of computers. However, we have seen very few benefits from these connections while most of the fundamental knowledge comes from pure sciences, mathematics, or fields based on mathematics like signal processing, image or control robotics. It does interact with artificial intelligence and machine learning in the sense that many algorithms in these fields are used for learning models or making decisions, which somewhat reflects the simplest abilities of reasoning and inference of humans. Nevertheless, a system that is able to imitate the full, or coarse, mechanism of receiving and processing the visual information in human has never been revealed in computer vision. Therefore, it is substantially necessary to improve the connections with sciences related to human vision, for example, the neurobiology, cognitive vision and biological vision. These fields study and model the physiological processes behind visual perception in humans. The computer vision, on the other hand, tries to describe the processes by implementation in software and hardware behind artificial vision systems. Interdisciplinary exchange between biological research and computer vision are therefore highly expected to nurture both fields.

There are several discussions about the quantum computer, which is a device that makes direct use of quantum-mechanical phenomena to perform operations on data. Large-scale quantum computers are able to solve certain problems more much quickly than any classical computer by using advanced algorithms like integer factorization using Shor's algorithm or the simulation of quantum many-body systems. While machine learning and artificial intelligence have just touched the surface of the "human intelligence" iceberg because of the incredible complexity of the human brain, the emergence of this technology offer us good opportunities to gain more insights. It allows modeling and testing several hypotheses with greater complexity, hence enabling the involvement of a large-scale visual analytics process in order to discover valuable knowledge.

Bibliography

[1] 53 objects database. URL http://www.vision.ee.ethz.ch/datasets/index.en.html

[2] Affine covariant features. URL http://www.robots.ox.ac.uk/~vgg/research/affine/

[3] Feature detector evaluation database. URL http://lear.inrialpes.fr/people/mikolajczyk/Database/

[4] Illumination dataset and liop binaries. URL http://vision.ia.ac.cn/Students/wzh/publication/liop/index.html

[5] Mrogh and mrrid binaries. URL http://www.sigvc.org/bfan/

[6] Pets 2009 dataset. URL http://www.cvg.rdg.ac.uk/PETS2009/a.html

[7] Recognition benchmark database. URL http://vis.uky.edu/~stewe/ukbench/

[8] Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S.M., Szeliski, R.: Building rome in a day. Commun. ACM **54**(10), 105–112 (2011)

[9] Ahonen, T., Hadid, A., Pietikäinen, M.: Face recognition with local binary patterns. In: Computer Vision - ECCV 2004, *Lecture Notes in Computer Science*, vol. 3021, pp. 469–481. Springer Berlin Heidelberg (2004)

[10] Ahonen, T., Hadid, A., Pietikäinen, M.: Face description with local binary patterns: Application to face recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **28**(12), 2037–2041 (2006)

[11] Ahonen, T., Matas, J., He, C., Pietikäinen, M.: Rotation invariant image description with local binary pattern histogram fourier features. In: Image Analysis, *Lecture Notes in Computer Science*, vol. 5575, pp. 61–70. Springer Berlin Heidelberg (2009)

[12] Ahonen, T., Pietikäinen, M.: Soft histograms for local binary patterns. In: Proceedings of Finnish Signal Processing Symposium (FINSIG) (2007)

[13] Anbarjafari, G.: Face recognition using color local binary pattern from mutually independent color channels. EURASIP Journal on Image and Video Processing **2013**(1), 1–11 (2013)

[14] Baf, F., Bouwmans, T., Vachon, B.: Type-2 fuzzy mixture of gaussians model: Application to background modeling. In: Proceedings of International Symposium on Advances in Visual Computing, ISVC '08, pp. 772–781. Springer-Verlag, Berlin, Heidelberg (2008)

[15] Bar-Hillel, A., Levi, D., Krupka, E., Goldberg, C.: Part-based feature synthesis for human detection. In: Computer Vision–ECCV 2010, vol. 6314, pp. 127–142. Springer (2010)

[16] Benenson, R., Mathias, M., Timofte, R., Van Gool, L.: Pedestrian detection at 100 frames per second. In: Proceedings of Computer Vision and Pattern Recognition (CVPR), pp. 2903–2910 (2012)

[17] Benenson, R., Mathias, M., Tuytelaars, T., Gool, L.V.: Seeking the strongest rigid detector. In: Proceedings of Computer Vision and Pattern Recognition (CVPR), pp. 3666–3673 (2013)

[18] Biswas, S., Sil, J., Sengupta, N.: Background modeling and implementation using discrete wavelet transform, a review. ICGST International Journal on Graphics, Vision and Image Processing, GVIP **11**, 29–42 (2011)

[19] Bouwmans, T.: Subspace learning for background modeling: A survey. Recent Patents on Computer Science **2**(3), 223–234 (2009)

[20] Bouwmans, T.: Recent advanced statistical background modeling for foreground detection-a systematic survey. Recent Patents on Computer Science **4**(3), 147–176 (2011)

[21] Brown, M., Lowe, D.G.: Automatic panoramic image stitching using invariant features. International Journal of Computer Vision **74**(1), 59–73 (2007)

[22] Bucak, S.S., Gnsel, B., Gursoy, O.: Incremental non-negative matrix factorization for dynamic background modelling. In: Proceedings of International Workshop on Pattern Recognition in Information Systems (PRIS), pp. 107–116 (2007)

[23] Butler, D., Sridharan, S., Bove V.M., J.: Real-time adaptive background segmentation. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol. 3, pp. III–349–52 vol.3 (2003)

[24] Chakravarty, P., Jarvis, R., et al.: People tracking from a moving panoramic camera. In: Australian Conference on Robotics and Automation (ACRA) (2008)

[25] Chan, C.H., Goswami, B., Kittler, J., Christmas, W.: Local ordinal contrast pattern histograms for spatiotemporal, lip-based speaker authentication. IEEE Transactions on Information Forensics and Security **7**(2), 602–612 (2012)

[26] Chang, R., Gandhi, T., Trivedi, M.: Vision modules for a multi-sensory bridge monitoring approach. In: Proceedings of IEEE International Conference on Intelligent Transportation Systems, pp. 971–976 (2004)

[27] Cheung, S.C.S., Kamath, C.: Robust techniques for background subtraction in urban traffic video. In: Proceedings of SPIE, vol. 5308, pp. 881–892 (2004)

[28] Cordes, K., Rosenhahn, B., Ostermann, J.: Increasing the accuracy of feature evaluation benchmarks using differential evolution. In: Proceedings of IEEE Symposium on Differential Evolution (SDE), pp. 1–8 (2011)

[29] Cristani, M., Farenzena, M., Bloisi, D., Murino, V.: Background subtraction for automated multisensor surveillance: a comprehensive review. EURASIP Journal on Advances in signal Processing **2010**, 43 (2010)

[30] Culibrk, D., Marques, O., Socek, D., Kalva, H., Furht, B.: Neural network approach to background modeling for video object segmentation. IEEE Transactions on Neural Networks **18**(6), 1614–1627 (2007)

[31] Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proceedings of Computer Vision and Pattern Recognition (CVPR), vol. 1, pp. 886–893 vol. 1 (2005)

