PHD THESIS FROM UNIVERSITÉ PIERRE ET MARIE CURIE

Speciality

Computer Science

École Doctorale Informatique, Télécommunications et Électronique (Paris)

Presented by

Ana Ştefania CĂLĂRĂŞANU

To obtain the title of

DOCTOR FROM UNIVERSITÉ PIERRE ET MARIE CURIE

Improvement of a text detection chain and the proposition of a new evaluation protocol for text detection algorithms

Defended on December 11, 2015 in front of a jury composed of:

Jean-Marc OGIER	Reviewer
Lionel PREVOST	Reviewer
Nicole VINCENT	Examiner
Beatriz MARCOTEGUI	Examiner
Nicolas BREDECHE	Examiner
Christopher KERMORVANT	Examiner
Séverine DUBUISSON	Thesis director
Jonathan FABRIZIO	Supervisor

Ana Ştefania CĂLĂRĂŞANU
Improvement of a text detection chain and the proposition of a new evaluation protocol for text detection algorithms
École Doctorale Informatique, Télécommunications et Électronique (Paris) , December 11, 2015
Reviewers: Jean-Marc OGIER and Lionel PREVOST
Supervisors: Séverine DUBUISSON and Jonathan FABRIZIO

PHD THESIS FROM UNIVERSITÉ PIERRE ET MARIE CURIE
Computer Science

Abstract

The objective of this thesis is twofold. On one hand it targets the proposition of a more accurate evaluation protocol designed for text detection systems that solves some of the existing problems in this area. On the other hand, it focuses on the design of a text rectification procedure used for the correction of highly deformed texts.

Text detection systems have gained a significant importance during the last years. The growing number of approaches proposed in the literature requires a rigorous performance evaluation and ranking. In the context of text detection, an evaluation protocol relies on three elements: a reliable text reference, a matching set of rules deciding the relationship between the ground truth and the detections and finally a set of metrics that produce intuitive scores. The few existing evaluation protocols often lack accuracy either due to inconsistent matching procedures that provide unfair scores or due to unrepresentative metrics. Despite these issues, until today, researchers continue to use these protocols to evaluate their work. In this Ph.D thesis we propose a new evaluation protocol for text detection algorithms that tackles most of the drawbacks faced by currently used evaluation methods. This work is focused on three main contributions: firstly, we introduce a complex text reference representation that does not constrain text detectors to adopt a specific detection granularity level or annotation representation; secondly, we propose a set of matching rules capable of evaluating any type of scenario that can occur between a text reference and a detection; and finally we show how we can analyze a set of detection results, not only through a set of metrics, but also through an intuitive visual representation. We use this protocol to evaluate different text detectors and then compare the results with those provided by alternative evaluation methods.

A frequent challenge for many Text Understanding Systems is to tackle the variety of text characteristics in born-digital and natural scene images to which current Optical Character Recognition (OCR)s are not well adapted. For example, texts in perspective are frequently present in real-word images because the camera capture angle is not normal to the plane containing text regions. Despite the ability of some detectors to accurately localize such text objects, the recognition stage fails most of the time. Indeed, most OCRs are not designed to handle text strings in perspective but rather expect horizontal texts in a parallel-frontal plane to provide a correct transcription. All these aspects, together with the proposition of a very challenging dataset, motivated us to propose a rectification procedure capable of correcting highly distorted texts.

Acknowledgements

Everything you want in life you can achieve. That's what my mom always says. What I know is that wanting and believing in something gets you closer to yours goals, but what really pushes you to the finish line are the people you have around you, and I have many to thank.

I first want to express my deepest gratitude to my two supervisors Jonathan FABRIZIO and Séverine DUBUISSON for giving me the opportunity of doing this thesis and for guiding me throughout these three years. Jonathan, your analytical judgement pushed me into consistently questioning and improving my work. Séverine, your positivism, hard work and incredible reactivity has made working with you an enriching experience. You both nourished me with encouragements, helping me persevere even in the most delicate moments of this thesis.

I want to sincerely thank Jean-Marc OGIER and Lionel PREVOST for accepting to review this manuscript as well as for their valuable feedback. I would also want to thank my defense jury examiners: Nicole VINCENT, Beatriz MARCOTEGUI, Nicolas BREDECHE and Christopher KERMORVANT for participating at the evaluation of my work. I would also want to mention Isabelle BLOCH whose feedback during my thesis mid-term evaluation contributed to the improvement of this work.

I would like to express my sincere thanks to all the members of the LRDE family, notably to the lab head Olivier RICOU which made this thesis possible and to my bureau colleagues Myriam, Edwin, Yongchao and Theo for always creating a joyful atmosphere at work. I also want to mention here Etienne for his numerous tips and encouragements, and Didier for being a gentleman and always taking off his glasses first :). A very special thank you goes to Daniela, for being wonderfully helpful throughout these years.

Lastly, I want to share this achievement with my closest ones. The biggest thank you in the world goes to my super hero mom, my number one supporter, who has never ceased to believe in me. Vielen dank to the #meyersenjoinglife team, *moja sestrica* and my brother-in-law Artjom, for sharing this moment with me. Another danke goes to Ciobo and Erika for simply being there for me since ever. I want to particularly thank my wonderful friend Martina for her support, encouragements and endless chats about 7737373s, radios and how to be a boss in a world full of "mazers" and "parles plus fort". I want to thank my high school sweethearts: Olivia, Teo and Roxi for their unconditional support and friendship. Finally, all my thoughts go to Alex who has been by my side from the very first day of this three year journey, embracing me with love, support and patience. You strengthen me like nobody else does.

I hope that wherever you are, I made you proud. This one is for you dad.

Contents

List of acronyms

AUC Area Under the Curve

CC Connected Component

DetEval DetEval

DIA Document Image Analysis

EMD Earth Mover's Distance

EvaLTex Evaluating the Localization of Text

FN False Negative

FP False Positive

GT Ground Truth

ICDAR International Conference of Document Analysis and Recognition

ICR Intelligent Character Recognition

LSM Least Square Method

OCR Optical Character Recognition

PDA Portable Digital Assistant

ROC Receiver Operating Characteristic

RRC Robust Reading Competition

STUS Scene Text Understanding Systems

TN True Negative

TP True Positive

TUS Text Understanding Systems

List of notations

$Area(x)$ The area in pixels of a surface x

G_i A GT object

Gr_i A reduced GT object

Ge_i An extended GT object

D_j A detection object

N_G Number of GT objects in a dataset

N_D Number of detections in a dataset

$\mathscr{G}$ The set of GT objects defined as $\mathscr{G} = \{G_i\}_{i=1..N_G}$

$\mathscr{D}$ The set of detections defined as $\mathscr{D} = \{D_j\}_{j=1..N_D}$

s_i The split level of a GT object G_i

m_j The merge level of a detection object D_j

$(G_i \bowtie D_j)$ A *one-to-one* matching

$(G_i \bowtie D_{j_1}...D_{j_l})$ A *one-to-many* matching

$(G_{i_1}...G_{i_k} \bowtie D_j)$ A *many-to-one* matching

$(G_{i_1}...G_{i_k} \bowtie D_{j_1}...D_{j_l})$ A *many-to-many* matching

1 General introduction and contributions

> *We cannot solve our problems with the same thinking we used when we created them.*
>
> — **Albert Einstein**

This chapter's objective is to present the subject of this PhD *thesis and place it in the context of the Document Image Analysis (*DIA*) research field. We expose the diversity of topics and applications that are part of this field, with a focus on the Scene Understanding Systems as they represent the challenge of this thesis. Lastly, we highlight the main problems that guided this work and list our contributions.*

The subjects of this PhD thesis is the improvement of a text detection system and the proposition of a new evaluation protocol for text localization algorithms. The aim of the thesis is twofold. First, it consists of the proposition of a new evaluation protocol designed for text localization algorithms. Today, no accurate protocol permits a reliable evaluation of such algorithms. The few existing protocols used in the literature are not able to cope with the complexity of text detection scenarios and provide poor metrics that produce unrepresentative scores. Hence, it is difficult to evaluate individually the performances of a text localization system as well as to compare it with other systems. Secondly, we focus on the improvement of a text detection chain by rectifying the text detection results to maximize the performance of the text recognition process. When dealing with natural or born-digital images, texts can have different orientations, or be subject to different deformations. Common OCRs have difficulties in correctly recognizing such texts. This is why we propose a complex rectification method that can deal and correct different text deformations.

In this chapter we will first present the different topics and applications linked to the DIA domain to better understand the context and the importance of this work. We will then expose a variety of challenges of natural and born-digital images that contain textual information. Next, we will introduce the concept of a scene text understanding system. Finally, we will conclude this introduction by enlisting the contributions proposed in this PhD thesis.

1.1 Document Image Analysis (DIA)

The goal of DIA is to process and extract information (semantics or content) from documents by applying image analysis, computer vision, artificial intelligence and/or pattern recognition tools. Documents can for example be images of scanned papers (e.g. newspapers, books), camera captures or video

frames containing textual information (e.g. captions). Research topics in document analysis include many fields, such as document layout analysis, document structure extraction, document segmentation, document binarization, document deskewing, text detection and localization, text rectification, text extraction, character and word recognition, symbol and graphic recognition, signature verification, writer identification, handwritten text, mathematical formula identification and recognition, stroke recovery from documents, or forensic document analysis.

A variety of applications are derived from the DIA technologies:

- Auto-driving systems such as self-driven cars that need to interpret automatically signs and boards.
- Mobile mapping systems such as the GOOGLE® car that matches extracted text from streets to indexed GOOGLE® maps.
- Aid systems for visually impaired people to help them in their natural indoor and outdoor environments.
- Navigation systems that can automatically "read" maps.
- Tourist assistant systems that help tourists to face to unfamiliar environments or unknown languages.
- Automatic document indexing with applications such as large document database sorting or web search engines.
- Dematerialization such as book conversion to digital libraries for space saving.
- Signature verification.
- Automatic license plate reading to deliver speed fees or to check parking entrances.
- Gender prediction from writing.
- Optical music recognition (OMR) applications that automatically interpret music score sheets and transform them into common audio formats.
- Various PDA and smartphone related applications.

