SPIM

Thèse de Doctorat

école doctorale **sciences pour l'ingénieur et microtechniques**

UNIVERSITÉ DE BOURGOGNE

Recognition of Reflective Manufactured Object in non Controlled Complex Environment

QINGLIN LU

SPIM

Thèse de Doctorat

école doctorale **sciences pour l'ingénieur et microtechniques**

UNIVERSITÉ DE BOURGOGNE

N° X X X

THÈSE présentée par

QINGLIN LU

pour obtenir le

Grade de Docteur de

l'Université de Bourgogne

Spécialité : **Instrumentation et informatique de l'image**

Recognition of Reflective Manufactured Object in non Controlled Complex Environment

Soutenue publiquement le 17 December 2015 devant le Jury composé de :

CHRISTOPHE DUCOTTET	Examinateur	Professeur à l'Université Jean Monnet
NICOLE VINCENT	Rapporteur	Professeur à l'Université Paris Descartes
PHILIPPE CARRÉ	Rapporteur	Professeur à l'Université de Poitiers
ERIC FAUVET	Co-Encadrant	Maitre de conférence à l'Université de Bourgogne
ANASTASIA ZAKHAROVA	Co-Encadrante	Maitre de conférence à l'INSA de Rouen
OLIVIER LALIGANT	Directeur de thèse	Professeur à l'Université de Bourgogne

CONTENTS

1

General Introduction

The work is carried out in the context of industrial project and supported by company Alithéon SAS. Company Alithéon SAS is a french consulting company for operation system building, software application developing, as well as project planification and conception providing. The project which I was involved in is building an internet of things to link the manufacturers, distributors and dealers in order to facilitate the communication among them. In this project, an automatic manufactured object recognition system is imperative to release the manpower. This computer vision and machine learning based system is required to recognize the manufactured objects in non-controlled environment such as in the factory or in a store. The developed system is supposed to be applied in a watchmaking company. The main purpose of my work was to study a system that is able to recognize and identify luxury manufactured watches.

In general, object recognition is one of the most important topics in computer vision. It aims to find and identify objects in an image or video sequence. Humans recognize a multitude of objects in images with a little effort, despite the fact that the image of the objects varies from different view points, differs in size and scale or even when the objects are translated or rotated. Moreover, objects can be recognized even when they are partially obstructed from the view. However, for the machine vision, accurately recognizing the objects under various conditions is still a challenging task. This thesis addresses the problem of reflective manufactured objects recognition based on machine vision. For the industrial applications, recognizing products by machine vision provides advantages for object indexing, quality control, and machine manipulating. However, many manufactured objects contain specular surfaces that make the objects partially or entirely reflective. The recognition of the objects with the presence of reflection is a long standing problem in

computer vision. In the image/video, the reflection conceals the original color and texture of the objects, duplicates the environment on the object surface.

In consequence of non-available of the often-used features, we aim to develop a system that is specifically targeting the reflective manufactured objects. The system has to address the following tasks: (1) object localization in the video (2) object model recognition (3) object reference number detection (4) object identification by recognition of detected reference number.

1.1/ OBJECTIVE

As previously mentioned, the objective of this thesis is to develop a system that recognizes and identifies manufactured objects in complex environment. The object recognition and identification play an important role in multiple tasks such as anti-fake, object representation as well as tracking the object and recording the place where it is sold. The mentioned complex environment contains various illumination sources and non-unify background. Due to the complex illumination condition and the tiny reference number engraved on the reflective object surface, recognition of the object by reading the reference number based on human vision may be error-prone. Thus, introducing a machine vision system to identify the object is expected to release the manpower and be more accurate. The expected system contains two main tasks that are object model recognition and object identification. For most of the manufactured products, one object model may be produced numerous times: the recognition process aims to recognize the object model, and the identification process aims to identify the individual object.

Object model recognition Generally, the machine vision based object recognition usually takes into account of objects' characteristics such as contours, shape, color, consisted components, etc... By comparing the characteristics from one candidate object with that from the reference objects, the candidate object can be classified and its model could be identified. In our system, we aim to recognize the object model by comparing the candidate object features with the template features, then classify the candidate object into the correct model class.

Figure 1.1: Reflective object recognition

Object identification Computer vision and machine learning based object identification mainly tackle the problem of optical character recognition. Indeed the reference number of a manufactured object carries unique information for each individual model. In our system, we aim to first localize the reference number in each frame of the input video, then recognize every single characters contained in the reference. With the recognized reference number, the individual object in the input video can be identified.

An example of the system usage is shown in Fig. 1.1, where in a jewelry shop. Sales present a product to the client with a tablet. In order to recognize the individual product. The sales pass the product in front of the camera of the tablet. The camera captures video and tablet sends the video to the service. The expected system recognizes the object model and identifies the individual product, then sends the ID of the object to the service. Finally the service displays the detailed presentation of the product by the tablet.

1.2/ PROBLEM DEFINITION AND LIMITATION OF EXISTING WORKS

We mainly focus on manufactured objects: numerous of them contains specular surfaces. Thus reflection is present on the surfaces of the objects. In acquired images and videos, the reflection brings negative effects on object's color, texture, and structure information.

Moreover, reflecting the complex environment on the object surface raises the challenge of object segmentation from the background. Therefore, the presence of reflection has been a long standing problem for object recognition, segmentation, and tracking in computer vision. Many investigations have been done in order to deal with reflection in the image. Most of them consider reflection as noise and try to remove/reduce it. Some of them analyze the chromaticity to distinguish the reflection and the background from a single image, some of them study the gradients with variation across a group of images, some of them use the polarization of the reflection. However, each idea has its own constraints such as specifically designed setup, the need of regular textures, or even the use of polarizes. Indeed, most of them achieved promising results with their own conditions. However, in a object recognition system, those numerous setups and strict constraints for removing the reflection are not suitable since they are difficult to be satisfied in various industrial conditions.

Basically, the object model recognition is to find out the corresponding model class of one unknown object. Recently, training deep convolutional neural networks (Deep-CNN) to classify the object is the most popular technique in object recognition. Within the increase of computer calculation speed, more and more layers and parameters can be fitted in and trained to recognize more complicate objects. In consequence of that, Deep-CNN achieved impressive results in huge dataset such as ImageNet [1] and challenging tasks such as object recognition from more than 220,000 classes. Instead of extracting specific features to present the target object, multiple-scale convolutional masks are employed in different layers to gain descriptors from the image. Most of these descriptors represent the gradient information. However, the presence of reflection creates false gradients in the image which leads to the inaccurate descriptors extraction. Moreover, Deep-CNN model requires enormous training data to avoid the under fitting problem. However, the construction of this kind of dataset from zero is normally not affordable. On the other hand, classical object recognition methods are based on extracting representative features from the object such as key points, textures, colors, and shape. These features can be very robust without the presence of the reflection. However, the reflection in the image can be the shadow of environment, as well as the saturation of illumination. Thus the mentioned features can be frequently covered, or changed, or even destroyed due to the presence of the reflection.

The object identification relies on the text detection and character recognition. Numerous contributions have been made for text detection and character recognition in the nature scene. Among the most successful methods, they attempt to detect text candidates by specifically designed features such as Maximum Stable Extremal Regions (MSER), Stroke Width Transform (SWT), or Histogram of Oriented Gradient (HOG)... Strong classifiers as deep neural network or deep metric learning are further employed to filter the false positives. Finally the text line is detected by geometric assumptions such as uniform SWT or specific projection profile. However, once the text in the nature scene is influenced by the reflection, those features are no more robust for the candidates detection at the very beginning of the system. Another problem concerns the geometric assumptions. Many authors assume that text contains uniform stroke width and always stay in lines. Uniform stroke width is not valid for multiple text zones in the image and text stay in lines is not valid according to many engraved characters on manufactured products. Hence, the rejection of geometric assumptions is indispensable for high performance text detection and character recognition system.

1.3/ MAIN CONTRIBUTIONS

The main contributions of our work rely on two aspects: object model recognition and object identification. We first build an initial prototype that aims to recognize the manufactured object. Then, with regard to the limitations of the initial prototype, we investigate in these problematic in order to improve both the object recognition and identification performances.

The initial prototype is a complete object recognition system which takes video sequence as input and the individual object identification as output. The contained stages in the prototype are offline processing and online processing, where offline processing aims to create template for each reference object model, online processing aims to recognize and identify the individual candidate object. The initial step of online processing is to track the candidate object in the video. In order to accurately localize the reflective object in the video, we proposed an fine-grained object tracking method based on parameterized external contour for tracking the reflective objects. It has two main advantages comparing to other tracking methods: (1) an accurate object tracking which provides the capability of

following details on the objects; (2) an efficient tracker for the entire reflective objects. With this parameterized external contour based tracking method, additionally with template matching process and prior knowledge about the exact location of the reference number on the template, the candidate object is able to be identified.

Hereafter, during the utilization of this initial prototype, several limitations are observed. In object model recognition phase, the extracted features are coarse-grained and evade from the reflection problem. On the other hand, in object identification phase, reference number detection is completely relying on previous object tracking and template matching which results in lacking of robustness. Moreover, the character recognition is conducted by an existing software that is not specifically created for recognizing the characters on reflective surfaces. Thus we investigate in improving the prototype addressing the limitations of the prototype.

Object model recognition For the object model recognition, the main limitations are caused from the lack of specific features. In order to boost the recognition performance, two methods which extract specific features for reflective objects are proposed:

The first one is a novel reflection based method to estimate the profile of specular surface with few constraints such as a straight light source. It is a local geometry analysis between object surface curvature and reflection curvature with known positions of camera, light source, and object. Within the information of local surface curvature, the accuracy of the object model recognition is significantly boosted.

The second feature is the reflective surface structure information which is based on object elementary surface segmentation. The proposed segmentation method has three main advantages: (1) an effective sub-segmentation method for the reflective surface structure understanding (on both specular and transparent surfaces); (2) Instead of removing reflection, we study the reflection motion and we consider it as additional information for sub-segmentation; (3) We use the reflection motion features as spatiotemporal coherence for video segmentation and fine attributes for object surface structure understanding.

Object identification For the object identification, the limitations of the prototype are from the delicate text detection and character recognition. In order to improve the object identification, two methods are presented concerning straightforward text detection and

reflective character recognition, respectively. To the best of our knowledge, detection and recognition text on reflective surfaces have not been studied yet.

For the text detection, the proposed method is based on extracting and clustering low level features and then learning these features with a strong classifier. It has 4 main advantages: (1) Low level features detect more possible text candidates which leads to less false positives. (2) No geometric constraints that allow the system to detect text with various orientations. (3) Strong classifier provides accurate results not only for the text detection on reflective surfaces, but also in the natural scenes. (4) The challenging dataset concerning texts on the reflective surfaces is released for further research.

For recognizing engraved characters on the reflective surfaces, a two cascaded SVM (support vector machine) framework is proposed. Comparing to other OCR (optical character recognition) systems, the proposed framework has the following advantages: (1) The extracted local geometric features are adapted to make the recognition scale invariant and less sensitive to the reflection. (2) The main contribution consists in boosting the recognition performances by introducing two cascaded SVM models based on previously analyzed accuracy rate. (3) The challenging dataset is also released for the further research purposes.

1.4/ DOCUMENT ORGANIZATION

The rest of the thesis is organized in 6 chapters. The chapter 2 and 3 present the dataset and the initial prototype. The chapters 3, 4, 5, 6 present four solutions of improvement according to the limitations of the prototype.

Chapter 2: Set up and Dataset We first study the image and video acquisition in an controlled environment. In order to do this, a experimental box is designed and constructed. With this box, different illumination and acquisition conditions can be modeled, such as the positions and the orientations of the light source, objects, and the camera, respectively. The variation of object, camera, and light source poses provide more training data from various object poses in multiple views through different illumination conditions. The dataset contains more than 1000 images and 45 videos of 10 different watches from 6 different models. The dataset is used for the model template creation, object model

recognition, as well as the generation of reflective characters dataset for the further object identification.

Chapter 3: Object Recognition - Initial Prototype In this chapter, we present the initial prototype for object recognition. The prototype contains 3 main stages: **offline processing** attempts to creat object model template based on the external contour. **online processing** addresses object tracking and template matching based on global features and contour segment features. **Object identification** aims to localize the reference number in the video and recognize the characters contained in the reference by using Tesseract OCR engine. In conclusion, we summarize the prototype and discuss its limitations. To trackle these limitations, some new research topics are further studied in the next chapters.

Chapter 4: Local Surface Curvature Analysis In this chapter, we aim to extract additional features that are local surface curvature to fulfill the lack of robust representative features. The proposed method focuses on analyzing specular surface curvature profile using a single line source. This single line source could be any straight line in the environment whose position can be easily measured. By studying the geometric relation in the system, the local surface curvature of the experimented object can be estimated according to the distortion of the line that are reflected by the object surface. For the reflective object recognition, the local surface curvature provides a novel representative and reliable feature that can be used to discriminate different object models.

Chapter 5: Object Surface Structure Understanding In this chapter, we address the limitation of escaping from reflection. The proposed method first tracks the moving reflection particles in the video, then uses the motion trajectories as surface labels, and finally segments the elementary continuous surfaces based on the trajectories. After the object surface segmentation, the graph which describes the surfaces distribution is constructed as a new feature for the object representation. The proposed segmentation method provides a new perspective concerning reflection in computer vision. Extracting information from reflections instead of removing/reducing them is pioneering the work in a new direction. Within the surface graph features, the object recognition and template matching

performances can be significantly improved.

Chapter 6: Detection of Text on Reflective Surfaces In this chapter, we present a novel method to detect text on the reflective surfaces. The method initially extracts low level features such as points, contours, and regions. Then similar features are clustered in order to precisely select the text candidates. Afterwards a powerful classifier, deep convolutional neural network is trained to predict text zone in the image. This method liberates the constraint that the reference number localization has to rely on the object contour tracking, template matching, as well as the prior knowledge. The detection is straightforward and also suitable for the text in the nature scene. The proposed text detection method not only improves the reference number localization accuracy, also simplifies the prototype. In this regard, the object identification stage can be conspicuously ameliorated.

Chapter 7: Recognition of Characters Engraved on Reflective Surfaces In this chapter, we propose a novel method to recognize characters that are engraved on reflective surfaces. This method first adapts several successful local geometric features to make the recognition scale invariant and less sensitive to the reflection. Then the classification performances of SVM classifier are analyzed with three decision boundaries accompanied by individual and combination of the features. The main contribution lies on boosting the recognition performances by introducing two cascaded SVM model based on the previously analyzed accuracy rate. Multiple evaluation results show that the proposed method outperforms single classifier based methods for the recognition of characters engraved on reflective surfaces. Moreover, a challenging dataset is released for further research purpose. With the proposed method that is specifically designed for recognizing characters on reflective surfaces, the tesseract OCR engine can be replaced. In this case, the reference number recognition based object identification can achieve a better accuracy.

1.5/ PUBLICATIONS

International Journal:

1. Q. Lu, E. Fauvet, A. Zakharova, O. Laligant : "Entire Reflective Object Structure Understanding," *ELSEVIER: Pattern Recognition Letters*, 2015.
2. Q. Lu, O. Laligant, E. Fauvet, A. Zakharova : "Reflective Manufactured Object Reference Localization," *Journal of Electronic Imaging*, 2015. **Under review**

International Conference:

1. Q. Lu, O. Laligant, E. Fauvet, A. Zakharova : "Entire Reflective Object Structure Understanding based on relfection motion features," *IEEE Proceeding: British Machine Vision Conference (**BMVC'15**)*, Swansea, UK, 2015.
2. Q. Lu, O. Laligant, E. Fauvet, A. Zakharova : "Manufactured Object Sub-Segmentation based on Reflection Motion Estimation," *IAPR Machine Vision and Applications (**MVA'15**)*, Tokyo, Japan, 2015.
3. Q. Lu, O. Laligant, E. Fauvet, A. Zakharova : "Local Surface Curvature Analysis based on Reflection Estimation," *SPIE Proceding: International Conference of Digital Image Processing (**ICDIP'15**)*, Los Angeles, USA, 2015.
4. Q. Lu, O. Laligant, E. Fauvet, A. Zakharova : "Local Surface Orientation Analysis," *SPIE Proceding: Quality Control by Artificial Vision (**QCAV'15**)*, Le Creusot, France, 2015.

Invited Talk:

1. E.Fauvet, A.Hostein, O.Laligant, Q.Lu, F.Truchetet. "CIL XIII,2657 and new technologies," Oxford, UK, 2015.

I

Part I: An initial Prototype

2

SET UP AND DATASET

In this chapter, we present the experiment set up, the equipment used in this project, and the acquired dataset. This database of the objects (watch) was later used in order to simulate and estimate the recognition and identification performance of the prototype. In this regard, we aim to build an environment in which different illumination and image acquisition conditions could be controlled. With this controlled system, not only the recognition problem can be formulated and the limitation of the prototype can be estimated, but also more accurate templates can be created for each object model.

2.1/ CONTROLLED ENVIRONMENT

The objective of building a controlled environment for the image acquisition is to simulate the limit of the recognition and provide an accurate user instruction for the prototype. The instruction indicates the constraints that need to be satisfied during the utilization of the prototype such as distance between object and camera, orientation of the object, as well as the illumination condition.

We build an experimental box for the image acquisition, shown in figure 2.1. The dimension of the acquisition box is $50cm \times 40cm \times 40cm$, with an $40cm \times 40cm$ underside and $50cm$ height of the side boards. The interior of the box is black in order to avoid the complex background in the acquired images. On the side boards, four battens are fixed at different levels of height in order to produce different lightning conditions. The support of the camera is rotateable and also slideable both in horizontal and vertical directions. The sliding of the camera provides different acquisition positions, and the rotation of the camera provides different points of view. The support of the object is a $10cm \times 10cm$ tray which

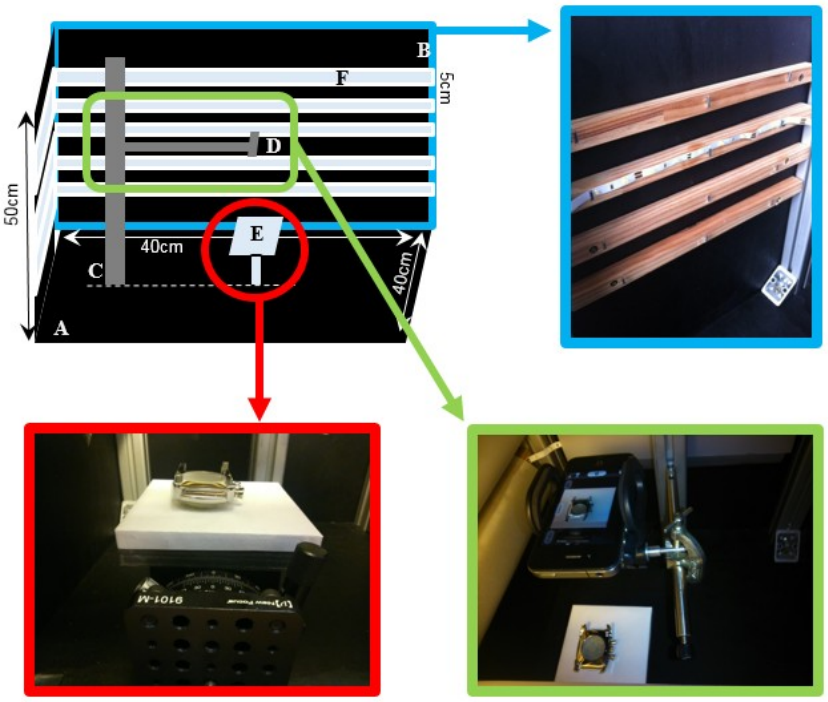


Figure 2.1: controlled environment in acquisition box

is white in order to facilitate the object segmentation. The tray is hold up by a rotation angular gyroscope which is able to rotate by an accurate angle; it is used for providing the multiple object pose for the image acquisition.

2.2/ EQUIPMENTS

The equipments include the camera, the light source, the support of the camera, and the support of the object. The camera used in the experiments is the i-sight camera of mobile device which has 5 mega pixels resolution. For the light source, as shown in Fig. 2.1, two LED grow lights are used for fading the shadows of the object and equipments. It consists of 30 light dots and these dots can be of any shape, here we use a circle one (see in Fig. 2.1). Additionally, two light spots are used to focus the illumination on the object. The support of the camera consists of two metal poles and a camera holder which clamps the camera. The distance between the camera and the object can be adjusted from $5cm$ to $30cm$, the camera orientation can be adjusted between -30 to 30 degrees. The support of the object is a $10cm \times 10cm$ trap which is fixed on a rotation angular gyroscope (see in Fig. 2.2). The gyroscope offers smooth rotation for mounted components or 1 inch optics

Figure 2.2: Illumination equipments: two light spots and two LED grow lights.

with sensitivity and control over the standard-performance alternative, which is shown in figure 2.2.

Figure 2.3: Rotation equipments: rotation angular gyroscope used in the experiments.

2.3/ DATASET

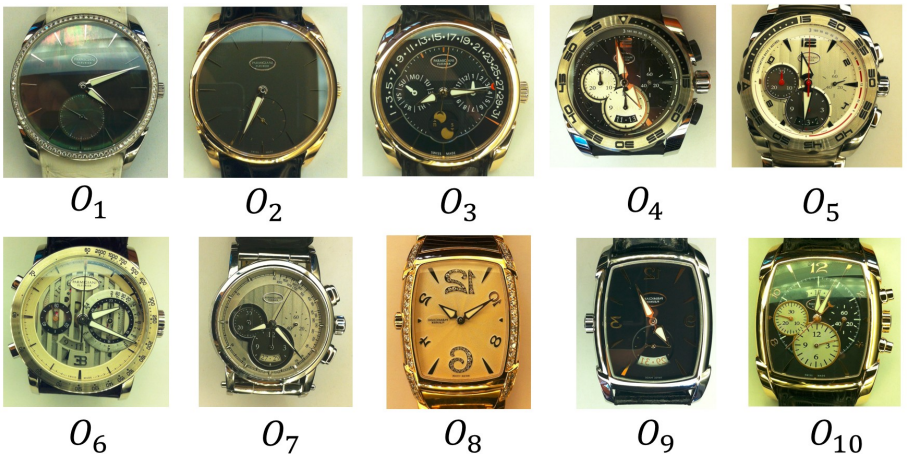


Figure 2.4: Models of the dataset

The dataset is collected from 10 watches belonging to 6 models. It contains more than

1000 images and 45 videos. For each model of the watch, 120 images and 4 videos are acquired, 2 videos capture the front side of the watch, 2 other videos capture the back side of the watch.

The acquired images for each watch are captured from different points of view (varying both distances between the camera and object and the orientation of the camera) and multiple object poses (different object orientations). The objects are captured from both front side and back side as shown in figure 2.3. The images of the front side are used for template creation and model recognition. The images from the back side are used for the reference number recognition in order to identify the object.

The acquired videos for each watch are from 3 scales, $10cm$ and $20cm$, respectively. In each video, the orientation of the object is smoothly moved from -25 to 25 degrees. All the videos have the same resolution of 1280×720. Furthermore, as the rotation of the object is smooth, in order to cover all the object poses, the duration of the video is between 20 to 40 seconds.

2.4/ CONCLUSION

In this chapter, we presented the experimental setup and the acquired dataset. The experimental setup allows us adding more light sources in order to simulate divers illumination environments (simple / complex). The dataset will be used to evaluate the further presented object recognition and identification prototype.

Figure 2.5: Examples of the acquired images, from both front side and back side of the watch.

3

Object Recognition System

In this chapter, we present the initial prototype for object recognition. The prototype contains 3 main stages: the **Offline processing** attempts to create object model template based on the external contour. It is supposed to be performed right after the manufacture for the template creation of new object models. The **Online processing** addresses object tracking and template matching based on global features and contour segment features. It is performed in the store for the object model recognition. The **Object identification** aims to localize the reference number in the video and recognize the characters contained in the reference by using Tesseract OCR engine. It is the advanced stage of online processing which helps the sale to identify the individual object. Hereafter, in the conclusion of this chapter, we discuss the prototype functionality as well as its limitations. To tackle these limitations, some new research topics are further studied in the next chapters.

3.1/ Introduction

In this chapter, an industrial driven framework is presented to address the recognition of manufactured objects. In general, the reference number of manufactured object carries unique information for each model, thus localizing and recognizing this number is essential to recognize the object. However, as the object reference number are often tiny, a straightforward localization of the reference number in the video is excessively challenging. In consequence of that, first track the object, then localize the reference number with the help of the prior knowledge and object template is more reasonable. Due to the presence of reflection on the object surfaces, the object color, texture, and interest points are highly effected. As shown in Fig. 3.1, the object surface reflects the environment which is

(a) (b) (c) (d)

Figure 3.1: Entire reflective objects: 4 models of watch and their reference numbers.

not the real texture on the object. These negative effects raised by reflection lead to a fact that often-used features such as key points [2, 3, 4], contours [5, 6], histogram of oriented gradient [7] are no more available. In this case, in order to track the object and further localize its reference number, we extract features from the object external contour which is the only robust signature against the reflection. The external contour is obtained by the pre-processing and divided into multiple individual segments that are represented by elementary geometric shapes. These contour segments are used not only for the template construction, also for the template matching. A tracker inspired from Kalman filter [8] is used to localize all the contour segments respectively. Afterwards, the combination of the segments tracking provides a fine-gained object tracking. Based on an accurate object tracking and prior knowledge about the reference number position on the template, the reference number can be localized in the video sequence. Finally, within the character recognition of the reference number, the object can be recognized. Although the dataset is challenging, the framework accomplishes meaningful results. The result evaluation is carried out in four aspects that are template matching, target object tracking, reference number localization, and optical character recognition.

The rest of this chapter is orgnized as follows: we first survey some related works in contour based object tracking and template matching based object recognition. Then the object recognition framework is presented in details. Hereafter, the experiment results as well as the evaluations are illustrated in the previously mentioned four aspects. Finally, in the conclusion, we summarize the object recognition framework and discuss its limitations. To tackle these limitations, four further problematic are studied and presented in the following chapters.

3.2/ RELATED WORKS

3.2.1/ CONTOUR BASED OBJECT TRACKING

Visual object tracking is a predominating subject in computer vision and widely studied during the last decades. As previously explained, visual tracking methods which are based on key points, colors, histograms are not suitable in our case. Thus we mainly review the related work on contour based object tracking. Contour-based representations have a long history in object recognition and computer vision. Considerable effort was spend in the past years on matching geometric shape models of the objects to image contours. The methods of representing an object contour can be roughly classified into two categories: non-parameterized contour and parameterized contour.

Most of the non-parameterized methods consider the contour tracking as an energy minimization problem. As the most successful contour based tracking in early years, condensation algorithm [9, 10, 11] is proposed by Micheal and Andrew. It is a fusion of the statistical factored sampling algorithm for static non-Gaussian problems with a stochastic model of the object motion. Even though condensation had a great success, it still suffers from tracking a complex shape model. Later on, Myung-cheol et al. [12] proposed an accurate object tracking method based on boundary edge selection. It is reliable to handle a sudden change of the tracked subject shape in a complex scene. While the main goal of this method is to improve the accuracy of tracking a textured object, once the texture of the object is influenced by the reflection, the boundary searching leads a significant mis-matching.

In tracking a parameterized contour, the object contour is represented using parameters. Many of these approaches use snake models [13], such as Kalman snake [14] and occlusion adaptive motion snake [15]. Natan [14] proposes to combine Kalman filter [8] and active contour model [15] to track nonrigid objects in combined spatio-velocity space. It employs measurements of the gradient-based image potential and the optical flow along the contour as system measurement. Nguyen et al. [16] performed a normalized correlation template tracking in the modulation domain. For each frame of the video sequence, they compute a multi-component AM-FM image model that characterizes the local texture structure of objects and backgrounds. The work closest to our approach is proposed by Erkut et al. [17]. It embeds FragTrack [18] into a particle tracker framework, associates

a reliable value to each fragment that describes a different part of the target object, then dynamically adjusts these reliabilities at each frame with respect to the current context.

Besides considering the contour as a integrated object descriptor, using different features to describe different elements of the target object is another strategy [19, 20, 21]. Splitting an object into parts introduces a kind of supplementary contour information regarding the target object by providing the relative spatial arrangements of different object sections. This offers an important advantage over the classical trackers which contain a loss of spatial information. In our case, this supplementary spatial information is considered as the object tracking features.

3.2.2/ TEMPLATE MATCHING BASED OBJECT RECOGNITION

Object detection and recognition is an active research topic in computer vision. Generally speaking, existing object recognition methods can be roughly categorized into learning-based approach and template-based approach. As the learning based methods rely considerably on local features, whereas the most local features extraction methods are not robust for reflection effected objects. Thus the learning based methods are not crucial in our case. We mainly review the template-based approachs which are more adapted to our case. Basically, in template based method, objects are described explicitly by templates and the task of object recognition becomes to find the best matching template given an input image. The templates can be represented as intensity/color images [22, 23], when the appearance of the object has to be considered. Appearance templates are often specific and suffering from the lack of generalization because the appearance of an object is usually subject to the lighting condition and surface property of the object. Therefore, binary templates representing the contours of the objects are often used in object recognition since the shape information can be well captured by the templates [24, 25, 26]. Given a set of binary templates representing the object, the task of recognition eventually becomes the calculation of the matching score between each reference template and the candidate object. The commonly used matching method is known as ”Chamfer matching” which computes the ”Chamfer distance” [27] or ”Hausdorff distance” [28, 29].

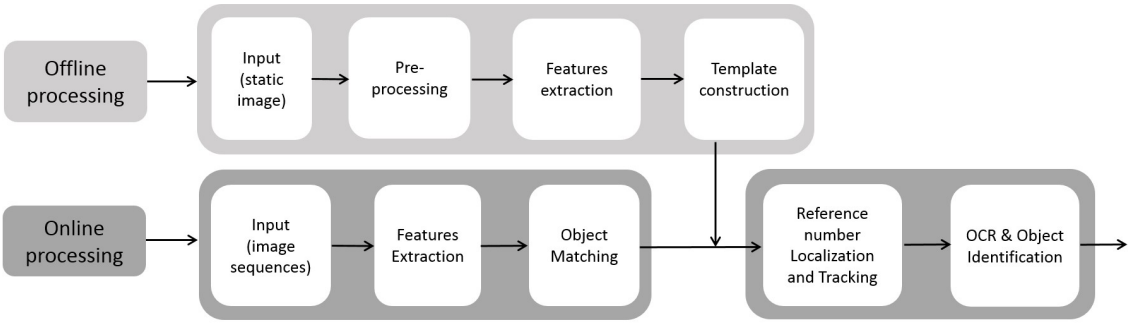


Figure 3.2: proposed framework for reference number localization in image sequence.

3.3/ OBJECT RECOGNITION IN VIDEO SEQUENCE

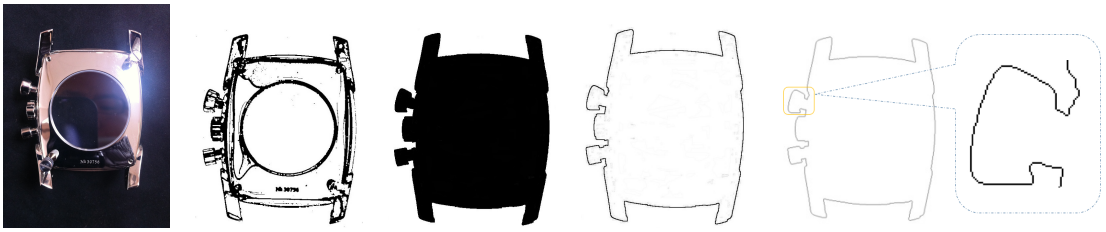
The pipeline of the framework is shown in Fig. 3.2. Offline processing is for the object template construction using 'ideal' static image. In pre-processing, the object external contours are extracted as skeleton, followed by the contour feature extraction. It divides the whole contour into multiple segments and represents these segments by elementary geometric shapes such as line segment and circle arc. Then the template is constructed by the contour segment features. On the other hand, online processing takes a video sequence which takes an unknown moving target object as input. The feature extraction is slightly different from the one in offline processing due to the lack of pre-processing. The details of online feature extraction is presented in the following section. The template matching is used for the purpose of initialization of the object location in the video. Once the contour features are matched to one of the templates, the tracking of contour segments is activated in order to localize the target object in the entire video.