[32] D'Angelo, A., Dugelay, J.L.: Color based soft biometry for hooligans detection. In: Proceedings of International Symposium on Circuits and Systems (ISCAS), pp. 1691–1694 (2010)

[33] Dollár, P.: Caltech pedestrian detection benchmark. URL http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/

[34] Dollár, P., Appel, R., Belongie, S., Perona, P.: Fast feature pyramids for object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence p. To appear (2014)

[35] Dollár, P., Appel, R., Kienzle, W.: Crosstalk cascades for frame-rate pedestrian detection. In: Proceedings of European Conference on Computer Vision (ECCV), ECCV'12, pp. 645–659 (2012)

[36] Dollár, P., Belongie, S., Perona, P.: The fastest pedestrian detector in the west. In: Proceedings of British Machine Vision Conference (BMVC), pp. 68.1–68.11 (2010)

[37] Dollár, P., Tu, Z., Perona, P., Belongie, S.: Integral channel features. In: Proceedings of British Machine Vision Conference, pp. 91.1–91.11 (2009)

[38] Elgammal, A.M., Harwood, D., Davis, L.S.: Non-parametric model for background subtraction. In: Proceedings of the 6th European Conference on Computer Vision-Part II, ECCV '00, pp. 751–767. Springer-Verlag, London, UK (2000)

[39] Elhabian, S.Y., El-Sayed, K.M., Ahmed, S.H.: Moving object detection in spatial domain using background removal techniques-state-of-art. Recent patents on computer science **1**(1), 32–54 (2008)

[40] Fan, B., Wu, F., Hu, Z.: Rotationally invariant descriptors using intensity order pooling. IEEE Transactions on Pattern Analysis and Machine Intelligence **34**(10), 2031–2045 (2012)

[41] Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. IEEE Transactions on Pattern Analysis and Machine Intelligence **32**(9), 1627–1645 (2010)

[42] Feng, J.: Combining minutiae descriptors for fingerprint matching. Pattern Recognition **41**(1), 342 – 352 (2008)

[43] Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: a statistical view of boosting. Annals of Statistics **95**(2), 337–407 (2000)

[44] Froba, B., Ernst, A.: Face detection with the modified census transform. In: Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition, pp. 91–96 (2004)

[45] Fu, X., Wei, W.: Centralized binary patterns embedded with image euclidean distance for facial expression recognition. In: Proceedings of Fourth International Conference on Natural Computation - Volume 04, ICNC '08, pp. 115–119. Washington, DC, USA (2008)

[46] Guo, Z., Zhang, D., Zhang, D., Zhang, S.: Rotation invariant texture classification using adaptive lbp with directional statistical features. In: Proceedings of International Conference on Image Processing (ICIP), pp. 285–288 (2010)

[47] Guo, Z., Zhang, L., Zhang, D.: Rotation invariant texture classification using {LBP} variance (lbpv) with global matching. Pattern Recognition **43**(3), 706 – 719 (2010)

[48] Gupta, R., Mittal, A.: Smd: A locally stable monotonic change invariant feature descriptor. In: Proceedings of European Conference on Computer Vision (ECCV), pp. 265–277 (2008)

[49] Gupta, R., Patil, H., Mittal, A.: Robust order-based methods for feature description. In: Proceedings of Computer Vision and Pattern Recognition (CVPR), pp. 334–341 (2010)

[50] Hadid, A., Pietikäinen, M.: Combining appearance and motion for face and gender recognition from videos. Pattern Recognition **42**(11), 2818 – 2827 (2009)

[51] Hafiane, A., Seetharaman, G., Zavidovique, B.: Median binary pattern for textures classification. In: Image Analysis and Recognition, *Lecture Notes in Computer Science*, vol. 4633, pp. 387–398. Springer Berlin Heidelberg (2007)

[52] Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, second edn. Cambridge University Press (2004)

[53] Heikkil, M., Pietikinen, M., Heikkil, J.: A texture-based method for detecting moving objects. In: Proceedings of British Machine Vision Conference, pp. 187–196 (2004)

[54] Heikkila, M., Pietikäinen, M.: A texture-based method for modeling the background and detecting moving objects. IEEE Transactions on Pattern Analysis and Machine Intelligence **28**(4), 657–662 (2006)

[55] Heikkilä, M., Pietikäinen, M., Schmid, C.: Description of interest regions with local binary patterns. Pattern Recognition **42**(3), 425–436 (2009)

[56] Huang, D., Wang, Y., Wang, Y.: A robust method for near infrared face recognition based on extended local binary pattern. In: Advances in Visual Computing, *Lecture Notes in Computer Science*, vol. 4842, pp. 437–446. Springer Berlin Heidelberg (2007)

[57] Iakovidis, D., Keramidas, E., Maroulis, D.: Fuzzy local binary patterns for ultrasound texture characterization. In: Image Analysis and Recognition, *Lecture Notes in Computer Science*, vol. 5112, pp. 750–759. Springer Berlin Heidelberg (2008)

[58] Jabid, T., Kabir, M., Chae, O., et al.: Robust facial expression recognition based on local directional pattern. ETRI journal **32**(5) (2010)

[59] Ji, R., Xu, P., Yao, H., Zhang, Z., Sun, X., Liu, T.: Directional correlation analysis of local haar binary pattern for text detection. In: Proceedings of International Conference on Multimedia and Expo (ICME), pp. 885–888 (2008)

[60] Jiang, N., Xu, J., Goto, S.: Pedestrian detection using gradient local binary patterns. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences **E95-A**(8), 1280–1287 (2012)

[61] Jin, H., Liu, Q., Lu, H., Tong, X.: Face detection using improved lbp under bayesian framework. In: Proceedings of IEEE First Symposium on Multi-Agent Security and Survivability, pp. 306–309 (2004)

[62] Ke, Y., Sukthankar, R.: Pca-sift: a more distinctive representation for local image descriptors. In: Proceedings of Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 506–513 (2004)

[63] Kellokumpu, V., Zhao, G., Pietikäinen, M.: Recognition of human actions using texture descriptors. Machine Vision and Applications **22**(5), 767–780 (2011)

[64] Kim, D.Y., Kwak, J.Y., Ko, B., Nam, J.Y.: Human detection using wavelet-based cs-lbp and a cascade of random forests. In: Proceedings of International Conference on Multimedia and Expo (ICME), pp. 362–367 (2012)

[65] Kim, K., Chalidabhongse, T.H., Harwood, D., Davis, L.: Real-time foreground-background segmentation using codebook model. Real-Time Imaging **11**(3), 172–185 (2005)

[66] Ko, B., Kim, D., Nam, J.: Detecting humans using luminance saliency in thermal images. Optics letters **37**(20), 4350–4352 (2012)

[67] Lazebnik, S., Schmid, C., Ponce, J.: A sparse texture representation using local affine regions. IEEE Transactions on Pattern Analysis and Machine Intelligence **27**(8), 1265–1278 (2005)

[68] Lee, B., Hedley, M.: Background estimation for video surveillance. In: Image and vision computing New Zealand, pp. 315–320 (2002)

[69] Li, B., Meng, M.Q.H.: Texture analysis for ulcer detection in capsule endoscopy images. Image and Vision Computing **27**(9), 1336 – 1342 (2009)