In document analysis, the extracted information can be divided into two categories: textual information (text elements) and graphics (symbols, diagrams, logos, *etc.*) [O'Gorman, 1997]. Based on the targeted applications, the textual analysis scope can be further classified into two categories. The first one involves an OCR conversion to get the textual transcription of characters and words into a digital format. A more advanced type of OCR is an ICR system, designed to handle handwritten texts. The second category is the layout analysis to identify the different structure elements of a document. It involves the segmentation of the whole document to separate text blocks from the non-textual ones and then requires their reordering for correct reading. Such techniques are mostly used for well formatted documents, usually machine printed ones (newspapers, invoices, books, *etc.*) to extract the different structural zones

(e.g. title, author, paragraph, keyword, abstract, table of contents) at different levels (word, line, text region).

A particular type of document analysis can also be performed on born-digital images, natural scene images and video frames. In such cases, the layout analysis usually targets the localization and extraction of the textual information that can later be processed by an OCR.

1.2 Challenges of daily life text

The wide availability of PDAs, digital cameras, mobile phones or robot vision systems allow the acquisition of high resolution pictures at a relatively low cost. Most of them are taken from natural environments, such as indoor places (e.g. homes, institutions, medical centers, kitchens, *etc.*) or outdoor scenes (e.g. streets, roads, *etc.*). These images are usually referred to as *real* images, or also *natural scene* images and considered as an important category of documents in the DIA field.

The texts present in natural scenes can be, among others, street signs, shop names or vehicle license plates. Searching for such "clues" can be a difficult task, not only for automated machines, but also for the human brain. Compared to traditional document images, urban scenes require more complex DIA technologies, due to the challenges imposed by the outdoor environment. Different conditions can influence the analysis of texts in natural scene images that are listed below.

Capture angle — a non parallel capture can lead to perspective deformations.

Lighting — text objects subject to shadows, brightness (specularity) or reflections can be hard to extract or recognize.

Text variety — the artistic design of many scene text objects that can contain many colors, fonts or sizes.

Text orientation — text can be inclined, vertical or even multi-oriented (in circle or curve).

Cluttered background — a non-uniform background (bricks, fences, trees, *etc.*) can lead to the over segmentation of an image and to the extraction of false text zones.

Occlusion — text objects can be partially occluded which can decrease the detection performances.

Image resolution and quality — poor resolution and quality can decrease the recognition accuracy of an OCR.

Natural scene text can then be considered as any text captured in the wild (real world) having no prior knowledge on any of the conditions mentioned above.

1.3 Scene Text Understanding Systems

STUS combine the layout analysis and the features of an OCR to recognize the textual information in real-world images. Although *Text Detection*, *Text Localization* or *Text Recognition* terms have been assigned to describe such systems, they can be misleading as they refer to specific stages of a STUS.

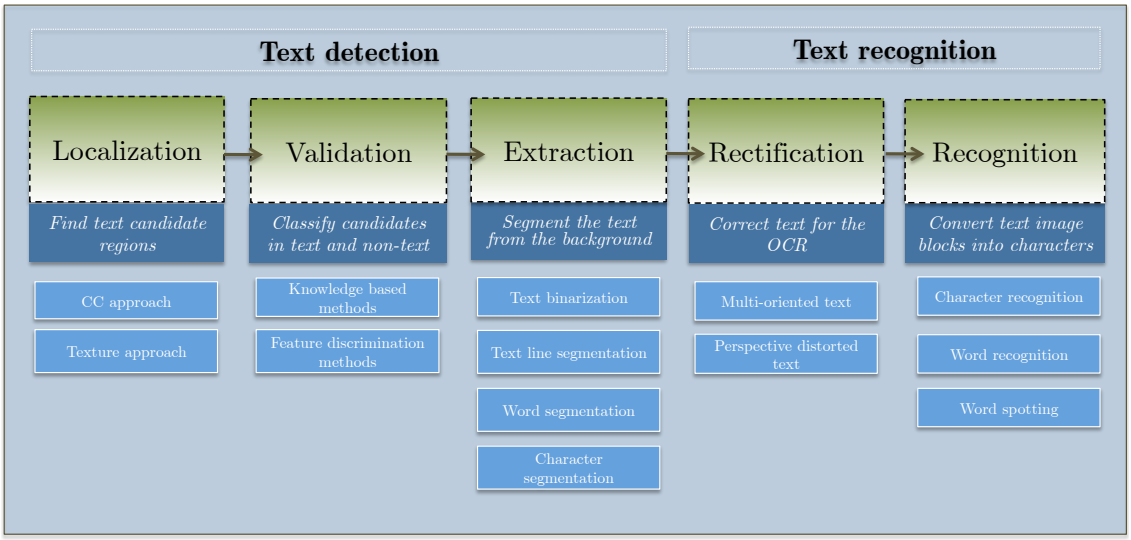


Fig. 1.1: A global framework dedicated to a Scene Text Understanding System.

A common framework for STUSs is divided into five main steps (see Figure 1.1): localization, validation, extraction, rectification and recognition. During the localization stage, text region candidates are first searched. They are then classified into text or non-text during the validation stage. The validated text is segmented from the surrounding background to get the accurate boundaries of text zones. The detection outputs can however be distorted and are then corrected during a rectification step. Finally, the recognition stage converts the extracted text regions into characters. In the next paragraphs, we give more details about these steps.

Text Localization. Text localization is the basis of any STUS as its objective is to localize the text candidate regions in pictures. There mainly exists two families of methods for this localization: connected component (CC) and texture based approaches. The connected component analysis consists in segmenting characters separately based on different characteristics, such as size or color and then in grouping them into text regions. A pre-validation stage is sometimes required for that latter. The texture based approaches use a sliding window to extract features from image blocks that are next classified into positive and negative text regions

Text Validation. During the localization step, a number of false text regions are detected. In the work described in [Ye and Doermann, 2015] the validation techniques are divided into knowledge based methods and feature discrimination methods. The knowledge based methods presume a prior knowledge on the size, color or projection profile of the text and hence the validation is done based on some predefined rules. On the other hand, the feature discrimination methods make no assumption on the

text characteristics. In such situations, different features are extracted from potential text regions and then validated using a classifier.

Text Extraction. The extraction step, often referred to as segmentation, is the stage during which accurate bounds of text zones are produced. We can consider different kinds of granularity for text extraction. For example, at a pixel level, the stage is called binarization. The extraction can also be at character, word or line level granularity. Depending on the text detection strategy, a grouping step can be necessary to gather text regions into larger ones.

Text Rectification. Current OCRs can only handle horizontal texts. However, in natural scene images, texts are often subject to perspective deformations. Many texts, due to the design format or to the capture angle, can also appear inclined or vertical. In some situations we can also face curved texts or texts in circle.

Text Recognition. The final step of a complete STUS consists in translating candidate text blocks into ASCII values. At this stage, the recognition can be done. The character recognition targets the classification of each CC separately based on different features, while word recognition also integrates a language dictionary that predicts the translation of words based on various statistical analysis. A special type of STUS applications that entirely rely on the text recognition step, *word spotting* methods, consists in matching image text blocks to words of a lexicon.

Note. We mention here that the discussed STUS in this section is not a generic framework, as its structure can differ from one case to another. For example, in some approaches, the validation stage can be included into the localization one, or the rectification step can be part of the extraction process. Moreover, some systems only focus on the detection stage and are commonly referred to as *text detection* algorithms. Conversely, systems that include both detection and recognition stages are usually referred to as *end-to-end text recognition* methods.

The evaluation of a STUS can be done at two moments (see Figure 1.2): after the detection stage to only quantify the localization performance quality or after the recognition stage and then the quality of recognition is also evaluated. Of course, the validation and rectification stages can also be evaluated separately.

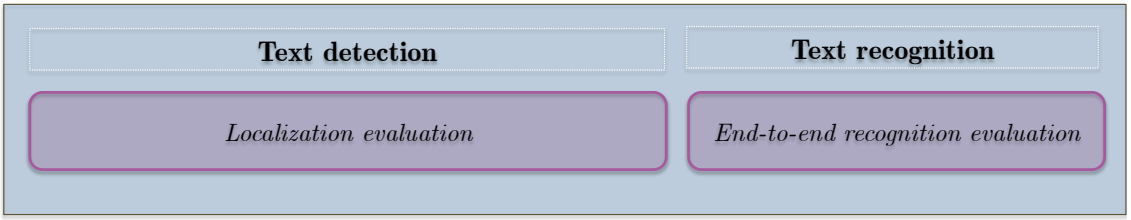


Fig. 1.2: Levels of evaluation of a text understanding system.

1.4 Contributions of the thesis

In this thesis we tackle two main problems of Text Understanding Systems. The first one, that motivated this work, refers to the unreliable manner text detection systems are nowadays being evaluated. Such an evaluation focuses on analyzing the performance of a detector to precisely provide the localization of text regions in an image. Text detectors are often severely penalized and wrongly scored despite their correct results. This happens for a number of reasons. The lack of accurate metrics and matching strategies between the results and the ground truth derive unrepresentative scores. Currently, text detectors follow the rules imposed by different evaluation protocols and adapt their results such that their methods are not penalized. Hence, in this thesis we try to provide a different view of this problem and propose an alternative evaluation approach which satisfies the challenges imposed by the diversity of text detection methods.

Another problem that Text Understanding Systems are facing is the variety of text characteristics in born-digital and natural scene images for which current OCRs are not well adapted. For example, texts in perspective are frequently present in real-word images because the camera capture angle is not normal to the plane containing text regions. Despite the ability of some detectors to accurately localize such text objects, the recognition stage fails in most of the time. Indeed, most OCRs are not designed to handle text strings in perspective but rather expect horizontal texts in a parallel-frontal plane to provide a correct transcription. All these aspects, together with the proposition of a very challenging dataset, motivated us to propose a rectification procedure capable of correcting highly distorted texts.

This manuscript is divided into two parts. The first part, which represents the core of this thesis, tackles the problem of text detection evaluation and proposes a new protocol designed to cope and solve many of the inconsistencies that current protocols are facing. The second part of this work consists in the proposition of a text rectification procedure needed for enhancing the performance of traditional OCRs.

In Chapter 2 we explain the common way text detection systems are being evaluated. We first introduce the elementary notions of an evaluation protocol: a GT annotation, a set of performance metrics and a matching strategy. We then introduce some of the most common problems that evaluation protocols are dealing with and conclude this chapter by giving a detailed state of the art.