3.3.1/ OFFLINE PROCESSING - TEMPLATE CONSTRUCTION

The objective of the offline processing is to track the reflective objects. Due to the moving reflection on the object surface, the object external contours are the only reliable features. In consequence of that, the features extracted for the object representation are based on external contours only.

3.3.1.1/ PRE-PROCESSING

Since the part-based trackers require the template of the target object in order to be initialized, the object contour template is first created. Therefore, we choose images which present distinct object external contour as the input. As shown in Fig. 3.3, 3.3(a) is the original image, 3.3(b) is the edge detection and binarization of the original image, then we fill the object to obtain 3.3(c), and detect the edge again to obtain the external contour of the object 3.3(d). Finally, we applied morphological operator to obtain the skeleton of the external contour as shown in 3.3(e).



(a) (b) (c) (d) (e)

Figure 3.3: Pre-processing. (a) original image (b) edge detection and binarization (c) fill object (d) edge detection (e)skeleton.

During the pre-processing stage, the Canny filter [5] is employed as the contour detector. It contains a 5×5 Gaussian filter to reduce the noise and derivative operators in the horizontal direction G_x and the vertical direction G_y. From this, the edge gradient G can be determined as:

$$G = \sqrt{(G_x)^2 + (G_y)^2}. \tag{3.1}$$

The morphology formula to obtain the skeleton of a continuous image $X \subset \mathbb{R}^2$ is given by:

$$S(X) = \bigcup_{\rho>0} \bigcap_{\mu>0} \left[(X \ominus \rho B) - (X \ominus \rho B) \circ \mu \overline{B} \right], \tag{3.2}$$

where $\ominus$ and $\circ$ are the morphological erosion and opening, respectively. ρB is an open ball of radius ρ, and $\overline{B}$ is the closure of B. After the pre-processing, we obtain the contour skeleton of the object. The connected pixels which present the skeleton are 8-connection.

3.3.1.2/ CONTOUR DETECTION AND FUSION

The contour skeleton is a continuous edge which presents the global shape of the object. However, features directly extracted from the global contour lead to a significant tracking error once the movement of the object contains a rotation. In order to avoid this problem and furthermore make the tracker available for observing the object rotation, we divide global contour into segments. As the contour is represented by its skeleton, we firstly describe the skeleton by a Freeman code [30], then we cut the contour into segments until the second change of Freeman direction. In this case, each segment contains two Freeman directions. After the division process, the global contour is cut into segments which are line segment or circle arc.

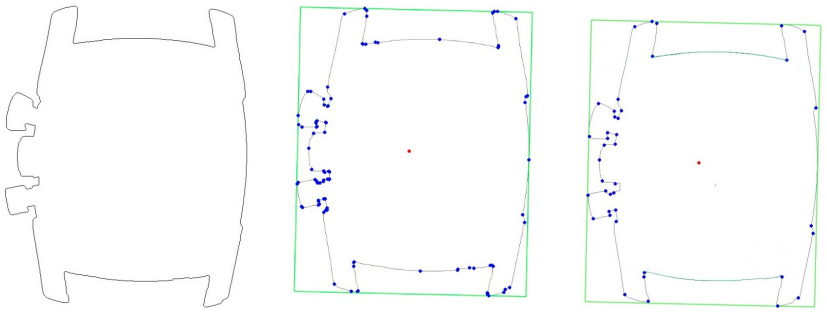


Figure 3.4: Contour segmentation: (a) contour skeleton (b) segmented skeleton (c) re-grouped skeleton.

However, simply cut the contour based on the Freeman code is not an optimized solution, many segments are over divided, for example, a smooth contour can be represented by several segments. This phenomenon is due to the sensitivity of the tiny Freeman direction change in the division process. Thus we present a segments fusion method in order to optimize the contour presentation. The fusion algorithm is illustrated in algorithm 3, for each segment in the divided contour set $S_i, i \in [1, N]$, where N is the number of segments in the contour set, the circle fitting is based on least square error measurement. Finding the least square circle corresponds to finding the center of the circle (X_c, Y_c) and its radius R_c which minimizes the residual function defined below:

$$Ri = \sqrt{(x - X_c)^2 + (y - Y_c)^2}, \qquad residu = \sum (R_i - R_c)^2, \tag{3.3}$$

Algorithm 1 Segments fusion

1. Segments fitting
    1. fit segments into circle and line
    2. calculate fitting error E_i^{fl} and E_i^{fc}
    3. return segment type T_i which has smaller fitting error.
2. Morphology segments fusion
    1. regroup neighbor segments which has the same type
    2. fit regrouped segment into this type
    3. calculate fitting error E_i^r
    4. compare with threshold parameter α to decide regroup/not group
3. Iteration
    1. compare N and N'
    2. if $N! = N'$ process again
    3. else give the output

where R_i denotes the measured radius, (x, y) denotes the position of the pixels on the contour skeleton. Additionally, one candidate segment S_i is fitted to a line and to a circle. The errors of these two fitting are denoted as E_i^{fl} and E_i^{fc}, respectively. The elementary geometric shape of this segment T_i is defined as:

$$T_i = \begin{cases} 0 \quad \text{(line)} & if \quad E_i^{fl} < E_i^{fc} \\ 1 \quad \text{(circle)} & otherwise, \end{cases} \tag{3.4}$$

During the morphology segment fusion step, the similar shape neighbor $S_j, j \in [i-1, i+1], j \neq i$ of S_i is tested. S_j and S_i are first grouped as one segment S_{ij}, then S_{ij} is fitted into the commum shape of these two candidate segments. The fitting error of S_{ij} is compared to a threshold parameter α, where the value of α is set to be 28 through various experiments. After every S_i is experimented, the contour segment set becomes $S'_i, i \in [1, N'], N' \leq N$. If $N' < N$, then a new iteration is activated, if $N' = N$, the segment fusion is terminated. In the Fig. 3.4, 3.4(a) is the original object contour skeleton and 3.4(b) presents the contour division based on Freeman code where the blue points are the ends of contour segments. 3.4(c) is the regrouped contour obtained by our method. The advantage of contour division and regroupment is to distribute a complex shape into multiple elementary shapes (lines and arcs). As the elementary shapes can be repre-

sented by simple formulation, the further template matching and target object tracking can be significantly accelerated.

3.3.1.3/ TEMPLATE CONSTRUCTION

After the contour division and fusion, we obtain a concise representation of the object contour with multi-segments. Each segment contains the local information concerning the object shape: size (Z), shape (T), distance to object gravity center (D), orientation (O), and the include angle (A). Here the (Z) and D are on pixel level, the T is line or arc, presented by 1 or 2. Orientation (O) is the angle from the middle of segment to circle center for arc, or the right angle of the line segment, respectively. The angle (A) is equal to 0 for a line and the include angle for an arc. Thus the feature vector denoted $f(S_i^t)$ of segment i at time t extracted for each segment contains 5 descriptors:

$$f(S_i^t) = \left[Z_i^t, T_i^t, D_i^t, O_i^t, A_i^t\right]; \quad f(C_i^t) \in \mathbb{R}^5. \tag{3.5}$$

The template of an object contains the features of all the contour segments of the object. Each object model corresponds to one template. All the templates are saved in the database for the further object matching and tracking.

Except object external contour features, some global features corresponding to the object shape are also extracted and added into the object features such as global shape (rectangle or circle), number of buttons, as well as the prior knowledge about reference number localization.

3.3.2/ ONLINE PROCESSING - OBJECT MODEL RECOGNITION & REFERENCE NUMBER LOCALIZATION

In order to localize the target object in the sequence of frames, the object model matching is the initialization of the object tracking. In online processing, the object external contour is first found by an adaptive snake model and divided into multiple segments. Moreover, the features are extracted for each segment and used for the template matching. Furthermore, the tracking of all the contour segments provides the location of the target object in the video.

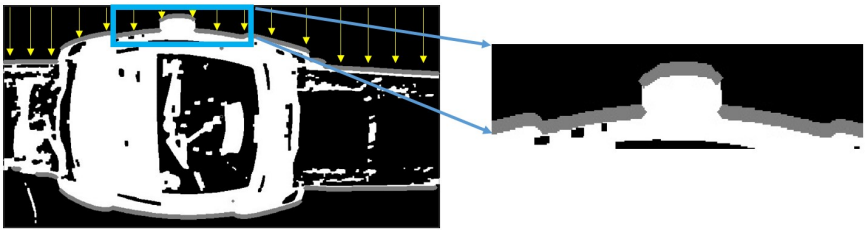


Figure 3.5: Contour searching in the video.

3.3.2.1/ FEATURE EXTRACTION

Compared to the feature extraction in offline processing, in online processing, the contour feature extraction is slightly different. As the external contour is obtained for each frame in the video, the pre-processing of skeleton computation is not available. We use an adaptive snake model to obtain the external contour on the two sides of the object. As shown in Fig. 3.5, two flexible lines (gray) are employed for enclosing the target object. One line moving from top to bottom and the other one moves from bottom to top. If the points on the line touch the object external contour, they stop moving and adhere on the contour. Otherwise, the points on the line keep on moving. As the external contour of the object is not a smooth curve, the two lines are separated into multiple segments. Subsequently, the features are extracted from these segments in the same way as in offline processing.

Based on parameterized external coutour segments, object model information such as global shape and number of buttons can be estimated to characterize the candidate object for the model recognition. These local geometrical features have to be specifically designed for each recognition application.

Global shape estimation The global shape of watch can be highly various and complicated. However, according to our data set, the global shapes of the object that we work with can be ellipse and rectangle (circle shape included in ellipse). The segments of the external contours have been collected during the object tracking. We assume that the long segments which are more centered in the object bounding box are likely to represent the global shape (Tg). Thus the longest segment which is close to the center of the object bounding box is fitted into the circle. For one candidate object ($CO_i, i = [1, 10]$), if the estimated radius $r(CO_i)$ is smaller than a threshold parameter $\rho(Tg)$, the global shape

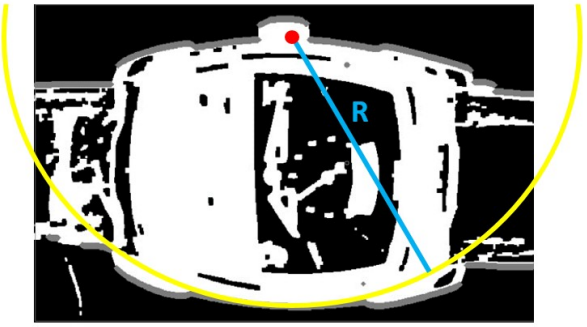


Figure 3.6: Object global shape estimation based on external contour segments.

of CO_i is considered as rectangle, assigned with $Tg(CO_i) = 1$. Otherwise, $Tg(CO_i) = 0$. The $\rho(Tg)$ is set to 300 according to the experimental experiences. The $Tg(CO_i)$ is saved as the first global shape feature. For both the ellipse and rectangle shapes, the radius $r(CO_i)$ is saved as the second global shape feature. As shown in Fig. 3.6, the gray curves are the detected external contour segments. The longest segment which is close to the bounding box center is fitted into the circle to compute the $r(CO_i)$.

Button feature extraction The other local geometrical features are the information about the buttons on the object. They contain the number of buttons and the sizes of each button. According to the external contour segments, a button in the image creates gaps between the ends of two neighbor segments. As shown in Fig. 3.7, the gap between segments S_{k-1} and S_k is hb_i^1, the gap between segments S_k and S_{k+1} is hb_i^2, the gap between segments S_{k-1} and S_{k+1} is wb_i. The hypothesis of being a button is based on the value of these gaps, with the help of threshold parameters ρ_h and ρ_w, where is ρ_h is the height threshold and ρ_w is the width threshold. Through various observations, ρ_h is set to be 24 and ρ_w is set to be 30. However, just setting thresholds for the height and width to verify the detection of the button is not robust. As the movements of the object may contain rotation, all parameters may vary. Thus the positions of the center of all the segments on each side of the object is tracked to estimate the object pose. For the video which contains t frames, the euclidean distance Dc between two centers are computed for each frame. For the whole video, a set of center distance $Dc_i, i \in [1, t]$ is obtained. As the more the object is rotated, the smaller Dc will displayed in the image plane (detailed explanation in Annexe 1). Thus the set is sorted decreasely for searching the frames in which the objects have been less rotated. Then the button verification is affected on

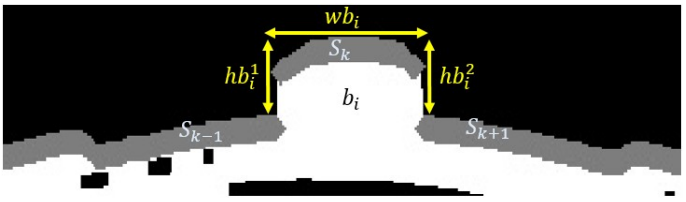


Figure 3.7: Button detection based on external contour segments.

the frames which has the 10% of the Dc in the front of the sorted set. Subsequently, a max voting of decisions for all the button verification is proceeded to obtain the button features. The button features contain the number of the buttons, the side of the button on the object, as well as the width and height.

As the external contour segments are obtained in both offline and online processes, the local geometrical features which are extracted based on contour segments can be used for both reference object and candidate object. Besides the global shape features and button features, previously extracted contour segments features are also used for the further object model recognition.

3.3.2.2/ TEMPLATE MATCHING

The template matching is the initial step for the further object tracking and reference number localization due to the fact that the corresponding model template provides the reliable reference number position. Due to the extracted features represent the object in an hierarchical way, the template matching is in terms of a decision tree as shown in Fig. 3.8. The nodes in the first layer of the decision tree attempt to verify the global shape. As the defined global shape is either ellipse or rectangle, the nodes output binary decision. Hereafter in the second layer, the nodes contain a measurement $\hat{M}_b$ which computes the error button feature matching in terms of $L_2 - norm$ as the equation 3.6. Denote the reference button feature as $F_B = \{f_1, f_2, ..., f_m\}$ where its cardinality $|F_B| = m, m \in \mathbf{R}$, the candidate button feature is as $F'_B = \{f'_1, f'_2, ..., f'_m\}$.

$$\hat{M}_b = \sum_{i=1}^{m} \|f_i - f'_i\|_2. \tag{3.6}$$

The decision in this layer is made by choosing the minimum measured error $\operatorname{argmin} \hat{M}_b$. Afterwards, the nodes in the third layer are measurement $\hat{M}_s$ of the contour segment

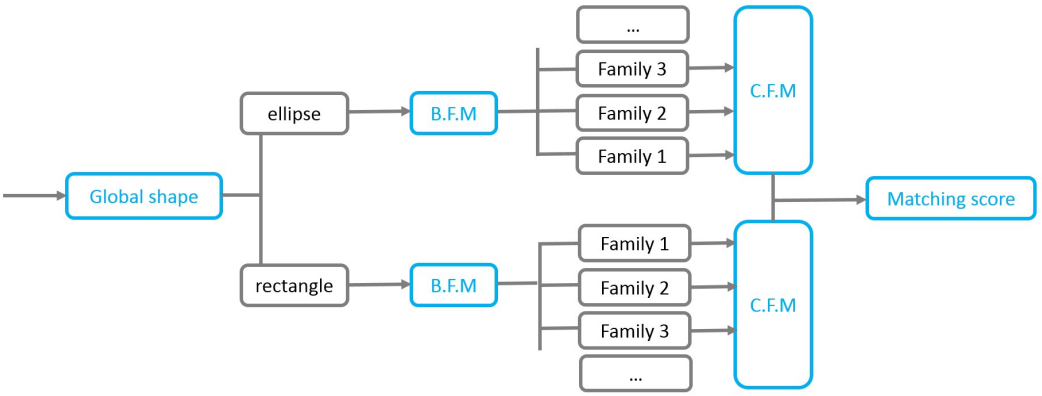


Figure 3.8: Template matching by decision tree. B.F.M: button feature matching; C.F.M: contour feature matching.

matching. As the contour segment detection in online processing is quite sensitive to noise, the longest k contour segments are used in the matching to minimize the error that is raised by the noise. In our case, as the object external contours are not extremely complex, the value of k is set to 5. Denote the reference segment feature as $G_S = \{g_1, g_2, ..., g_n\}$, where its cardinality $|G_S| = n, n \in \mathbf{R}$, the candidate segment feature as $G'_S = \{g'_1, g'_2, ..., g'_m\}$. The measurement is formulated as:

$$\hat{M}_s = \sum_{i=1}^{m} \|g_i - g'_i\|_2. \tag{3.7}$$

Then the matching score is obtained by choosing the minimum error $\operatorname{argmin} \hat{M}_s$ of the contour segment matching. Once the best match is found based on the decision tree, a template matching threshold parameter θ_m is presented to decide whether the object is correctly matched to the corresponding template. If it is correctly matched, the tracking process is activated, otherwise the matching is repeated in the next frame.

3.3.2.3/ CONTOUR SEGMENT TRACKING

Our tracker is composed by an iterative matching computation and inspired by Kalman filter [8]. The tracker is initialized for each detection, the state of a reference segment S_i^t is presented as $St(S_i^t) = \{p_i^t, d_i^t, v_i^t\}$, where p_i^t, d_i^t and v_i^t present the segment position, the moving direction and the moving velocity of the reference segment, respectively. The

state transition density is defined as follow:

$$p_i^t = p_i^{t-1} + v_i^{t-1} \times 1, \quad v_i^t = v_i^{t-1}. \tag{3.8}$$

The sampling processes a predictive circle window with the radius of δ and the center at the position predicted by equation 3.8. Instead of sampling candidate segment (note as Sc_j^t with its state $St(Sc_j^t) = \left\{pc_i^t, dc_i^t, vc_i^t\right\}$) with a weight which costs computationally expensive, a predictive sampling window is employed. Each reference segment in the predictive window is scored by the matching function (equation 3.6) with the reference segment, then $Argmin\left\{St(S_i^t, Sc_i^t)\right\}$ is computed to find the best match. By taking the advantage of tracking each segment independently in the video, the tracking of object is very accurate. In consequence of that, even the object movement contains slightly rotation, the object can be nevertheless correctly tracked.

3.3.3/ Object Identification - Optical Character Recognition

For an object whose model is recognized by the online processing, its reference number provides the unique information about the object ID. The localization and the recognition of the reference number are the key point for the object identification.

3.3.3.1/ Reference number localization

The location of the reference number is manually labeled as a prior knowledge as shown in the Fig. 3.9. The positions of the four vertices shown as purple points are saved in the features for each object. Based on the object template matching and the contour segment tracking, the object is tracked in the video sequence. Then by projecting the reference number locations from the template to the tracked object, the reference number on the candidate object can be located individually in each video frames. More specifically, the matched contour segments are registered to the segments in the template. A transformation matrix is obtained during the registration. The projection of reference number locations is conducted by mapping the vertices with the inverse of the transformation matrix. Once the four vertices are mapped in the video frames, a bounding box can be constructed by linking the vertices sequentially. This bounding box is considered as the detected reference location. However, due to the affine projection, the detected bounding

box is distorted, which means the located reference number in the bounding box is also distorted. As the characters recognition performs poorly with distorted reference numbers, the bounding box is projected again with the transformation matrix in order to obtain the regularized reference numbers.

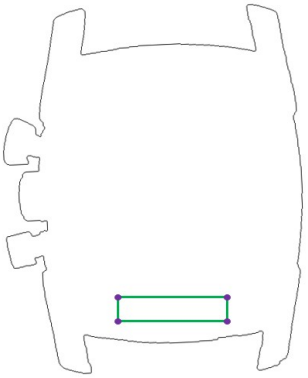


Figure 3.9: Labeled reference number location for the object

3.3.3.2/ OPTICAL CHARACTER RECOGNITION WITH TESSERACT OCR ENGINE

Recognition of the reference number is the last stage of identifying the object in the framework. In each frame, the reference number is cropped from the original frame according to the detected boundingbox. Denote the cropped image which contains the reference number as imagette. Each imagette is binarized and input into *Tesseract OCR engine* (more details refer to [31]) in order to recognize the reference number. The recognized characters from a video sequence is denoted as $CR = \{cr_1, cr_2, ..., cr_N\}$, where N is the total frame number of the video and $cr_i, i \in [1, N]$ is a string of characters. Then, among CR, a max voting process is employed to find the most repeated reference number as the output decision.

3.4/ RESULTS

As explained in the previous chapter, the experiments are conducted in using the camera with 5Mpx resolution. The videos contain from 50 to 200 frames. A rotation (-30 degrees to 30 degrees) platform is used to move and rotate the object. The experiments are conducted on 45 videos and the tested objects are 10 different models from different brands. The results of object template matching and target object localization are

presented, respectively.

3.4.1/ TEMPLATE MATCHING

According to the accuracy of template matching, the confusion matrix 3.10 is shown. For most of the objects, the accuracy of correctly matched object is 100%. However for the object O_3, O_8, and O_9, several of their test data are miss classified into object O_{10}. It is due to the similarity of these models, and the complexity of the contour segment template.

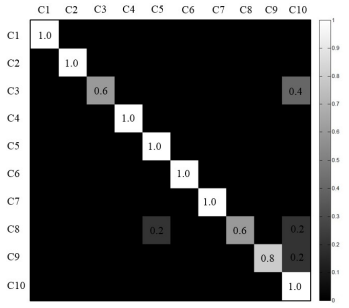


Figure 3.10: Confusion matrix for object model recognition.

The time consuming of the object model recognition is shown in table 3.1. For all the object models, the recognition time is under 0.15 second. Besides that, the recognition time is under 0.1 for object O_6, O_8, O_9, and O_{10}. It is because of the less complexity of the contour segment template which yields a simpler template matching.

Object	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8	O_9	O_{10}
time(s)	0.14	0.13	0.11	0.10	0.12	0.08	0.12	0.08	0.07	0.09

Table 3.1: Processing time for object model recognition.

3.4.2/ TARGET OBJECT TRACKING

According to the object localization, we mainly evaluate the object tracking performances. Due to the lack of space, 5 experiments results are shown in Fig. 3.11. It shows that the tracking of contour segments in all the experiments keeps reliable. However, while the segments of the contour are very small as in the third column, the tracked segments slightly change their positions on the contour. It is due to the low features matching

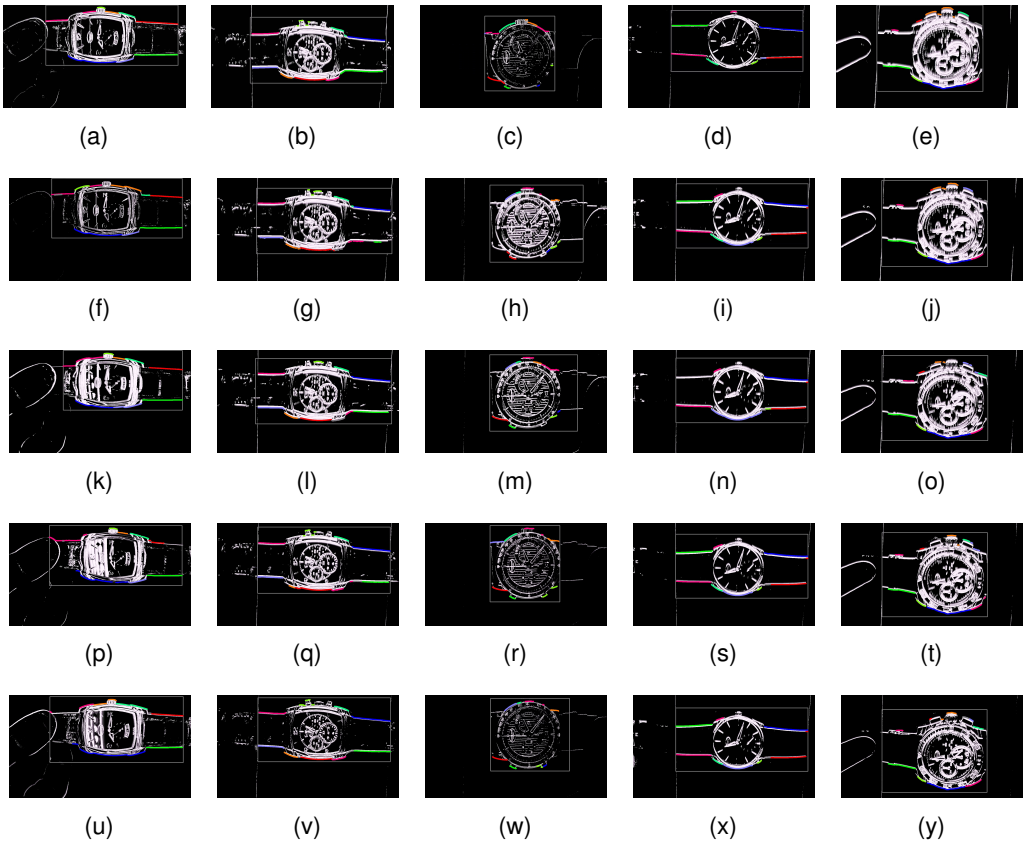


Figure 3.11: Object contour segments tracking: One color represents one contour segment tracked, gray box describes the measured object location.

threshold. Whereas the movement of the object contains perspective rotation, the matching threshold can not be extremely high.

To quantitatively evaluate our approach, we propose two criteria. One is the distance (D_t) between ground-truth bounding box center and estimated bounding box center. The evaluation is conducted with a pixel level threshold ϕ which signify the correct tracking in the current frame t, here we set ϕ to 30. The tracking performance T_t is computed as:

$$T_t = \begin{cases} 1 & D_t < \phi \\ 0 & D_t > \phi \end{cases} \tag{3.9}$$

Another criterion is the position of contour segments. For each i contour segments in the frame, Pos_i presents the center position of each segment, and $PosG_i$ is the center position of $i-th$ ground-truth contour segment. Then the center positions are compared based on the euclidean distance:

$$A(Pos, PosG) = \sum_{i=1}^{n} \sqrt{(Pos_i.x - PosG.x)^2 + (Pos_i.y - PosG.y)^2}. \tag{3.10}$$

This criterion concerns the spatial tracking performance. A pixel level threshold parameter δ is employed to signify the fine-gained tracking performance in the current frame, here δ is set to 20.

$$S_t = \begin{cases} 1 & A(Pos_i, PosG_i) < \delta \\ 0 & A(Pos_i, PosG_i) > \delta \end{cases} \tag{3.11}$$

Within these two criteria, our approach is evaluated by summing the correct bounding box and contour segments tracking in entire videos. The results are shown in the table 3.2. The results of box tracking rate confirms the performance of global object tracking. For objects O_1, O_2, and O_4, the tracking rate achieve 100% which means the object are correctly located in all the frames. Objects O_3 and O_5 are not perfectly located in all the frames, but the tracking rate are still both higher than 98%. For the segments tracking, our approach are not able to track all the segments in all the videos. Especially for the objects which contain small contour segments. However, our approach is still capable to track most of the segments correctly.

TAC	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8	O_9	O_{10}
$\hat{B}$(%)	100	100	98.90	100	98.25	100	100	98.90	100	98.25
$\hat{S}$(%)	98.26	95.83	89.67	93.85	92.04	98.26	95.83	89.67	93.85	92.04

Table 3.2: Quantification Results for object bounding box tracking and segments tracking. TAC is the tracking accuracy, $\hat{B}$ is the object bounding box tracking rate, $\hat{S}$ is the segment tracking rate.

3.4.3/ REFERENCE NUMBER LOCALIZATION

Another evaluation is performed by comparing the ground-truth window to the localized window by our proposed method. It aims to evaluate the performance of tracking details on the target object. The ground-truth window is set as the bounding box of the reference number, the localized window is based on the localized object bounding box. The results are shown in Fig. 3.12, first column contains the original frames 20, 101, 161, 171, and

222, respectively. The second column presents the segment tracking, the third column shows the localization of the target object, the blue window is the ground-truth and the green window is the detected reference by the proposed method. We can see that the segment tracking is very accurate and stable. Quantitatively, we present a superposition performance criteria $A(S_t, G_t)$ between two windows (the one we tracked S_t and the one of ground-truth G_t) which contain the reference number. The ground-truth is manually generated in order to perform the quantitative evaluation. These two windows are compared by the following equation:

$$A(S_t, G_t) = \frac{Surface(G_t \cap S_t)}{Surface(G_t \cup S_t)}, \tag{3.12}$$

then the tracking performance (AT) in the whole video is presented by:

$$AT = \frac{1}{N} \sum_{t=1}^{N} A(S_t, G_t). \tag{3.13}$$

The results of superposition are illustrated in table 3.3. The AT are between 71% and 82% for all the objects.

Object	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8	O_9	O_{10}
AT(%)	81.91	77.83	71.93	80.42	73.76	81.91	77.83	71.93	80.42	73.76

Table 3.3: Superposition evaluation.