[70] Li, L., Huang, W., Gu, I.H., Tian, Q.: Statistical modeling of complex backgrounds for foreground object detection. IEEE Transactions on Image Processing **13**(11), 1459–1472 (2004)

[71] Li, X., Hu, W., Zhang, Z., Wang, H.: Heat kernel based local binary pattern for face representation. IEEE Signal Processing Letters **17**(3), 308–311 (2010)

[72] Li, X., Hu, W., Zhang, Z., Zhang, X.: Robust foreground segmentation based on two effective background models. In: Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval, MIR '08, pp. 223–228. ACM, New York, NY, USA (2008)

[73] Liao, S., Chung, A.C.S.: Face recognition by using elongated local binary patterns with average maximum distance gradient magnitude. In: Proceedings of Asian Conference on Computer Vision - Volume Part II, ACCV'07, pp. 672–679 (2007)

[74] Liao, S., Zhao, G., Kellokumpu, V., Pietikäinen, M., Li, S.Z.: Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes. In: Proceedings of Computer Vision and Pattern Recognition (CVPR), pp. 1301–1306 (2010)

[75] Liao, S., Zhu, X., Lei, Z., Zhang, L., Li, S.: Learning multi-scale block local binary patterns for face recognition. In: Advances in Biometrics, *Lecture Notes in Computer Science*, vol. 4642, pp. 828–837. Springer Berlin Heidelberg (2007)

[76] Lim, J.J., Zitnick, C.L., Dollr, P.: Sketch tokens: A learned mid-level representation for contour and object detection. In: Proceedings of Computer Vision and Pattern Recognition (CVPR), pp. 3158–3165 (2013)

[77] Lin, H.H., Liu, T.L., Chuang, J.H.: A probabilistic svm approach for background scene initialization. In: Proceedings of International Conference on Image Processing, vol. 3, pp. 893–896 vol.3 (2002)

[78] Lindeberg, T.: Feature detection with automatic scale selection. International Journal of Computer Vision **30**(2), 79–116 (1998)

[79] Livingstone, M., Hubel, D.: Vision and Art: The Biology of Seeing. Harry N. Abrams (2008)

[80] Lovegrove, S., Davison, A.: Real-time spherical mosaicing using whole image alignment. In: Computer Vision ECCV 2010, *Lecture Notes in Computer Science*, vol. 6313, pp. 73–86. Springer Berlin Heidelberg (2010)

[81] Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60**(2), 91–110 (2004)

[82] Maddalena, L., Petrosino, A.: A self-organizing approach to background subtraction for visual surveillance applications. IEEE Transactions on Image Processing **17**(7), 1168–1177 (2008)

[83] Mäenpää, T., Pietikäinen, M.: Classification with color and texture: jointly or separately? Pattern Recognition **37**(8), 1629 – 1640 (2004)

[84] Maji, S., Berg, A., Malik, J.: Classification using intersection kernel support vector machines is efficient. In: Proceedings of Computer Vision and Pattern Recognition (CVPR), pp. 1–8 (2008)

[85] Marr, D.: Vision: A Computational Investigation into the Human Representation and Processing of Visual Information. Henry Holt and Co., Inc., New York, NY, USA (1982)

[86] Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: Proceedings of British Machine Vision Conference (BMVC), vol. 2, pp. 384–393 (2002)

[87] Mattivi, R., Shao, L.: Spatio-temporal dynamic texture descriptors for human motion recognition. In: Intelligent Video Event Analysis and Understanding, pp. 69–91. Springer (2011)

[88] McFarlane, N., Schofield, C.: Segmentation and tracking of piglets in images. Machine Vision and Applications **8**(3), 187–193 (1995)

[89] Messelodi, S., Modena, C., Segata, N., Zanin, M.: A kalman filter based background updating algorithm robust to sharp illumination changes. In: Image Analysis and Processing ICIAP 2005, *Lecture Notes in Computer Science*, vol. 3617, pp. 163–170. Springer Berlin Heidelberg (2005)

[90] Michelson, A.: Studies in Optics. Dover books on astronomy. Dover Publications (1995)

[91] Mihreteab, K., Iwahashi, M., Yamamoto, M.: Crow birds detection using hog and cs-lbp. In: Proceedings of International Symposium on Intelligent Signal Processing and Communications Systems (ISPACS), pp. 406–409 (2012)

[92] Mikolajczyk, K., Schmid, C.: An affine invariant interest point detector. In: Computer Vision ECCV 2002, *Lecture Notes in Computer Science*, vol. 2350, pp. 128–142. Springer Berlin Heidelberg (2002)

[93] Mikolajczyk, K., Schmid, C.: Scale and affine invariant interest point detectors. International Journal of Computer Vision **60**(1), 63–86 (2004)

[94] Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence **27**(10), 1615–1630 (2005)

[95] Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Gool, L.V.: A comparison of affine region detectors. International Journal of Computer Vision **65**(1-2), 43–72 (2005)

[96] Mortensen, E.N., Deng, H., Shapiro, L.: A sift descriptor with global context. In: Proceedings of Computer Vision and Pattern Recognition (CVPR), pp. 184–190 (2005)

[97] Mu, Y., Yan, S., Liu, Y., Huang, T., Zhou, B.: Discriminative local binary patterns for human detection in personal album. In: Proceedings of Computer Vision and Pattern Recognition (CVPR), pp. 1–8 (2008)

[98] Nanni, L., Brahnam, S., Lumini, A.: A local approach based on a local binary patterns variant texture descriptor for classifying pain states. Expert Systems with Applications **37**(12), 7888 – 7894 (2010)

[99] Nanni, L., Lumini, A., Brahnam, S.: Local binary patterns variants as texture descriptors for medical image analysis. Artificial Intelligence in Medicine **49**(2), 117 – 125 (2010)

[100] Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: Proceedings of Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 2161–2168 (2006)

[101] Ojala, T., Pietikäinen, M.: Unsupervised texture segmentation using feature distributions. Pattern Recognition **32**(3), 477–486 (1999)

[102] Ojala, T., Pietikäinen, M., Harwood, D.: Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In: Proceedings of International Conference on Pattern Recognition, vol. 1, pp. 582–585 vol.1 (1994)

[103] Ojala, T., Pietikäinen, M., Harwood, D.: A comparative study of texture measures with classification based on featured distributions. Pattern Recognition **29**(1), 51–59 (1996)

[104] Ojala, T., Pietikäinen, M., Mäenpää, T.: Gray scale and rotation invariant texture classification with local binary patterns. In: Computer Vision - ECCV 2000, *Lecture Notes in Computer Science*, vol. 1842, pp. 404–420. Springer Berlin Heidelberg (2000)

[105] Ojala, T., Pietikäinen, M., Mäenpää, T.: Multire solution gray-scale and rotation invariant texture classication with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence **24**(7), 971–987 (2002)

[106] Ojala, T., Valkealahti, K., Oja, E., Pietikäinen: Texture discrimination with multidimensional distributions of signed gray level differences. Pattern Recognition **34**, 727–739 (2001)

[107] Oliver, N., Rosario, B., Pentland, A.: A bayesian computer vision system for modeling human interactions. IEEE Transactions on Pattern Analysis and Machine Intelligence **22**(8), 831–843 (2000). DOI 10.1109/34.868684