Chapter 3 is dedicated to presenting the core of this manuscript, consisting of the proposition of an alternative evaluation protocol, EvaLTeX, designed to handle many of the unsolved issues faced by other evaluation methods. First, we discuss the contributions related to the GT annotation. Next, we explain how the matchings between the GT and a set of detections are being treated. We then discuss the choice of using a set of global performance metrics that can capture the complexity of a detection. Finally, we show that our proposed protocol can be applied to any text representation. Namely, we explain its functioning on text detections annotated with free-form masks.

The goal of Chapter 4 is to propose a visual representation of a text detector's efficiency through histograms. We show that this representation can provide additional information about the behavior of a detector that cannot be captured by a set of performance metrics. We also introduce the use of the Earth Mover's Distance as an alternative evaluation method to the one proposed in Chapter 3.

In Chapter 5 we present the experimental results obtained with the proposed evaluation methods introduced in the two previous chapters. To validate our solutions we propose a series of comparisons with other commonly protocols used in the literature. The comparisons are done at two-levels. First, we compute different performance scores on individual images. Secondly, we analyze the scores obtained on a set of images.

Chapter 6 is dedicated to a introduction to the context of text rectification procedures and describes the role of such a procedure in the global framework of a text understanding system. To do so, we illustrate the challenges due to the different deformations that texts are often subject to in both born-digital and natural scene images. Next, we list the related works done in this research area and present our contributions.

The description of the proposed rectification method is detailed in Chapter 7. The proposed approach, dedicated to text strings in perspective, relies on a well-known projective transformation that maps the coordinates of the deformed text onto the world coordinate system. We show that for an accurate rectification we need a precise approximation of the boundaries of the text. This approximation represents one of the main contributions of this chapter for which we propose a robust solution that can be used to rectify highly distorted texts. This chapter also proposes a simple and efficient method to correct some curved text strings. It consists in approximating the orientation of a character with respect to the location of its neighbors.

The experimental results that validate our proposed rectification method are shown in Chapter 8. The evaluation performance of the rectification process is done based on the results obtained on the two datasets proposed during the ICDAR 2015 *Competition on Scene Text Rectification*. In this chapter we also show the advantages and the drawbacks of our method.

Finally, Chapter 9 provides some conclusions on the works introduced in this thesis. A general discussion of all the aspects presented in this work are reviewed and possible future works are proposed.

Part I

Contribution to performance evaluation

2 How are text detection chains being evaluated

Contents

This chapter's objective is to introduce the notion of performance evaluation in the context of text detection systems. Firstly, we will describe the component elements of an evaluation protocol necessary for the comprehension of this manuscript: the ground truth annotation, its associated dataset, the performance metrics and the matching strategies. Secondly, we will discuss the limitations of commonly used evaluation methods that motivated our work. Finally, we will give a detailed overview of the evaluation frameworks used by recent text detection algorithms in the literature.

2.1 Introduction

The fast development of text detection systems in the last years has led to many approaches and consequently to a variety of evaluation protocols. As in many fields of computer vision, evaluating

text detection methods relies on a number of elements: firstly, the use of a pertinent *dataset*, built based on the specificities of a text detection task; secondly, a *text reference*, commonly known as the GT, which should be annotated as precise as possible; lastly, a solid *protocol* that estimates the accuracy of a detector by evaluating the correspondence between its output and the GT.

Evaluating text detection systems can be done in different manners. While end-to-end text detection systems imply a text recognition final stage, the text localization results should not be evaluated at the end of a system's chain, but rather separately as the detection accuracy might be distorted by the efficiency of the used OCR. Moreover, the text transcription is not always necessary, as many applications are only interested in the detection stage, to perform, for example text enhancement, license plate blurring, *etc.* The text localization outputs can be evaluated based on a segmentation result. This requires a true binarization reference of a text, that can vary depending on the stroke thickness. Here, the evaluation does not only focus on the detection but it also evaluates the binarization method. The best compromise to evaluate the localization of text seems to be the approximation of a text contour at the character, word, line or region level, depending on the targeted application.

In the following, we will introduce the elements of an evaluation protocol. We will start by pointing out in Section 2.2 the different levels and representations of a GT. We will then list, in Section 2.3, the existing datasets on which most of the text detection methods in the literature have been evaluated. Section 2.4 is dedicated to the definition of an evaluation protocol and the description of its elements: matching strategies and performance metrics. The existing evaluation protocols in the literature are listed and discussed in Section 2.5. Finally, a series of recent text detectors and with their datasets and evaluation strategy is given in Section 2.6.

2.2 Ground truth annotation

The GT is a notion designating a standard of accuracy. In text detection, it represents the text reference to which all detections will be compared. An annotation level, also called *granularity*, as well as a *text representation*, are required to label a GT text. The granularity refers to the minimum element to be labeled as text. The representation on the other hand, describes the geometric form used to annotate the text object. We hereby enlist the text annotation levels and text representations used in the literature and illustrated some of them in Figure 2.1.

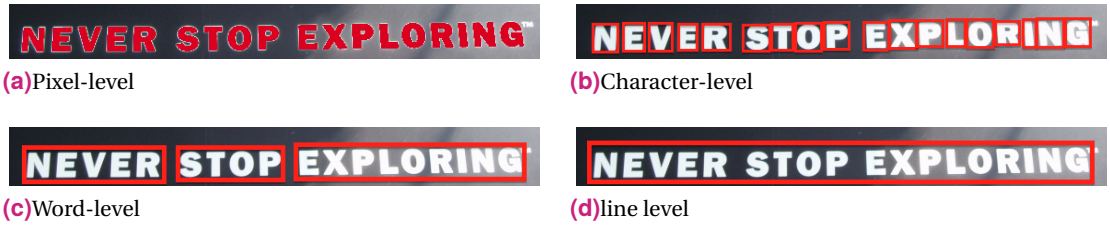
Pixel level — When using a pixel level annotation the GT text objects are usually annotated by irregular masks. This annotation is mostly used for evaluating segmentation tasks.

Character level — Characters are usually annotated by bounding boxes, circles, ellipses or oriented polygons.

Word level — Probably one of the most used granularities, the word annotation implies grouping multiple characters into bounding boxes, most of the time.

Line level — A line level annotation implies grouping multiple words together. Text lines are usually annotated by rectangular boxes, or polygons.

Region level The region annotation [Liang et al., 2001] and [Shafait et al., 2008] is usually used for document analysis to detect specific sections of a document. In such cases, the labeling is mostly done with either rectangular boxes or polygons.

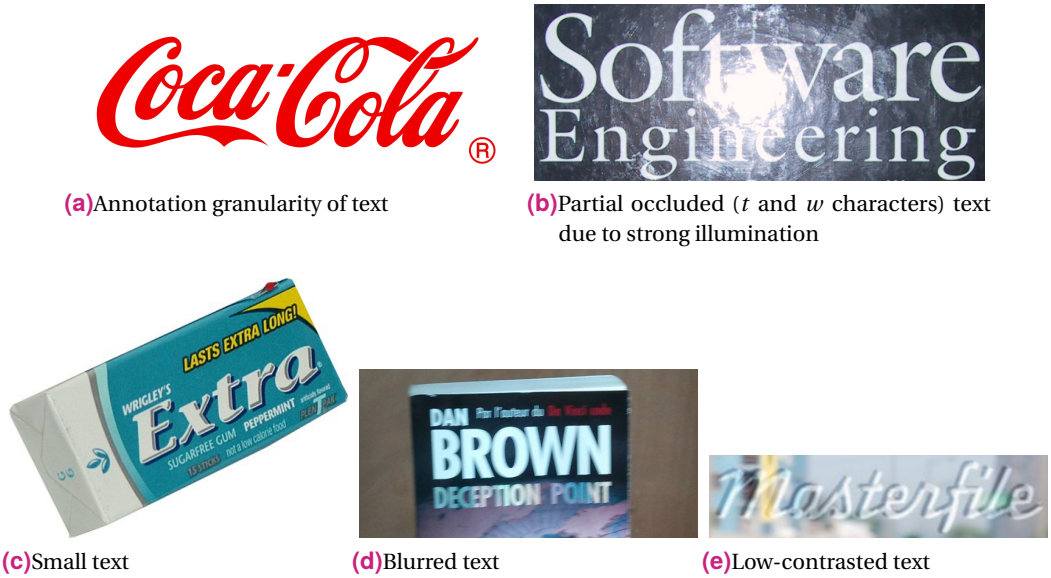


(a) Pixel-level **(b)**Character-level

(c) Word-level **(d)**line level

Fig. 2.1: Examples of text annotation levels using bounding boxes.

Annotating the GT is not an obvious task. It relies on the target text detection application or on the subjectivity of the person who manually annotates the GT. It is then sometimes difficult to choose, in certain contexts, which text objects should be annotated and which should not (see Figure 2.2). For example, what is the minimum text size we consider a detector should be able to deal with? How should the occluded text be annotated? How should the word "COCA-COLA" be annotated: as a single object or as two separate ones? What is the level of blurring or contrast allowed for a text object in order to be annotated in the GT? In order to tackle some of these problems, some evaluation protocols added text object characteristics in the ground truth metadata. However, assigning additional information to ground truth text is still a subjective task. For example, based on their vision strength, two different annotators can evaluate differently the visibility level of a text. Besides the subjectivity related issues,



(a) Annotation granularity of text **(b)**Partial occluded (t and w characters) text due to strong illumination

(c) Small text **(d)**Blurred text **(e)**Low-contrasted text

Fig. 2.2: Cases of text annotation ambiguities.

another annotation problem has been in the center of attention: the consistency between the granularity level and the GT text representation. For example, a horizontal box can not correctly fit a tilted or curved text: the surrounding box will also contain a large amount of non-text areas, and the annotation will not be precise enough. Despite the increasing interest in multi-oriented text detection systems, a large number of datasets still propose a GT annotation using rectangular bounding boxes and only few of them use a more flexible representation of text, as it will be shown in Section 2.3.

Note: Horizontal or inclined bounding boxes have the advantage of simplicity as they require only four coordinates. On the other hand, irregular polygons rely on the subjectivity of the annotator as different point configurations can be used to label the same text. Clear and simple rules need to be defined to annotate text.