3.4.4/ OPTICAL CHARACTER RECOGNITION

45 videos in the dataset were tested for the character recognition based object identification. More precisely, for one video, the reference number recognition decision is string of characters $Dr_i = x_1x_2x_3...x_n$, where i is the video index and n is the total number of the characters in the decision. As the object identification is based on the whole string of the recognized reference numbers, only the correct recognition of all the characters yields to the identification. Thus one false character recognition leads to mis-identification. The result is evaluated on the object identification accuracy as shown in Tab. 3.4. For the objects O_4, O_5, O_7, and O_9, the identification accuracy are 80% which are meaningful results. However, for the rest of the objects, the identification is not very accurate, especially the

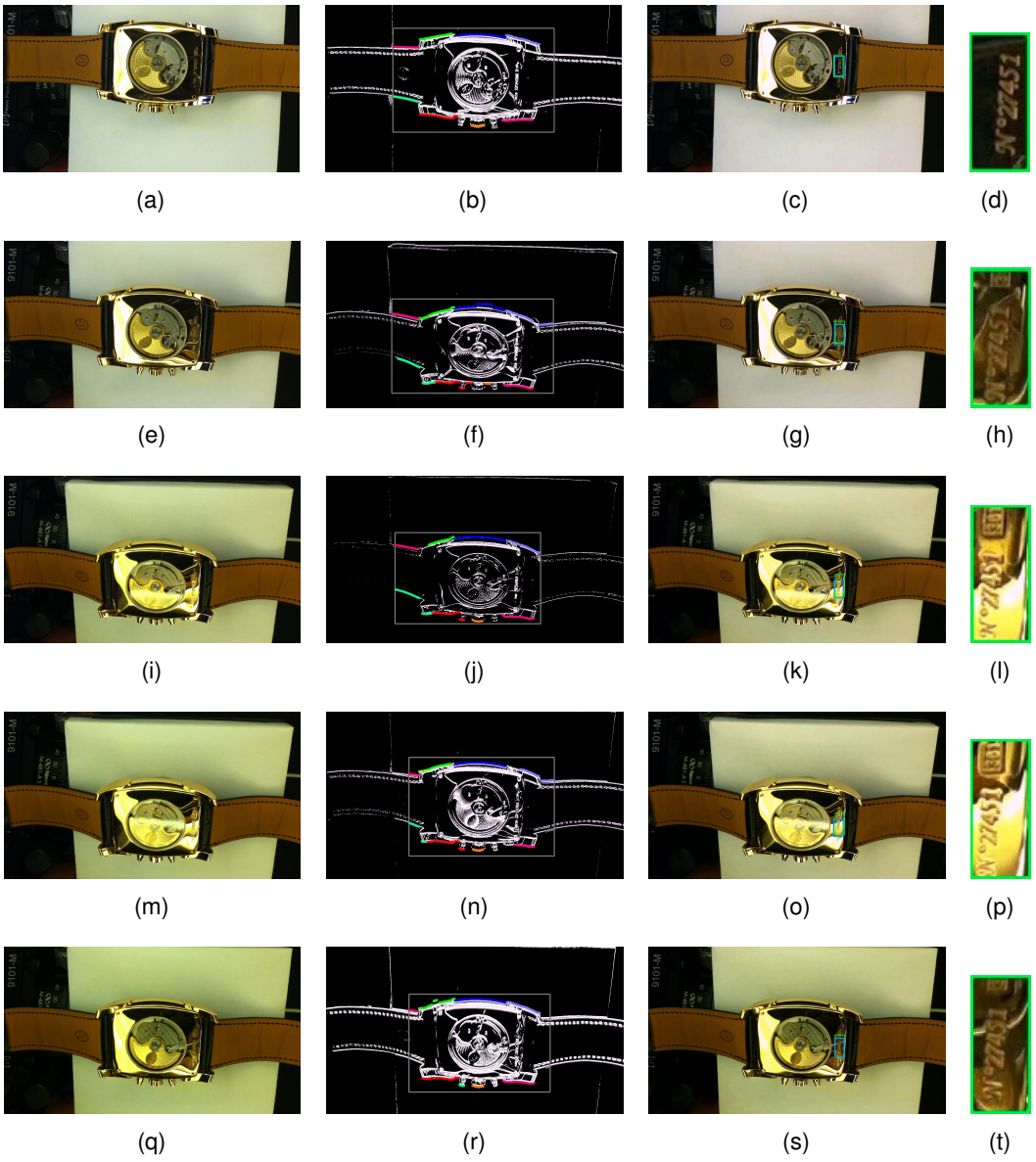


Figure 3.12: Object fine-gained tracking in the video. First column: original frames, second column: segments tracking, third column: reference zone localization.

object O_8 has never been identified in all its videos. The reasons which lead to the poor identification accuracy are various, such as low resolution, the tesseract OCR engine is not invariant to the reflection, as well as the low contrast between the characters and the background.

Object	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8	O_9	O_{10}
AC(%)	40	60	40	80	80	20	80	0	80	40

Table 3.4: Object identification accuracy.

3.5/ CONCLUSION

In this chapter, we present an object recognition framework to deal with manufactured reflective objects. The framework contains two preliminary stages and one latter stage: offline processing and online processing as the preliminary stages and object identification as the latter stage. The offline processing aims to build the template for the reference objects and online processing attempts to recognize an unknown object model through the input video. Then object identification finds the exact piece of the object through reading the reference number.

In the offline processing, the template construction is mainly based on the local descriptors and contour segment features. The advantage of using low level features is to gain the low computational cost and feature dimensions. Moreover, the extracted global shape features, button features, and contour segment features are degraded in terms of globability. In consequence of that, the features are organized in the forms of hierarchical decision tree that facilitates the further template matching. In online processing, external contours are detected by enclosing the object with two adaptive lines. Then the extracted features are matched with all the templates in order to find and localize the target object in the video. By tracking contour segments in the following frames, the object can be located. Hereafter in the object identification, the reference number is located in each frame with the prior knowledge of its position according to the template, followed by character recognition with tesseract engine. Once the reference number of the target object is correctly recognized, the object is identified. The results are evaluated through 4 aspects that are template matching, target object tracking, reference number localizing, and object identification. These 4 aspects evaluate sequentially the performances of the different functions of the prototype. Through the experiments on the acquired dataset, the first three performances are meaningful whereas the object identification is not satisfactory.

3.5.1/ LIMITATIONS AND RAISED TOPICS

The prototype has several limitations that yield the poor object recognition performance. However, these limitations raise interesting research topics that we can investigate in.

The first limitation is the features that are merely extracted from external contour. The presence of the reflection on the involved objects causes the non-available of most of

the often-used features in the state of the art. In this case, the first though about the solution is obviously using the only robust signature 'external contour'. However, using the features that extracted only based on the external contour are more likely to create enormous bias. Furthermore, if the reference objects have similar shape, these contour based features may scarcely be representative. Therefore, in order to achieve high object recognition performance, extracting and using additional features to describe the objects are imperative. According to this, we investigate in searching other features that are highly representative. Through various observations during the experiments, we find that the environment are distorted in the reflection and this deformation is strongly related with the object local surface curvature. In consequence of that, we aim to extract the object local surface curvature information by analyzing the deformation of the reflection. Since the object local surface curvature is a fine-gained feature, it is highly representative for distinguishing the similar object models.

The second limitation of the prototype is escaping from the problem raised by the reflection. Undeniably, reflection brings enormous negative effects that make the object recognition task more challenging. Most of the reflection-involved works in the state of the art attempt to remove or reduce the reflection before processing the method for their main task such as object recognition, tracking, segmenting. Very few works aim to employ the reflection as additional information. After an intensive study, we find that the reflection move smoothly along a continuous surface and irregularly while passing from one surface to another. Thus the reflection motion contains local geometric information about the object surface. In this regard, we propose a elementary surface segmentation method which extracts the reflection motion for the surface labeling. Building a graph based on the segmented elementary surfaces is a new strategy to represent the surface distribution of the object. Instead of removing/reducing reflection, taking its advantage is pioneering the work in a new direction.

The third drawback of the prototype is the reference number tracking. It is highly depending on the contour segment tracking and object model matching. Too many previous stages yield the inaccuracy of the reference number tracking. In order to solve the inaccuracy tracking problem and make sure about the reference number covering, the prior knowledge concerning about the reference number position is set to be 5 times bigger the reference itself. However, the oversize bounding box leads to another problem: insuf-

ficient resolution for the character recognition. Therefore, we aim to find a straightforward method that detects the text in the image without any prior knowledge or initial stages. Various efforts have been contributed on text detection in the image of natural scene. However, to the best of our knowledge, detecting text on reflective surfaces has not been studied yet. Thus, we present a text detection method which specifically handle the problem of reflective surfaces.

The last limitation is the lack of recognition system especially designed for reflective characters. Despite the fact that Tesseract OCR engine is one of the most successful framework for the character recognition, it is still far from handling the characters that are engraved on the reflective surfaces. Since the reference number recognition is the key of object identification, the poor OCR accuracy is obviously the short board of the object recognition prototype. Consequently, we present a novel method which is specifically for the reflective character recognition. The proposed method initially extracts static based features to make the recognition scale invariant and less sensitive to the reflection, then employs cascaded SVM classifier in order to boost the recognition accuracy. In order to validate the proposed method, a challenging dataset containing more that 1500 characters is built and released for further research purpose.

II

Part II: Further in problematic

4

Local Surface Curvature Analysis

In this chapter, we aim to extract additional features in order to fulfill the lack of robust representative features. As in the initial prototype, external contour features are the only extracted features for the object tracking, the template matching, and the object model recognition. Moreover, the movement of the object contains perspective rotation which yields the transformation of the external contours. Thus, the feature extraction process is indispensable to be enhanced by gaining more robust features. In consequence of that, the object local surface curvature is extracted as an additional feature. The proposed local surface curvature analysis method focuses on analyzing specular surface curvature profile using a single line source. This single line source could be any straight line in the environment whose position can be easily measured. By studying the geometric relation in the system, the local surface curvature of the experimented object can be estimated according to the distortion of the line that is reflected by the object surface.

4.1/ Introduction and related works

Object surface curvature can be another important feature for the objects model recognition. For non reflective objects, traditional 3D reconstruction methods such as using monocular camera or stereo-camera through Structure-from-Motion techniques provides nice results. However for reflective surface objects, such methods are not suitable because of the change of surface texture from different view poses and matching these distorted features from different scenes is not robust and accurate. Thus we present a

specular surface curvature analysis method in using few constraints such as a straight line source.

In literature, the constraints applied in most of the works are very restrictive in order to study the local geometry on specular objects. Savarese and Perona [32, 33] proposed an analysis of the relationship between a calibrated scene composed of lines through a point, and the geometry of curved mirror surface on which the scene is reflected. This analysis can be performed on a perfect reflective object by using a triangle check board. However, such a perfect textured and calibrated pattern as check board is unrealistic.

Another way to study specular objects is using the ideas of optical polarization. Barrois and Wohler [34] presented a method which incorporates different channels of information, one of which is a polarization angle of light reflected from the object surface that provides information on rotation of an object relative to the camera. Atkinson et al. [35] worked on exploiting polarization by surface reflection, using image of smooth dielectric objects to recover surface normal and height. However, some other works use textured light sources as [36, 37, 38, 39]. Zheng and Murata [40] developed a system in which extended lights illuminate a rotating specular object whose surface is reconstructed by analyzing the motion of highlight stripes. In [39], Halsead et al. proposed a reconstruction algorithm where a surface model is fitted to a set of normals by imaging a specularly reflective surface with a light pattern.

Our method focuses on analyzing specular surface curvature profile using a single straight line source. This single line source could be any straight line in the environment whose position can be easily measured. Fewer constraints make the experiment easy to manipulate. Also, a single line source does not require the feature extraction to be extremely accurate. In other words, the object surface does not need to be a perfect mirror. In consequence, our approach performs correct analysis for specular surface, even for transparent objects.

4.2/ THE GEOMETRY OF THE REFLECTION

We start by estimating geometric information from a specular surface. Our framework includes a camera (C) to capture the reflection of a straight-line light source (S) from a hemisphere object. The fixed LED grow lights are covered by a semi-transparent white

paper to produce a homogeneous lightening condition. The border of the cover paper is a straight line which is fixed on the support in the experimental box. In Fig. 4.1, the solid arc denotes the profile of maximum cross section later referred as center section. The dash-dot arc represents the profile of the section crossing the middle of radius later referred as middle section. In the profile view (left part of Fig. 4.1), O is the original of the coordinate (and also the projection of camera on X axis), while A and B are the reflection points. Lines $\overline{AA'}$ and $\overline{BB'}$ are perpendicular to Y axis, with their lengths denoted by D_a and D_b respectively. The projection of normal vector on point A is overlapping the normal vector on point B. To the top view, we denote the object radius R_s, the reflection radius R_r, the angle of incident light which passes through the object center and line source α_{os}, the angle of reflection light which passes through object center and camera α_{oc}, the angle between Y axis and normal vector α_{on}.

Our setup separates a geometry relation into two steps: 1. The Fig. 4.1 profile view helps us to study geometrical relations between the position of specular object, the reflection point and the camera. 2. From top view, the relationship between the object surface curvature and the reflection curvature is discovered. With these two measurements, the object surface curvature can be computed from the reflection curvatures.

4.2.1/ Assumption and objective

In our setup, the positions of the camera, light source, and object are known. The line light source is reflected by the object surface to the image plane of the camera. According to the local curvature of the object surface, the line light source is reflected as a curve on this local surface. Our objective is to estimate the curvature of the object local surface from the reflection curvature by analyzing their geometrical relationship.

4.2.1.1/ Definition and basic specular reflection constraints

We put profile view in Fig. 4.1 into 2D coordinates, the projection of camera on X axis being original point. The angles mentioned above can be expressed by the positions of the components in the setup as follows:

$$\alpha_{os} = \tan^{-1}(\frac{S.y - O.y}{S.x - O.x}), \qquad \alpha_{oc} = \tan^{-1}(\frac{C.y - O.y}{C.x - O.x}), \tag{4.1}$$

$$\alpha_{on} = \frac{\alpha_{os} + \alpha_{oc}}{2}, \qquad \beta = abs(\frac{\phi}{2} - \alpha_{on}). \tag{4.2}$$

As the object is a hemisphere, the object center is on the horizontal axe. Moreover, the camera is placed on the vertical axe, which means:

$$O.y = 0, \qquad C.x = 0. \tag{4.3}$$

By combining equations 4.1, 4.2 and 4.3, β can be computed by:

$$\beta = abs(\frac{\phi}{2} - \frac{\tan^{-1}(\frac{S.y}{S.x-O.x}) + \tan^{-1}(-\frac{C.y}{O.x})}{2}). \tag{4.4}$$

In top view, point A'' presents the projection of A on center section, O_r is center of circle fitted by reflection, D_r is the distance between O_r and object center. Reflection points A and C are both on middle section, point B is on center section. As A, B and C are all covered by the reflection trace, the reflection curvature can be measured. Moreover, the distance between A and $A''$is a half of the object radius, so the expression of object curvature based on reflection curvature can be established.

The projection distances D_a, D_b can be expressed as:

$$D_b = \frac{\sqrt{3}}{2} R \sin(\beta), \qquad D_a = R \sin(\beta). \tag{4.5}$$

The geometric relationships between projection distances and curvatures are:

$$\overline{AA''} = \frac{1}{2} R, \qquad R_r = D_b + D_r. \tag{4.6}$$

As $\Delta AA''O_r$ is a right triangle, from Pythagorean Theorem, we obtain:

$$\overline{AO_r}^2 = \overline{AA''}^2 + (D_a + D_r)^2. \tag{4.7}$$

As $\overline{AO_r}$ is radius of the reflection R_r, equation 4.7 can be transformed as:

$$R_r^2 = (\frac{R}{2})^2 + (D_a + D_r)^2. \tag{4.8}$$

By combining equations 4.5 4.6 with 4.8, we obtain

$$D_r = \frac{\cos(\beta)^2 R}{4(2 - \sqrt{3})\sin(\beta)}. \tag{4.9}$$

Once D_r is computed, by combining the equations 4.6, 4.1 and 4.2, it is straightforward to conclude that the object curvature R and the reflection curvature R_r are parameterized by β:

$$R_r = R\sin(\beta) + \frac{\cos(\beta)^2 R}{4(2 - \sqrt{3})\sin(\beta)}. \tag{4.10}$$

From this equation, we can compute the surface curvature from the reflection curvature if the given object is a hemisphere. We consider non-hemisphere objects in the next section.

4.2.2/ EXTENSION TO NON-HEMISPHERE OBJECTS

For a non-hemisphere object, we assume that the surface locally corresponds to a sphere. The object center is no longer on the horizontal axis. So there is one additional unknown parameter which is the distance from bottom surface to sphere center, denoted by h. For convenience, we call it center height. In order to estimate object surface profile, we need to estimate the surface curvature R and the center height h in the meantime. As we can obtain only one equation from geometry, we need one more equation. The first solution is to move the object, since the curvatures of reflection change while object surface is moving. We take two frames in which object is located at two known positions to obtain two reflection curvatures. In this case, two equations can be found to solve surface curvature and center height. Another possible solution is to use the fact that in one frame, we have a pair of identical surfaces which are reflecting the same line source. Since the geometry in both cases is exactly the same, we present only two identical spheres case to avoid the redundancy.

Similarly to the case of one sphere, described in the section 4.2.1.1, we can obtain these angles by known position parameters as:

$$\alpha_{o_1s} = \tan^{-1}(\frac{S.y-h}{S.x-O_1.x}), \quad \alpha_{o_2s} = \tan^{-1}(\frac{S.y-h}{S.x-O_2.x}), \quad \alpha_{o_1c} = \tan^{-1}(\frac{C.y-h}{C.x-O_1.x}), \quad \alpha_{o_2c} = \tan^{-1}(\frac{C.y-h}{C.x-O_2.x}), \tag{4.11}$$

$$\alpha_{o_1c} = \tan^{-1}(\frac{C.y-h}{C.x-O_1.x}), \quad \alpha_{o_2c} = \tan^{-1}(\frac{C.y-h}{C.x-O_2.x}), \quad \alpha_{o_1n} = \frac{\alpha_{o_1s} + \alpha_{o_1c}}{2}, \quad \alpha_{o_2n} = \frac{\alpha_{o_2s} + \alpha_{o_2c}}{2}, \tag{4.12}$$

$$\beta_{o1} = abs(90 - \alpha_{o_1n}), \quad \beta_{o2} = abs(90 - \alpha_{o_2n}). \tag{4.13}$$

We can see that finally β_{o1} and β_{o2} are parameterized only by h. Based on this geometric information, we could turn to the relationship among reflection curvatures, object curvatures and object center height. According to the equation 4.10, the relationship is presented as:

$$R_{r1} = R(\sin(\beta_{o1}) + \frac{\cos(\beta_{o1})^2}{4(2 - \sqrt{3})\sin(\beta_{o1})}), \quad R_{r2} = R(\sin(\beta_{o2}) + \frac{\cos(\beta_{o2})^2}{4(2 - \sqrt{3})\sin(\beta_{o2})}), \tag{4.14}$$

$$R_{r2} = R(\sin(\beta_{o2}) + \frac{\cos(\beta_{o2})^2}{4(2 - \sqrt{3})\sin(\beta_{o2})}), \tag{4.15}$$

then by combining equations 4.14, 4.15 and by eliminating R, we obtain:

$$\frac{R_{r1}}{\sin(\beta_{o1}) + \frac{\cos(\beta_{o1})^2}{4(2-\sqrt{3})\sin(\beta_{o1})}} = \frac{R_{r2}}{\sin(\beta_{o2}) + \frac{\cos(\beta_{o2})^2}{4(2-\sqrt{3})\sin(\beta_{o2})}}. \tag{4.16}$$

In this equation, β_{o1} and β_{o2} could be written as function of h, R_{r1} and R_{r2} could be computed from the image, so h can be deduced from equation 4.16. Once h is obtained, R can be deduced from either equation 4.14 or 4.15.

4.3/ SURFACE RECOVERY

Now we assume that we can measure the object surface curvature R, as well as the center height h. The object center position and reflection point positions can be obtained as functions of these two parameters.

As illustrated by the Fig. 4.3 for both spherical and paraboloid surfaces, once the surface curvature and center height are known, the center of local curved surface O_s can be measured. As the object center position on horizontal axe is a known parameter, with the obtained height, the center position in 2D coordinate is known. Also, the local object surface curvature R is computed, hence we can draw a local circle (dash-dot arcs in the Fig. 4.3) with the center positions and local surface curvatures, reflection point on the surface can be estimated as the intersection of local circle and object surface. Once the reflection point is found, the direction of normal vector can be calculated, furthermore, the tangent of observed local curve at reflection point is measured.

4.4/ EXPERIMENTAL RESULTS

In our setup, the object position, light source position, and the camera position are controlled, additionally with the object parameters offered by the manufacture, the surface curvature of the watch feet can be estimated. The reason of choosing watch feet as the main studying surface is that every watch has symmetrical feet for connecting the bracelet. Moreover, the surface curvatures of the feet are uniform. These conditions perfectly fit the assumption of two symmetrical surfaces with the same curvatures.

In table 4.1, R_M is the notation for the measured radius while R_E is the value of the radius obtained by our method. CH is the notation of Center height. AC_r and AC_h are the accuracy of radius and height, respectively. The results show that our approach works not only on spherical surface, but also on paraboloid surface. In the first experiment on metallic balls, two identical balls are located next to each other, so the distance between two centers is equal to the diameter of the ball. The radius and center height are measured by a measuring instrument. In the second experiment on sunglasses, the radius, the distance between two centers and minimum thickness of the glass are given by the optometrist. As the glasses are flat-wise on the support table, the center height can be measured

by taking the sum of the glass radius and the minimum thickness. The accuracy of the measurement of object radius and height are all above 98%.

Object	R_M	R_E	AC_r	CH_M	CH_E	AC_h
ball	1.60cm	1.5955cm	0.9972	1.10cm	1.113cm	0.9882
glasses	125.0cm	123.73cm	0.9898	125.2cm	124.13cm	0.9915

Table 4.1: Quantification Results.

In figure 4.4, the first column contains the original images with reflection fitted circle. The second column contains measured surface (red) and estimated surface by our approach (blue). From the original images, we can see that the surface of metallic balls is not a perfect mirror and the glasses are transparent. According to the results above, just with one straight line source, the measurements of our proposed method are very accurate for both spherical and paraboloid surface, even the surface is not perfect mirror or a transparent one.

Object	O_1	O_2	O_2	O_2	O_2	O_6	O_7	O_8	O_9	O_{10}
R_M	2.50	3.15	2.65	2.50	2.84	3.15	2.50	2.94	3.66	2.32
R_E	2.46	3.21	2.64	2.48	2.92	3.11	2.44	2.99	3.43	2.29
AC_r	0.98	0.98	0.99	0.99	0.97	0.99	0.98	0.98	0.94	0.98
CH_M	0.42	1.12	0.56	0.42	0.96	1.12	0.42	1.13	0.78	0.98
CH_E	0.41	1.10	0.56	0.41	0.95	1.11	0.42	1.10	0.81	0.99
AC_h	0.98	0.98	1.0	0.99	0.99	0.99	1.0	0.97	0.96	0.99

Table 4.2: Results for local surface curvature estimation.

Based on this method, we also tested the objects in the dataset. Since for each watch, the feet are mostly symmetric and its curvature is highly representative. Moreover, the curvatures of the feet are very different from these of one model to those of another one, thus we extract the local curvature of the feet as additional feature and the results are shown in the Table. 4.2. The ground-truth local curvatures and the center heights are given by the manufacture.

4.5/ CONCLUSION

In this chapter, we have described an estimation method for computing specular surface curvature with one straight line source. It is based on a local geometry analysis between

object surface curvature and reflection curvature with known positions of camera, light source, and object. As we need only one light source as a pattern, many kinds of sources in the real scene can be used to produce reflections on the surface. We used a LED grow light for one experiment and a roof window in another one. Experiments carried out are on both spherical surface and paraboloid surface. We obtain curvature of local surface on reflection point and the height of reflection center as results which describe the local surface profile. The method is experimented for the project involved manufactured objects (watches). The feet curvatures of each watch model have been accurately estimated by the proposed method. Moreover, the local surface curvature features can be used as highly representative descriptor for the object recognition. Regarding to the future investigation, the local surface curvature analysis leads to the 3D reconstruction of the entire reflective object and estimating the surface curvature of a complex shape under few constraints. Moreover, to enhance the analysis of local surface curvature, understanding the surface structure of an entire reflective object is quite crucial. Thus we will propose a novel segmentation based method to extract object surface graph in the next chapter. The presented local surface curvature analysis is published in International Conference of Digital Image Processing in 2015.

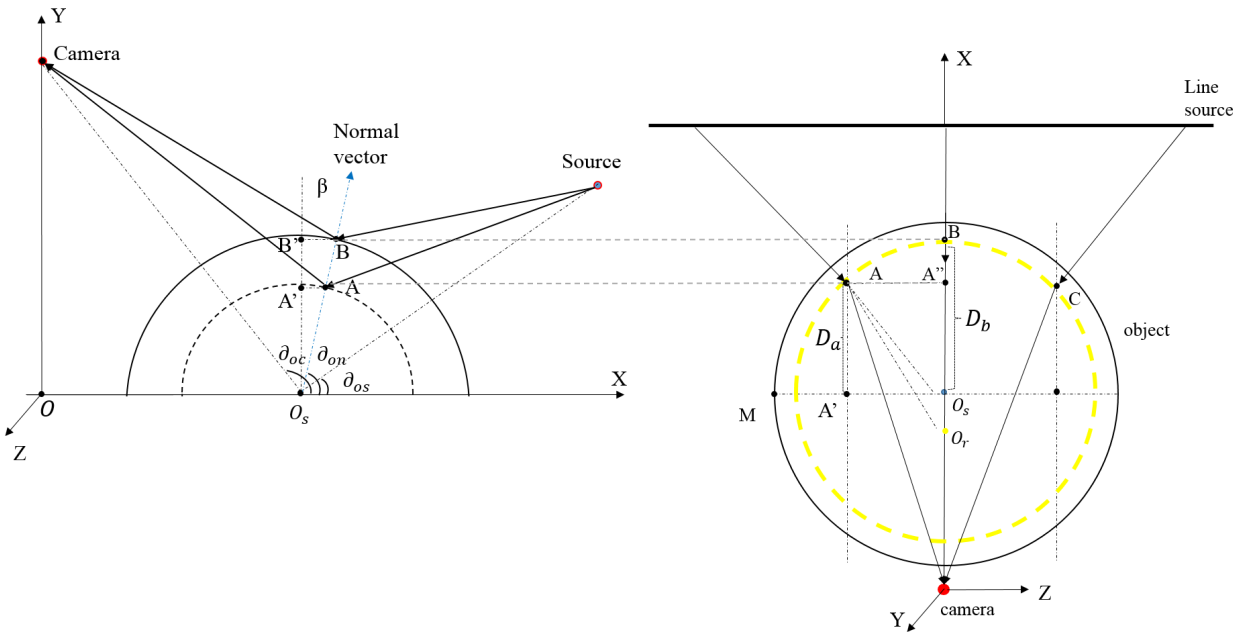


Figure 4.1: Analysis of hemisphere specular surface. The figure on the left is the profile view, the view direction is parallel to line source, so the line source is shown as a point.O is the original of the coordinate (and also the projection of camera on X axis), while A and B are the reflection points. Lines $\overline{AA'}$ and $\overline{BB'}$ are perpendicular to Y axis, with their lengths denoted by D_a and D_b respectively. The projection of normal vector on point A is overlapping the normal vector on point B. The figure on the right is top view. we denote the object radius R_s, the reflection radius R_r, the angle of incident light which passes through the object center and line source α_{os}, the angle of reflection light which passes through object center and camera α_{oc}, the angle between Y axis and normal vector α_{on}.

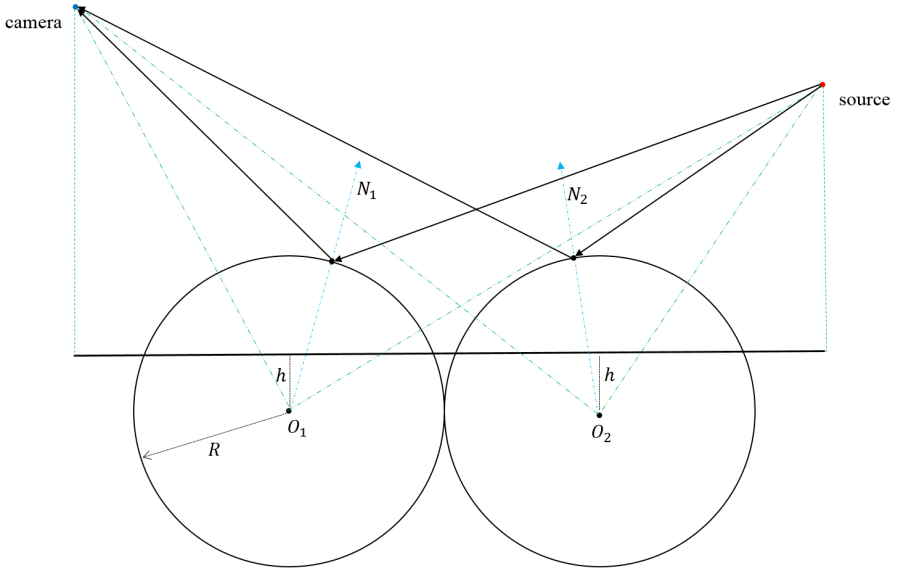


Figure 4.2: Geometry of two identical surfaces. Radius of both spheres are R, two centers are denoted as O_1 and O_2, object center height is h, normal vectors who pass reflection points are N_1 and N_2, angles of incident lights which pass object centers and line source are α_{o_1s} and α_{o_2s}, angles of reflection lights which passes object centers and camera are α_{o_1c} and α_{o_2c}, angles between vertical axe and normal vectors are α_{o_1n} and α_{o_2n}, angles between vertical axe and normal vectors are β_{o1} and β_{o2}.

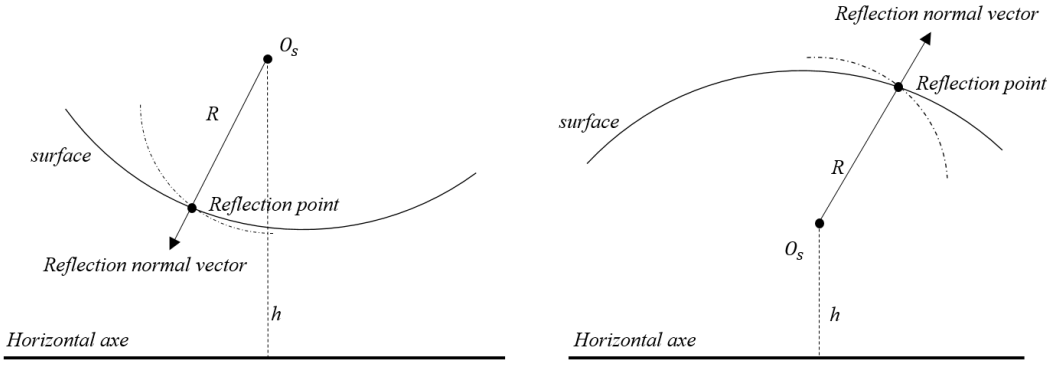


Figure 4.3: Object surface recovery by curvature and center height. The image on the left presents the geometry of a spherical surface; the image on the right presents the geometry of a paraboloid surface.

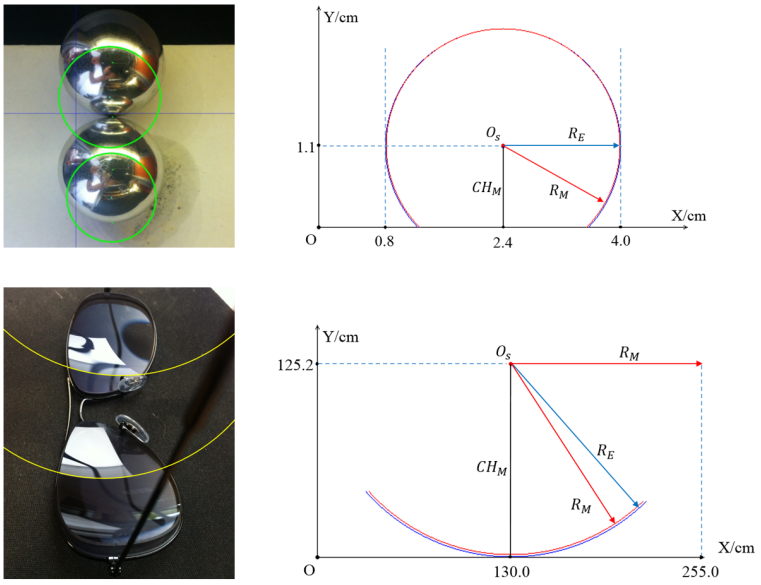


Figure 4.4: surface curvature and center height estimation. First row: spherical surface; second row: paraboloid surface. CH is the notation of Center height. AC_r and AC_h are the accuracy of radius and height, respectively.

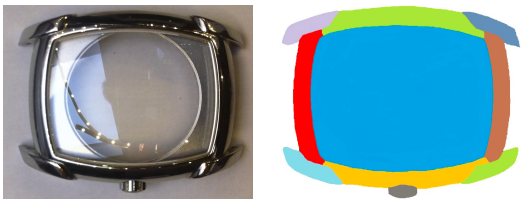
5

Reflective Object Surface Structure Understanding

In this chapter, we will discuss how the reflections appearing on the image can be treated. Obviously, reflection brings negative effects that makes the robust feature extraction more challenging and the object recognition more complicated. The most common way is to remove or to reduce the reflections. However, removing or reducing reflections does not add any information concerning the image. It means that in our case it would be better to try to extract supplementary information from the reflections. Such information could provided by moving the reflection trace along the surface and studying it's moving trajectories. The proposed method first tracks the moving reflection particles in the video, then uses the motion trajectories as surface labels, and finally segments extracting the additional information from the motion trajectories is definitely. After the object surface segmentation, the graph which describes the surfaces distribution is constructed as a new feature for the object representation. The proposed segmentation method provides a new perspective of reflection treatment in computer vision without any prior knowledge. Extracting the additional information from the motion trajectories of the particle is definitely. Concerning the object recognition, a reliable graph feature which describes the surfaces distribution is constructed.