[108] Ouyang, W., Wang, X.: Single-pedestrian detection aided by multi-pedestrian detection. In: Proceedings of Computer Vision and Pattern Recognition (CVPR), pp. 3198–3205 (2013)

[109] Paisitkriangkrai, S., Shen, C., Zhang, J.: Fast pedestrian detection using a cascade of boosted covariance features. IEEE Transactions on Circuits and Systems for Video Technology **18**(8), 1140–1151 (2008)

[110] Palmer, S.E.: Vision science: Photons to phenomenology. The MIT press (1999)

[111] Petpon, A., Srisuk, S.: Face recognition with local line binary pattern. In: Proceedings of International Conference on Image and Graphics (ICIG), pp. 533–539 (2009)

[112] Pietikäinen, M., Nurmela, T., Mäenpää, T., Turtinen, M.: View-based recognition of real-world textures. Pattern Recognition **37**(2), 313–323 (2004)

[113] Pietikäinen, M., Ojala, T., Nisula, J., Heikkinen, J.: Experiments with two industrial problems using texture classification based on feature distributions. In: Photonics for Industrial Applications, pp. 197–204. International Society for Optics and Photonics (1994)

[114] Pietikäinen, M., Ojala, T., Xu, Z.: Rotation-invariant texture classification using feature distributions. Pattern Recognition **33**, 43–52 (2000)

[115] Pietikäinen, M., Zhao, G., Hadid, A., Ahonen, T.: Computer Vision Using Local Binary Patterns. No. 40 in Computational Imaging and Vision. Springer (2011)

[116] Prioletti, A., Mogelmose, A., Grisleri, P., Trivedi, M., Broggi, A., Moeslund, T.: Part-based pedestrian detection and feature-based tracking for driver assistance: Real-time, robust algorithms, and evaluation. IEEE Transactions on Intelligent Transportation Systems **14**(3), 1346–1359 (2013)

[117] Russell, B.C., Martin-Brualla, R., Butler, D.J., Seitz, S.M., Zettlemoyer, L.: 3D Wikipedia: Using online text to automatically label and navigate reconstructed geometry. ACM Transactions on Graphics (SIGGRAPH Asia 2013) **32**(6) (2013)

[118] Sabzmeydani, P., Mori, G.: Detecting pedestrians by learning shapelet features. In: Proceedings of Computer Vision and Pattern Recognition (CVPR), pp. 1–8 (2007)

[119] Schaffalitzky, F., Zisserman, A.: Viewpoint invariant texture matching and wide baseline stereo. In: Proceedings of IEEE International Conference on Computer Vision (ICCV), vol. 2, pp. 636–643 (2001)

[120] Schindler, K., Wang, H.: Smooth foreground-background segmentation for video processing. In: Proceedings of Asian Conference on Computer Vision (ACCV), pp. 581–590. Hyderabad, India (2006)

[121] Schwartz, W., Kembhavi, A., Harwood, D., Davis, L.: Human detection using partial least squares analysis. In: Proceedings of International Conference on Computer Vision (ICCV), pp. 24–31 (2009)

[122] Shi, W., Chen, S., Fang, L.: Local pixel class pattern based on fuzzy reasoning for feature. WSEAS Transactions on Signal Processing **9**(2), 31–40 (2013)

[123] Song, T., Li, H.: Local polar dct features for image description. IEEE Signal Processing Letters **20**(1), 59–62 (2013)

[124] Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. In: Proceedings of Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 246–252 (1999)

[125] Szeliski, R.: Computer Vision: Algorithms and Applications, 1st edn. Springer-Verlag New York, Inc., New York, NY, USA (2010)

[126] Taj, M., Cavallaro, A.: Multi-view multi-object detection and tracking. In: Computer Vision, *Studies in Computational Intelligence*, vol. 285, pp. 263–280. Springer Berlin Heidelberg (2010)

[127] Takala, V., Ahonen, T., Pietikäinen, M.: Block-based methods for image retrieval using local binary patterns. In: Image Analysis, *Lecture Notes in Computer Science*, vol. 3540, pp. 882–891. Springer Berlin Heidelberg (2005)

[128] Takala, V., Pietikainen, M.: Multi-object tracking using color, texture and motion. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–7 (2007)

[129] Tan, X., Triggs, B.: Enhanced local texture feature sets for face recognition under difficult lighting conditions. IEEE Transactions on Image Processing **19**(6), 1635–1650 (2010)

[130] Tang, D., Liu, Y., kyun Kim, T.: Fast pedestrian detection by cascaded random forest with dominant orientation templates. In: Proceedings of British Machine Vision Conference (BMVC), pp. 58.1–58.11 (2012)

[131] Tang, F., Lim, S., Chang, N., Tao, H.: A novel feature descriptor invariant to complex brightness changes. In: Proceedings of Computer Vision and Pattern Recognition (CVPR), pp. 2631–2638 (2009)

[132] Tavakkoli, A., Nicolescu, M., Bebis, G.: A novelty detection approach for foreground region detection in videos with quasi-stationary backgrounds. In: Advances in Visual Computing, *Lecture Notes in Computer Science*, vol. 4291, pp. 40–49. Springer Berlin Heidelberg (2006)

[133] Thaler, M., Bailer, W.: Real-time person detection and tracking in panoramic video. In: Proceedings of Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1027–1032 (2013)

[134] Thomas, J., Cook, K.: Illuminating the Path: The Research and Development Agenda for Visual Analytics. IEEE Computer Society Press (2005)

[135] Tola, E., Lepetit, V., Fua, P.: Daisy: An efficient dense descriptor applied to wide baseline stereo. IEEE Transactions on Pattern Analysis and Machine Intelligence **32**(5), 815–830 (2010)

[136] Toyama, K., Krumm, J., Brumitt, B., Meyers, B.: Wallflower: principles and practice of background maintenance. In: Proceedings of International Conference on Computer Vision (ICCV), vol. 1, pp. 255–261 (1999)

[137] Trefnỳ, J., Matas, J.: Extended set of local binary patterns for rapid object detection. In: Proceedings of the Computer Vision Winter Workshop, vol. 2010 (2010)

[138] Tuytelaars, T., Gool, L.V.: Matching widely separated views based on affine invariant regions. International Journal of Computer Vision **59**(1), 61–85 (2004)

[139] Tuytelaars, T., Mikolajczyk, K.: Local invariant feature detectors: a survey. Foundations and Trends® in Computer Graphics and Vision **3**(3), 177–280 (2008)

[140] Tuzel, O., Porikli, F., Meer, P.: Pedestrian detection via classification on riemannian manifolds. IEEE Transactions on Pattern Analysis and Machine Intelligence **30**(10), 1713–1727 (2008)

[141] Viola, P., Jones, M.J.: Robust real-time face detection. International Journal of Computer Vision **57**(2), 137–154 (2004)

[142] Wagner, D., Mulloni, A., Langlotz, T., Schmalstieg, D.: Real-time panoramic mapping and tracking on mobile phones. In: Proceedings of Virtual Reality Conference (VR), pp. 211–218 (2010)