2.3 Datasets

The increasing development in the text detection and recognition field (see [Ye and Doermann, 2015] for a complete survey), has pushed the research community to propose numerous datasets for a variety of tasks and applications. We list hereby, in the chronological order of their publication, a number of datasets used for both detection and recognition purposes and summarize their characteristics. Some of them are illustrated in Figure 2.3. A summary of these methods is also given in Table 2.1.

HUA'S DATASET

The dataset proposed in [Hua et al., 2001][1] consists of 45 video clips for a total of 6,750 frames and 158 text boxes, belonging to Spanish TV RTVE and to the Ministry of Education of Singapore. Three clips do not contain any textual information. The GT annotation is done using the *Ground Truth Generator* framework through which one can manually assign attributes (*Text String*, *Height Variance*, *Skew Angle*, *Color/Texture*, *String Density*, *Recognizability Index*) to each text object. The dataset consists of horizontal graphic and natural scene texts in English, Spanish and Chinese languages. The dataset is proposed with an evaluation protocol discussed in Section 2.5.6.

RRC'03
RRC'05

The RRC'03[2] [Lucas et al., 2003] and RRC'05 [Lucas, 2005] datasets have been designed for the *Robust Reading Competitions* during ICDAR 2003 and ICDAR 2005 and, until present, are still widely used. They contain 509 samples of scene text images for a total of 2,276 GT objects. The datasets are divided into two subsets: a training subset containing 258 images (and 1,100 GT text boxes) and a testing subset with 251 images (and 1,156 text boxes). The datasets mainly contain horizontal English words. The GT annotation is done at character and word levels.

CHARS74K

The CHARS74K[3] dataset [de Campos et al., 2009] is a character recognition database containing English and Kannada symbols used for training purposes. As its name suggests, this dataset contains 74k (74,107) images, each one with one character (0-9, a-z, A-Z) from natural scene or synthetic images.

SIGN EVALUATION DATA

In [Weinman et al., 2009], the authors propose a dataset[4] containing signs captured in a downtown area. The dataset consists of 95 text regions for a total

[1] http://www.cs.cityu.edu.hk/~liuwy/PE_VTDetect/
[2] http://algoval.essex.ac.uk/icdar/Datasets.html
[3] http://www.ee.surrey.ac.uk/CVSSP/demos/chars74k/
[4] http://www.cs.grinnell.edu/~weinman/

of 215 English words and 1,209 characters. It proposes a character level GT annotation using bounding boxes.

101 VIDEO IMAGES

In [Phan et al., 2009], the authors proposed a video dataset containing images with horizontal text lines with various font colors and backgrounds. The video frames taken from daily news programs, sports videos and movie clips contain both graphic and scene texts in different languages (English, Chinese and Korean), and the image sizes range from 320 × 240 to 816 × 448 pixels.

EPSHTEIN'S DATASET

The dataset proposed in [Epshtein et al., 2010][5] is another database that focuses on text in street view scenes and contains 307 color images of sizes ranging from 1024 × 768 to 1024 × 1360 pixels. The database is considered harder to deal with than other common natural scene datasets due to the cluttered backgrounds (repeating pattern objects and vegetation).

OXFORD CORNMARKET SCENE TEXT

The OXFORD CORNMARKET SCENE TEXT[6] dataset [Posner et al., 2010] contains using images of a busy street scene. The GT is labeled at word level using bounding boxes. However, due to the complex environment, some text areas, considered as difficult to detect, were not annotated. All images are resized to a fixed size of 640 x 480 pixels.

KAIST

The KAIST[7] dataset [Lee et al., 2010], designed for segmentation, localization and recognition tasks, contains 3000 samples of indoor and outdoor scene images, all resized to a fixed size of 640 × 480 pixels. The images are taken under various lighting conditions (night, day, shadow). The dataset contains English and Korean text objects, annotated using bounding boxes at both character and word levels.

SVT

The SVT[8] (*Street View Text*) dataset [Wang and Belongie, 2010] is dedicated to text string in the wild benchmarks. Its data comes from GOOGLE Street View engine and are used for both text detection and recognition purposes, making the database useful for end-to-end systems. The majority of the natural scene texts are frontal and captured at a middle distance [Ye and Doermann, 2014]. The image samples were chosen such that the skew of text objects is minimized [Wang et al., 2011]. The GT annotation is exclusively done at word level. It contains 350 images (100 training images for a total of 257 GT bounding boxes and 250 testing images for a total of 647 GT text objects). The dataset is composed of multi-oriented and horizontal English text. However, as stated

[5] http://research.microsoft.com/enus/um/people/eyalofek/text_detection_database.zip
[6] http://www.robots.ox.ac.uk/~posnerhi/TextSpotting/pmwiki.php/Results/IROS10
[7] http://www.iapr-tc11.org/mediawiki/index.php/KAIST_Scene_Text_Database
[8] http://tc11.cvc.uab.es/datasets/SVT_1

in [Yao et al., 2014], this database provides incomplete word annotation.

NEOCR — The NEOCR[9] dataset contains 659 samples of real world images with 5,238 annotated GT bounding boxes. This dataset presents the particularity to be multilingual (eight different languages). The GT is labeled both with rectangular boxes for horizontal texts and quadrilaterals for oriented text zones.

OSTD — The OSTD[10] (*Oriented Scene Text Dataset*) dataset [Yi and Tian, 2011b] focuses on multi-oriented natural scene texts (indoor views, logos, street scenes), and contains 89 samples for a total of 218 GT objects.

MULTILINGUAL DATASET — The dataset proposed in [Pan et al., 2011a] targets the performance evaluation of detectors of English and Chinese texts. It consists of 248 training images and 239 test images captured from natural scenes.

SIGNS-N800 — The SIGNS-N800[11] dataset [Bouman et al., 2011] contains 241 images (81 samples of training images and a testing subset of 160 images) of flyers, road signs and posters acquired by a VGA camera. Two GT annotations are available: firstly, each character within a sign region is manually segmented; secondly, each sign region is separately manually segmented.

RRC'11 — The *Robust Reading Competition*[12] dataset used during ICDAR′11 contains every sample of ICDAR′03 and ICDAR′05 databases, except for a couple of images. It consists of two subsets: RRC'11-BD contains 552 born-digital images (420 training samples for a total of 3,583 GT text objects and 102 testing samples for a total of 918 GT objects); and RRC'11-SI, contains 484 natural scene images (229 training samples for a total of 848 GT text boxes and 255 testing samples for a total of 1,189 GT objects). The dataset is composed of mainly English texts captured at a short distances [Ye and Doermann, 2014]. The GT annotation, which is done at word level, was revised due to some annotation inconsistencies in ICDAR′03 and ICDAR′05 datasets. The main challenges of this database consists in detecting texts of various sizes and in various illumination conditions.

SVHN — The SVHN[13] (*Street View House Numbers*) dataset [Netzer et al., 2011] was designed for recognition tasks and contains 10 classes of digits (1 for each digit).

[9] http://www.iapr-tc11.org/mediawiki/index.php/NEOCR:_Natural_Environment_OCR_Dataset
[10] http://media-lab.engr.ccny.cuny.edu/cyi/project_scenetextdetection.html
[11] https://engineering.purdue.edu/~ace/kbsigns/
[12] http://robustreading.opendfki.de/
[13] http://www.iapr-tc11.org/mediawiki/index.php/The_Street_View_House_Numbers_(SVHN)_Dataset

There are 73,257 digits in the training set and 26,032 digits in the testing set, and an additional of 531,131 less difficult digit samples.

SHIVAKUMARA DATASET — The authors in [Shivakumara et al., 2012] proposed an independent dataset for video text detection purposes. This dataset contains 220 samples of non-horizontal text images (176 scene text images and 44 graphics text images) and 800 samples of horizontal text images (160 Chinese text, 155 scene text and 485 English text images).

IIIT5K Word — The images in IIIT5K Word[14] dataset [Mishra et al., 2012] are collected from the GOOGLE® image search engine, based on queries such as *billboards, signboard, house numbers, house name plates or movie posters.* The dataset contains 5,000 images (cropped words) for a total of 5,000 GT boxes (2,000 training GT objects and 3,000 testing GT objects). The dataset contains distorted English text strings.

MSRA-TD500 — The MSRA-TD500[15] [Yao, 2012] dataset contains 500 natural scene images (300 training images for a total of 1,068 GT text boxes and 200 testing images for a total of 651 GT objects) and is used for very complex scene text detection tasks. The images are taken from both indoor (e.g. signs, doorplates, caution plates) and outdoor (e.g. guide boards and billboards) environments. The dataset contains multi-oriented English and Chinese texts over complex backgrounds. The GT annotation is done at line level rather than word level due to the difficulty of partitioning Chinese text lines into individual words.

MSRA-TD500 WORD — MSRA-TD500 WORD[16] dataset [Phan et al., 2013] was proposed as an extension of MSRA-TD500 database, which provides only line level GT annotations. MSRA-TD500 WORD preserves the images from MSRA-TD500 dataset but proposes a word-level labeling of English texts.

SVT-PERSPECTIVE — The StreetViewText-Perspective[17] dataset [Phan et al., 2013] was designed to fulfill the need of evaluating perspective text recognition systems. It is based on the original SVT database which was proposed in [Wang and Belongie, 2010]. The images were taken at the same places as in the SVT dataset, but only side-view angles were chosen to capture the scenes. For each image in the dataset, the words present in the lexicon were manually annotated using quadrilaterals.

[14] http://cvit.iiit.ac.in/projects/SceneTextUnderstanding/IIIT5K.html
[15] http://pages.ucsd.edu/~ztu/Download_front.htm
[16] https://www.comp.nus.edu.sg/~phanquyt/
[17] http://www.comp.nus.edu.sg/~phanquyt/

RRC'13

ICDAR'13 dataset[18] [Karatzas et al., 2013] is known as one of the most common datasets in the literature. It is used for localization, segmentation and recognition tasks and contains of two subsets. The first subset, RRC'13-BD corresponds to *Challenge 1* and contains 551 samples of born-digital images (410 training images with 3,564 GT objects, and 141 testing images for a total of 1,439 GT text boxes). The image size ranges from 194×30 to 660×476 pixels. The second subset, RRC'13-SI corresponds to *Challenge 2* and consists of 462 samples of natural scene images: 229 are training images (containing 848 GT text regions) and 233 testing images (containing 1,095 GT objects). Both datasets are annotated at word level and mainly contain preponderantly horizontal English words.