5.1/ Introduction

The *object surface structure* (OSS) describes the geometric distribution of the elementary continuous surfaces of an object (The definition of elementary continuous surface



(a) (b)

Figure 5.1: reflective object structure understanding: (a) original image (b) manually sub-segmented ground-truth image

is given in section 5.3.4). It is a highly representative feature obtained by performing a sub-segmentation of the surface. For non-reflective objects, the OSS can be easily recognized due to the distinguished contour, texture, and color. However, for the entire reflective objects, the reflective effects make the understanding of OSS extremely complicated. For instance, Fig.5.1(a) is the original image of an entire reflective object which consists of both specular and transparent surfaces; Fig.5.1(b) is the ground-truth of the manual sub-segmentation according to the OSS. We can see that due to the reflection on the object, the boundaries are barely observable and the OSS is hard to recognize. Moreover, because of the transparent surface, undesired components inside the object are also visible. Thus, the sub-segmentation of the object is not a trivial task.

In the proposed method, the reflection motion features are extracted in the image sequence as spatiotemporal information. Then the object is segmented by taking these features in order to understand the OSS. Both the camera and object are fixed, the light source is moving around the object in order to produce the *reflection particles* (RP) on the object surface. The surface is supposed to be piece-wise elementary continuous, i.e. it consists of several elementary continuous subsurfaces.

We assume that while the RP are moving on the object surface, their positions, directions, and velocities are extracted in each frame as reflection motion features. These features are matched in all the frames for tracking the RP in the whole sequence. The trajectories of RP are smooth along the subsurfaces. While they are passing through the boundary of two subsurfaces, irregular features (jumps) appear. Thus, we stop tracking when the trajectories are not smooth enough with respect to the previous frames. This guarantees that the trajectory of a moving RP stays on the same elementary continuous subsurface. Then, the surface is segmented by employing flood fill method [41] which takes the positions in the trajectory as seeds. As this process iteratively covers all the trajectories,

different surfaces of the object could be respectively labeled. With the segmented object, the arrangement of the elementary continuous surfaces and the surface size proportion can be extracted as the shape features for further object model recognition.

Different versions of the proposed method have been published in International Conference of Machine Vision and Applications(**MVA'15**), British Conference on Machine Vision(**BMVC'15**), and Pattern Recognition Letters.

5.2/ RELATED WORKS

Dealing with reflection: Many works have been done in dealing with the reflection in the image. The most common idea is to consider the reflection as noise, then try to remove or reduce it, such as the methods proposed in [42, 43, 44, 45]. However, several attempts have been made to use information contained in the reflection to extract object features. Savarese and Perona [46, 47] propose an analysis of the relationship between a calibrated scene composed of lines through a point, and the geometry of a curved mirror surface on which the scene is reflected. This analysis is used to measure the object surface profile. DelPozo and Savarese [48] use static specular flows features to detect specular surfaces on natural image. Barrois and Wohler [49] present a method which incorporates different channels of information, one of which is a polarization angle of light reflected from the object surface. It provides information on the rotation of an object with respect to the camera.

Video object segmentation: Many methods have been proposed for video object segmentation. Most existing methods attempt to exploit the temporal and spatial coherence in the image sequence, in which pixels with similar appearance and spatiotemporal continuity are grouped together over a video volume [50, 51, 52]. There are also some works [53, 54] that adapt graph-based image segmentation to video segmentation by building the graph in the spatiotemporal volume. Shi and Malik [55] use nystrom normalized cuts, in which the nystrom approximation is applied to solve the normalized cut problem for spatiotemporal grouping. Grundmann et al. [56] apply hierarchical graph-based approach in segmenting 3D RGBD point clouds by combining depth, color, and temporal information. Moreover, another scene segmentation method using RGBD data was proposed by Bergamasco et al. [57] who employ a game-theoretic clustering schema which benefits

from the macropixels pairwise similarities to combine color and depth information.

Object sub-segmentation in detail: Approaches closest to ours investigate in extracting fine-gained attributes for object recognition [58, 59, 60, 61, 62]. Deng and Feifei [59] present an attribute-based framework for describing object in details which can be generalized across object categories. Bourdev and Malik [58] use 3D data of human body which is annotated into different body parts to recognize the posture. Vedaldi et al. [63] propose a method for understanding objects in detail by studying the relation between part detection and attribute prediction. It diagnoses the performance of classifier that pools information from different parts of an object. However, the attributes used by these authors are no more accurate in presence of reflection, thus these methods are not robust in object segmentation in case of reflective surfaces.

The proposed approach extracts reflection motion features in the image sequence as spatiotemporal information, then sub-segments object by taking these features as fine-gained attributes in order to understand object surface structure. Comparing to other reflection dealing methods, we do not use any prior knowledge like calibrated camera or textured environment. Furthermore, to the best of our knowledge, the use of reflection motion features as spatiotemporal coherence for video segmentation and fine-attributes for object structure understanding has not been yet studied.

5.3/ PROPOSED METHOD

Our goal is to transform the motion of reflections into useful information that can help to segment different continuous surfaces of an object and further extract shape features from the segmented objects. The proposed pipeline is made up of three main tasks depicted in Fig. 5.2. The first step is the RP motion feature extraction, followed by a RP tracking process, finally the sub-segmentation is performed by taking the RP motion trajectories as labeling information.

5.3.1/ ESTIMATION OF REFLECTION

The motion of RP provides temporal information, thus in order to employ the RP moving information for object sub-segmentation, we firstly extract motion features of all the

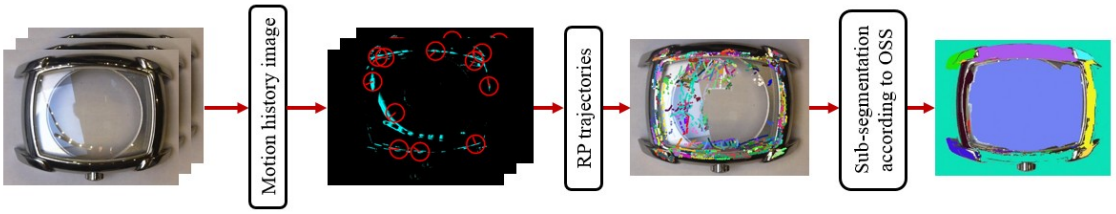


Figure 5.2: Illustration of the proposed pipeline (see text for details)

moving RP in the video.

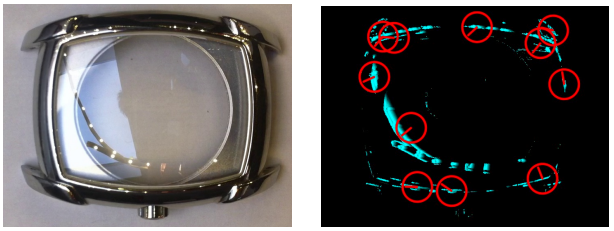
Since both our object and camera are fixed, in the video, movements could only be produced by reflections due to the movement of the light source. We use the motion history image (*MHI*) [64, 65] to extract RP. The *MHI* $H_\tau(x,y,t)$ can be computed from an update function $\Psi_\tau(x,y,t)$:

$$H_\tau(x,y,t) = \begin{cases} \tau & if \quad \Psi_\tau(x,y,t) = 1 \\ max(0, H_\tau(x,y,t-1) - \delta) & if \quad \Psi_\tau(x,y,t) = 0 \end{cases} \tag{5.1}$$

Precisely, if the pixel at position (x,y) in t-th frame has moved, $\Psi_\tau(x,y,t) = 1$, otherwise $\Psi_\tau(x,y,t) = 0$. The duration τ decides the temporal extent of the movement, and δ is the decay parameter. For more details concerning the definition of *MHI*, one can refer to [64, 65]. This leads to a static scalar valued image where the more recently moving pixels are brighter. Then the moving direction can be efficiently calculated by convolution with separable Sobel filters in the X and Y directions yielding the spatial derivatives: $F_x(x,y)$ and $F_y(x,y)$, respectively. The gradient orientation (Ø) of the pixel is:

$$Ø = \arctan \frac{F_y(x,y)}{F_x(x,y)}. \tag{5.2}$$

Note that these gradient vectors will point orthogonally to moving object boundaries at each step in the *MHI*. It gives us a normal optical flow representation. After that, a downward stepping flood fill [41] is used to label motion regions connected to the current *MHI*. In this method, one puts a connected RP to be a family of neighbor pixels having similar motion direction. From the frame at time t, we extract the n moving RP (later denoted by $C_i^t, i \in [1:n]$) as 8-connected pixels of the similar motion. From each C_i^t, a motion feature vector $f(C_i^t) = \left\{d_i^t, p_i^t\right\}$ is extracted, where d_i^t and p_i^t present the direction and the position of the RP, respectively. d_i^t is obtained by taking the average direction of all the pixels in C_i^t



(a) (b)

Figure 5.3: (a) Original frame; (b) motion history image of current frame. White pixels represent moving reflection particles. Red clocks represent moving directions of corresponding reflection particles.

while p_i^t is the center of a bounding box that contains C_i^t. The motion features are used in the following section to match and track each RP in the image sequences.

5.3.2/ REFLECTION PARTICLES MATCHING

As the motion features $f(C_i^t)$ are extracted independently from each frame, the matching should be adapted to link the temporal information and to filter the impossible matches. The matching of a reference particle feature $f(C_i^t)$ and a candidate particle feature $f(C_j^{t+\Delta t})$ needs to satisfy two following constraints:

$$err_p(i,j) = \sqrt{(p_i^t.x - p_j^{t+\Delta t}.x)^2 + (p_i^t.y - p_j^{t+\Delta t}.y)^2} < \delta, \tag{5.3}$$

$$err_d(i,j) = (d_i^t - d_j^{t+\Delta t})^2 < \alpha. \tag{5.4}$$

We set $\alpha = 20$ and $\delta = 10$ based on the experiment results. The equation 5.3 describes the condition of particle position,where $err_p(i,j)$ is the position difference between C_i^t and $C_j^{t+\Delta t}$. The equation 5.4 gives the condition of moving direction, where $err_d(i,j)$ is the direction difference between C_i^t and $C_j^{t+\Delta t}$. From a pair of matched features, a velocity feature $v_j^{t+\Delta t}$ is computed by:

$$v_i^{t+\Delta t} = \frac{\sqrt{(p_i^t.x - p_j^{t+\Delta t}.x)^2 + (p_i^t.y - p_j^{t+\Delta t}.y)^2}}{\Delta t}. \tag{5.5}$$

Then the updated motion feature of the reference particle is $f(C_i^{t+\Delta t}) = \left\{d_i^{t+\Delta t}, p_i^{t+\Delta t}, v_i^{t+\Delta t}\right\}$. The matching algorithm is the following:

Algorithm 2 Reflection particles matching

Input: $f(C_i^t) = \left\{d_i^t, p_i^t\right\}$, $f(C_j^{t+\Delta t}) = \left\{d_j^{t+\Delta t}, p_j^{t+\Delta t}\right\}$.
Output: $f(C_i^{t+\Delta t})$.

do matching $f(C_i^t)$ and $f(C_j^{t+\Delta t})$ with equations 5.3 and 5.4
if matching is true **then**

1. compute $v_i^{t+\Delta t}$ by using equation 5.5
2. update $f(C_i^t)$ to $f(C_i^{t+\Delta t})$
3. return $f(C_i^{t+\Delta t})$

else

1. $f(C_i^{t+\Delta t}) = f(C_i^t)$
2. return $f(C_i^{t+\Delta t})$

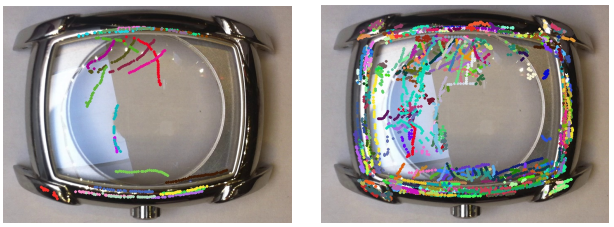
That is, if no candidate particle features can be matched to reference particle feature, $f(C_i^{t+\Delta t})$ will be updated using the previous reference feature $f(C_i^t)$. On the other side, if there exist several candidate particle features which could be matched to $f(C_i^t)$, the C_i^t is computed as $Argmin\{err_d(i,j)\}$.

5.3.3/ Reflection particles tracking

The tracking of RP suffers from several problems: the high frequency of appearance and disappearance of the RP, the shape evolution of the RP, as well as multiple reference RP needed to be tracked in the same time. Our tracker is composed by an iterative matching computation. The tracker is initialized for each detection, the state of a reference RP (C_i^t) is presented as $S(C_i^t) = \left\{p_i^t, d_i^t, v_i^t\right\}$. The state transition density is defined by:

$$p_i^t = p_i^{t-1} + v_i^{t-1} \times 1, \qquad v_i^t = v_i^{t-\Delta t}. \tag{5.6}$$

The sampling processes a predictive circle window with the radius of δ and the center at the position predicted by the equation 5.6. It is due to the RP motion features have already been extracted in each frame. Instead of sampling candidate RP (note as cc_j^t with its feature $f(cc_j^t) = \left\{dc_i^t, pc_i^t, vc_i^t\right\}$) with a weight which costs computational extremely expensive, a predictive sampling window is employed. Then each reference RP and candidate RP pair in the predictive window is scored by the difference of the moving



(a) (b)

Figure 5.4: Reflection moving trajectories. (a) fifteen longest trajectories. (b) all the trajectories.

direction:

$$err_d^c(c_i^t, cc_j^t) = (d_i^t - dc_j^t)^2, \tag{5.7}$$

and the $Argmin\left\{err_d^c(c_i^t, cc_j^t\right\}$ is computed to find the best match. Here we also present a threshold parameter β to break current reference RP tracking when the RP moving direction hugely changes. In our experiments, the value of β is set to 30. This tracking phase guarantees to keep all the associated RP on the same surface.

During tracking RP in frames, positions of all tracking results are saved as the moving trajectory. The trajectory of C_i is denoted as $T(C_i) = \left\{p_i^1, p_i^2, ...p_i^t, \right\}$. One trajectory is considered as one label for a continuous surface on the object. As the RP could go through one surface in different directions, we save trajectories respectively for each direction. In this case, it ensures that one trajectory labels only one surface. On the other hand, some trajectories label the same surface. In Fig. 5.4, where one color presents one trajectory of moving reflection, image 5.4(a) contains 15 longest trajectories, image 5.4(b) contains all the trajectories.

5.3.4/ ELEMENTARY CONTINUOUS SURFACES SEGMENTATION

In order to solve the problem of multi-labelled surfaces, an iterative surface segmentation of the object is performed based on RP moving trajectories. For convenience, we introduce a notation of an elementary continuous surface. It is defined according to the variation of γ of the object surface, γ being the difference between two neighbor normals. Then at each point of the surface, if the corresponding γ is below the threshold parameter ψ, the surface is considered as an elementary continuous surface, otherwise it is not. In

fact, ψ denotes the limit of a surface normal variance which is not visible in the image. We put the distance between two neighbor points on the object surface equal to 1 mm. After experiments on various objects, ψ is set to be equal to 2.2 degrees. As shown in Fig. 5.5, 5.5(a) and 5.5(b) are both discontinuous surfaces since their variations of γ are beyond the threshold parameter ψ, while 5.5(c) is an elementary continuous surface since γ is small enough.

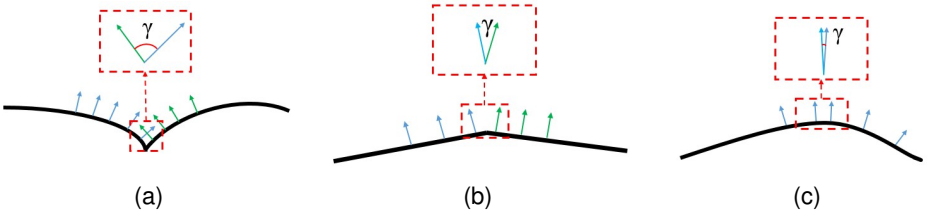


Figure 5.5: (a)(b) discontinuous surfaces (c) elementary continuous surface

Segmentation of elementary continuous surfaces serves to describe the surface structure of the object. As some trajectories are labeling the same surface, an iterative flood fill function is applied to merge the segmentation results of different trajectories on the same surface. The seeds which need to be flood filled are systematically sampled positions with a skip of 5 in the trajectory. Since the surface is elementary continuous, a trajectory can cover the surface regions with different brightness levels, the flood fill produces only one surface and the reflection does not produce additional seb-segments. The flood fill method which we used during the segmentation is the same for the reflection particle detection. The pixel value $I(x, y)$ is considered to belong to the labeling domain if:

$$I(x', y') - d_l < I(x, y) < I(x', y') + d_h, \tag{5.8}$$

where d_l and d_h stand for maximum lower/upper brightness difference between the current observed pixel and one of its neighbors belonging to the surface, respectively. The algorithm 3 illustrates the segmentation process:

Since the trajectories do not have the same length and they may contain numerous positions, we order trajectories by increasing lengths and then systematic sample the positions by a skip of 5. Finally the flood fill is performed by starting from the sampled seeds in shorter trajectories to the sampled seeds in longer trajectories. In case of an elementary continuous surface, the segments containing shorter trajectories could be merged into other segments if there exists a suitable longer trajectory which covers all the segments.

Algorithm 3 Segmentation process

1. Trajectory sampling
    1. Sort trajectories by size in increasing order
    2. Systematically sample of each trajectory with a skip of 5
2. Segmentation
    1. Update filling color to the color of $T(C_i)$
    2. Flood fill all $p_i^t \in T(C_i)$ with current filling color
3. Morphology component regrouping
    1. Update current filling color to the color of $T(C_j)$
    2. Regroup and fill all the components passed by $T(C_j)$ with current filling color ($i < j$)
4. Final processing
    1. Fill holes which are surrounded by segmented regions with the surrounding color

In this case, segments containing the seeds of shorter trajectories are relabeled according to the labeling of seeds of longer trajectory. As the reflection on the surface is highly variable, the segmentation phase might not cover the whole surfaces. In consequence, the final processing fills the holes which are surrounded by segmented regions with the surrounding color.

5.4/ SEGMENTATION RESULTS AND EVALUATION

The experiments are conducted by using the camera with the resolution of 5 Mpx. A LED grow light is used to produce reflections on the object. Note that the light source consists of multiple light dots and it can be of any shape, here we use a circle one. For the outdoor experiments, two projectors are used. The number of acquired frames depends on the complexity of the object surfaces and the number of light sources. In order to keep a reasonable number of acquired images, our LED grow light contains 30 light spots. The ground-truth images are manually labeled according to the 3D models of the objects which are obtained by a non-contact 3D digitizer VI-910.

As the considered objects are reflective and/or transparent, the images contain many high-variability regions. Three of the comparison segmentation methods are graph-based methods [53, 50]. They are based on k nearest neighbors, adjacent, and hierarchical graph, respectively. The graph-based methods are chosen since they have the ability

to preserve detail in low-variability image regions while ignoring detail in high-variability regions. The forth comparison method is EM segmentation [66]. It is a pixel clustering method in a joint feature space. It segments the image with the information from different aspects (color-texture-position). Over 20 objects have been processed, 7 of them are shown in the Fig. 5.7. Due to the similarity of the three graph-based results and the lack of space, only KNN graph-based results are illustrated in Fig. 5.7. The objects *cover*, *ball* and *car2* have completely specular surfaces, the third object *scotch* is transparent, and the other three objects contain both specular and transparent surfaces. The experiments for two cars are carried out outdoors. From the results, we can see that graph-based methods work reasonable in segmenting the object, but the sub-segmentation of the object surface does not provide meaningful results. EM segmentation preserves very well the contour of the objects but also the contour of the reflection that yields the poor sub-segmentation performance. Conspicuously, the results obtained by our method are more accurate. In consequence of a high sub-segmentation performance, the OSS is well presented.

5.4.1/ QUANTITATIVE EVALUATION

The purpose of our object surfaces segmentation is to understand the structure of the reflective objects. Therefore, to evaluate the proposed method, we manually labeled all the elementary continuous surfaces of the object to generate the ground-truth image as reference. Then we verify the segmentation performance with a pixel-wise evaluation.

To evaluate our method in details, we calculate true positives (*TP*), false positives (*FP*), false negatives (*FN*), precision and recall for each surface, which are computed as follows:

$$TP = \frac{NTP}{PG}, FP = \frac{NFP}{PD}, \tag{5.9}$$

$$precision = \frac{NTP}{NTP + NFP}, \tag{5.10}$$

$$recall = \frac{NTP}{NTP + NFN}, \tag{5.11}$$

where NTP, NFP, NFN stand for the number of the true positive pixels, false positive

pixels and false negative pixels, respectively; PD, PG, ND, NG stand for number of positives detected, number of positives in ground-truth mask, number of negatives detected and number of negatives in ground-truth. After computing precision and recall for each surface, a weighted combination of evaluations on each surface is proposed to verify the entire performance for a whole object. The total pixel number N of the ground-truth object is computed as:

$$N = \sum_{i=1}^{n} PG(i), \tag{5.12}$$

where n is the number of surfaces. Then a weight w_i is defined by the percentage of the pixel number of current surface on that of the whole object, where i is surface index.

$$w_i = \frac{PD(i)}{N}. \tag{5.13}$$

With the weights of each surface, the precision ($precision_o$) and recall ($recall_o$) of the object can be computed as:

$$precision_o = \sum_{i=1}^{n} precision_i \times w_i; \tag{5.14}$$

$$recall_o = \sum_{i=1}^{n} recall_i \times w_i; \tag{5.15}$$

Then, we generate the *receiver operating characteristic* (ROC curves) for objects in the experiment by varying the parameters d_l and d_h of the flood fill method. We use 5 different values for $d_l \in [1.5, 2.5, 3.5, 4.5, 5.5]$ and 3 different values for $d_h \in [6.5, 7.5, 8.5]$. From the ROC curves, we can see that for *Scotch*, *Ball* and *Phone*, the precision values keep very high at the beginning and suddenly go down during the raising of recall values. This is due to the fact that these objects all have two surfaces. Within the change of parameters of flood fill method, the labeling color of one surface overfills the other surface. Then the sudden overfilling makes precision value suddenly drop down. For the other objects, as they have approximately ten surfaces, the curves are more smooth. For all the indoor experiments (except the one of the car), the precision values reach 0.99 and recall values are more than 78. For the outdoor experiments on the cars, under a nature en-

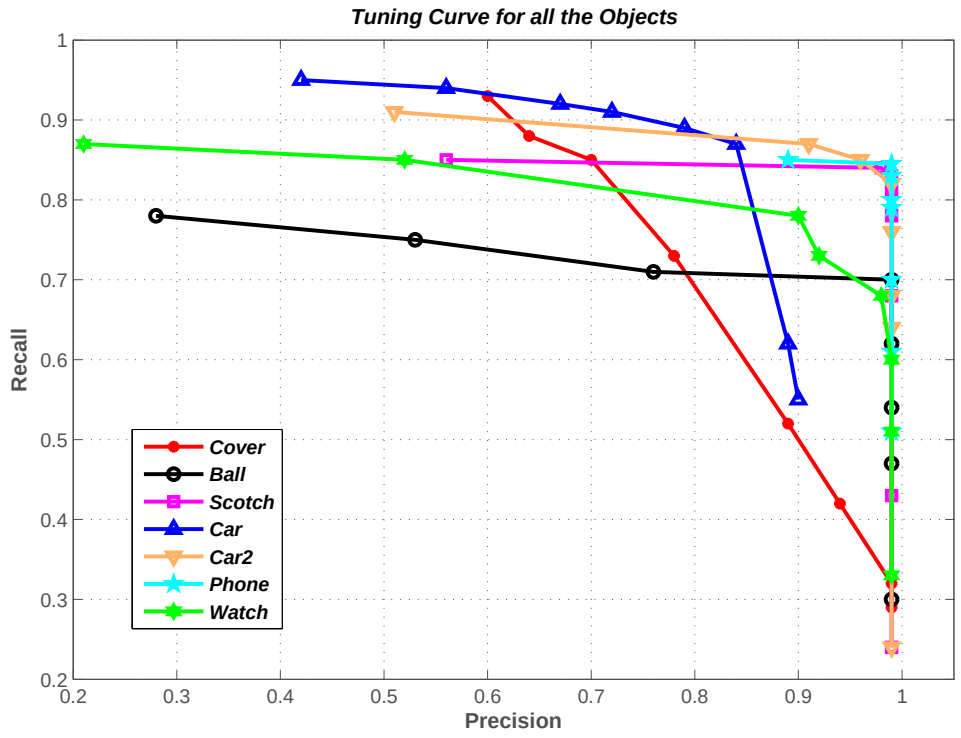


Figure 5.6: ROC curve for the objects. All the curves were generated in using d_l from 1.5 to 5.5, d_h from 6.5 to 9.5. Each point corresponds to one combination of d_l and d_h. Objects with more subsurfaces have smoother curves.

vironment without controlling illumination condition except our light source, the precision values reach 0.99 and the recall values are more than 0.88. These results illustrate the robustness of our segmentation method in OSS understanding under different experiment conditions and of various objects.

5.4.2/ COMPARISON WITH OTHER WORKS

To our best knowledge, no segmentation method is designed for dealing with reflective objects, thus it is not a trivial task to compare with other methods. Among existing methods, graph-based and region based segmentation methods are most likely to treat the case of reflective surfaces. Three graph based (KNN, adjacent, and hierarchical) methods and one region based (EM) method are chosen for the comparison. We would like to point out that only the proposed method and hierarchical graph-based method [50] take advantage of temporal information while the other two methods use static data. We did

Table 5.1: Best f-score of the objects.

f-score	Cover	Ball	Scotch	Car	Car2	Phone	Watch
KNN graph [53]	0.56	0.38	0.48	0.73	0.83	0.51	0.74
Ajdacent graph [53]	0.48	0.34	0.54	0.66	0.79	0.48	0.75
EM [66]	0.17	0.41	0.46	0.54	0.27	0.79	0.47
Hierarchical graph [50]	0.46	0.32	0.39	0.72	0.81	0.43	0.44
our method	**0.76**	**0.84**	**0.89**	**0.86**	**0.93**	**0.91**	**0.84**

not compare with contour based methods since in this case, reflections would produce false true negative contours which leads to a poor segmentation. To evaluate the segmentation performance, we employ the f-score as criterion which is a harmonic mean of precision and recall. It is computed as:

$$\text{f-score} = 2 \times \frac{precision \times recall}{precision + recall}. \tag{5.16}$$

Therefore, we choose f-score as the criterion of segmentation performance evaluation in order to compare our proposed method with the state-of-the-art approaches. In table 5.1, we compare our proposed method to 4 well known segmentation methods. We can see that the f-score of object *Cover* is 0.76, which is much lower than for the other objects computed by all the methods. This is due to the fact that the surfaces of this object are concave, moving reflection vanish extremely quick even though the surfaces are smooth and moving trajectories are split into smaller trajectories. On the other hand, f-score of *Ball* is also only 0.84 because of the presence of high intensity variations in small regions. In the experiment of object *Phone*, despite the fact that intensity variations are important on the whole object, it is not the case for small sub-regions. Thus, the final processing of our method can fill the holes and yields the value of f-score to 0.91. As for the two outdoor experiments, both provide meaningful results. The f-score of car2 reaches 0.93 which means high rate in both precision and recall. We would like to emphasize that, while dealing with reflective and transparent objects, our method outperforms significantly (at least 9% higher) the state-of-the-art methods.

With the proposed elementary continuous surface segmentation method, a surface is represented by a graph. In consequence of that, the object model recognition performance of the prototype can benefit from the object surface graph features.

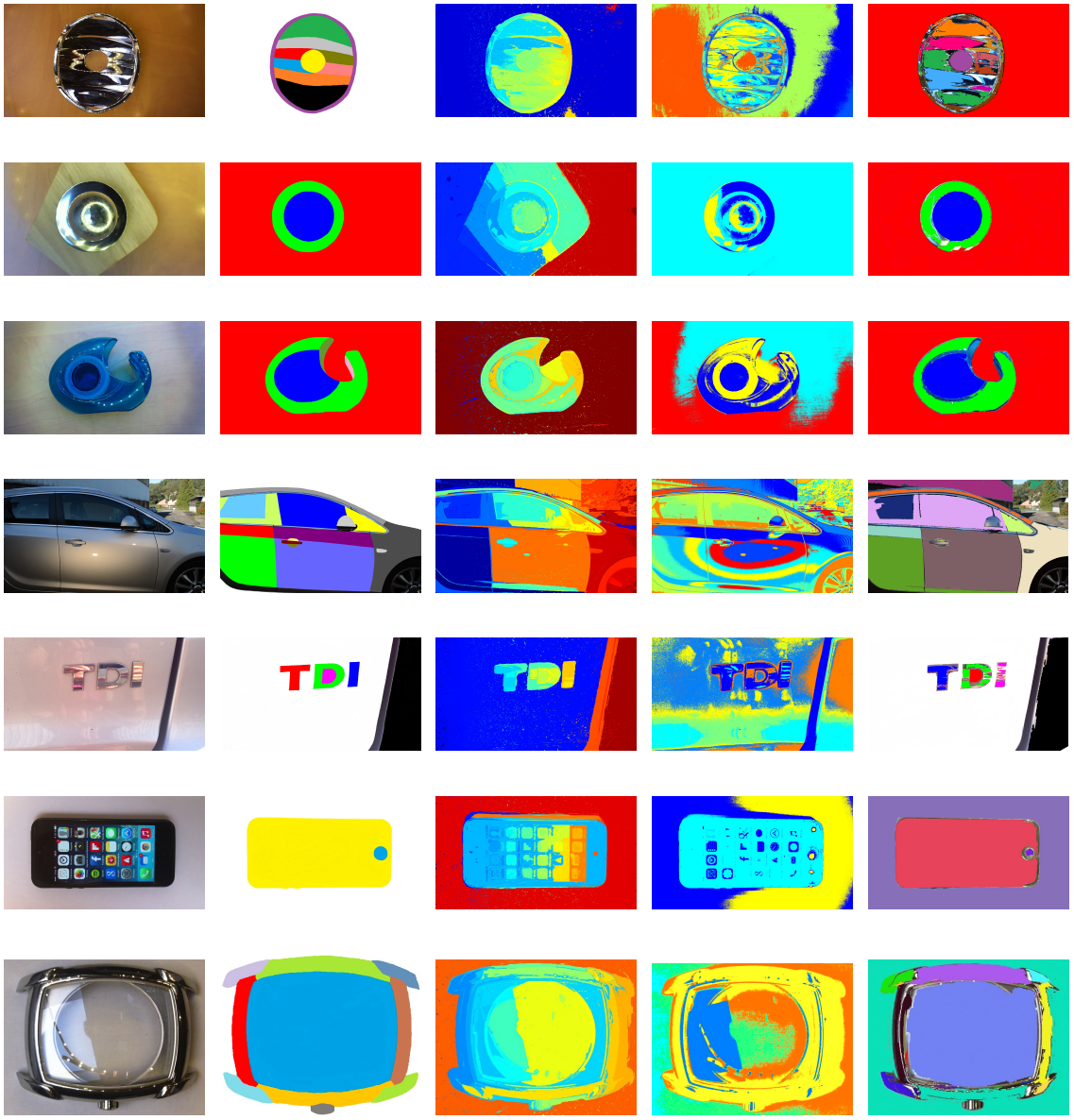


Figure 5.7: First column: original images. Second column: ground-truth segmentation. Third column: k nearest neighborhood graph-based segmentation [53]. Forth column: EM segmentation [66]. Last column: Segmentation by our proposed method based on reflection motion estimation. (better see in color)

5.5/ GRAPH REPRESENTATION OF ELEMENTARY SURFACES

Graphs are widely used as a general and powerful representation in a variety of scientific fields and many problems can be formulated as attributed graph matching. Building a graph representation based on the elementary surface segmentation provides more features for the object model recognition. Basically, a graph $G = (V, E)$ is composed by a

set of vertices $V = a_i, i = [1, N]$ where N is the number of vertices and a set of edges $E = b_i j, i = [1, N], j = [1, N], i \neq j$. Each vertex a_i and edge $b_i j$ are represented by a set of attributes, respectively. The set of the attributes of vertices and edges form the graph representation for a object model.