[143] Walk, S., Majer, N., Schindler, K., Schiele, B.: New features and insights for pedestrian detection. In: Proceedings of Computer Vision and Pattern Recognition (CVPR) (2010)

[144] Wang, H., Suter, D.: A re-evaluation of mixture-of-gaussian background modeling. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 1017–1020. Pennsylvania, USA (2005)

[145] Wang, J., Bebis, G., Nicolescu, M., Nicolescu, M., Miller, R.: Improving target detection by coupling it with tracking. Machine Vision and Applications **20**(4), 205–223 (2009)

[146] Wang, L., Wu, H., Pan, C.: Adaptive εlbp for background subtraction. In: Proceedings of the 10th Asian Conference on Computer Vision - Volume Part III, ACCV'10, pp. 560–571. Springer-Verlag (2011)

[147] Wang, X., Han, T., Yan, S.: An hog-lbp human detector with partial occlusion handling. In: Proceedings of International Conference on Computer Vision (ICCV), pp. 32–39 (2009)

[148] Wang, Z., Fan, B., Wu, F.: Local intensity order pattern for feature description. In: Proceedings of International Conference on Computer Vision (ICCV), pp. 603–610 (2011)

[149] Welsh, B.C., Farrington, D.P.: Public area cctv and crime prevention: An updated systematic review and meta-analysis. Justice Quarterly **26**(4), 716–745 (2009)

[150] Winder, S., Brown, M.: Learning local image descriptors. In: Proceedings of Computer Vision and Pattern Recognition (CVPR), pp. 1–8 (2007)

[151] Winder, S.A.J., Hua, G., Brown, M.: Picking the best daisy. In: Proceedings of Computer Vision and Pattern Recognition (CVPR), pp. 178–185 (2009)

[152] Wolf, L., Hassner, T., Taigman, Y.: Descriptor based methods in the wild. In: Real-Life Images workshop at the European Conference on Computer Vision (ECCV) (2008)

[153] Wren, C.R., Azarbayejani, A., Darrell, T., Pentland, A.P.: Pnder: Real-time tracking of the human body. IEEE Transactions on Pattern Analysis and Machine Intelligence **19**, 780–785 (1997)

[154] Wu, B., Nevatia, R.: Optimizing discrimination-efficiency tradeoff in integrating heterogeneous local features for object detection. In: Proceedings of Computer Vision and Pattern Recognition (CVPR), pp. 1–8 (2008)

[155] Xu, J., Wu, Q., Zhang, J., Tang, Z.: Fast and accurate human detection using a cascade of boosted ms-lbp features. IEEE Signal Processing Letter **19**(10), 676–679 (2012)

[156] Xue, G., Song, L., Sun, J., Wu, M.: Hybrid center-symmetric local pattern for dynamic background subtraction. In: Proceedings of International Conference on Multimedia and Expo (ICME), pp. 1–6 (2011)

[157] Yamazaki, M., Xu, G., Chen, Y.W.: Detection of moving objects by independent component analysis. In: P. Narayanan, S. Nayar, H.Y. Shum (eds.) Computer Vision ACCV 2006, *Lecture Notes in Computer Science*, vol. 3852, pp. 467–478. Springer Berlin Heidelberg (2006)

[158] Yao, C.H., Chen, S.Y.: Retrieval of translated, rotated and scaled color textures. Pattern Recognition **36**(4), 913 – 929 (2003)

[159] Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. ACM Comput. Surv. **38**(4) (2006)

[160] Zambanini, S., Kampel, M.: A local image descriptor robust to illumination changes. In: Image Analysis, *Lecture Notes in Computer Science*, vol. 7944, pp. 11–21. Springer Berlin Heidelberg (2013)

[161] Zhang, B., Gao, Y., Zhao, S., Liu, J.: Local derivative pattern versus local binary pattern: Face recognition with high-order local pattern descriptor. IEEE Transactions on Image Processing **19**(2), 533–544 (2010)

[162] Zhang, S., Yao, H., Liu, S.: Dynamic background modeling and subtraction using spatio-temporal local binary patterns. In: Proceedings of IEEE International Conference on Image Processing, pp. 1556–1559 (2008)

[163] Zhang, W., Shan, S., Gao, W., Chen, X., Zhang, H.: Local gabor binary pattern histogram sequence (lgbphs): a novel non-statistical model for face representation and recognition. In: Proceedings of International Conference on Computer Vision, vol. 1, pp. 786–791 (2005)

[164] Zhao, G., Pietikäinen, M.: Dynamic texture recognition using local binary patterns with an application to facial expressions. IEEE Transactions on Pattern Analysis and Machine Intelligence **29**(6), 915–928 (2007)

[165] Zhao, S., Gao, Y., Zhang, B.: Sobel-lbp. In: Proceedings of International Conference on Image Processing (ICIP), pp. 2144–2147 (2008)

[166] Zheng, J., Wang, Y., Nihan, N.L., Hallenbeck, M.E.: Extracting roadway background image: Mode-based approach. Transportation Research Record: Journal of the Transportation Research Board **1944**(1), 82–88 (2006)

[167] Zheng, Y., Shen, C., Hartley, R., Huang, X.: Pyramid center-symmetric local binary/trinary patterns for effective pedestrian detection. In: Proceedings of Asian Conference on Computer Vision - Volume Part IV, ACCV'10, pp. 281–292 (2011)

[168] Zhou, B., Wang, X., Tang, X.: Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents. In: Proceedings of Computer Vision and Pattern Recognition (CVPR), pp. 2871–2878 (2012)

[169] Zhu, C., Bichot, C.E., Chen, L.: Image region description using orthogonal combination of local binary patterns enhanced with color information. Pattern Recognition **46**(7), 1949 – 1963 (2013)

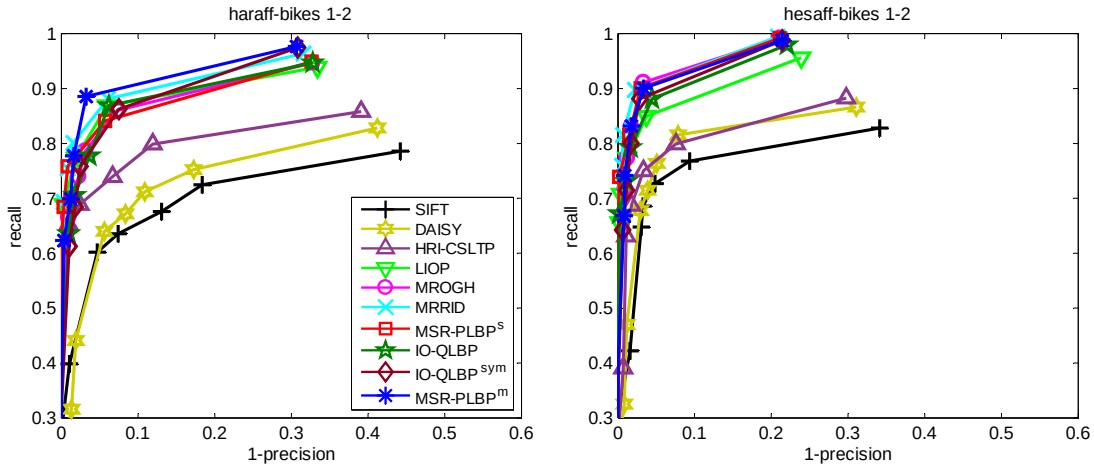
[170] Zhu, Q., Yeh, M.C., Cheng, K.T., Avidan, S.: Fast human detection using a cascade of histograms of oriented gradients. In: Proceedings of Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 1491–1498 (2006)