SPORTS-10K, TV SERIES-1M

SPORTS-10K AND TV SERIES-1M[19] are two large video datasets (TV SERIES-1M contains more than 1 million images, SPORTS-10K contains 10,000 images) [Mishra et al., 2013] designed for text retrieval tasks. SPORTS-10K dataset contains frames taken from sport video clips with advertisement signboards. The GT is based on manually annotating the queries contained in each frame.

IIIT STR

The IIIT STR[20] (*Scene Text Retrieval*) dataset [Mishra et al., 2013] is composed of 10,000 images collected from the GOOGLE® and FLICKR image search engines. Images containing texts were collected using GOOGLE$_{circledR}$ engine based on 50 query words such as *department, police, Microsoft building or motel*. Images with no text were extracted from FLICKR based on queries such as *sky or building*. This dataset is dedicated to benchmark text retrieval systems (word spotting). Consequently, the GT consists in manually annotating whether there is a query word or not in each image of the dataset.

MASTER

The MASTER[21] (*Multi-script And Scene Text Reading*) dataset, introduced in [Kumar et al., 2013], was designed for text localization, segmentation and recognition tasks and contains both training and testing data. The localization task is done on 167 training and 167 testing images, annotated using bounding boxes at word level. For recognition purposes, the dataset is divided into two subsets: a subset for English word recognition task with 67 camera-captured scene images, containing 495 training GT text regions and 645 GT testing objects; a second subset, for Kannada word recognition task, containing 300 training images and 243 samples of test images.

CUTE80

The CUTE80[22] (*Curved Text 80*) dataset [Risnumawan et al., 2014], consists of 80 indoor and outdoor images with curved text lines. The GT annotation

[18] http://rrc.cvc.uab.es/
[19] http://cvit.iiit.ac.in/projects/STR/videoSTR.html
[20] http://cvit.iiit.ac.in/projects/STR/IIITSTR.html
[21] http://mile.ee.iisc.ernet.in/mrrc/
[22] http://web.fsktm.um.edu.my/~cschan/downloads_CUTE80_dataset.html

is done using a set of points for each text line. The dataset is characterized by complex backgrounds, low resolution and perspective distortions. It also includes an evaluation protocol (see Section 2.4).

YOUTUBE VIDEO TEXT (YVT)

The YOUTUBE VIDEO TEXT dataset, introduced in [Nguyen et al., 2014], is a collection of YouTube text images: overlay texts, such as captions, song titles, logos and scene texts (street and business signs). The GT annotation is done with bounding boxes, using the VATIC framework [Vondrick et al., 2013]. The dataset contains 30 videos, each one at 30 frames per second.

HUST-TR400

HUST-TR400[23] dataset, proposed by [Yao et al., 2014], contains 400 natural scene images with English letters and Arabic numbers of different colors, fonts, orientations and sizes and was designed for end-to-end scene text recognition systems. The images were taken from three different sources (images captured by volunteers in different cities of the U.S.A., from FLICKR and from MSRA-TD500 datasets). This database is designed to evaluate end-to-end systems. The GT annotation is done at word level.

BBC NEWS FOOTAGE

This dataset[24] of 2.3 million frames from BBC News footage is used to test the robustness of the text detector proposed in [Jaderberg et al., 2014c], and generally for text spotting tasks. It contains images related to queries such as *Hollywood, Boris Johnson, Vision, Police, Oxford, United*. However, no associated GT metadata are provided.

SOUTH INDIAN LANGUAGES DATASET

SOUTH INDIAN LANGUAGES dataset [Pavithra and Aradhya, 2014] is a collection of 114 multilingual (Kannada, Tamil, Telugu and Malayalam) text images (text book and novel covers, magazines, posters) with varying complex backgrounds, different font colors and sizes.

MJSYNTH

The MJSYNTH[25] Synthetic Word dataset [Jaderberg et al., 2014a], [Jaderberg et al., 2014b] contains 9,000,000 images of 90,000 synthetically generated English words. This dataset is used for text recognition purposes.

FUJITSU

The FUJITSU dataset [Wang et al., 2014] is a multilingual benchmark that contains 208 scene text images captured with a smart phone and a digital camera. The text objects are horizontal, vertical, inclined and of different languages. The GT annotation level is not specified, and the text is labeled using inclined

[23] http://mc.eistar.net/
[24] http://www.robots.ox.ac.uk/~vgg/research/text/
[25] http://www.robots.ox.ac.uk/~vgg/data/text/

bounding boxes.

RRC'15-IST

ICDAR′15 dataset[26] [Karatzas et al., 2015] is the dataset proposed during the most recent ICDAR RRC. The novelty with respect to the previous dataset RRC'13 consists of a new image set designed for incidental scene text detection, recognition and end-to-end tasks. It contains 1670 images (with 17,548 text regions) among which 1500 were made publicly available (1000 training images and 500 images for testing) while the remaining 170 images are private. The GT annotation is done at word level using quadrilaterals. Some words in the dataset were annotated using a "do not care" tag, namely texts in non-Latin scripts, non-readable or one and two-character words. The evaluation on this dataset is made using the Pascal evaluation protocol described in Section 2.5.12.

TRW'15

The TRW'15[27] dataset was proposed for the ICDAR 2015 Text Reading in the Wild competition [Zhou et al., 2015]. The dataset is focused on multilingual (English and Chinese) text detection and recognition in complex natural scenes. It contains around 1000 natural scene images, taken from the Internet or by volunteers divided into: a testing subset of 484 images and a training subset of 500 images. The annotation is done at line level using polygons. Text regions have been divided into four categories: "translucent English", "translucent other", "non-translucent English" and "non-translucent other". The translucent text regions encode website links, describe shop names or contact information. The dataset also contains "do not care" text regions. The evaluation on this database is done using the ICDAR′03 protocol.

[26] http://rrc.cvc.uab.es/
[27] http://icdar2015.imageplusplus.com/

Fig. 2.3: Text image samples from different datasets. From top to bottom: KAIST, III5K, MSRA-I, MSRA-TD500, OSTD, SVHN, SVT and CHARS74K datasets.

Tab. 2.1: Summary of existing datasets used for text segmentation (S), localization (*L*), recognition (*R*), end-to-end (*EE*) and spotting (*SP*) tasks ordered chronologically.

Dataset	Year	Task	Nb. images	GT	Characteristics	URL
HUA'S DATASET	2001	*L*	6,750	-	Horizontal graphic and natural scene texts	http://www.cs.cityu.edu.hk/~liuwy/PE_VTDetect/
RRC'03/RRC'05	2003/2005	*L*	509	Character and word level annotation using bounding boxes	horizontal English texts	http://algoval.essex.ac.uk/icdar/Datasets.html
CHARS74K	2009	*R*	74k	-	Natural scene and synthetic characters	http://www.ee.surrey.ac.uk/CVSSP/demos/chars74k/
SIGN EVALUATION DATA	2009	*L*	-	Character level annotation using bounding boxes	Images containing signs	-
101 VIDEO IMAGES]	2009	*L*	-	-	Graphic and scene texts in English and Chinese from videos	-
EPSHTEIN'S DATASET	2010	*L*	307	-	Cluttered backgrounds	http://research.microsoft.com/enus/um/people/eyalofek/text_detection_database.zip
OXFORD CORNMARKET SCENE TEXT	2010	*L*	-	Word-level annotation using bounding boxes	Complex environments	http://www.robots.ox.ac.uk/~posnerhi/TextSpotting/pmwiki.php/Results/IROS10
KAIST	2010	*S*, *L*, *R*	3000	Character an word level annotation using bounding boxes	-	http://www.iapr-tc11.org/mediawiki/index.php/KAIST_Scene_Text_Database
SVT	2010	*L*, *R*	600	Word level annotation	Horizontal and multi-oriented natural scene texts	http://tc11.cvc.uab.es/datasets/SVT_1
NEOCR	2011	*L*	659	Bounding boxes and quadrilaterals	Multilingual texts	http://www.iapr-tc11.org/mediawiki/index.php/NEOCR:_Natural_Environment_OCR_Dataset

Tab. 2.1: Summary of existing datasets used for text segmentation (S), localization (L), recognition (R), end-to-end (EE) and spotting (SP) tasks ordered chronologically.

Dataset	Year	Task	Nb. images	GT	Characteristics	URL
OSTD	2011	L?	89	Irregular polygons	Natural scene texts	http://media-lab.engr.ccny.cuny.edu/cyi/project_scenetextdetection.html
MULTILINGUAL DATASET	2011	L	487		Natural scene texts in English and Chinese	-
SIGNS-N800	2011	L	241	Character and region level annotation	Images containing signs	https://engineering.purdue.edu/~ace/kbsigns/
RRC'11-BD	2011	S, L, R	552	Word level annotation using bounding boxes	Graphic texts	http://robustreading.opendfki.de/
RRC'11-SI	2011	S, L, R	484	Word level annotation using bounding boxes	Natural scene texts	http://robustreading.opendfki.de/
SVHN	2011	R	~ 53k	-	10 classes of digits	http://www.iapr-tc11.org/mediawiki/index.php/The_Street_View_House_Numbers_(SVHN)_Dataset
SHIVAKUMARA DATASET	2012	L	1020	-	Horizontal and non-horizontal texts	-
IIIT5K Word	2012	SP	5k	-	English distorted texts from GOOGLE® engine	http://cvit.iiit.ac.in/projects/SceneTextUnderstanding/IIIT5K.html
MSRA-TD500	2013	L	500	Line level	Multi-oriented texts on complex backgrounds	http://pages.ucsd.edu/~ztu/Download_front.htm
MSRA-TD500 WORD	2013	L	500	Word level	Multi-oriented texts on complex backgrounds	https://www.comp.nus.edu.sg/~phanquyt/
SVT-PERSPECTIVE	2013	R	600	Word level annotation using quadrilaterals	Perspective texts	http://www.comp.nus.edu.sg/~phanquyt/
RRC'13-BD	2013	S, L, R	551	Word level annotation using bounding boxes	Graphic texts	http://rrc.cvc.uab.es/?ch=1&com=introduction

Tab. 2.1: Summary of existing datasets used for text segmentation (S), localization (*L*), recognition (*R*), end-to-end (*EE*) and spotting (*SP*) tasks ordered chronologically.