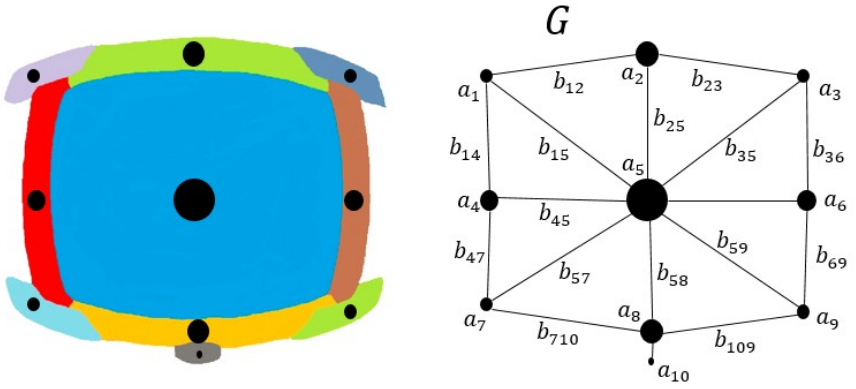


Figure 5.8: Graph representation of the object surface. The subsurfaces are considered as nodes, then linked by the edges

In our case, based on the elementary surfaces segmentation, the center of each surface is considered as a vertex, the connection of two neighboring surfaces centers is considered as the edge. Moreover, the size of the vertex signifies the size of the corresponding surface. As shown in Fig. 5.8, the graph of the object model is denoted as $G = (V, E)$. 10 elementary surfaces are represented by vertex $V = a_i, i = [1, 10]$, the connections of the elementary surface centers are represented by edges $E = b_i j, i = [1, 10], j = [1, 10], i \neq j$. In general, any graph representation satisfying the condition of dot product similarity [ ]. However, not all potential representations are effective in representing data in the context of learning. As we aim to recognize the object model, a learnable graph representation is prioritized. Thus we employ *histogram-attributed relational graph* (HARG), wherein all edge attributes are represented by histogram distributions. In this regard, for the further graph matching, the similarity value between two attributes in this graph is then computed as their dot product. In order to keep the object model recognition orientation invariant, the length of the edge is chosen as the edge attribute.

For the edge attribute construction, as widely done in computer vision, we assume that each interest point can be assigned a characteristic scale. Consider an edge b_{ij} from node a_i to node a_j, as shown in Fig. 5.9, the vector from a_i to a_j can be expressed into polar coordinates as (ρ_{ij}, θ_{ij}). We transform this into a histogram-based attribute, which is

invariant to the characteristic scale of a_i. We use uniform bins of size n_L in the log space with respect to the position and scale of a_i, making the histogram more sensitive to the position of nearby points. The log-distance histogram L_{ij} is constructed on the bins by a discrete Gaussian histogram centered on the bin for ρ_{ij}:

$$L_{ij}(k) = f_L(k-m), \quad \text{s.t.} f_L(x) = \mathcal{N}(0,\rho_L), \rho_{ij} \in bin_\rho(m), \tag{5.17}$$

where $\mathcal{N}(\mu,\rho_L)$ represents a discrete Gaussian window of size of ρ centered on μ, and $bin_\rho(k)$ denotes the kth log-distance bin from the center of a_i. We use a window size $\rho_L = \rho_P = 5$ so that $\mathcal{N}(0,5) = 1.0$, $\mathcal{N}(\pm 1,5) = 0.4578$, $\mathcal{N}(\pm 2,5) = 0.0439$, and 0 otherwise.

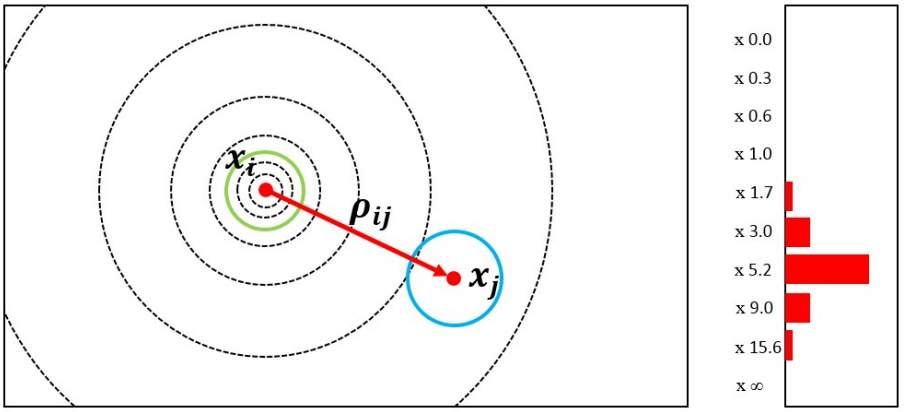
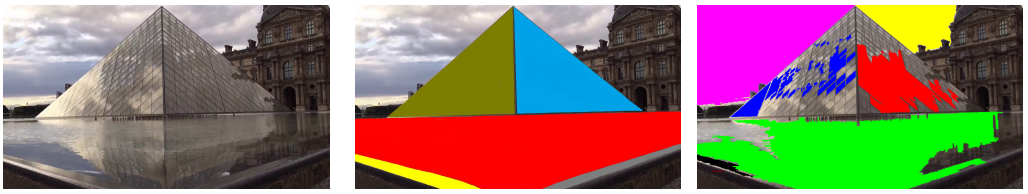


Figure 5.9: length of edge b_{ij} and its histogram attribute

Our representation has several advantages for the further matching. When used with local invariant features, it becomes geometrically invariant in scale. Moreover, from the viewpoint of learning, the nonparametric nature of histogram allows us to represent multi-modal distribution of the edge length through the learning process, which means that the graph features can be learned for the further object model recognition.

5.6/ CONCLUSION

We have presented a segmentation method based on reflection motion features in order to deal with reflective and transparent objects. Due to a simple constraint (object and camera are fixed), our method can be widely used in the industry for object recognition and retrieving. More importantly, instead of removing and reducing reflections, taking its advantage is pioneering work in a new direction. The results show that the reflec-



(a) (b) (c)

Figure 5.10: Future work: object segmentation by employing fully nature light source.

tion motion features can be used as a robust signature for labeling continuous surfaces on reflective and transparent objects. In comparison with conventional segmentation approaches, our method can overcome the difficulties produced in case of reflective and transparent objects and leads to higher performances in terms of accuracy and robustness. This efficiency has been proved through multiple experiments over various objects and under different type of illumination conditions (indoor and outdoor). This series of test highlight the advantage given by our approach against the state-of-the-art methods. Moreover, the use of graph to present surface structure distribution extracts more robust feature for the object recognition.

Regarding future work, we intend to use nature illumination source for the object segmentation. An example is shown in figure 5.10, where no man-made light source is used. It is a time lapse video of 10 seconds which requires 4 hours of image acquisition. The reflection motion of the cloud is used to perform a surface segmentation. However, the faces of the pyramid do not satisfy our constraint of the elementary continuous surface, thus they are not detected as entire surfaces. One of possible directions of future research will be to adapt our method to such surfaces. We are also interested in exploring the evolution of reflection shape.

6

TEXT DETECTION ON REFLECTIVE SURFACES

In this chapter, we present a novel method to detect text on the reflective surfaces. As explained in the chapter 3, the object reference number relies on contour tracking and template matching. Too many previous stages yield the inaccuracy of the reference number localization. Therefore, we propose a straightforward method to detect the text in the image without any prior knowledge or previous stages. The method initially extracts low level features such as points, contours, and regions. Then similar features are clustered in order to precisely select the text candidates. Afterwards a powerful classifier is trained to predict text zone in the image. By using this method, we get rid of the constraint that the reference number localization has to rely on the object contour tracking, template matching, and the prior knowledge is no longer needed. The detection is straightforward and also suitable for the text in the nature scene.

6.1/ INTRODUCTION

For object identification by reading the reference number, localizing the reference number is the essential task. Furthermore, the great success of smart phones and large demands in content-based image understanding have made text detection a crucial task. It is desirable to build a system that can robustly detect text under various conditions. As the text contains large variation in language, font, color, scale and orientation in complex scene, the detection remains unsolved. The difficulties mainly come from two aspects: (1) the diversity of the texts and (2) the complexity of the backgrounds. On one hand, text

is a high level concept but better defined than the generic objects [8]; on the other hand, repeated patterns (such as windows and barriers) and random clutters (such as grasses and leaves) may be similar to texts, and thus lead to potential false positives. In our case, as shown in Fig. 6.1, the presence of the reflection ruins the homogeneity of the text characteristics, and weakens the text saliency on the surface. It makes the problem of text detection on reflective surfaces even more complicated. Thus we want to build a detection system which is not only able to deal with text in nature scene, but also with the text on reflective surfaces. To the best of our knowledge, detection of text on reflective surface has not yet been studied.

(a) (b)

Figure 6.1: Text detection on reflective surfaces

6.2/ RELATED WORKS

Most of the existing text detection methods have focused on detecting text on contrast background in nature images and videos. In recent years, many successful methods consist of two main stages: text candidate detection and text/non-text classification.

In the text candidate detection, various features have been proposed. These features can be roughly classified into three groups: region based features, stroke width based features, and contour based features. For region based features, the method of Maximally Stable Extremal Regions (MSER) was proposed by Matas et al. [67] for establishing correspondences between a pair of images taken from different viewpoints. Since then, it has been widely used for text detection [68, 69, 70]. MSER is an efficient (near linear complexity) and practically fast detection algorithm, since the texts in nature images have uniform color, thus MSER produces meaningful text detection results. As for stroke width based methods, one may notice that the stroke width of text in the nature scene is usually

homogeneous. One of the most popular algorithm is the stroke width transform (SWT) introduced by Epshtein et al. [71], which became popular for text detection thanks to its efficiency [72, 73]. A multi-scale representation of this transform called Strokelets [74] was developed by Cong for scene text recognition. We end by mentioning contour based features even if they are less used because of huge data amounts and increasing scene complexity. Despite of that, NLFS contour detector proposed by Laligant [75] can still provide significant information to distinguish text from non-text patterns.

In text/non-text classification, multiple classifiers have produced promising results. Since Yann Lecun proposed a series of learning techniques such as back-propagation [76, 77], programmable topology [78, 79, 80], Convolutional Neural Network [81, 82], neural network based classification has been dominating the learning field during the last decade. A great quantity of works based on neural network have been presented. During the last 10 years, within the increase of computational speed, deep learning achieved significant success in many applications including text detection [83, 84, 85, 86]. Some other classifiers have been also studied to solve the problem of text detection such as Support Vector Machine (SVM) [82], metric learning [87], and bag-of-words [88].

Two problems arise when thinking about text detection algorithms. First of all, most of the popular text detection methods assume that text lines are straight. Within this assumption, some of these methods use geometric filters such as Adaptive Run Length Smoothing (ARLS) [89], symmetry property of character groups [90]. Some other methods use specially designed geometric features such as rotation invariant features [91]. Also, some other methods take the advantage of candidate clustering to solve the problem raised by arbitrary orientation text such as [92] and High Order Correlation Clustering [93]. However, this assumption is an ill-posed hypothesis, in many cases, the text line can be curved, especially on manufactured objects.

Another problem is that none of the methods in the state-of-the-art are designed for detecting text on the reflective surfaces. In order to fill this gap and provide a solution for reflective manufactured object identification, we propose an novel method that uses no geometric filters and is specifically designed for detecting text on reflective surfaces.

6.3/ PROPOSED TEXT DETECTION METHOD

The proposed pipeline is made up of four main tasks depicted in Fig. 6.2. The first step is to extract low level features from the input static image. Then the similar features are clustered together to provide more reliable information. Within the clustered features, a local consistency map is computed to select the text candidates. Finally, with a deep Convolutional Neural Network, false positive candidates are filtered out and the text zone can be detected in the image.

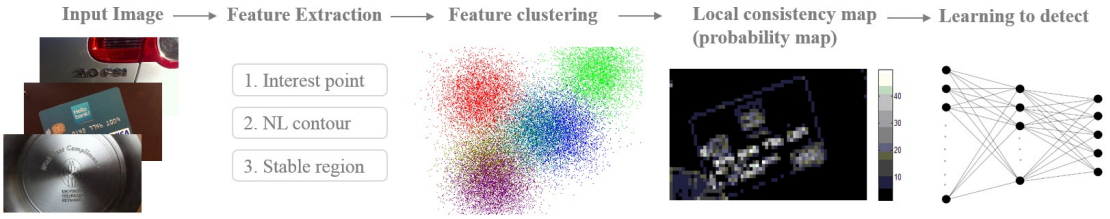


Figure 6.2: Object identification by recognizing characters engraved on the surfaces.

6.3.1/ FEATURE EXTRACTION

As mentioned in the previous section, most of the existing text detection methods employed geometric constraints. These methods work especially well for detecting the text that is horizontal or near horizontal. However, text of arbitrary orientations in complex natrual images has received much less attention and remains a challenge for most practical systems. Thus in our method, we reject all the geometric constraints and extract the low level features. We assume that everything in the image can be represented by a combination of points, contours, and regions. Different scenes have different compositions of these three basic elements, thus the extracted features are based on these three basic elements.

6.3.1.1/ POINT FEATURES

In image processing, key points are of high value for visual tracking, object recognition, and object indexing. Numerous key point detection methods have enjoyed great success in the last century, they are often represented by the corners which contain large gradient

values in both horizontal and vertical directions. We employ SIFT [3] as the key points since it does not only detect the key points, but also describe the characteristics of the key points that based on the gradient. In our case, after extensive study the property of key points on the text, we assume two hypothesis:

hypothesis 1: Key points stay close on the text in the images.

hypothesis 2: The HOG features of the reference key point have high variation comparing to that of neighboring key points.

Figure 6.3: The property of key points on the text.

In the Fig. 6.3, the key points are very dense in the text zone and usually very sparse on the non text zone except the repeated patterns. The image on the left is the HOG features on the repeated patterns, the image on the right is the HOG features on the text. The HOG features of the key points on the repeated patterns are similar while that of the text are very different.

In order to formulate the hypothesis, we denote a set of key points as $P_i, i = 1, 2, 3, ..., n$, their HOG features are $H(P_i)$. For one reference key point P_i, its m nearest neighbors are $\phi(P_i) = \{P_j, j = 1, 2, ..., m\}$, where m is set to 5 in our case and P_j is defined as

$$P_j = \underset{j=1,2,...,m; j \neq i}{\operatorname{argmin}} \|P_i - P_j\|. \tag{6.1}$$

The difference between the HOG features of a reference key point and that of its neighbor point set is computed by bhattacharyya distance [94] and denoted by $D_b(H(P_i), H(P_j))$.

$$D_b(H(P_i), H(P_j)) = -ln(BC(H(P_i), H(P_j))), \tag{6.2}$$

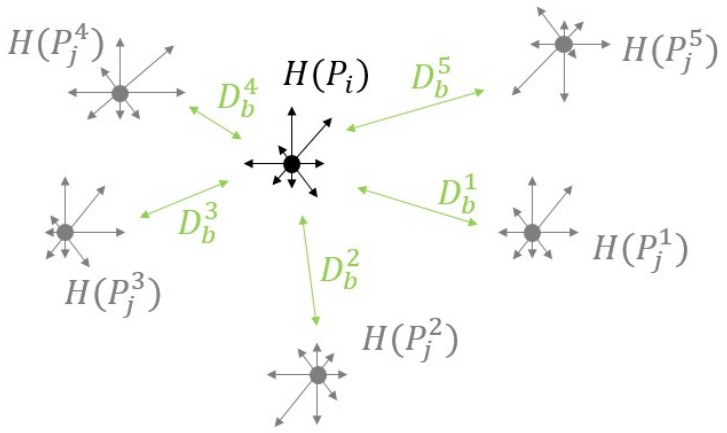


Figure 6.4: Visualization of hypothesis 2.

where $BC(H(P_i), H(P_j))$ is the bhattacharyya coefficient and computed by:

$$BC(H(P_i(x)), H(P_j(x))) = \sum_{x \in X} \sqrt{P_i(x) P_j(x)}. \tag{6.3}$$

As shown in Fig. 6.4, the $D_b(H(P_i), H(P_j))$ describes the difference between the HOG features of the reference key point P_i and that of one neighboring key point P_j. Then, the analysis of all the $D_b(H(P_i), H(P_j))$ describes the variation of the HOG features of this set of points.

In order to extract this variation, the mean $\mu(P_i)$ of all the differences $D_b(H(P_i), H(P_j))$ is computed:

$$\mu(P_i) = \frac{1}{m} \sum_{j=1}^{m} D_b(H(P_i), H(P_j)). \tag{6.4}$$

All the key points $P_i, i = 1, 2, 3, ..., n$ have a corresponding $\mu(P_i)$. According to the hypothesis 2, higher $\mu(P_i)$ indicates higher confidence rate of a key point being in text zone. Thus we compute the maximum and minimum values of $D(P_i)$:

$$\hat{M} = \text{argmax}_{i=1,2,3,...,n}\, D(P_i); \quad \hat{m} = \text{argmin}_{i=1,2,3,...,n}\, D(P_i); \tag{6.5}$$

to normalize the $D(P_i)$. Then compute the confidence rate $f(P_i)$ for each P_i:

$$f(P_i) = \exp - \frac{(\hat{M} - D(P_i))^2}{(D(P_i) - \hat{m})^2 + \epsilon}, \tag{6.6}$$

where ϵ is a control parameter for keeping the denominator non-zero, it is set to 0.001 in our case.

Within the confidence rate computed for each key point P_i, the local consistency map can be constructed. A window of 10×10 is set around each P_i, the intensity values of all the pixels in the window is set to $f(P_i)$. An example of a local consistency map of an input image is shown in Fig. 6.5.

(a) (b)

Figure 6.5: Local consistency map obtained by the key points

6.3.1.2/ CONTOUR FEATURES

The contour is another widely used feature in image processing. It presents the object shape, textures, and the composition of the elementary components. Many successful contour detection methods have been proposed in the last century. We employ NLFS contour detection method [6] proposed by Laligant because of its ability of describing the contour energy, obtaining both noise reduction and edge detection, as well as its low computation cost. As in the image, the text zone contains intensive contours which provide high contour energy.

The main idea of NLFS contour detector is based on the nonlinear combination of two polarized derivatives G_+ and G_-:

$$G_+ = \begin{pmatrix} G_{x+} \\ G_{y+} \end{pmatrix} = \begin{pmatrix} T(F_+(z)I(z,.)) \\ T(F_+(z)I(.,z)) \end{pmatrix}, \tag{6.7}$$

$$G_- = \begin{pmatrix} G_{x-} \\ G_{y-} \end{pmatrix} = \begin{pmatrix} -T(F_-(z)I(z,.)) \\ -T(F_-(z)I(.,z)) \end{pmatrix}, \tag{6.8}$$

where T is a threshold function which is :

$$T(x) = \begin{cases} 0 & if \quad x < 0 \\ x & if \quad x > 0 \end{cases} \tag{6.9}$$

Then the contour detection modulus $|G|$ can be obtained from the polarized gradient images as follows:

$$|G| = \sqrt{(G_{x+} + G_{x-})^2 + (G_{y+} + G_{y-})^2}. \tag{6.10}$$

(a) (b)

Figure 6.6: NLFS contour detection [75]

After applying the NLFS contour detection method, the contour energy of the image is shown in Fig. 6.6. Generally, a small text zone contains more contour energy comparing to non text zone of the same size, except on repeated patterns. According to that, we use a sliding window to detect the zone which contains more contour energy. However, due to the different scales of the text in different images, the size of the sliding window needs to be adaptive. In consequence of that, we take a convolution of three windows of different sizes (20×20, 30×30, 50×50) on the image to obtain 3 contour energy based local consistency maps, and then merge them with weights.

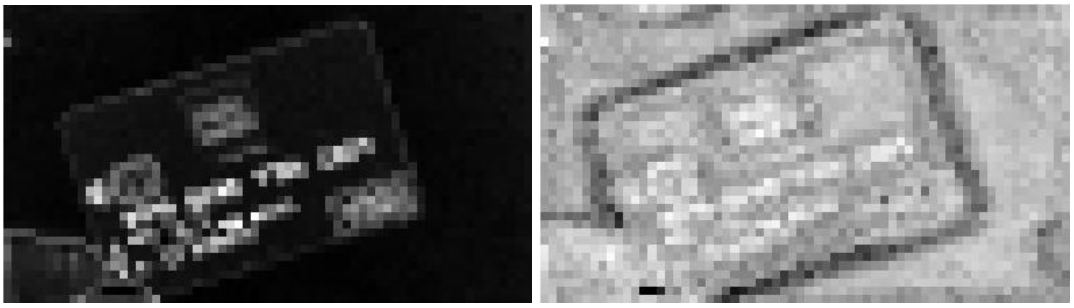
Each time a sliding window goes through the image, the shifting is set to 50% of overlapping in order find a balance between creating an accurate consistency map and low computational cost. We denote the first window by C_0, then the window obtained after i shifts is denoted by C_i. The contour energy contained in C_i is denoted by $f(C_i)$. For each C_i, a histogram $M(C_i) = \sum_{j=1}^{9} m_j$ of 9 bins is computed. Then we compare two criteria to represent the contour energy: entropy $E(C_i)$ (illustrating the variation of the contour energy contained in the window C_i) and mean $Mean(C_i)$ of the histogram $M(C_i)$. $M(C_i)$ represents the energy contour very straightforward, while $E(C_i)$ illustrates the variation of

the contour energy of the corresponding window.

$$E(C_i) = \sum_{j=1}^{9} m_j \times \log \frac{1}{m_j}; \tag{6.11}$$

$$Mean(C_i) = mean(M(C_i)). \tag{6.12}$$

The local consistency maps for this sliding window computation are obtained based on these two criteria, shown in Fig. 6.7. $6.7a$ is the mean map and $6.7b$ is the entropy map. We can see that the mean map describes the text zone while the entropy map defines well the contours of the card and text contours. It comes out that the mean map represents better the text zone thus in what follows it is used as local consistency map.



(a) (b)

Figure 6.7: Local consistency map obtained based on the contours, (a) is the mean map, (b) is the entropy map.

6.3.1.3/ REGION FEATURES

Region features are recently studied for blob detection and text detection, and object recognition. The most successful method is Maximually Stable Extremal Regions (MSER) proposed by Malik et al [67]. This technique searches the stable regions which is made of connected components in the static image I. An extremal region Q is defined as a region such that either for all pixels $p \in Q, q \in Q : I(p) > I(q)$ (maximum intensity region) or for all $p \in Q, q \in Q : I(p) < I(q)$ (minimum intensity regions). Then let $Q_1, Q_2, ...Q_i...$ be a sequence of nested extremal regions ($Q_i \subset Q_{i+1}$). Extremal region Q_{i*} is maximally stable if and only if $q(i) = |Q_{i+\Delta} \mathbb{Q}_{i-\Delta}|/|Q_i|$ has a local minimum at Q_{i*}. (Here $|\cdot|$ denotes cardinality). Δ is a threshold parameter. The equation checks for regions that remain stable over a certain number of thresholds. If a region $Q_{i+\Delta}$ is not significantly larger than

a region $Q_{i-\Delta}$, region Q_i is taken as a maximally stable region.

(a) (b)

Figure 6.8: Maximually Stable Extremal Regions detection in the image.

As shown in Fig. 6.8, all the texts can be detected by the MSER. The texts without any reflection are perfectly labeled. However, the texts with reflection, due to the high intensity variation, are labeled by multiple MSERs. In consequence of that, another technique employed additionally with MSER is Stroke Width Transform. It is a local image operator which computes per pixel the width of the most likely stroke containing the pixel as shown in Fig. 6.9.

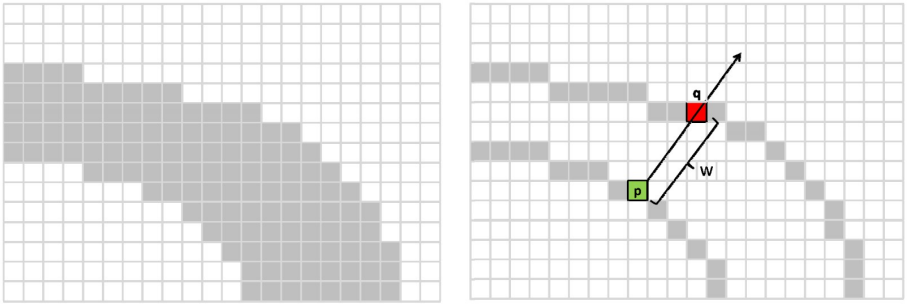


Figure 6.9: Stroke Width Transform [71].

As in the nature images, texts are mostly machine printed, text groups have self-similarity since adjacent characters bear similar stroke width. According to that, the property of the MSERs on reflective text can be summarized as: smash, gathered, but has similar stroke width. Thus the features that we attempt to extract for each detected region R_i are:

1) Average $\mu_{ed}(R_i)$ of euclidean distance between R_i and its N neighbors.

2) Stroke width variation $Var_{st}(R_i)$ between R_i and its N neighbors.

For R_i, smaller $\mu_{ed}(R_i)$ and $Var_{st}(R_i)$ values indicate higher likelyhood of R_i being text. Thus the region local consistency map is computed based on $\mu_{ed}(R_i)$ and $Var_{st}(R_i)$. We denote

the maximum and minimum values of $\mu_{ed}(R_i)$ and $Var_{st}(R_i)$ as $\hat{M}_{\mu_{ed}}$, $\hat{m}_{\mu_{ed}}$, $\hat{M}_{Var_{st}}$, and $\hat{m}_{Var_{st}}$, respectively. For each P_j in R_i, the confidence rate $f(P_j)$ is computed:

$$f(P_j) = \frac{1}{2}\exp -\frac{(\hat{M}_{\mu_{ed}} - \mu_{ed}(P_j))^2}{(\mu_{ed}(P_j) - \hat{m}_{\mu_{ed}})^2 + \epsilon_{\mu_{ed}}} + \frac{1}{2}\exp -\frac{(\hat{M}_{Var_{st}} - Var_{st}(P_j))^2}{(Var_{st}(P_j) - \hat{m}_{Var_{st}})^2 + \epsilon_{Var_{st}}}, \tag{6.13}$$

where $\epsilon_{\mu_{ed}}$ and $\epsilon_{Var_{st}}$ are control parameters for keeping the denominaters non-zero, they are set to 0.001 in our case. Within the confidence rate computed for each pixel P_j in the image, the region local consistency map can be constructed as shown in Fig. 6.10.

(a) (b)

Figure 6.10: Maximually Stable Extremal Regions detection in the image.

6.3.2/ FEATURE CLUSTERING

In all the three feature extraction processes, the features are extracted from each individual key points, sliding windows, and MSERs. However, text can be better identified by the properties of a group rather than of individual characters because (1) individual element varies a lot and tends to cause false positive; (2) a group of similar elements provides more robust statics for discriminating text from noise. Accordingly, we attempt to cluster individual features into groups which have similar elements.

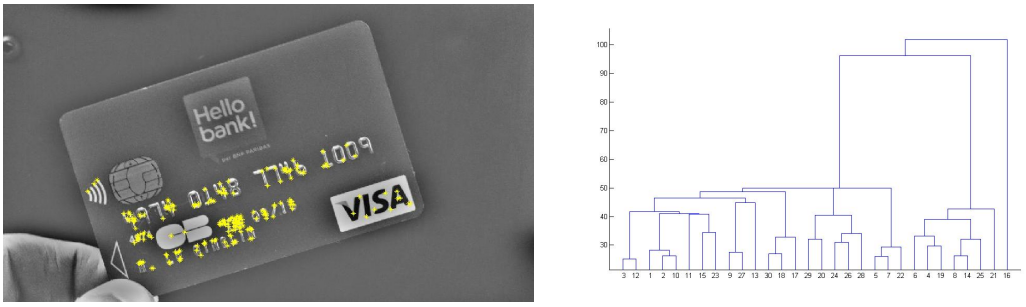
For each extracted feature from the input image, the number of clusters is unknown. Accordingly, many centroid based clustering methods such as K-means clustering [95], mean shift clustering [96] and distribution based methods such as expectation-maximization algorithm [66] are not available because their requirement of the number of classes. We employ Agglomerative Hierarchical cluster tree [97] for the features clustering because that it does not need to provide the number of classes. Considering one type of extracted features denoted as $x_1, x_2, ..., x_n \in \mathbb{R}^n$, the goal of clustering is to group

them into reasonable classes. First the features are placed into their own singleton group, then the cluster iteratively merges two closest groups according to the similarity estimation until all the features are merged into a single group. Here the similarity estimations $dist()$ are different for different types of features: euclidean distance and HOG value variation are used for key points similarity estimation; entropy and mean value are used for contours based sliding windows similarity estimation; euclidean distance and stroke width variation are used for MSERs. The following notation is used to describe the linkage of two groups: group r is formed from group p and q, n_r is the number of features in the group r, x_{ri} is the i-th feature in the group r.

Then *average linkage* is used to merge the most similar two groups:

$$d(r,s) = \frac{1}{n_r n_s} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} dist(x_{ri}, x_{sj}). \tag{6.14}$$

After iteratively compute the similarities and link the groups, all the features can be placed in a hierarchical tree, key points clustering are illustrated in Fig. 6.11.



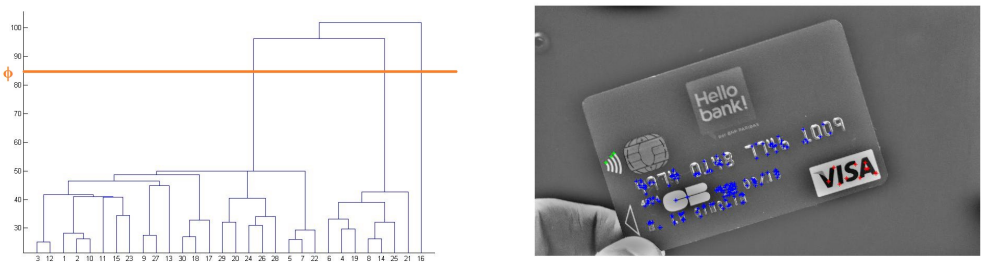
(a) (b)

Figure 6.11: Agglomerative Hierarchical Cluster Tree of key points.

Then a threshold parameter ϕ is used to cut the tree into different groups. In our case, ϕ is set to 10 times of the minimum distance. Then the hierarchical cluster tree is cut into three groups which means key points are clustered in three groups as shown in Fig. 6.12.

The same clustering is computed also for contour features and region features as shown in Fig. 6.13. For each group of features, average value is considered as the final feature.

After clustering features, the local consistency maps of key points, contours, and regions are updated by taking the average value of each group of features. Then these three



(a) (b)

Figure 6.12: Clustered key points by Agglomerative Hierarchical Clustering.