[171] Zivkovic, Z.: Improved adaptive gaussian mixture model for background subtraction. In: Proceedings of International Conference on Pattern Recognition (ICPR), vol. 2, pp. 28–31 (2004)

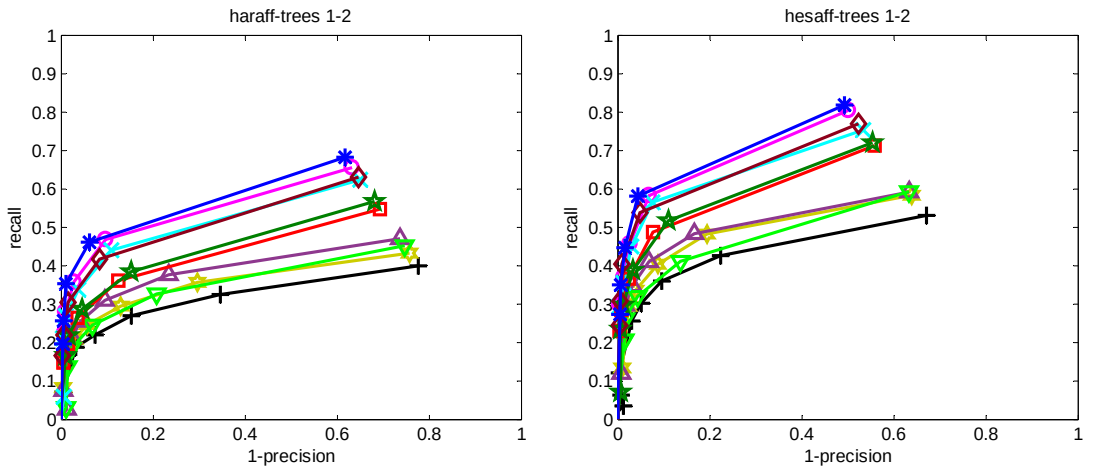
Appendix A

Image matching results

A.1 Matching results on the Oxford dataset

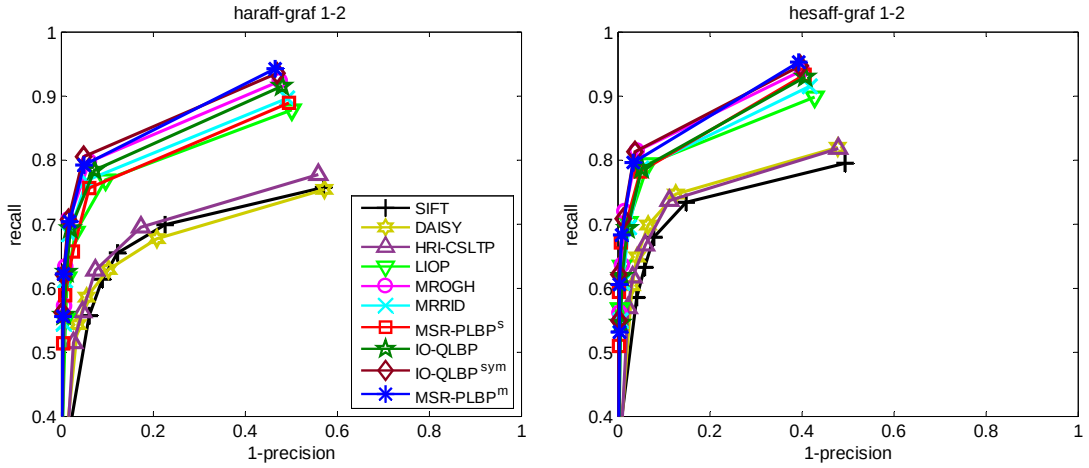


(a) bikes: image blur

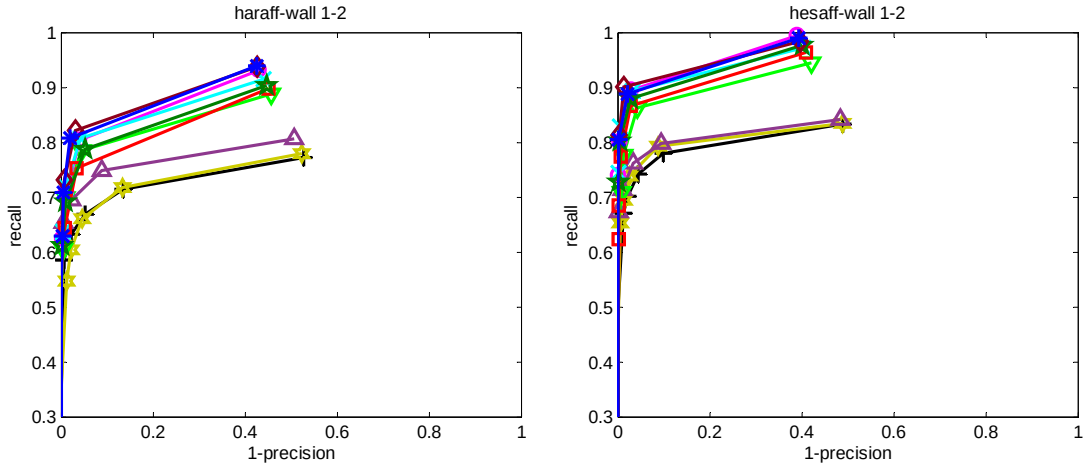


(b) trees: image blur

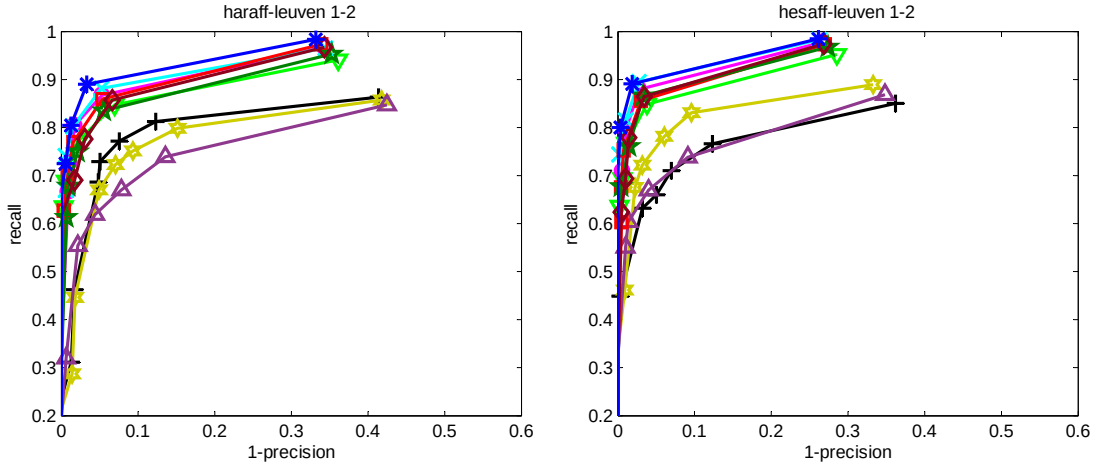
Figure A.1: The performance of evaluated descriptors on the $1^{st} - 2^{nd}$ pairs of the Oxford benchmark (Part 1/3). The scales are different through figures.