Dataset	Year	Task	Nb. images	GT	Characteristics	URL
RRC'13-SI	2013	*S, L, R*	462	Word level annotation using bounding boxes	Natural scene texts	http://rrc.cvc.uab.es/?ch=2&com=introduction
SPORTS-10K/ TV SERIES-1M	2013	*SP*	10k/1 million		Sports video clips	http://cvit.iiit.ac.in/projects/STR/videoSTR.html
IIIT STR	2013	*SP*	10k	-	Images from GOOGLE® and FLICKR	http://cvit.iiit.ac.in/projects/STR/IIITSTR.html
MASTER	2013	*L, S, R*	334 (TL) ,310 TR	Word level annotation using bounding boxes	English and Kannada texts	http://mile.ee.iisc.ernet.in/mrrc/
CUTE80	2014	*R*	80	Polygon points	Curved text lines	http://web.fsktm.um.edu.my/~cschan/downloads_CUTE80_dataset.html
YOUTUBE VIDEO TEXT (YVT)	2014	*L*	900 frames	Bounding boxes	Captions, song titles, logos and scene texts	-
HUST-TR400	2014	*R*	400	Word-level	Images from FLICKR and MSRA-TD500 dataset	http://mc.eistar.net/
BBC NEWS FOOTAGE	2014	*SP*	2.3 million	-	Daily news video frames	http://www.robots.ox.ac.uk/~vgg/research/text/
SOUTH INDIAN LANGUAGES	2014	*L*	114	-	Multilingual texts	-
MJSYNTH	2014	*R*	9 million	-	Synthetical English words	http://www.robots.ox.ac.uk/~vgg/data/text/
FUJITSU	2014	*L*	208	Inclined bounding boxes	Multilingual and multi-oriented texts	-
RRC'15-IST	2015	*L,R,EE*	1670	Word-level annotation using quadrilaterals	Incidental and perspective distorted texts in natural scenes	http://rrc.cvc.uab.es/
TRW'15	2015	*L,R*	~ 1000	Line-level annotation using polygons	Multi-lingual and natural scene texts	http://icdar2015.imageplusplus.com/

2.4 What is an evaluation protocol?

An evaluation protocol is a system that determines the relationships between a set of references (or ground truth) and a set of detection outputs. In this section we introduce the elementary components of an evaluation framework needed for the better comprehension of the manuscript. First, we enlist the different metrics (*confusion matrix, receiver operation characteristic, area under curve and Euclidean distance comparison*) that underlay the current evaluation approaches used in text detection. Next, we define the different matching scenarios between the GT and the detection results.

2.4.1 Metrics

Confusion matrix. Nowadays, most of the common metrics used in the object detection area and particularly for text detection performance evaluation are derived from the confusion matrix. The confusion matrix, also known as the error matrix [Stehman, 1997] or as the contingency table, is a tool for evaluating the performance of a classification system. It quantifies the number of correct and incorrect detections made by a classifier with respect to the GT associated to a dataset. Table 2.2 shows a 2×2 confusion matrix for a two class classification case with the following entries:

TP: the number of correct predictions that a detection is an actual GT text object;

FN: the number of incorrect predictions that a detection is not a GT text object;

FP: the number of incorrect predictions that a detection is a GT text object;

TN: the number of correct predictions that a detection is not a GT text object.

Tab. 2.2: A two-class confusion matrix

DETECTIONS / GROUND TRUTH	TEXT	NON-TEXT
TEXT	TRUE POSITIVE (TP)	FALSE NEGATIVE (FN)
NON-TEXT	FALSE POSITIVE (FP)	TRUE NEGATIVE (TN)

In the following we will enumerate the different performance measurements that can be directly derived from the confusion matrix.

Precision/Positive Predictive Value/Confidence is the proportion of the predicted positive cases that were correct:

$$P = \frac{TP}{TP + FP} \tag{2.1}$$

Negative Predictive Value is the inverse of Precision and computes the proportion of negative predictions that are really negative.

$$NPR = \frac{TN}{TN + FN} \tag{2.2}$$

Recall/True Positive Rate/Hit Rate/Sensitivity is the proportion of positive cases that were correctly identified and is defined as:

$$R = \frac{TP}{TP + FN} \tag{2.3}$$

True negative rate/Specificity is the inverse of Recall and quantifies the proportion of negative cases that were correctly classified.

$$SPC = \frac{TN}{TN + FP} \tag{2.4}$$

False positive rate/Fallout is the proportion of negative cases that were incorrectly classified as positive:

$$FPR = \frac{FP}{FP + TN} \tag{2.5}$$

False negative rate/Miss Rate is the proportion of positive cases that were incorrectly classified as negatives:

$$FNR = \frac{FN}{FN + TP} \tag{2.6}$$

False Discovery Rate is the proportion of false positives among all positive predictions.

$$FDR = \frac{FP}{FP + TP} \tag{2.7}$$

Accuracy is the total number of correct predictions :

$$AC = \frac{TP + TN}{TP + TN + FN + FP} \tag{2.8}$$

F-Score [Rijsbergen, 1979] is defined as the harmonic mean of Recall and Precision:

$$F1 = 2 \cdot \frac{R \cdot P}{R + P}$$

The $F-$Score, also known as $F1 - Score$, is a particular case of the F_β metric that favors Precision if $\beta > 1$ and Recall if $\beta < 0$ and is given by:

$$F_\beta = (1 + \beta^2) \cdot \frac{P \cdot R}{(\beta^2 \cdot P) + R} \tag{2.9}$$

The F-Score uniformly balances the importances of Precision and Recall.

G-measure [David, 2011] is the the geometric mean of Recall and Precision:

$$G = \sqrt{P \cdot R} \tag{2.10}$$

Matthews correlation coefficient [Matthews, 1975] is the correlation coefficient between the GT and detection binary classifications that can take values between −1 (inconsistency between the GT and detections) and 1 (perfect prediction):

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{2.11}$$

ROC ROC graphs are an alternative tool used for the comparison of classification models. The ROC plot represents the FP rate on the X-axis and the TP rate on the Y-axis. The classification model can depend on a parameter that gives more or less importance to TP compared to FP. Each (FP,TP) configuration leads to a different ROC curve. If the classifier does not use any parameter, the ROC plot is represented by a single point which corresponds to a (FP,TP) pair. An example of such a curve is given in Figure 2.4.

An ideal classifier, that correctly detects all the texts, should be represented by a curve that climbs fast toward the $(0,1)$ point (top left corner of the plot). Then, the false positive rate is 0 while the true positive is 1. A classifier that outcomes all detections to be positive is represented by the $(1,1)$ point. Similarly, the point $(0,0)$ depicts a model whose detections are all negative. Finally, the classifier for which all detections are incorrect is represented by the point $(1,0)$. One of the advantages of the ROC plot is that, as stated in [Swets, 1988], it summarizes all the data in the confusion matrix. Moreover, it provides a visualization of two main characteristics: the classifier's capacity of correctly detecting and the proportion of negative texts that are incorrectly detected.

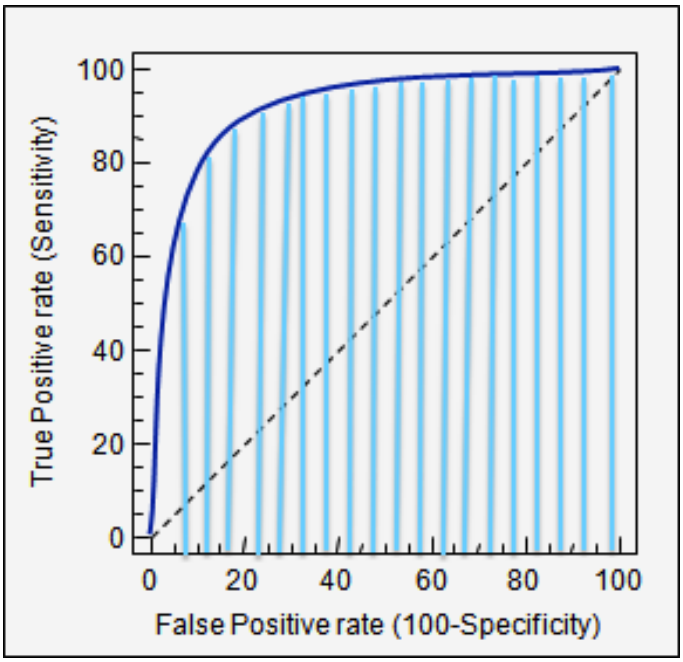


Fig. 2.4: AUC measure corresponding to the surface under the ROC curve[28] (dark blue) depicted with vertical blue lines.

Area under the curve (AUC). The two-dimensional representation given by the ROC plots provides a straightforward visualization of a classification output. The accuracy of the classification model

[28] Credit: http://i.stack.imgur.com/5x3Xj.png

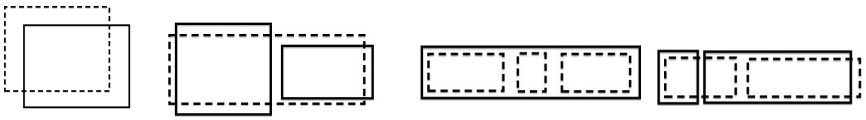


Fig. 2.5: Matching cases (GT is represented by dashed rectangles and detections by plain line rectangles).

depicted with a ROC plot, can further be computed using the area under the ROC curve [Swets, 1988]. This measure is called Area Under the Curve (ROC). An illustration is given in Figure 2.4.

Euclidean distance comparison. A different approach used for calculating the performance of a classifier from a ROC curve is to use the Euclidian distance. Based on the Euclidean distance (d_E) between the "ideal" point (0,1) and a given point (FP,TP), an accuracy performance measure is derived as:

$$d_E = \sqrt{W \cdot (1 - TP)^2 + (1 - W) \cdot FP^2} \tag{2.12}$$

W is a weight parameter that assigns the importance given to FP, respectively TP. The values for d_E range from 0 (perfect classification model) to $\sqrt{2}$ for a model that has incorrectly detected all texts.