(a) (b)

Figure 6.13: Clustered contour sliding windows and MSERs by Agglomerative Hierarchical Clustering.

maps are merged with a uniform weight to construct a final probability map as shown in Fig. 6.14.

(a) (b)

Figure 6.14: Constructed final probability map for text detection.

Once the probability map is constructed, a threshold parameter is used to select the candidate text windows which will be further tested in a strong classifier.

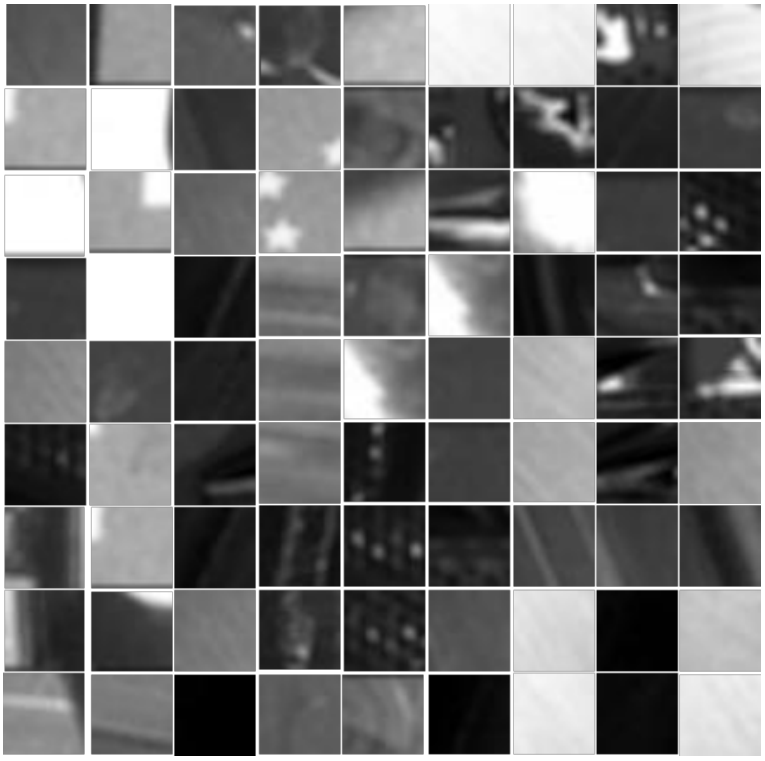
6.3.3/ LEARNING TO DETECT

As previously mentioned, the candidate text windows are selected based on the probability map that consists of 3 features. This candidate selection process filtered most of the non-text windows, however, all the candidates in reserve are highly probably text. Moreover, due to the reflection effect, most of the candidates contain noise. Thus, a strong classifier is needed to detect text from all the candidate windows. In consequence of that, we use deep neural network as the classification framework on account of its ability of highly non-linear classification. Furthermore, as the previously extracted features are used for the candidate selection, using them again for the classification is not efficient. We employ deep convolutional neural network (Deep-CNN) to extract new features from all the candidate images for training and validation. The main advantage of using Deep-CNN for the text candidate classification is bringing more additional features and generating more robust classification results.

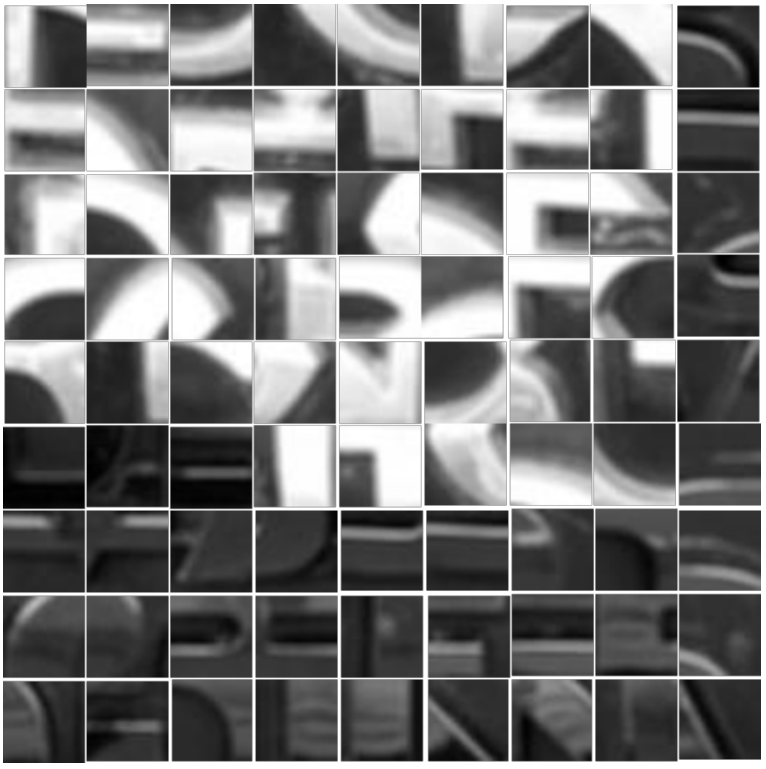
The data set contains 12,000 images where 6000 are text images and 6000 are non-text images. In Fig. 6.15, the data size are 25×25, the first image shows the non-text data and the second image shows the text data. In Fig. 6.16, the data size are 50×50, the first image shows the text data and the second image shows the non-text data. During training and validation, all the input images are normalized to size 50×50 in order to generate uniform features for classification.

The used Deep-CNN architecture is the MatConvNet tool box implemented by Andrea Vedaldi and Andrew Zisserman [98]. This tool box contains multiple significant contributions in deep learning field in recent years such as [99], [100] and inspired the most successful deep learning architectures such as [101], [102], and [103]. This Deep-CNN architecture consists of 8 layers that are 4 convolutional layers, 2 pooling layers, 1 Rectified Linear Unit layer, and 1 decision layer using softmax loss function.

Convolution As the inputs are images, each of them will be a real array of pixels and channels per pixel. Hence the first two dimensions of the array span space, while the last one spans channels. Note that only the input of the network is an actual image, while the remaining data are intermediate feature maps. To visualize the feature map, the output of the first convolutional layer is shown in the Fig. 6.17. The convolutional layer applies



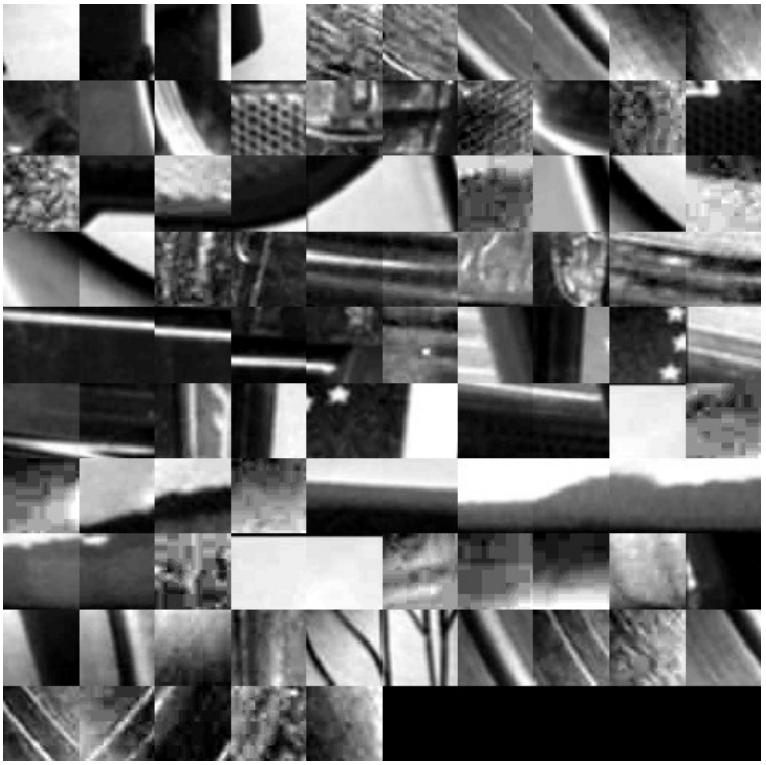
(a)



(b)

Figure 6.15: Data set of reflective text. (a) is the 25×25 non-text data, (b) is the 25×25 text data.

(a)



(b)

Figure 6.16: Data set of reflective text. (a) is the 50×50 text data, (b) is the 50×50 non-text data.

to the input map an operator that is local and translation invariant. Here, convolutional operators are applying on a bank of 10 linear filters.

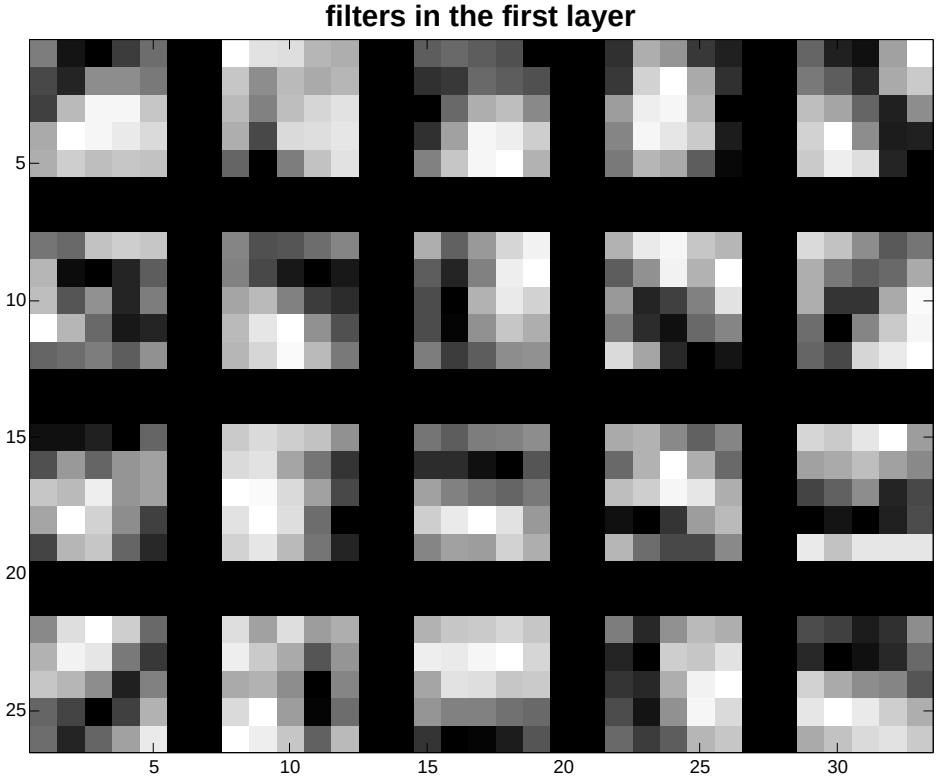


Figure 6.17: First layer.

Rectified Linear Unit In order to solve complex classification problems, Deep-CNN composes several different functions. In addition to the linear filters as convolution, there are several non-linear operators as well such as sigmoid/logistic function or Rectified Linear Unit (ReLU). The simplest non-linearity is obtained by following a linear filter by a non-linear gating function, applied identically to each component (i.e. point-wise) of a feature map. In this work, the non-linearity function is the ReLU.

Pooling A pooling operator operates on individual feature channels, coalescing nearby feature values into one by the application of a suitable operator. Common choices include max-pooling (using the max operator) or sum-pooling (using summation). In our Deep-CNN architecture, max-pooling is used.

6.4/ TEXT DETECTION RESULTS

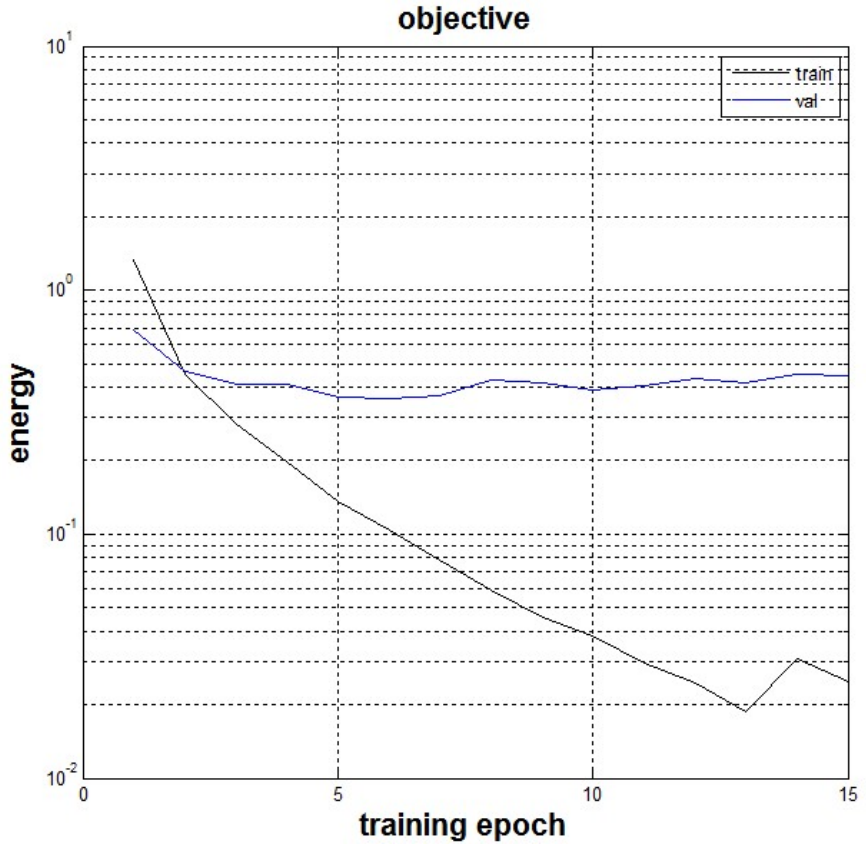
The text detection is evaluated based on the text/non-text classification and experimented on our data set of 12,000 images. The whole data set is randomly partitioned into 5 folds, in each experiment, one fold is left out, three folds are used as training data and the last one fold is used as validation data. The final classification result is the average of all the experiments which reaches to 91.40 accuracy.

The objective function and classification error are evaluated within the training epoch and shown in Fig. 6.18. Both the objective and error curves start to be smooth and steady from the 10th iteration, it illustrates the learning process is neither over fitting nor under fitting. The correct learning process provides the reliable classification results.

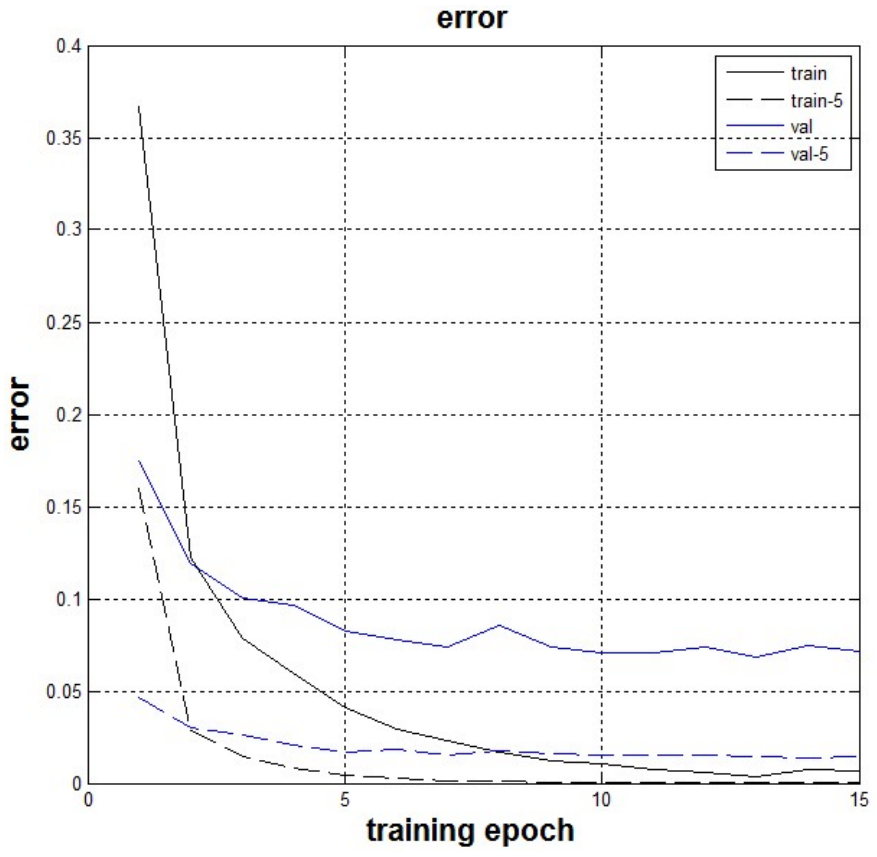
The qualitative results of text detection is shown in Fig. 6.19. The left column is original image and the right column is detected text. In most of the experiments, text zones are correctly detected even though some of the text lines are not straight. It is due to the advantage of low level features for the candidate selection and avoiding geometric filters. However, some of the text are missing such as in the forth experiment, it is due to the low contrast of text and background.

6.5/ CONCLUSION

We propose a novel text detection method which contains two main stages: text candidate selection by low level features to avoid geometric filters, text candidate classification based on Deep-CNN to detect text zone in the image. The main contributions of this method are: (1) efficiency for text detection on reflective surfaces. (2) release from the constraint of geometric filters which assume the text lines need to be straight. According to the future work, as the proposed method focus on detecting text on reflective surfaces, automatically reference number detection on the watch in a straightforward manner without matching template. In this case, the reference number based object identification can be significantly improved. Furthermore, more training and validation data are needed to adapt our method to nature images which contain text on both reflective surfaces and non-reflective surfaces.

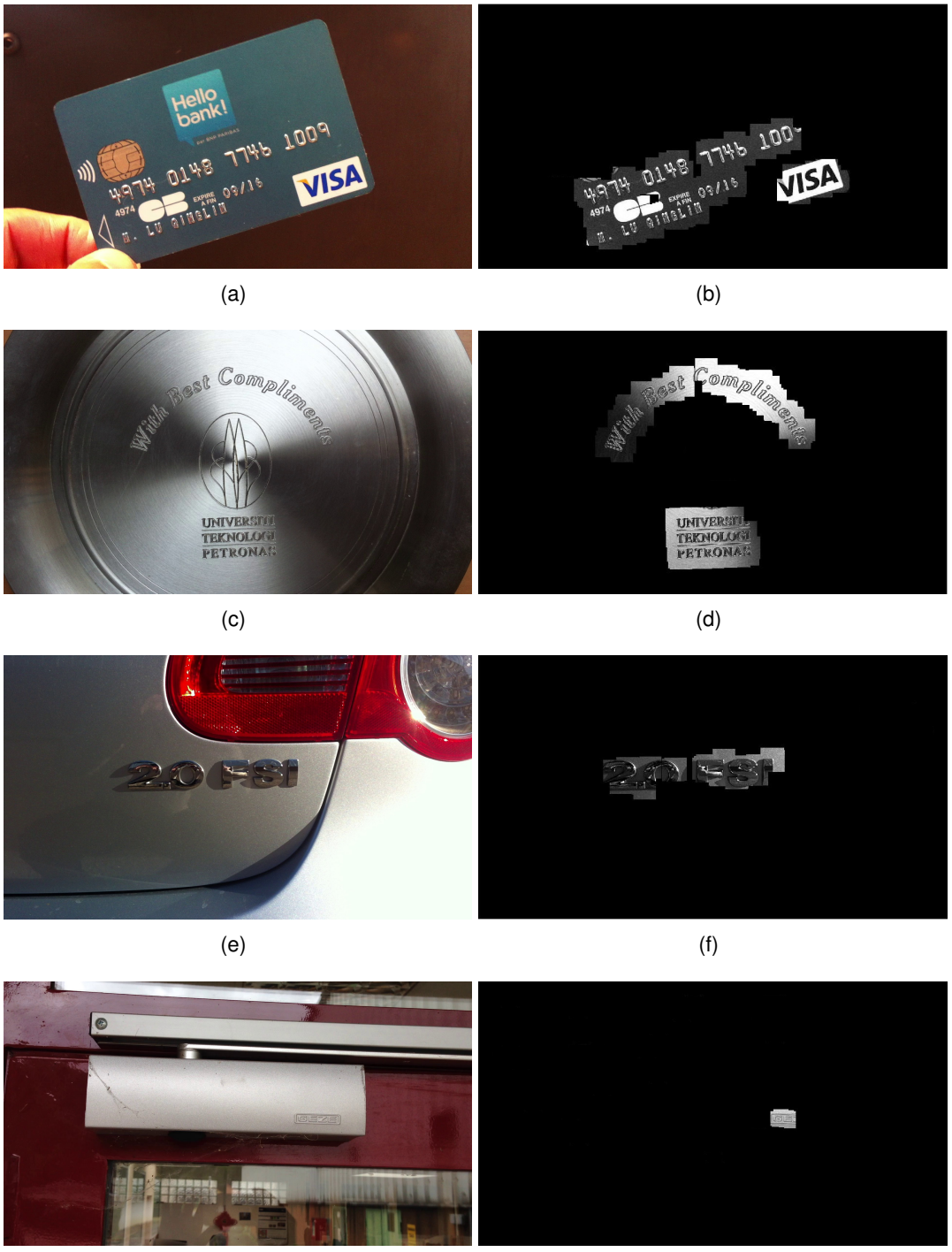


(a)



(b)

Figure 6.18: Evaluation of learning results. (a) is the evaluation of objective function, (b) is the evaluation of validation error.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 6.19: Quantitative results of text detection.

7

Reflective Character Recognition

In the previous chapter we have developed approaches and methods for text areas detection on various reflective surfaces. In this chapter we focus on character recognition present in these areas. In the initial prototype, the character recognition is conducted by tesseract OCR engine that is not specifically designed for solving our problem. Thus we attempt to propose a recognition system to recognize characters that are engraved on the reflective surfaces. This method first adapts several successful local geometric features to make the recognition scale invariant and less sensitive to the reflection. Then the classification performances of SVM classifier are analyzed with three decision boundaries accompanied by individual and combination of the features. The main contribution lies on boosting the recognition performances by introducing two cascaded SVM model based on the previously analyzed accuracy rate. Multiple evaluation results show that the proposed method outperforms single classifier based methods for the recognition of characters engraved on reflective surfaces. Moreover, a challenging dataset is released for further research purpose.

7.1/ Introduction and related works

On various manufactured objects such as coins, signboard, and jewellery, identification signatures which are often characters are engraved on reflective surfaces. These characters contain important semantic information about the object model or ID. Thus, the recognition of these characters is considered as a building block for the object recognition

and identification. In Fig. 7.1, a schematic diagram of object identification by recognizing the engraved characters is depicted. Due to the presence of reflection, high variations change the consistency of local features of the characters. Moreover, they make the surfaces and the characters non-contrasted. Therefore, the recognition of characters engraved on reflective surfaces is not a trivial task.

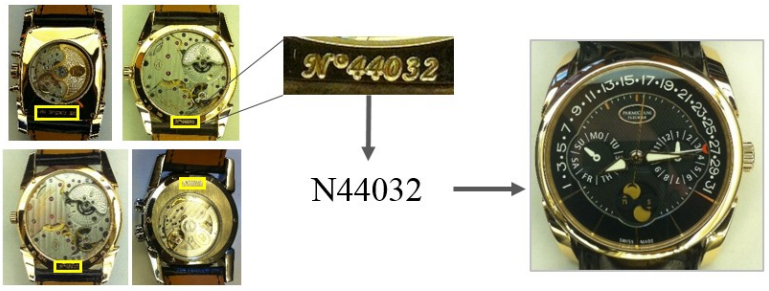


Figure 7.1: Object identification by recognizing characters engraved on the surfaces.

Two main cases being treated in the literature are the recognition on the contrast background and the recognition on the background representing a natural scene. For the recognition of characters on the contrast background, significant results have been achieved. We will mention some of them: Barczak et al. uses moments up to 4^{th} order and a AdaBoost classifier to recognise digits from the MNIST[1] dataset [104]. Ebrahimzadeh et al. by using linear SVM and HOG features achieved an accuracy of 97% [105]. Celar et al. used a different approach by creating bag of words of SIFT features and neural network as classifier, and achieving an accuracy of 94% [106].

As to the second case, the results were also impressive. Epshtein et al. uses the strokes to identify the characters in a text and further recognising the text [71]. A comparative study is carried out in [107] for scene text character recognition, by using HOG feature descriptor and SVM classifier on two datasets Chars 74K[2] and ICDAR 2003[3]. Su et al. proposed a novel technique using convolutional co-occurrence HOG which proved to be robust in detecting text in natural images using SVM [108]. An unsupervised feature learning method (k-means clustering) is used with linear SVM [109]. In terms of classification, cascading classifiers together achieved promising results compared to the single classifier. Zheng et al. used Haar like cascaded classifier with feature HOG; an accuracy

[1] http://yann.lecun.com/exdb/mnist/
[2] http://www.ee.surrey.ac.uk/CVSSP/demos/chars74k/
[3] http://algoval.essex.ac.uk/icdar/Datasets.html

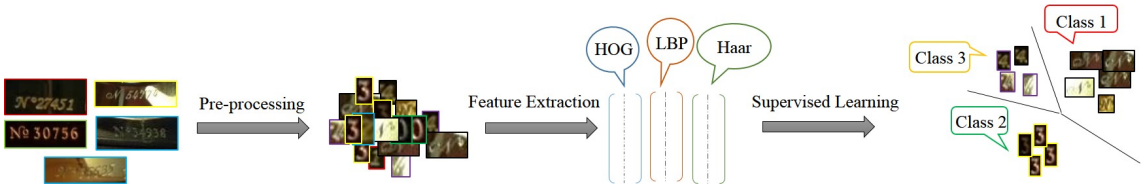


Figure 7.2: A schematic representation of methodology used for recognition of engraved characters on specular surfaces.

of 94% was achieved by their method [110], whereas a cascade classifier approach was used by Heitz et al. for for holistic scene understanding [111]. Sulan et al. use two stage cascade model with a combination of CNN-MPL and CNN-SVM classifiers for recognizing handwritten digit recognition and achieving an accuracy of 99.82%.

The problematic of engraved character recognition is mainly raised due to reflection, which limits us in using frequently employed features and simple learning algorithms. We propose a framework which deals with recognizing engraved characters and depicts characteristics of specularities. A set of traditional character representation features, such as Histogram of Oriented Gradient (HOG), Haar like features and, Local Binary Pattern (LBP) are adapted to be scale invariant and less sensitive to reflection. Furthermore, a two cascade SVM classifier is used for character recognition. The selection of each model is based on the experiments of different combination of selected input features and different SVM decision boundaries. Two models with their respective parameters and input features with highest recognition accuracy are selected. The final recognition results using two cascade SVM models are evaluated through 5-fold cross validation.

7.2/ RECOGNITION OF ENGRAVED CHARACTER ON REFLECTIVE SURFACES

The proposed pipeline consists of three main steps which is illustrated in Fig. 7.2. The initial step preprocessing generates the dataset, where samples are manually collected from industrial acquired videos. The detailed description of dataset is given in Section 7.3.1. Then the features are extracted and adapted to be scale invariant and less sensitive to the reflection. Afterwards, different combinations of SVM classifiers with three decision boundaries with selected input features are evaluated for the cascade model selection

(supervised learning). Finally the character recognition is performed on the proposed cascaded classifier.

7.2.1/ ADAPTED FEATURE EXTRACTION

Based on previous work regarding character recognition with contrast background, we initiate with few similar features such as HOG [112], LBP [113], and Haar-like features [114, 115]. These features led to promising results in licence plate detection [116, 117, 118] and handwriting recognition [119, 105, 120].

7.2.1.1/ HISTOGRAM OF ORIENTED GRADIENTS

Histogram of Oriented Gradients (HOG) features were first introduced by Dalal and Triggs [112] for human detection. Since then, it has been widely used in other applications such as face recognition [121] and pedestrian detection [112, 122, 123]. The HOG features are based on computing the gradient of each pixel in the cell in both X and Y directions. These gradient values are further employed to compute the magnitude and direction values. Then the image is divided into cells, where in our case the cell sizes vary from 2×2 to 5×5 in order to manage the different character scales. Then blocks are presented to combine the gradient information from the cells, each block contains 4 (2×2) cells. The block goes through the whole image by convolution of 50% overlapping (see in Fig. 7.3). The gradient values are between 0 and 180. Thus in each block, the gradients are collected to construct a 9-bins histogram, each bin has a range of $[0, 20]$, as shown in Fig. 7.4. The priority assigned to the direction value to fall in the respected bin depends upon its corresponding magnitude value. The obtained histogram is further normalized and the HOG features are given by this histogram.

7.2.1.2/ LOCAL BINARY PATTERN

The Local Binary Pattern (LBP) [124] offers the advantage to encode information about the local neighbourhood around the selected pixels. The LBP descriptor studies mainly the texture information and it has been used in applications such as attitude recognition [125], character recognition in Chinese licence plate detection [118] etc. The advantage

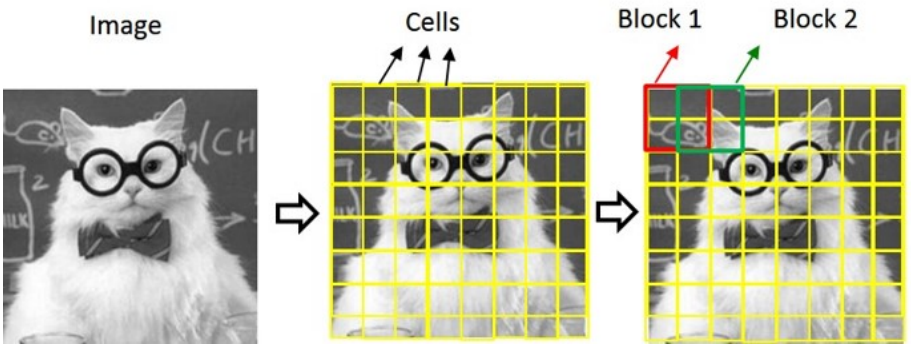


Figure 7.3: Illustration of blocks and cells convolution on the image.

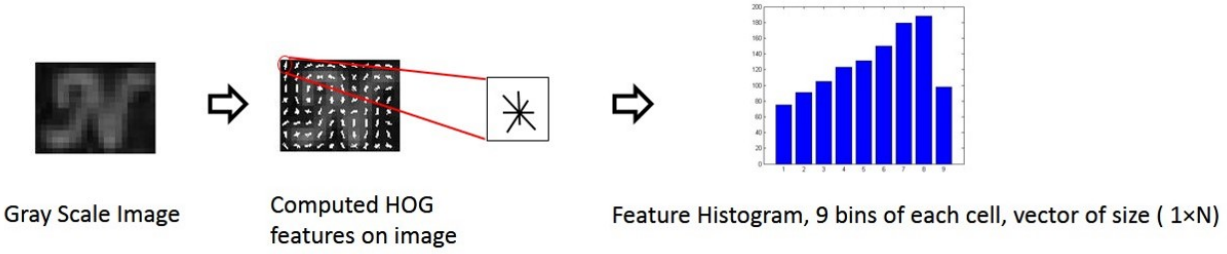


Figure 7.4: Illustration of HOG features computation.

of LBP features is its invariance to illumination conditions. The LBP feature is calculated by comparing the central pixel value within its 8 neighboring pixel values. The central value in the mask is denoted as V_C and its 8 neighboring values are denoted as V_C^n, where $n \in [1, 8]$. The neighbor labeling value of V_C is denoted as BV_C^n, and the computed LBP value of V_C is denoted as $\hat{V_C}$. The computation of $\hat{V_C}$ is presented by equations 7.2.1.2 and 7.2.1.2:

$$BV_C^n = \begin{cases} 0 & if \quad V_C^n <= V_C \\ 1 & if \quad V_C^n > V_C \end{cases} \tag{7.1}$$

$$\hat{V_C} = \sum_{n=0}^{7} 2^7 \times BV_C^n. \tag{7.2}$$

The generated binary pattern is further converted into a decimal value which is in the interval $[0, 255]$, followed by arranging into a histogram (As shown in Fig. 7.5). In our case, 3×3 mask is used due to the small image size.