(a) graf: viewpoint change

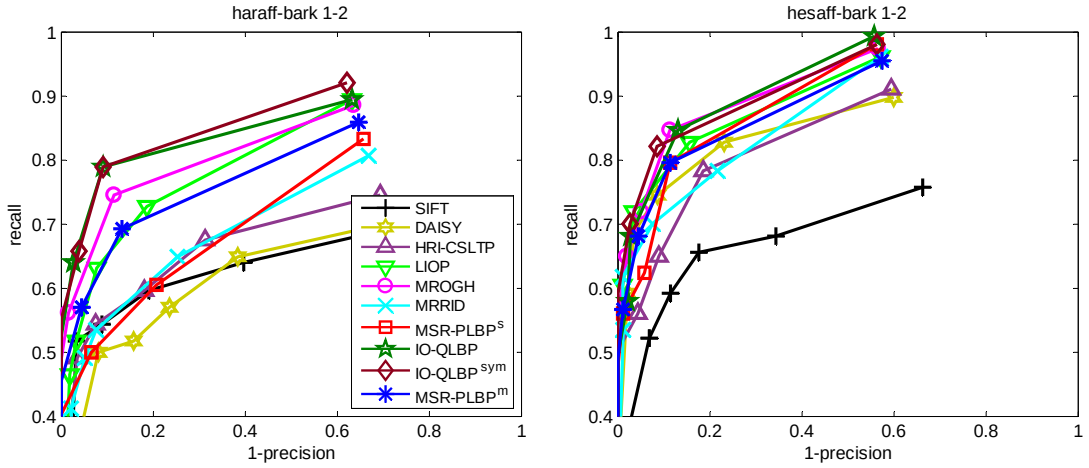


(b) wall: viewpoint change

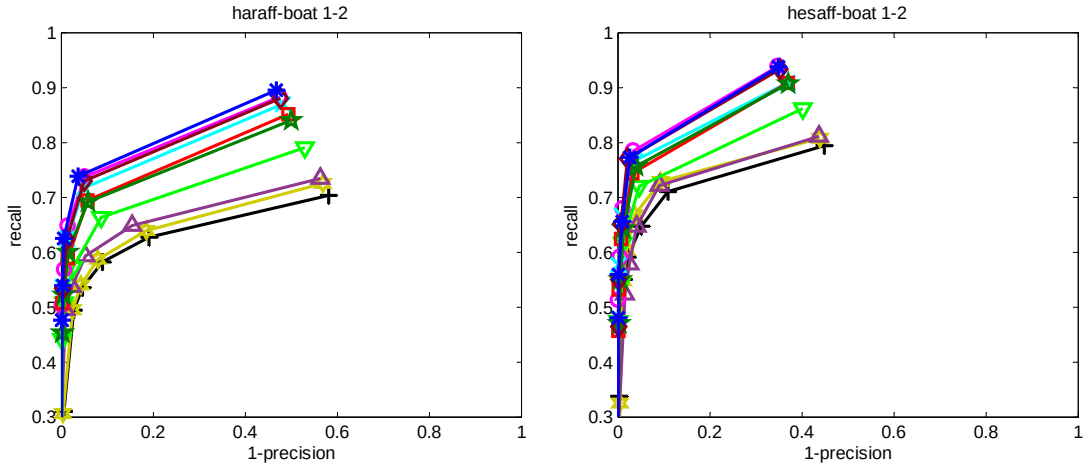


(c) leuven: illumination change

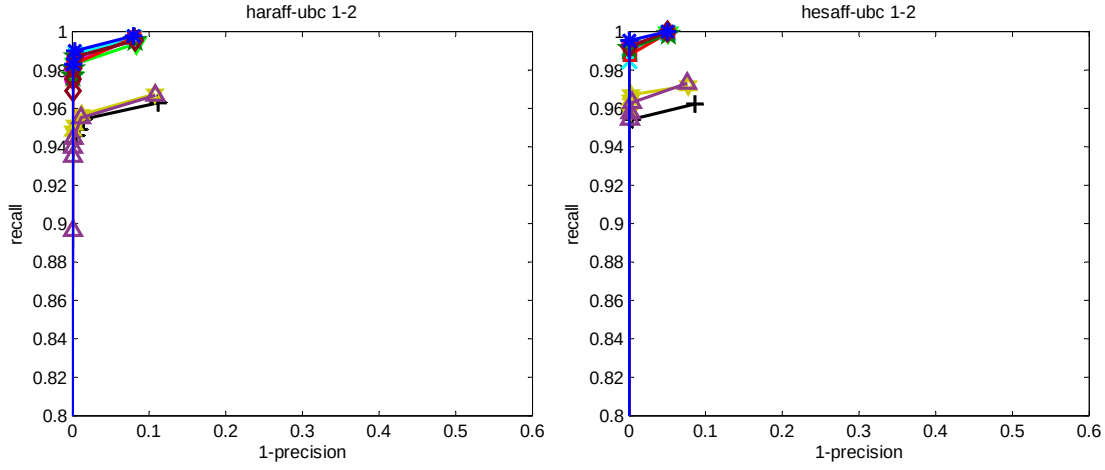
Figure A.2: The performance of evaluated descriptors on the $1^{st} - 2^{nd}$ pairs of the Oxford benchmark (Part 2/3). The scales are different through figures.