Despite the measurement diversity offered by the confusion matrix, the most used metrics for text detection evaluation remain the Recall, Precision and $F-$Score. In text detection, the Recall is the proportion of correctly detected texts with respect to the total number of GT texts, while the Precision represents the proportion of correctly detected texts with respect to the total number of detections. Over-estimating the number of detections decreases the Precision, while under-estimating this number decreases the Recall.

2.4.2 Matching strategy

Besides the performance measurements, an evaluation protocol relies on a *matching strategy*, that defines the relationship between a set of GT objects and a set of detections. Four types of matchings are considered as illustrated in Figure 2.5:

(a) *one-to-one*: one detection matches exactly one text object;

(b) *one-to-many*: multiple detections match one text object;

(c) *many-to-one*: one detection matches multiple text objects;

(d) *many-to-many*: conditions (b) and (c) are simultaneously satisfied.

Two more scenarios can also appear: a *false positive* represents a detection that has no correspondence in the GT; a *missed detection* denotes a text object that has no correspondence in the detection set.

In order to describe the matching type between the text objects and the detections, one needs to rely on a local evaluation, done at object level. Unlike the metrics discussed in the previous section, that are

computed globally, the local evaluation consists in computing, for each pair of text object and detection, a matching value, commonly known as the *overlap area ratio*.

Quality matching evaluation. One of the first local measurements introduced for text localization evaluation is the Jaccard index [Jaccard, 1901]. It measures the similarity between two sets A and B and is defined as the ratio between the intersection and the union of these two sets: $J(A,B) = \frac{A \bigcap B}{A \bigcup B}$. Given a GT object G and detection D, the Jaccard index can be seen as the intersection area between the GT text object and the detection divided by their union area:

$$J(G,D) = \frac{Area_G \bigcap Area_D}{Area_G \bigcup Area_D} \tag{2.13}$$

The Jaccard index can take values in the unitary interval: a perfect matching will get the value 1, while a mismatch will be evaluated to 0. Two common overlap area ratio coefficients, that are derived from the Jaccard index, have also been used to locally evaluate the quality of a matching. The *coverage* coefficient measures the proportion of matched surface with respect to the GT object area, defined as:

$$Cov = \frac{Area_G \bigcap Area_D}{Area_G} \tag{2.14}$$

The *accuracy* coefficient on the other hand, measures how precise is the matching area with respect to the detection surface:

$$Acc = \frac{Area_G \bigcap Area_D}{Area_D} \tag{2.15}$$

We can divide the *one-to-one* mapping into four categories (see Figure 2.6):

perfect detection: the detection perfectly matches the GT object;

partial detection: the coverage area between the detection and the GT object is smaller than the area of both objects individually;

over-detection: the detection area is larger than the GT object's one and covers it entirely;

under-detection: the detection area is smaller than the GT object's one and is entirely included within.

Note. One can observe that when dealing with an over-detection, the Jaccard index becomes the accuracy rate because the detection includes the GT object and hence their union is the detection surface itself. Correspondingly, evaluating an under-detection using the Jaccard index is equivalent to use the coverage measurement because here, the GT object encloses the detection box and hence their union is equal to the GT surface itself.

2.5 Evaluation protocols in the literature

This section is dedicated to the existing evaluation protocols in the literature. In a first instance, we review the main approaches used by the evaluation methods. Then, we individually analyze the protocols by presenting their matching process as well as their advantages and disadvantages.

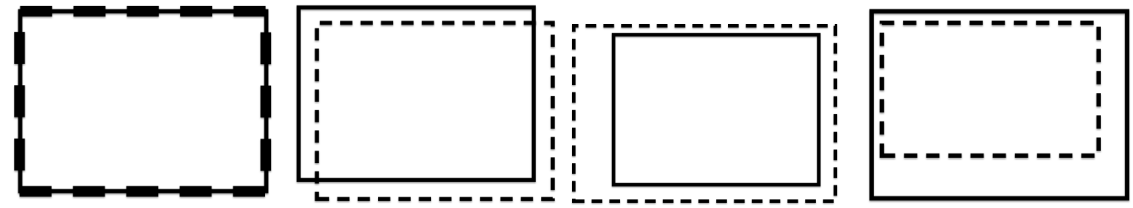


Fig. 2.6: *One-to-one* detection types; GT are represented by plain rectangles and detections by dashed rectangles.

Minimum area constraint. Most of the current algorithms consider a detection as valid (respectively a GT text object as matched) if the local measurements (overlap area ratios) satisfy a minimum overlap [Clavelli et al., 2010], [Karatzas et al., 2013], [Mariano et al., 2002], [Lucas et al., 2003], [Lucas, 2005], [Nascimento and Marques, 2006], [Wolf and Jolion, 2006], [Shahab et al., 2011]. Such an approach validates all matchings for which the local measurement is higher than a predefined threshold and rejects all others. This is however a problem, because most of the time, the detection is evaluated in a binary manner with scores equal to 1 or 0, depending if the minimum overlap constraint is satisfied or not. Hence, if we compare two localization systems, one that partially detects a text (without satisfying the overlap constraint) and one that entirely misses the text both will get the same score, which makes their comparison unfair, as seen in Figure 2.7. In other cases, if the minimum area constraint is not satisfied the detections can even be counted as FPs, decreasing the Precision rate. The overlapping area ratio constraint misclassifies many GT text boxes during the matching protocol which results in low scores, even when the detected boxes substantially overlap the GT ones. Also, the scattering scenarios are poorly treated. For example, if a GT text box is matched with multiple detections, only the detections that satisfy the area constraint will be considered, while the other ones will be rejected.

Text not detected; ICDAR′13 Inkam method

Partial detection (red rectangle); ICDAR′13 Text_detector_CASIA method

Fig. 2.7: An example of irrelevant score. Both methods get Recall and Precision scores equal to 0 during the ICDAR 2013 RRC evaluation protocol because none of them satisfied the constraint.

Best match approach. Beside the minimum area constraint, some evaluation protocols imply a best match approach which consists in assigning only one GT object to a detection, regardless of the real number of matched GT objects. In many cases, when the GT annotation is done at word level, a text detector that produces line level results can be frequently penalized as evaluation protocols cannot deal (or deal too coarsely) with granularity differences.

Score normalization approaches. Usually the final scores, such as the Recall or Precision, are a result of a normalization of the sum of local measurements. Sometimes, this normalization is done with respect to the number of images in a given dataset. While this seems to be the natural way of doing this, it can severely distort the scores. For example, an image containing a single GT text will weight more in the computation of final scores than an image containing one detected GT object and one undetected GT object. The normalization could also be performed with respect to the total surface of all GT and detection objects. In such case, small objects would contribute less to the final scores than the larger ones. In text detection, smaller objects are not necessarily easier to detect than the larger ones. Other approaches normalize the local measurements with respect to the total number of GT objects and detections. Then, all GT objects, respectively all detections, are treated equally, regardless of their surface. This last approach remains the best compromise as all objects equally contribute to the performance of a detector.

In the following sections we detail the existing evaluation protocols, in an alphabetical order, by providing their matching strategy and performance measurements. Also, for each protocol, we summarize their advantages and drawbacks. Fifteen evaluation frameworks are being analyzed: Anthimopoulos's protocol (Section 2.5.1), Clavelli's protocol (Section 2.5.2), CLEAR metrics (Section 2.5.3), CUTE80 (Section 2.5.4), DETEVAL (Section 2.5.5), Hua's protocol (Sectio 2.5.6), ICDAR'03 (Section2.5.7), Ma's protocol (Section 2.5.8), Mariano's protocol (Section 2.5.9), MSRA-TD500 (Section 2.5.10), Nascimento's protocol (Section 2.5.11), PASCAL metrics (Section 2.5.12), Shivakumara's protocol (Section 2.5.13), VACE metrics (Section 2.5.14), Yi's protocol (Section 2.5.15) and ZoneMap (Section 2.5.16).

From now on, let us consider the set of GT objects $\mathscr{G}$ defined as $\mathscr{G} = \{G_i\}_{i=1..N_G}$ and the set of detections $\mathscr{D}$ defined as $\mathscr{D} = \{D_j\}_{j=1..N_D}$, where G_i represents a GT object and D_j its corresponding detection. N_G denotes the number of GT objects in $\mathscr{G}$, and N_D the number of detections in $\mathscr{D}$. $Area(x)$ will be used to denote the area (in pixels) of an object (GT text or detection) x.

2.5.1 Anthimopoulos's evaluation protocol

In [Anthimopoulos et al., 2010] an evaluation method was proposed based on the estimation of the number of characters n_c in a text line computed as $n_c = \frac{r_t}{r_c+r_s}$, where r_c and r_s are two constants, representing the character and space ratios. The number of characters in a text line r_t is here considered as proportional to the ratio width w_t to height h_t of that text line. Based on this assumption, the contribution of each box to the overall evaluation is defined as $r_t = w_t/h_t$. The overall performance is then computed based on the Recall and Precision of the area coverage, normalized by the approximation of the number of characters for every text line. This gives the following redefinition of Recall and Precision:

$$Recall_{ecn} = \frac{\sum_{i=1}^{N_G} \frac{GDI_i}{hg_i^2}}{\sum_{i=1}^{N_G} \frac{Area(G_i)}{hg_i^2}}, \qquad Precision_{ecn} = \frac{\sum_{j=1}^{N_D} \frac{DGI_j}{hd_j^2}}{\sum_{j=1}^{N_D} \frac{Area(D_j)}{hd_j^2}} \tag{2.16}$$

where hg_i is the height of GT object G_i and hd_j the height of the corresponding detection box D_j. GDI_i and DGI_j are the corresponding intersections computed such that minor inconsistencies between the GT and the detection sets are not penalized:

$$GDI_i = \begin{cases} Area(G_i) & \text{if } \frac{Area(G_i \cap (\bigcup_{j=1}^{N_D} D_j))}{Area(G_i)} \geq th \\ Area(G_i \cap (\bigcup_{j=1}^{N_D} D_j)) & \text{if } \frac{Area(G_i \cap (\bigcup_{j=1}^{N_D} D_j))}{Area(G_i)} < th \end{cases} \tag{2.17}$$

$$DGI_i = \begin{cases} Area(D_j) & \text{if } \frac{Area(D_j \cap (\bigcup_{i=1}^{N_G} G_i))}{Area(D_j)} \geq th \\ Area(D_j \cap (\bigcup_{i=1}^{N_G} G_i)) & \text{if } \frac{Area(D_j \cap (\bigcup_{i=1}^{N_G} G_i))}{Area(D_j)} < th \end{cases} \tag{2.18}$$

Finally, the F-Score metric is computed to get a global measurement.