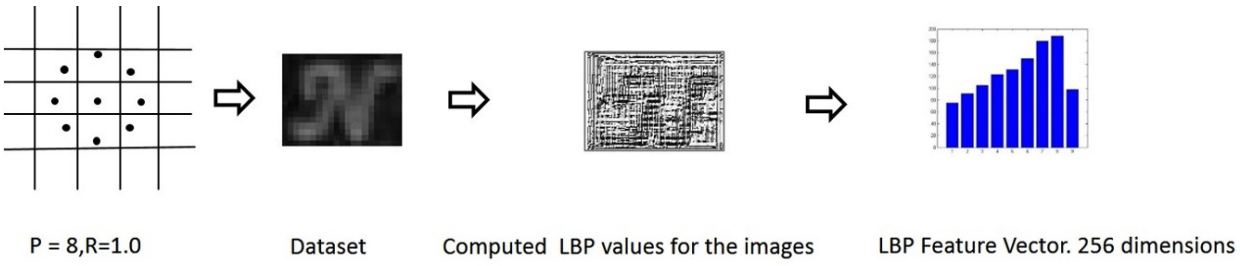


Figure 7.5: Illustration of LBP features computation.

7.2.1.3/ HAAR-LIKE FEATURES

Haar-like features were first introduced by Viola and Jones [126] and used for pedestrain detection [115] and object detection [114]. Haar-like features employ different masks to study the gradient information in the image (see Fig. (7.6)). The intensity is summed in the white and black regions respectively and then subtracted between these two regions.

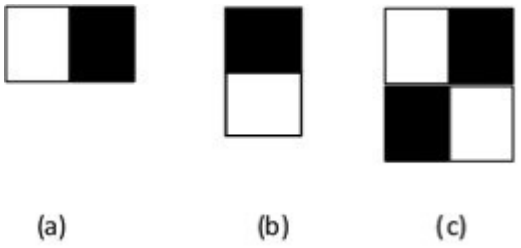


Figure 7.6: Masks employed by Haar-like features

In our case, vertical, horizontal and diagonal Haar like features have been used to extract features. Using these three features the edges are computed for all the directions. All values are normalized by equation 7.2.1.3 so that the feature values were positive. Then for each mask, the Haar values are collected to build the histogram.

$$HaarValue = 128 \times \frac{diffVal}{510} + 128. \tag{7.3}$$

These three features are extracted as an essential characteristics of the characters. However, as the data set contains images with different sizes, the dimension of computed features vector would be different according to different sizes of the images. It leads to the confusion of the learning process for the character recognition. In order to solve this

problem, we rearrange the features in a statistic way to adapt the features to be scale invariant and less sensitive to the reflection. For LBP features and Haar-like features, as the dimensions for all the images are the same, we convert the histograms into 9 bins to reduce the feature dimension. Thus the LBP features are now represented by a vector of dimension 9, Haar-like features are represented by 3 vectors of dimension 9. For HOG features, due to the various dimensions for different images, the features need to be normalized. Therefore, resizing all the images into equal size is one strategy. However, it spoils the geometrical properties of the characters in the image. Thus we adapt the features by 4 central moments: mean value, standard deviation(STD), skewness and kurtosis. The histogram which is computed for each cell contains 9 bins which are aligned one below the other so that each column would correspond to each bin, as shown in Fig. 7.7. Further the mean, standard deviation, skewness and kurtosis are calculated for each of the respected bins. This modifed HOG features lead to equal feature dimension for each image.

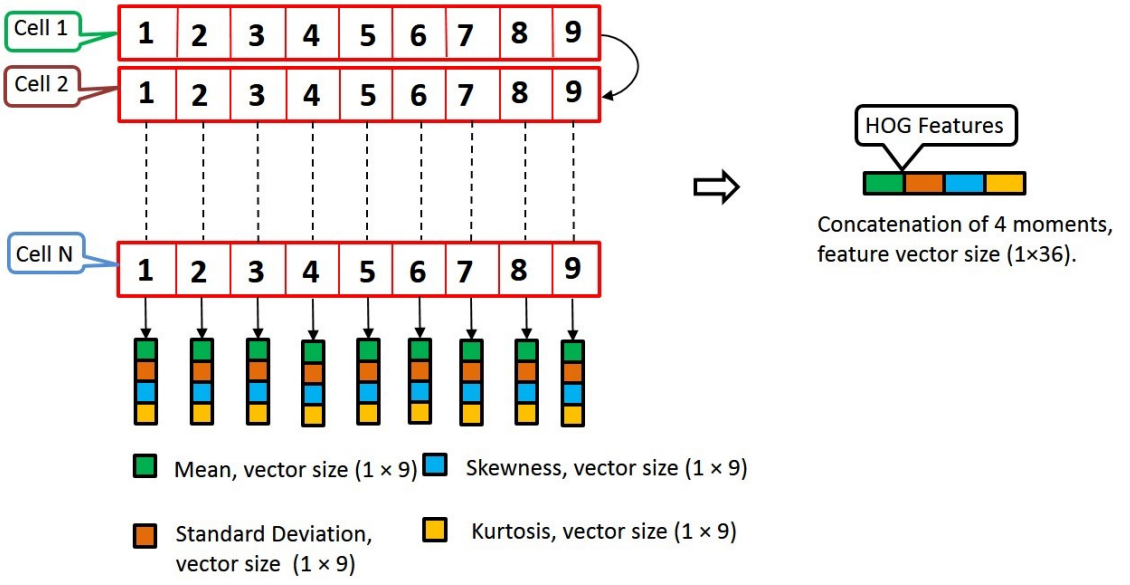


Figure 7.7: Adapted Haar-like features

The used central moments describe the characteristics of a distribution. The central moment of order k is defined by:

$$m_k = E(x - \mu)^k, \tag{7.4}$$

where $E(x)$ is the expected value of x.

Mean (μ) is the central value of the probability density function (PDF), which can be calculated by:

$$\mu = \frac{1}{N}\sum_{i=0}^{N} A_i, \tag{7.5}$$

where N is the number of samples A_i.

Standard Deviation (S) shows the variation from the sample to the mean of the PDF i.e. to see how wide is the distribution of the samples. S is the square root of the variance:

$$S = \sqrt{\frac{1}{N}\sum_{i=0}^{N} |A_i - \mu|^2}. \tag{7.6}$$

Skewness (sk) is the 3rd order moment giving information on the symmetry properties. If the tail of the distribution is longer on the left side of the mean, the skewness is negative and if the tail of the distribution is longer on the right the skewness is positive. The skewness of a distribution is calculated based on equation:

$$sk = \frac{E(x-\mu)^3}{S^3}. \tag{7.7}$$

Kurtosis (k) is the 4th order central moment and is related to the sharpness of the peak of the distribution. For a distribution prone to outliers, the kurtosis will be greater than 3 and otherwise if the distribution is not prone to outliers it will be less than 3. The kurtosis of a distribution is computed based on equation:

$$k = \frac{E(x-\mu)^4}{S^4}. \tag{7.8}$$

With these four central moments, the dimension of the HOG features for all the input images is normalized from more than 1000 into 36.

Finally, as shown in Fig. 7.8, for each input image, the features extracted with HOG, Haar, and LBP are concated into a 1D vector with $72(36 + 27 + 9)$ values. Subsequently, these features are employed by a cascaded two SVM model for supervised learning and further the character recognition.

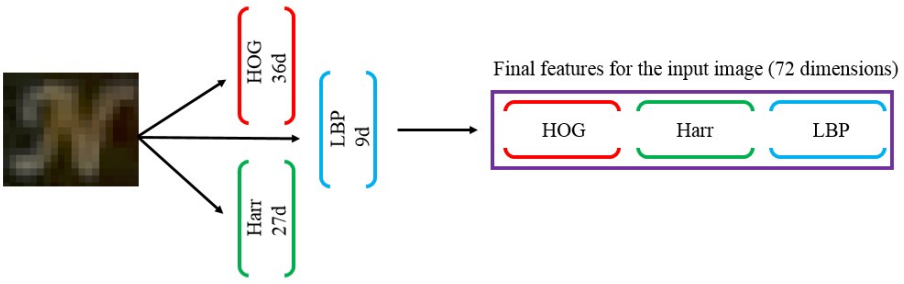


Figure 7.8: Ordered concatenation of features HOG + Haar + LBP.

7.2.2/ CASCADED SVM MODEL

Within the extracted features, a classification is performed to recognize the characters. Since the size of dataset is not enormous, SVM classifiers based on decision boundaries are employed. As mentioned in the literature, cascaded classifier models boost the classification accuracy and generally outperform single classifier model. Thus a cascaded two SVM model is proposed in order to increase the number of strongly confident samples and further boost the recognition accuracy.

The proposed cascaded model is illustrated in Fig. 7.9. The two models are initially trained by the training data (described in 7.3.1). The testing data is first introduced to model 1 where a recognition confidence (RC) is acquired (detailed explanation in section 7.2.2.2). The samples (images) with high recognition confidence are denoted as high confidence samples and the rest are treated as low confidence samples. Then the low confidence samples are further fed to model 2 as testing data. The final results are combination of RC obtained from model 1 and model 2, later discussed in section 7.2.2.3. A threshold parameter θ is introduced, which determines the confidence of the sample (high or low confidence).

7.2.2.1/ MODEL SELECTION

The linear SVM, rbf (radial basis function) and polynomial SVM kernels are evaluated within all the combinations of extracted features (see Table 7.1). As the feature dimension is bigger than 500, SVM with linear and rbf kernel perform better as compared to taking a polynomial of degree $>= 1$. It is due to the highly non linear property of polynomial kernel which is less competitive for high dimensional features. The parameters

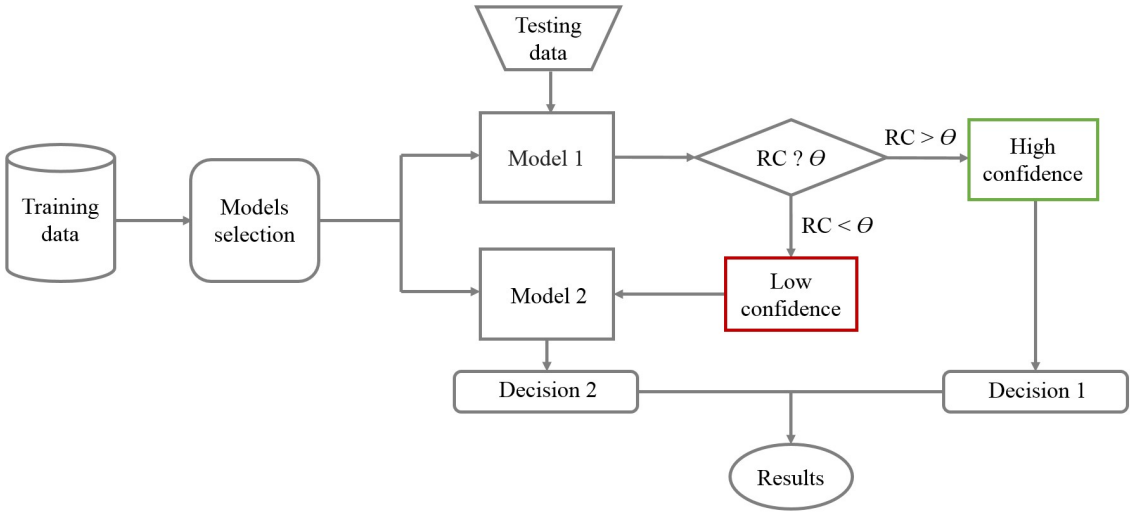


Figure 7.9: A schematic block diagram representation of two cascaded SVM models.

of the SVM classifiers are estimated by using grid search method. The linear SVM with the combination of features (HOG + LBP) which achieve the best classification rate is selected as model 1. Whereas, rbf SVM with the combination of features (LBP + Haar-like) is selected as model 2.

Features	Linear	rbf	Polynomial
LBP	75.32	71.05	66.49
Haar-like	58.02	59.58	33.27
HOG	71.71	44.92	65.82
HOG+LBP	**79.70**	18.98	70.81
LBP+Haar-like	77.30	**79.46**	57.17
HOG+Haar-like	75.14	46.31	71.47
LBP+HOG+Haar-like	78.68	46.13	73.33

Table 7.1: Single SVM model with selected features evaluation using 5-fold cross-validation on dataset, for cascade model selection.

7.2.2.2/ RECOGNITION CONFIDENCE

When the training data is introduced to model 1, a *Recognition Confidence* (RC) which is, the probabilities of each sample belonging to all the classes. In our case, as there are 11 classes, RC is a vector of 11 probability values, the sum being equal to 1.0. The maximum RC value for each sample is selected and compared to the threshold θ. The samples which have RC value bigger than θ are denoted as high confidence samples,

otherwise, the samples are denoted as low confidence samples and pass to model 2 for further classification. In the experiments, different values of θ are tested to obtain the best RC threshold (discussed in results).

7.2.2.3/ DECISION MAKING

The decision (D) of samples is based on combining the classification rate from the two models. The RC value of the high confidence samples and low confidence samples obtained by the model 1 are denoted by M_1^h and M_1^l, respectively. The RC value of low confidence samples obtained by model 2 are denoted by M_2^l. Two combination methods are proposed: (i) **Max-pooled cascaded model:** For each low confidence sample selected by model 1, the RC from model 2 is compared to the RC of model 1. The higher RC is accepted as the predicted class. (ii) **Simple cascaded model:** The decision from model 2 is accepted and the final decision is made from the predicted class of high confidence samples from model 1 and that of low confidence samples in model 2. These two combination methods are illustrated by the equations 7.9 and 7.10.

$$D_{max-pooled} = M_1^h \cup max\left\{M_1^l, M_2^l\right\}, \tag{7.9}$$

$$D_{simple} = M_1^h \cup M_2^l. \tag{7.10}$$

7.3/ CHARACTER RECOGNITION RESULTS

The experiments are conducted in the industrial non-controlled lighting environment. All the datasets were captured from the shiny metallic surface of the industrial objects with engraved characters.

7.3.1/ DATASET

The characters in our dataset are engraved on reflective surface, thus high intensity variations caused by reflection are present in the images. A part of the dataset is shown in Fig. 7.10. The dataset contains ten classes of digits (0-9) and one class of letter (N). The

resolution of the collected data varies from $[8 \times 15]$ to $[28 \times 18]$ and the number of samples is 1666. This dataset with respective labels is available on http://visor.udg.edu/i2cvb/ for further research purpose.

7.3.2/ RECOGNITION RESULTS

In the proposed cascaded classification model, the threshold θ plays an important role for defining the sample confidence. The recognition performances are evaluated by varying θ (see Fig. 7.11). The accuracy becomes stable at 0.84 when θ reaches 0.8 in max-pooled cascaded model. The accuracy decreases when θ crosses 0.6 in simple cascaded model. It is due to the fact that the max-pooled cascaded model selects the better maximum RC value between model 1 and 2. It also guarantees that the weak samples are classified with a higher recognition confidence.

On the other hand, the classification details for each class are illustrated in the form of a confusion matrix for both proposed methods in Fig. 7.12 and Fig. 7.13. The X-axis represents the actual classes and Y-axis represents the predicted classes. The proposed model performs meaningful classification for classes C_0, C_2, C_7, and C_N. However, for both classes C_8 and C_9, the model misclassified 10% of the samples into C_3 and C_4, which leads to a lower accuracy. Nevertheless, the total recognition accuracy of 84.08% in such a dataset still remains significant.

In order to qualitatively visualize the character recognition performances of the proposed method, the results for both correctly classified samples and misclassified samples are shown in Fig. 7.14. Note that the presence of reflection breaks the intensity consistency of both character and the background, making the feature learning process very difficult. In spite of this fact, even the most challenging samples in our case is still correctly classified. On the other hand, the misclassified cases are mainly due to artefacts (plastic cover, scratches) and neighbouring character parts. Orientation dependent features used in our experiments could be an another reason for misclassification. Note that the misclassified box of class C_0 is empty, it means that the recognition accuracy of class C_0 is 100%.

Finally, a comparison study with several promising single classifiers based on Random Forest (RF) and K-Nearest Neighbour (KNN) is shown in Table. 7.2. The evaluation is performed by using 5-fold cross validation for all the experiments. The final result for each

classifier is shown in the last column which is the average of recognition accuracy for 5 folds. For RF, the experiments are conducted by varying the number of trees to obtain the best possible recognition accuracy 70.08% (using 500 trees). In case of KNN, the number of clusters is set to 11. The results show that the proposed method significantly outperforms the other competitors. Such high level accuracy in our case is due to the combination of selected features and cascaded architecture of the classifier.

Method	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	All Folds
Linear SVM	79.88	79.28	77.78	80.18	81.38	79.70
rbf SVM	80.48	77.47	74.77	81.98	82.58	79.46
Random Forest	74.47	69.07	72.37	68.17	69.97	70.81
KNN classifier	63.06	62.46	60.66	64.86	63.36	62.88
our method (Simple model)	82.88	81.98	79.88	81.38	82.28	81.68
our method (MP model)	84.98	84.38	82.58	81.98	86.49	**84.08**

Table 7.2: Comparison with promising classifiers. The max-pooling model obtains the best recognition accuracy based on the average of all folds.

7.4/ CONCLUSION

We presented two cascaded SVM models for character recognition on reflective surfaces. We first adapted the local features to be scale invariant and less sensitive to reflection, then studied the recognition performances of the different combinations of features trained by single SVM classifiers with three decision boundaries. Based on the performance of decision boundaries and feature combinations, two best models were selected. Later, these models were used in a cascaded architecture whose final results were chosen in two different strategies. Multiple evaluation results on 1666 samples show that the max-pooled cascaded model works better than the simple cascaded method. However both the strategies outperform than the competitors. Additionally, this dataset could be used for further research purpose. Regarding to future work, we strongly believe that higher image resolution would produce a better accuracy, thus a super resolution technique could be employed. Moreover, we will generate more training data by adding simulated reflective patterns on the original images and collect more data from various fonts. It will significantly increase the character recognition accuracy and improve the object identification performance.

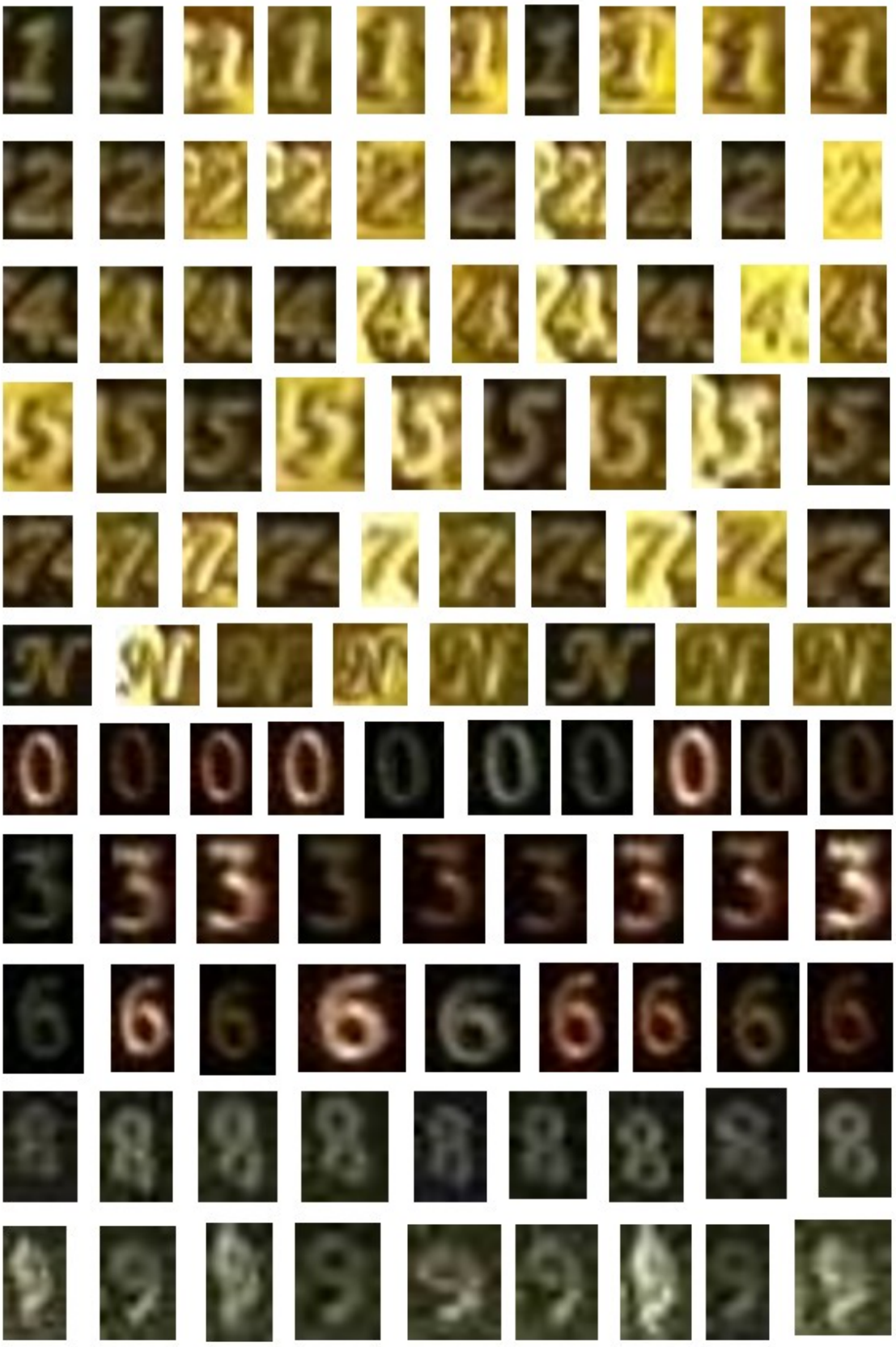


Figure 7.10: Part of the data set we used for the character recognition.

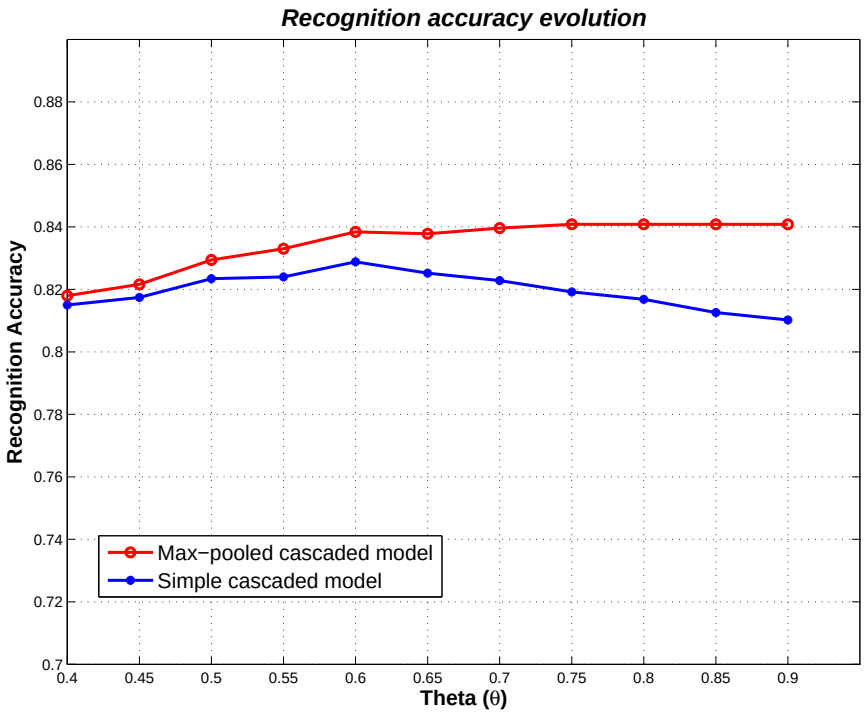


Figure 7.11: Evaluation of the recognition accuracy for both cases. The accuracy becomes stable when θ reaches 0.7 in Max-pooled cascaded model, but it decreases when θ reaches 0.6 in simple cascaded model.

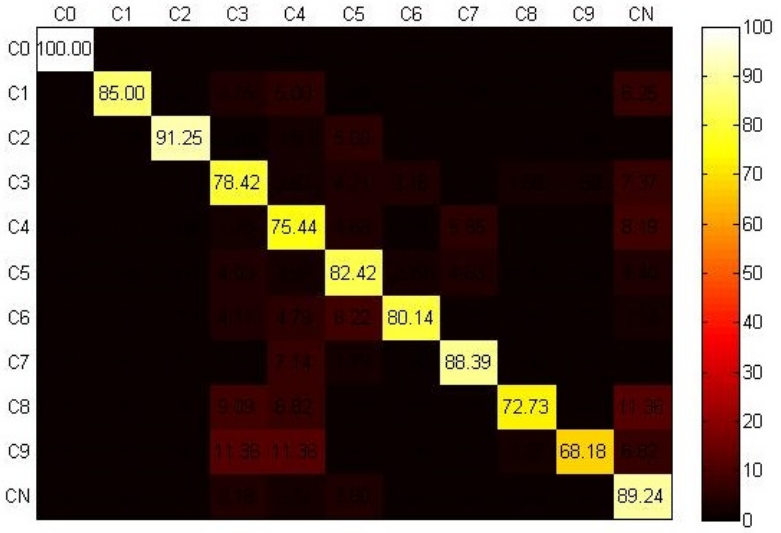


Figure 7.12: Confusion matrix depicting performances of max-pooled cascaded model.

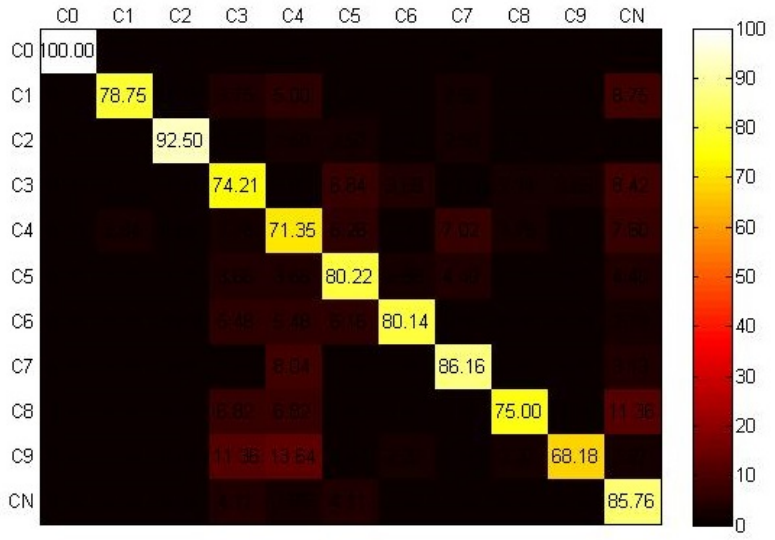


Figure 7.13: Confusion matrix depicting performances of simple cascaded model.

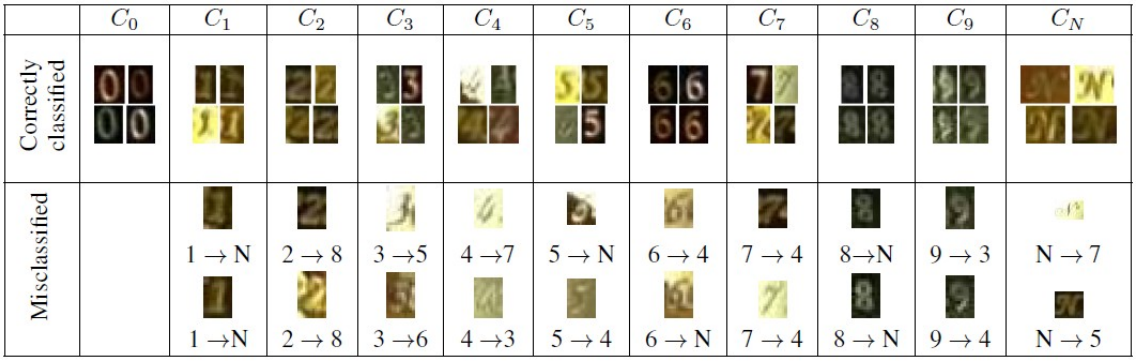


Figure 7.14: Qualitative results of character recognition. Most of the misclassified samples are highly affected by the reflection.

8

Conclusion

The presented works in this thesis address the reflective manufactured object recognition and identification. We aim to develop efficient and accurate algorithms for the real industrial applications in order to automatize and facilitate the object recognition and identification prototype both in the manufactures and the stores. To fulfill the objective, we initially proposed an prototype that contains database construction, object template matching for the object model recognition, and optical character recognition for the object identification. Furthermore, aiming at the limitations of the prototype, we studied and presented four algorithms in succession to improve both the object recognition and identification performance of the prototype.

Setup and initial prototype The setup is built in terms of a box, in what the lighting condition, object pose, and camera pose are controlled. Using this setup, a dataset consists of over 1000 images and 45 videos for 10 different watches from 6 different models collected in the controlled environment. The initial prototype which contains template creation, object model recognition, and object identification is implemented and evaluated on the acquired dataset.

Setup and Database construction

We first study the image and video acquisition in an controlled environment. In order to do this, a experimental box is designed and constructed. With this box, different illumination and acquisition conditions can be modeled, such as the positions and the orientations of the light source, objects, and the camera, respectively. The variations of object, camera, and light source poses provide more training data from various object poses in multiple views through different illumination conditions. The dataset contains more than 1000

images and 45 videos of 10 different watches from 6 different models. The dataset is used for the model template creation, object model recognition, as well as the generation of reflective characters dataset for the further object identification.

Initial prototype

The prototype contains 3 main stages: offline processing attempts to create object model template based on the external contour. Online processing addresses object tracking and template matching based on global features and contour segment features. Object identification aims to localize the reference number in the video and recognize the characters contained in the reference by using Tesseract OCR engine. In conclusion, we summarize the prototype and discuss its limitations.

Improvement of object model recognition The improvements of object model recognition mainly rely on extracting more representative features. The first feature is object local surface curvature. This feature extraction method is experimented on analyzing the surface curvature of the watch feet and achieved meaningful results. The second feature is the surface distribution graph based on object sub-segmentation. The segmentation method is tested on various reflective objects under different condition and performs promising results.

Object local surface curvature analysis

The proposed method focus on analyzing specular surface curvature profile using a single line source. This single line source could be any straight line in the environment whose position can be easily measured. By studying the geometric relation in the system, the local surface curvature of the experimented object can be estimated according to the distortion of the line that are reflected by the object surface. For the reflective object recognition, the local surface curvature provides a novel representative and reliable feature that can be used to discriminate different object models.

Object surface structure understanding

The proposed method first tracks the moving reflection particles in the video, then uses the motion trajectories as surface labels, afterwards segments the elementary continuous surfaces based on the trajectories. After the object surface segmentation, the graph which describes the surfaces distribution is constructed as a new feature for the object repre-

sentation. The proposed segmentation method provides a new perspective concerning reflection in computer vision. Instead of removing/reducing reflection, taking its advantage is pioneering the work in a new direction. Within the surface graph features, the object recognition and template matching performances can be significantly improved.

Improvement of object identification The improvements of object identification mainly rely on two aspects: text detection and character recognition. Both of the two aspects are under the constraint of being on the reflective surface.