(a) bark: zoom and rotation



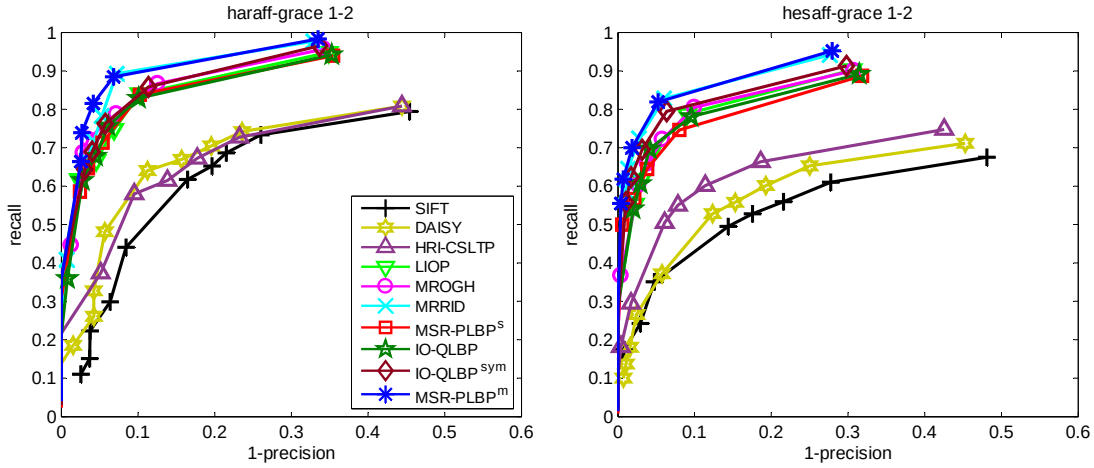
(b) boat: zoom and rotation



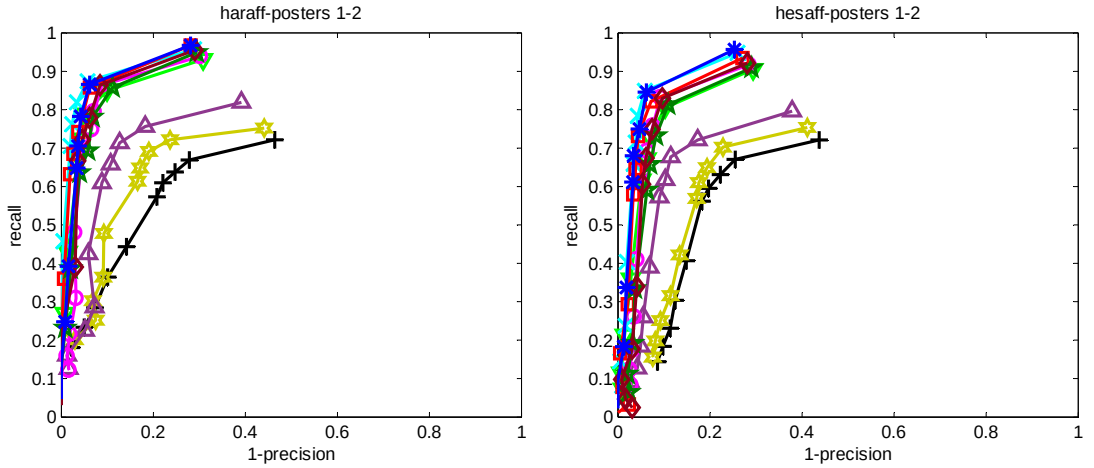
(c) ubc: JPEG compression

Figure A.3: The performance of evaluated descriptors on the $1^{st} - 2^{nd}$ pairs of the Oxford benchmark (Part 3/3). The scales are different through figures.

A.2 Matching results on the Viewpoint change dataset

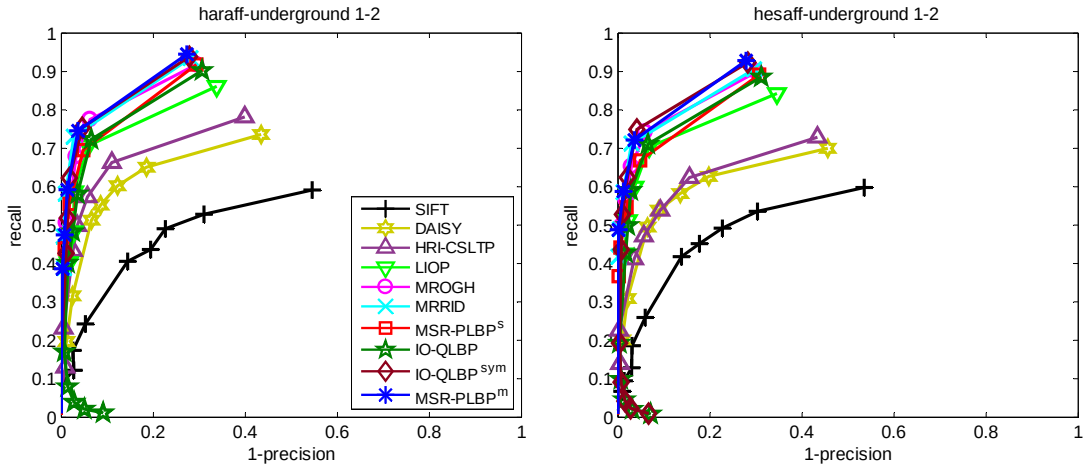


(a) grace

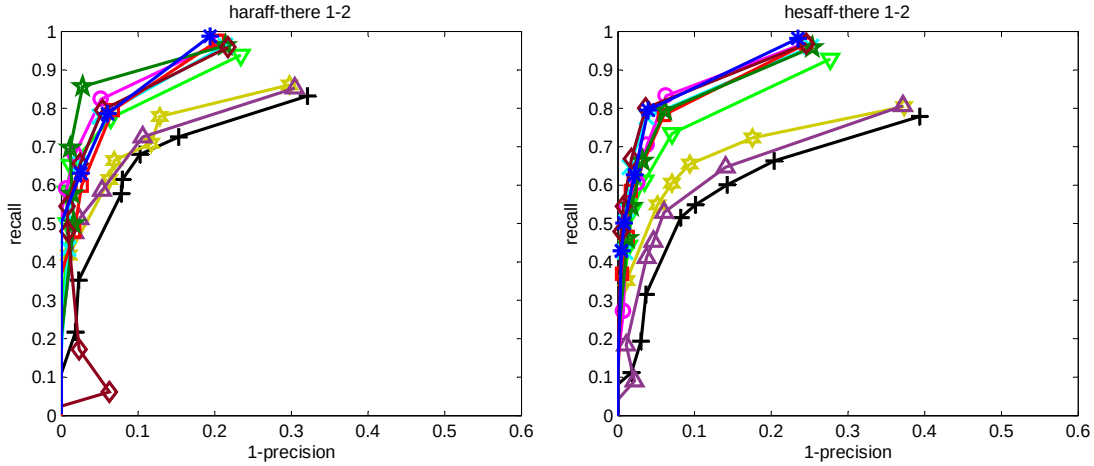


(b) posters

Figure A.4: The performance of evaluated descriptors on the $1^{st} - 2^{nd}$ pairs of the Viewpoint change dataset (Part 1/2). The scales are different through figures for better clarifying the plots.



(a) underground



(b) there

Figure A.5: The performance of evaluated descriptors on the $1^{st} - 2^{nd}$ pairs of the Viewpoint change dataset (Part 1/2). The scales are different through figures for better clarifying the plots.

List of publications

Reviewed International Conference Papers

1. Nguyen, T.N., Le, B., Miyata, K.: A multi-layer framework for background subtraction using extended local ternary patterns. In: Proceedings of International Workshop on Advanced Image Technology (IWAIT), pp. 476–481 (2012)

2. Nguyen, T.N., Le, B., Miyata, K.: Effective feature description using intensity order local binary pattern. In: Proceedings of International Conference on Cyberworlds (CW), pp. 175–182 (2013)

3. Nguyen, T.N., Miyata, K.: A novel local binary pattern feature for robust pedestrian detection. In: Proceedings of NICOGRAPH International 2014, pp.127-136 (2014)

International Journal

1. Nguyen, T.N., Le, B., Miyata, K.: A novel integration of intensity order and texture for effective feature description. IEICE Transactions on Information and Systems, **E97-D**(8), pp.2021-2029 (2014)

2. Nguyen, T.N., Miyata, K.: Multi-scale region perpendicular local binary pattern: An effective feature for interest region description. The Visual Computer Journal, **Accepted**, – (2014). DOI:10.1007/s00371-014-0934-5