Text detection on reflective surfaces

The proposed text detection method initially extracts low level features such as points, contours, and regions. Then similar features are clustered in order to precisely select the text candidates. Afterwards a powerful classifier is trained to predict text zone in the image. This method liberates the constraint that the reference number localization have to rely on the object contour tracking, template matching, as well as the prior knowledge. The detection is straightforward and also suitable for the text in the natural scene. The proposed text detection method not only improves the reference number localization accuracy, also simplifies the prototype. In this regard, the object identification stage can be conspicuously ameliorated.

Recognition of characters engraved on reflective surfaces

Our character recognition method first adapts several successful local geometric features to make the recognition scale invariant and less sensitive to the reflections. Then the classification performances of SVM classifier are analyzed with three decision boundaries accompanied by individual and combination of the features. The main contribution lies on boosting the recognition performances by introducing two cascaded SVM model based on the previously analyzed accuracy rate. Multiple evaluation results show that the proposed method outperforms single classifier based methods for the recognition of characters engraved on reflective surfaces. Moreover, a challenging dataset is released for further research purpose. With the proposed method that is specifically designed for recognizing characters on reflective surfaces, the tesseract OCR engine can be replaced. In this case, the reference number recognition based object identification can achieve a better accuracy.

The proposed prototype and the improvements aim to industrialize the computer vision and machine learning algorithms in aspects of manufactured object recognition and identification. In several years, the automatic prototypes will release more and manpower in both the industry and the stores. We expect our works can be considered as a building block for the further industrial applications.

8.1/ FUTURE WORK

The future work consists of two main aspects: algorithmic perspectives and industrial perspectives.

Regarding to the algorithmic perspectives, each of the proposed methods guide the work to a new direction. The experiment setup allows us adding more light sources in order to simulate divers illumination environments (simple / complex). The initial prototype provides us the problematic concerning to the object recognition and identification that we latter investigated. In object recognition, the local surface curvature analysis leads to the 3D reconstruction of the entire reflective object under few constraints. Furthermore, estimating the surface curvature of a complex shape will be also expected. On the other hand, by studying surface structure graph, we find an interesting topic: since the shape of reflection change because of the movement of light source, and hence its reflection on the object surface, thus to explore the evolution of reflection shape and extract additional reflection motion features from it will be very impressive and amusing. In object identification, the text detection on reflective surfaces needs to be tested on public dataset and compared with the state-of-the-art methods. More importantly, enhance the detection results by using random markov field [127] and obtain smooth text regions are crucial. For the character recognition, we will generate more training data by adding simulated reflective patterns on the original images and collect more data from various fonts. It will significantly increase the character recognition accuracy and improve the object identification performance.

Regarding to the future work for industry, integrating in the prototype the object recognition and identification improvements is the primary perspective. In object model recognition, more abundant features such as local surface curvature and surface distribution graph have been extracted for the object representation. Thus, using external contour

and the previous two additional features will lead to a boosted accuracy for object model recognition. In object identification, replacing the prior knowledge about reference number localization with our text detection method can simplify the dataset construction process and provide more accurate reference number bounding box. Moreover, as the proposed character recognition method outperforms tesseract OCR engine on our dataset that are engraved on reflective surfaces, replacing the tesseract OCR engine by the proposed method is supposed to give better object identification results. Furthermore, acquiring more dataset from different types of object besides watch is also important. The acquired dataset temporarily consists of only watches as manufactured object. However, within the development of the company, more and more manufactured objects such as jewelry, mobile phone, bank card, and even automobile will be included in our recognition and identification system. In consequence of that, enlarge and enrich the dataset on both quantity and variety is crucial.

9

ANNEXE

9.1/ ANNEXE 1: OBJECT GEOMETRIC PROPERTY IN ACQUIRED IMAGE

$$len_{img}(C_1C_2) = \frac{h'}{2h} \times \overline{C_1C_2}, \tag{9.1}$$

$$len_{img}(C_1'C_2') = \frac{h'}{2(h + \overline{C_1'C_2'}\sin\theta)} \times \overline{C_1'C_2'}\cos\theta + \frac{h'}{2(h - \overline{C_1'C_2'}\sin\theta)} \times \overline{C_1'C_2'}\cos\theta, \tag{9.2}$$

$$\overline{C_1C_2} = \overline{C_1'C_2'}, \tag{9.3}$$

$$C(\theta) = \frac{len_{img}(C_1C_2)}{len_{img}(C_1'C_2')} = \frac{\cos(\theta)h^2}{h^2 - \sin(\theta)^2\overline{C_1'C_2'}^2}. \tag{9.4}$$

The value of the rotation angle θ is limited in $[0, \phi/2]$. The corresponding values of $C(\theta)$ are ploted in Fig. 9.2, the evoluation of $C(\theta)$ starts from 1 and exponentially increases. It demonstrates that $len_{img}(C_1C_2)$ is always longer than $len_{img}(C_1'C_2')$ through the whole rotation. $len_{img}(C_1C_2)$ corresponds to the center distances from two sides of external contours when the object pose is perpenticular to the camera, $len_{img}(C_1'C_2')$ corresponds to the center distances from two sides of external contours when the object is rotated. The evaluation of $C(\theta)$ illustrate the fact that $len_{img}(C_1'C_2')$ decreases within the increase of rotation angle θ.

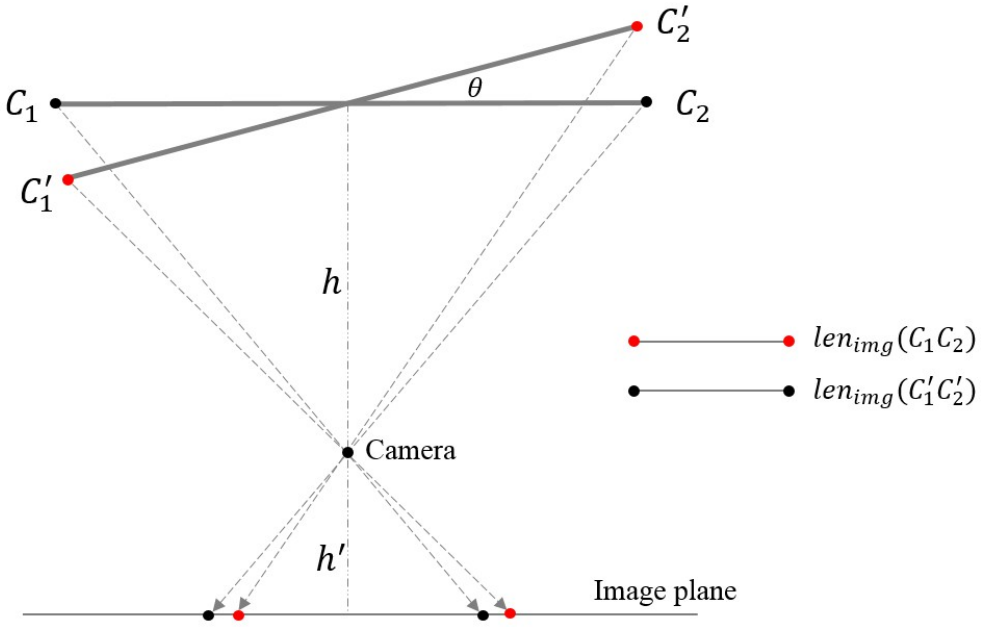


Figure 9.1: Object rotation and its image property.

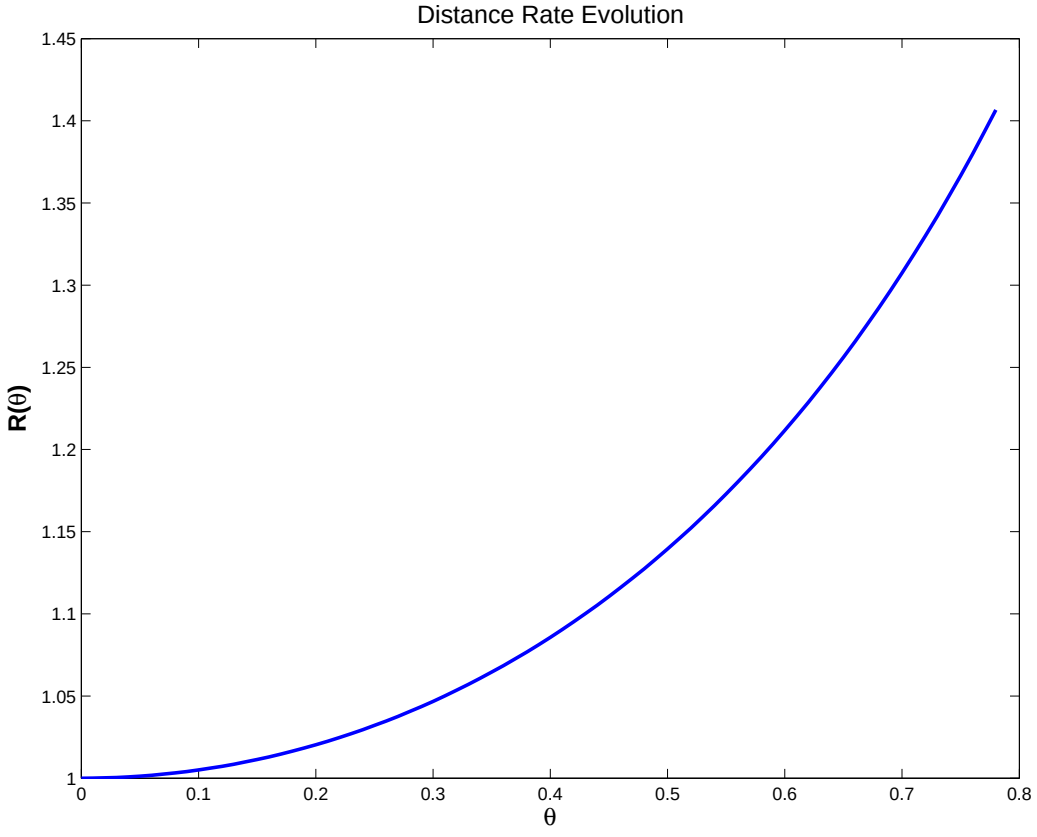


Figure 9.2: Rate

BIBLIOGRAPHY

[1] Jia Deng, Wei Dong, Richard Socher, Li jia Li, Kai Li, and Li Fei-fei. Imagenet: A large-scale hierarchical image database. In *In CVPR*, 2009.

[2] Chris Harris and Mike Stephens. A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, pages 147–151, 1988.

[3] D.G. Lowe. Object recognition from local scale-invariant features. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1150–1157 vol.2, 1999.

[4] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.

[5] J Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, June 1986.

[6] O. Laligant and F. Truchetet. A nonlinear derivative scheme applied to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(2):242–257, 2010.

[7] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, CVPR '05, 2005.

[8] R. E. Kalman. A new approach to linear filtering and prediction problems. *ASME Journal of Basic Engineering*, 1960.

[9] D. J. MacKay. Information theory, inference and learning algorithms. *Cambridge university press*, 2003.

[10] Michael Isard and Andrew Blake. Condensation—conditional density propagation forvisual tracking. *Int. J. Comput. Vision*, 29(1):5–28, August 1998.

[11] Michael Isard and Andrew Blake. Condensation - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29:5–28, 1998.

[12] Myung-Cheol Roh, Tae-Yong Kim, Jihun Park, and Seong-Whan Lee. Accurate object contour tracking based on boundary edge selection. *Pattern Recogn.*, 40(3):931–943, March 2007.

[13] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *INTERNATIONAL JOURNAL OF COMPUTER VISION*, 1(4):321–331, 1988.

[14] N. Peterfreund. Robust tracking of position and velocity with kalman snakes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(6):564–569, 1999.

[15] Y. Fu, A.T. Erdem, and A.M. Tekalp. Tracking visible boundary of objects using occlusion adaptive motion snake. *Image Processing, IEEE Transactions on*, 9(12):2051–2060, 2000.

[16] Chuong T. Nguyen, Joseph P. Havlicek, and Mark B. Yeary. Modulation domain template tracking. In *CVPR*. IEEE Computer Society, 2007.

[17] Erkut Erdem, Séverine Dubuisson, and Isabelle Bloch. Fragments based tracking with adaptive cue integration. *Comput. Vis. Image Underst.*, 116(7):827–841, July 2012.

[18] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 798–805, 2006.

[19] Kai Nickel and Rainer Stiefelhagen. In *Computer Vision ECCV 2008*, volume 5305, pages 514–526. 2008.

[20] S.M. Shahed Nejhum, J. Ho, and Ming-Hsuan Yang. Visual tracking with histograms and articulating blocks. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, 2008.

[21] P. Chockalingam, N. Pradeep, and S. Birchfield. Adaptive fragments-based tracking of non-rigid objects using level sets. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1530–1537, 2009.

[22] Shivani Agarwal and Dan Roth. Learning a sparse representation for object detection. In *Proceedings of the 7th European Conference on Computer Vision-Part IV*, ECCV '02, pages 113–130, 2002.

[23] Bastian Leibe, Edgar Seemann, and Bernt Schiele. Pedestrian detection in crowded scenes. In *In CVPR*, pages 878–885, 2005.

[24] D. M. Gavrila and et al. Real-time object detection for "smart" vehicles. In *INT'L CONF. ON COMPUTER VISION, CORFU*, pages 87–93, 1999.

[25] D.M. Gavrila. A bayesian, exemplar-based approach to hierarchical shape matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(8).

[26] Nguyen Duc Thanh, Philip Ogunbona, and Wanqing Li. Human detection based on weighted template matching. In *Proceedings of the 2009 IEEE International Conference on Multimedia and Expo*, ICME'09, pages 634–637, 2009.

[27] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'77, pages 659–663, 1977.

[28] Daniel P. Huttenlocher, Gregory A. Klanderman, Gregory A. Kl, and William J. Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:850–863, 1993.

[29] W.J. Rucklidge. Locating objects using the hausdorff distance. In *Computer Vision, 1995. Proceedings., Fifth International Conference on*, pages 457–464, 1995.

[30] Herbert Freeman. On the encoding of arbitrary geometric configurations. *Electronic Computers, IRE Transactions on*, EC-10(2):260–268, June 1961.

[31] R. Smith. An overview of the tesseract ocr engine. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, 2007.

[32] S. Savarese and P. Perona. Local analysis for 3d reconstruction of specular surfaces. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 2, pages II–738–II–745 vol.2, 2001.

[33] Silvio Savarese and Pietro Perona. In *Computer Vision ECCV 2002*, volume 2351 of *Lecture Notes in Computer Science*, pages 759–774. 2002.

[34] Bjrn Barrois and Christian Whler. 3d pose estimation based on multiple monocular cues. In *CVPR*. IEEE Computer Society, 2007.

[35] G.A. Atkinson and E.R. Hancock. Recovery of surface orientation from diffuse polarization. *Image Processing*, 15(6):1653–1664, June 2006.

[36] Ralph Seulin, Frederic Merienne, and Patrick Gorria. Simulation of specular surface imaging based on computer graphics: Application on a vision inspection system. *EURASIP J. Appl. Signal Process.*, 2002(1):649–658, January 2002.

[37] L.S. Kerrigan and W.J. Adams. The perception of gloss: A review. *Vision Research*, 109, Part B:221 – 235, 2015. Perception of Material Properties (Part I).

[38] Fleming RW, Torralba A, and Adelson EH. Specular reflections and the perception of shape. *Journal of Vision*, 4:798 – 820, 2004.

[39] Mark A. Halstead, Brain A. Barsky, Stanley A. Klein, and Robert B. Mandell. Reconstructing curved surfaces from specular reflection patterns using spline surface fitting of normals. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pages 335–342. ACM, 1996.

[40] Jiang Yu Zheng and Akio Murata. Acquiring a complete 3d model from specular motion under the illumination of circular-shaped light sources. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):913–920, 2000.

[41] A. Treuenfels. An efficient flood visit algorithm. *C/C++ Users Journal*, 12, 1994.

[42] X. Guo, X. Chao, and Y. Ma. Robust separation of reflection from multiple images. *CVPR*, 2014.

[43] M. D'Zmura and P. Lennie. Mechanism of color constancy. *JOSA*, 3:1162–1172, 1986.

[44] R.T. Tan and K. Ikeuchi. Separating reflection components of textured surfaces using a single image. *PAMI*, 25:178–193, 2005.

[45] R.T. Tan and K. Ikeuchi. Reflection components decomposition of textured surfaces using linear basis functions. *CVPR*, 2005.

[46] S. Savarese and P. Perona. Local analysis for 3d reconstruction of specular surfaces. *CVPR*, 2001.

[47] S. Savarese and P. Perona. Local analysis for 3d reconstruction of specular surfacespart ii. *ECCV*, 2002.

[48] A. Delpozo and S. Savarese. Detecting specular surfaces on natural images. *CVPR*, 2007.

[49] B. Barrois and C. Wohler. 3d pose estimation based on multiple monocular cues. *BenCOS*, pages 1–7, 2007.

[50] M. Grundmann, V. Kwatra, and I. Essa M. Han. Efficient hierarchical graph-based video segmentation. *CVPR*, 2010.

[51] S. Paris and F. Durand. A topological approach to hierarchical segmentation using mean shift. *CVPR*, 2007.

[52] C. Xu and J.J. Corso. Evaluation of seper-voxel methods for early video processing. *CVPR*, 2012.

[53] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59, 2004.

[54] Kleber Jacques Ferreira de Souza, Arnaldo de Albuquerque Arajo, Zenilton K.G. do Patrocnio Jr., and Silvio Jamil F. Guimares. Graph-based hierarchical video segmentation based on a simple dissimilarity measure. *Pattern Recognition Letters*, 47, 2014.

[55] J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, 22:888–905, 2000.

[56] M. Grundmann, V. Kwatra, M. Han, and I. Essa. efficient hierarchical graph-based segmentation of rbgd videos. *CVPR*, 2010.

[57] F. Bergamasco, A. Albarelli, A. Torsello, M. Favaro, and P. Zanuttigh. Pairwise similarities for scene segmentation combining color and depth data. *ICPR*, 2012.

[58] L. Bourdev and J. Malik. Body part detectors trained using 3d human pose annotations. *ICCV*, 2009.

[59] J. Deng and L. Feifei. Fine-grained crowdsourcing for fine-grained recognition. *CVPR*, 2013.

[60] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. *CVPR*, 2009.

[61] V. Ferrari and A. Zisserman. Learning visual attributes. *Advances in Neural Information Processing Systems*, 2007.

[62] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 32, 2010.

[63] A. Vedaldi, S. Mahendran, S. Tsogkas, S. Maji, B. Girshick, J. Kannala, E. Rahtu, I. Kokkinos, M. B. Blaschko, D. Weiss, B. Taskar, K. Simonyan, N. Saphra, and S. Mohamed. Understanding objects in detail with fine-gained attributes. *CVPR*, 2014.

[64] J. Davis. Hierarchical motion history images for recognizing human motion. *IEEE workshop DREV*, 2001.

[65] Md. AtiqurRahman Ahad, J. K. Tan, H. Kim, and S. Ishikawa. Motion history image: Its variants and applications. *Machine Vision and Applications*, 23:255–281, 2012.

[66] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:1026–1038, 1999.

[67] J Matas, O Chum, M Urban, and T Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761 – 767, 2004. British Machine Vision Computing 2002.

[68] Cunzhao Shi, Chunheng Wang, Baihua Xiao, Yang Zhang, and Song Gao. Scene text detection using graph model built upon maximally stable extremal regions. *Pattern Recognition Letters*, 34(2):107 – 116, 2013.

[69] Lukas Neumann and Jiri Matas. Real-time scene text localization and recognition. In *CVPR*, pages 3538–3545. IEEE Computer Society, 2012.

[70] Weilin Huang, Yu Qiao, and Xiaoou Tang. Robust scene text detection with convolution neural network induced mser trees. In *Computer Vision ECCV 2014*, volume 8692, pages 497–511. Springer International Publishing, 2014.

[71] Boris Epshtein, Eyal Ofek, and Yonatan Wexler. Detecting text in natural scenes with stroke width transform. In *CVPR*, pages 2963–2970, 2010.

[72] Luka Neumann and Jiri Matas. Scene text localization and recognition with oriented stroke detection. In *IEEE International Conference on Computer Vision (ICCV)*, 2013.

[73] Weilin Huang, Zhe Lin, Jianchao Yang, and Jue Wang. Text localization in natural images using stroke feature transform and text covariance descriptors. In *IEEE International Conference on Computer Vision (ICCV)*, 2013.

[74] Cong Yao, Xiang Bai, Baoguang Shi, and Wenyu Liu. Strokelets: A learned multi-scale representation for scene text recognition. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[75] Qixiang Ye, Qingming Huang, Wen Gao, and Debin Zhao. Fast and robust text detection in images and video frames. *Image and Vision Computing*, 23(6):565–576, 2005.

[76] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in Neural Information Processing Systems 2 (NIPS)*, 1989.

[77] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. In *Neural Computation*, volume 1, pages 541–551, 1989.

[78] B. Boser, E. Sackinger, J. Bromley, Y. LeCun, and L. D. Jackel. An analog neural network processor with programmable topology. In *IEEE Journal of Solid-State Circuits*, volume 26, pages 2017–2025, 1991.

[79] Y. LeCun. A theoretical framework for back-propagation. In *Artificial Neural Networks: concepts and theory, IEEE Computer Society Press*, 1992.

[80] H. Drucker and Y LeCun. Improving generalization performance using double back-propagation. In *IEEE Transaction on Neural Networks*, volume 3, pages 991–997, 1992.

[81] Y. LeCun and Y. Bengio. word-level training of a handwritten word recognizer based on convolutional neural networks. In *Proc. of the International Conference on Pattern Recognition*, 1994.

[82] Fu-Jie Huang and Yann LeCun. Large-scale learning with svm and convolutional nets for generic object categorization,. In *Proc. Computer Vision and Pattern Recognition Conference (CVPR)*, 2006.

[83] Utkarsh Porwal, Yingbo Zhou, and Venu Govindaraju. Handwritten arabic text recognition using deep belief networks. In *Proceedings of the 21st International Conference on Pattern Recognition, ICPR 2012, Tsukuba, Japan, November 11-15, 2012*, pages 302–305, 2012.

[84] A. Ray, S. Rajeswar, and S. Chaudhury. Text recognition using deep blstm networks. In *Advances in Pattern Recognition (ICAPR), 2015 Eighth International Conference on*, pages 1–6, 2015.

[85] Hailiang Xu and Feng Su. Robust seed localization and growing with deep convolutional features for scene text detection. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, Shanghai, China, June 23-26, 2015*, pages 387–394, 2015.

[86] A. Coates, B. Carpenter, C. Case, S. Satheesh, B. Suresh, Tao Wang, D.J. Wu, and A.Y. Ng. Text detection and character recognition in scene images with unsupervised feature learning. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, 2011.

[87] Xu-Cheng Yin, Xuwang Yin, and Kaizhu Huang. Robust text detection in natural scene images. *CoRR*, abs/1301.2628, 2013.

[88] Ming Zhao, Shutao Lia, and James Kwok. Text detection in images using sparse representation with discriminative dictionaries. *Image and Vision Computing*, 28:1590–1599, 2010.

[89] C. San Martin and S.-W. Kim. Using adaptive run length smoothing algorithm for accurate text localization in images. volume LNCS 7042, pages 149–156, 2011.

[90] Z. Zhang, W. Shen, C. Yao, and X. Bai. Symmetry-based text line detection in natural scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[91] Cong Yao, Xiang Bai, Wenyu Liu, Yi Ma, and Zhuowen Tu. Detecting texts of arbitrary orientations in natural images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[92] Lluis Gomez i Bigorda and Dimosthenis Karatzas. Multi-script text extraction from natural scenes. In *2013 12th International Conference on Document Analysis and Recognition, Washington, DC, USA, August 25-28, 2013*, pages 467–471, 2013.

[93] Sungwoong Kim, Sebastian Nowozin, Pushmeet Kohli, and Chang D. Yoo. Higher-order correlation clustering for image segmentation. In J. Shawe-Taylor, R.S. Zemel, P.L. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 1530–1538. 2011.

[94] A. Bhattacharyya. On a measure of divergence between two multinomial populations. *Sankhy: The Indian Journal of Statistics (1933-1960)*, 7(4):pp. 401–406, 1946.

[95] J. A. Hartigan and M. A Wong. A k-means clustering algorithm. *Journal of the Royal Statistical Society*, 28:100–108, 1979.

[96] Cheng Yizong. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, 1995.

[97] Wei Zhang, Deli Zhao, and Xiaogang Wang. Agglomerative clustering via maximum incremental path integral. *Pattern Recognition*, 46(11):3056–3065, 2013.

[98] A. Vedaldi and K. Lenc. Matconvnet – convolutional neural networks for matlab. 2015.

[99] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[100] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *CoRR*, abs/1405.3531, 2014.

[101] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *CoRR*, abs/1411.4038, 2014.

[102] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip H. S. Torr. Conditional random fields as recurrent neural networks. *CoRR*, abs/1502.03240, 2015.

[103] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

[104] AL Barczak, Martin J Johnson, and Chris H Messom. Revisiting moment invariants: rapid feature extraction and classification for handwritten digits. In *IVCNZ*, 2007.

[105] Reza Ebrahimzadeh and Mahdi Jampour. Efficient handwritten digit recognition based on histogram of oriented gradients and svm. *IJCA*, 104(9):10–13, 2014.

[106] Stipe Celar, Zeljko Stojkic, Zeljko Seremet, Zeljko Marusic, and Danijel Zelenika. Classification of test documents based on handwritten student id's characteristics. *Procedia Engineering*, 100:782–790, 2015.

[107] Chucai Yi, Xiaodong Yang, and Yingli Tian. Feature representations for scene text character recognition: A comparative study. In *ICDAR*, pages 907–911, 2013.

[108] Bolan Su, Shijian Lu, Shangxuan Tian, Joo Hwee Lim, and Chew Lim Tan. Character recognition in natural scenes using convolutional co-occurrence hog. In *ICPR*, pages 2926–2931, 2014.

[109] Adam Coates, Blake Carpenter, Carl Case, Sanjeev Satheesh, Bipin Suresh, Tao Wang, David J Wu, and Andrew Y Ng. Text detection and character recognition in scene images with unsupervised feature learning. In *ICDAR*, pages 440–445, 2011.

[110] Kuan Zheng, Yuanxing Zhao, Jing Gu, and Qingmao Hu. License plate detection using haar-like features and histogram of oriented gradients. In *ISIE*, pages 1502–1505, 2012.

[111] Geremy Heitz, Stephen Gould, Ashutosh Saxena, and Daphne Koller. Cascaded classification models: Combining models for holistic scene understanding. In *Advances in Neural Information Processing Systems*, 2009.

[112] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[113] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *PAMI*, 24(7):971–987, 2002.

[114] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 1, pages I–900–I–903 vol.1, 2002.

[115] Xinyi Cui, Yazhou Liu, Shiguang Shan, Xilin Chen, and Wen Gao. 3d haar-like features for pedestrian detection. In *Multimedia and Expo, 2007 IEEE International Conference on*, pages 1263–1266, 2007.

[116] RF Prates, G Cámara-Chávez, William R Schwartz, and David Menotti. Brazilian license plate detection using histogram of oriented gradients and sliding windows. *arXiv*, 2014.

[117] Hao Pan and Bailing Zhang. An integrative approach to accurate vehicle logo detection. *Journal of Electrical and Computer Engineering*, page 18, 2013.

[118] Lixia Liu, Honggang Zhang, Aiping Feng, Xinxin Wan, and Jun Guo. Simplified local binary pattern descriptor for character recognition of vehicle license plate. In *CGIV*, pages 157–161. IEEE, 2010.

[119] José A Rodrıguez and Florent Perronnin. Local gradient histogram features for word spotting in unconstrained handwritten documents. In *ICFHR*, 2008.

[120] Rafael MO Cruz, George DC Cavalcanti, and Tsang Ing Ren. Handwritten digit recognition using multiple feature extraction techniques and classifier ensemble. In *ICSSIP*, pages 215–218, 2010.

[121] Oscar Déniz, Gloria Bueno, Jesús Salido, and Fernando De la Torre. Face recognition using histograms of oriented gradients. *Pattern Recognition Letters*, 32(12):1598–1603, 2011.

[122] Massimo Bertozzi, Alberto Broggi, Mike Del Rose, Mirko Felisa, Alain Rakotomamonjy, and Frédéric Suard. A pedestrian detector using histograms of oriented gradients and a support vector machine classifier. In *ITSC*, pages 143–148, 2007.

[123] Yadong Mu, Shuicheng Yan, Yi Liu, Thomas Huang, and Bingfeng Zhou. Discriminative local binary patterns for human detection in personal album. In *CVPR*, pages 1–8, 2008.

[124] Dong-Chen He and Li Wang. Texture unit, texture spectrum, and texture analysis. *Geoscience and Remote Sensing, IEEE Transactions on*, 28(4):509–512, 1990.

[125] Mohamed Eisa, A ElGamal, R Ghoneim, and A Bahey. Local binary patterns as texture descriptors for user attitude recognition. *International Journal of Computer Science & Network Security*, 10(6):222–229, 2010.

[126] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. pages 511–518, 2001.

[127] Ross Kindermann and J. L. Snell. *Markov Random Fields and Their Applications*. AMS, 1980.

List of Figures

List of Tables

Document generated with LaTeX and:
the LaTeX style for PhD Thesis created by S. Galland — http://www.multiagent.fr/ThesisStyle
the tex-upmethodology package suite — http://www.arakhne.org/tex-upmethodology/

Abstract:

In machine vision, many applications such as object recognition, quality control, and fake detection... have been developed for manufactured objects. However, most of the applications are not able to deal with reflective objects. In this thesis, we aim to propose a recognition and identification system that is targeting the reflective manufactured objects. In the first part of the thesis, an initial prototype which consists of object model recognition and object identification is presented. In the second part of the thesis, we investigate into several important issues such as feature extraction, text detection, and character recognition in order to improve the recognition and identification performance. We propose four algorithms to tackle the problems and show how the proposed methods improve the system performance on reflective manufactured object recognition and identification.

Keywords: Reflective manufactured objects, Object model recognition, Object identification, Text detection, Character recognition

Résumé :

Dans le domaine de la vision artificielle, de nombreuses applications telles que la reconnaissance d'objet, contrôle de la qualité, et la détection de défauts ... ont été développés pour les objets manufacturés. Cependant, la plupart des applications ne sont pas en mesure de traiter des objets réfléchissants. Dans cette thèse, nous nous efforçons de proposer un système de reconnaissance et d'identification qui cible les objets manufacturés réfléchissantes. Dans la première partie de la thèse, un premier prototype qui se compose d'une phase de reconnaissance d'objets et d'une identification de l'objet est présenté. Dans la deuxième partie de la thèse, nous étudions plusieurs composantes importantes telles que l'extraction de caractéristiques, la détection de texte, et la reconnaissance de caractères afin d'améliorer la performance de reconnaissance et d'identification. Nous proposons quatre algorithmes pour résoudre les problèmes rencontrés et nous montrons comment les méthodes proposées améliorent les performances du système pour la reconnaissance de l'objet réflchissant et son identification.

Mots-clés : Object Manufacturé, Réfléchissant, Reconnaissance d'object, Idendification d'object, Détection de text, Reconnaissance de caractère

SPIM

■ École doctorale SPIM - Université de Bourgogne/UFR ST BP 47870 F - 21078 Dijon cedex
■ tél. +33 (0)3 80 39 59 10 ■ ed-spim@univ-fcomte.fr ■ www.ed-spim.univ-fcomte.fr

UNIVERSITÉ DE BOURGOGNE