Anthimopoulos's evaluation protocol: **advantages**(↗) and **disadvantages** (↙)

- ↗ relaxation of localization errors;
- ↗ non-binary local evaluation;
- ↙ not very accurate since based on approximations;
- ⚠ no details on how the different types of matchings are handled.

2.5.2 Clavelli's evaluation protocol

Clavelli *et al.* [Clavelli et al., 2010] proposed a multi-level annotation scheme which consists in representing text objects at pixel (text part), atom (e.g character), word and line levels. This framework can evaluate text segmentation tasks, when text objects are represented at pixel and part levels, as well as localization applications when text is represented at character level.

The matching protocol is based on two thresholds: T_{min} and T_{max}, used to validate the matchings between a GT and a detection represented by a set of connected components (CCs). The default values are set to: $T_{\text{min}} = 0.9$, $T_{\text{max}} = \min(5, 0.5 \cdot T)$, where T corresponds to the thickness of the text part. Based on this, the detection CCs are classified into several categories, presented in Table 2.3.

Tab. 2.3: The detection types handled in [Clavelli et al., 2010].

Detection type	Matching	Minimal coverage	Maximal coverage
background	false positive		
fraction	*one-to-one*	not satisfied	satisfied
whole	*one-to-one*	satisfied	satisfied
multiple	*many-to-one*	satisfied for all text parts	satisfied collectively for the combination of the covered text parts
fraction & multiple	*many-to-one*	not satisfied at least for one text part	satisfied collectively for the combination of the covered text parts
mixed	if any other case occurs		

The matching at word and line levels is done with respect to the ability of a detector to group character blocks. Recall, Precision and F-Score are computed with respect to the number of correctly extracted atoms with respect to the two coverage thresholds.

Clavelli's evaluation protocol: **advantages**(↗) and **disadvantages** (↙)

- ↗ separation of matching types;
- ↗ accurate evaluation due to the minimum text granularity level;
- ↙ binary local evaluation due to the use of thresholds $T_{\min}$ and $T_{\max}$;
- ↙ can not handle word and line level texts represented with a bounding box annotation;
- ↙ *one-to-many* scenarios are not handled;
- ↙ requires a character level detection and a grouping stage for word and line detections.

2.5.3 CLEAR metrics

The CLEAR metrics have been proposed by the authors in [Kasturi et al., 2009]. The accuracy of a detector is calculated based on the number of detection failures m_t and false positives, FP. Then, for each frame t, a *Multiple Object Detection Accuracy* ($MODA$) measure is computed in the following manner:

$$MODA(t) = 1 - \frac{c_m \cdot m_t + c_f \cdot FP}{N_G^{(t)}}, \tag{2.19}$$

where c_m and c_f are the cost functions corresponding to the missed detections and false positives and $N_G^{(t)}$ is the number of GT objects in the t^{th} frame. c_m and c_f are scalar weights that can be set depending on the application. For a set of frames, the accuracy is computed using a Normalized $MODA$ (N_MODA) metric:

$$N\_MODA = 1 - \frac{\sum_{t=1}^{N_{frames}} (c_m \cdot m_t + c_f \cdot FP)}{\sum_{t=1}^{N_{frames}} N_G^{(t)}} \tag{2.20}$$

The Jaccard index is used to compute the $MappedOverlapRatio$ between the GT and the detection results:

$$MappedOverlapRatio = \sum_{i=1}^{N_{mapped}^{(t)}} \frac{Area(G_i^{(t)} \cap D_i^{(t)})}{Area(G_i^{(t)} \cup D_i^{(t)})} \tag{2.21}$$

where $G_i^{(t)}$ is the i^{th} GT object in the t^{th} frame, $D_i^{(t)}$ is the detection object corresponding to $G_i^{(t)}$, and $N_{mapped}^{(t)}$ denotes the number of matched object pairs in frame t. The *Multiple Object Detection Precision* ($MODP$) for each frame t is computed as:

$$MODP(t) = \frac{(MappedOverlapRatio)}{N_{mapped}^{(t)}} \tag{2.22}$$

Similar to $N-MODA$, the Normalized $MODP$ is given by:

$$N-MODP = \frac{\sum_{t=1}^{N_{frames}} MODP(t)}{N_{frames}} \tag{2.23}$$

CLEAR metrics **advantages**(↗) and **disadvantages** (↙)

- ↗ provides both quantity ($MODA$) and quality measurements ($MODP$);

↗ used to evaluate other object detection applications;

↙ *one-to-many*, *many-to-one* and *many-to-many* matchings not treated;

↙ the normalization of $N-MODP$ is done with respect to the number of frames in the database.

2.5.4 CUTE80 evaluation protocol

The evaluation protocol described in [Risnumawan et al., 2014] and associated to CUTE80 dataset handles curved text lines represented by a set of polygon points. The matching strategy consists in establishing the minimum intersection area, a_i, between the GT G_i and the polygon area of a curved text line detection D_j, defined as:

$$a_i = \frac{Area(D_j)}{Area(D_j \bigcup G_i) - Area(D_j \bigcap G_i)} \tag{2.24}$$

Global scores are computed using the well-known Precision, Recall and F-Score metrics:

$$Recall = \frac{\sum_i a_i}{N_G} \tag{2.25}$$

$$Precision = \frac{\sum_i a_i}{N_D} \tag{2.26}$$

CUTE80 evaluation protocol: **advantages**(↗) and **disadvantages** (↙)

↗ handles curved text;

↗ non-binary local evaluation;

↙ only one local measurement is used for computing the recall and precision;

⚠ no details on the different matching scenarios;.

2.5.5 DetEval evaluation framework

DetEval is an evaluation protocol [Wolf and Jolion, 2006] used during ICDAR 2011 and ICDAR 2013 RRC (*Challenge 1* and *Challenge 2*). The local evaluation is done based on the area recall $A_r = \frac{Area(G_i \bigcap D_j)}{Area(G_i)}$ and area precision $A_p = \frac{Area(G_i \bigcap D_j)}{Area(D_j)}$ that need to satisfy the following conditions:

$$A_r \geq t_r \tag{2.27}$$

$$A_p \geq t_p \tag{2.28}$$

$$\sum_{i}^{N_G} A_r \geq t_r \tag{2.29}$$

$$\sum_{j}^{N_D} A_p \geq t_p \tag{2.30}$$

where t_r and $t_p \in [0,1]$ are the area recall and precision constraints. The matching between a GT object and a detection is then decided based on the following constraints:

$$Match_G(G_i, \mathscr{D}, t_r, t_p) = \begin{cases} 1 & \text{if } G_i \text{ matches exactly one detection box} \\ & \text{(Equations 2.27 and 2.28 satisfied);} \\ 0 & \text{if } G_i \text{ is not matched by any detection box;} \\ & \text{(Equations 2.27 and 2.28 not satisfied);} \\ f_{sc}(k) & \text{if } G_i \text{ is matched by } k \text{ detection boxes} \\ & \text{(Equations 2.28 and 2.29 satisfied)} \end{cases} \tag{2.31}$$

$$Match_D(D_j, \mathscr{G}, t_r, t_p) = \begin{cases} 1 & \text{if } D_j \text{ matches exactly one GT box;} \\ & \text{(Equations 2.27 and 2.28 satisfied);} \\ 0 & \text{if } D_j \text{ does not match any GT box;} \\ & \text{(Equations 2.27 and 2.28 not satisfied);} \\ f_{sc}(k) & \text{if } D_j \text{ matches } k \text{ GT boxes} \\ & \text{(Equations 2.27 and 2.30 satisfied);} \end{cases} \tag{2.32}$$

$f_{sc}(k)$ represents a fragmentation level applied if a GT object is matched multiple times. The Recall and Precision are then computed as following:

$$R_{OB}(\bar{G}, \bar{D}, t_r, t_p) = \frac{\sum_k \sum_i Match_G(G_i^k, D^k, t_r, t_p)}{\sum_k |\, G^k \,|} \tag{2.33}$$

$$P_{OB}(\bar{G}, \bar{D}, t_r, t_p) = \frac{\sum_k \sum_j Match_D(D_j^k, G^k, t_r, t_p)}{\sum_k |\, D^k \,|}, \tag{2.34}$$

The DETEVAL tool also proposes an alternative set of metrics, based on the AUC and a visual representation thought ROC plots, with the objective to capture the complexity of the results given by a detection algorithm. It consists in characterizing both the quality and the quantity natures of a detection set. Compared to the default version of DETEVAL, the only difference is the way of computing the global recall and precision values, while the local object matching rules remain the same. The recall and precision values are computed over a range of 20 different area threshold values used to obtain the AUC graph and then averaged to give two overall metrics, R_{OV} and P_{OV} in the following manner:

$$R_{OV} = \frac{1}{2T} \sum_{i=1}^{T} R_{OB}(\bar{G}, \bar{D}, i/T, t_p) + \frac{1}{2T} \sum_{i=1}^{T} R_{OB}(\bar{G}, \bar{D}, t_r, i/T) \tag{2.35}$$

$$P_{OV} = \frac{1}{2T} \sum_{i=1}^{T} P_{OB}(\bar{G}, \bar{D}, i/T, t_p) + \frac{1}{2T} \sum_{i=1}^{T} P_{OB}(\bar{G}, \bar{D}, t_r, i/T) \tag{2.36}$$

DETEVAL framework **advantages**(↗) and **disadvantages** (↙)

↗ quantity/quality characterization of detections;

- ↗ visual tool for representing the detector's performance.
- ↙ *many-to-one* matches are often dismissed due to Equation 2.30;
- ↙ *one-to-many* matches are often dismissed due to Equation 2.29;
- ↙ partial matchings are discarded when the thresholds in Equations 2.27 and 2.28 are not satisfied;
- ↙ binary local evaluation for *one-to-one* matchings.

2.5.6 Hua's evaluation protocol

The evaluation protocol introduced in [Hua et al., 2001] assigns a detection difficulty level to each GT object of a dataset: *Initial Level, Textbox Height, Textbox Width, Character Height Variance, Skew Angle, Color and Texture, Background Complexity, String Density* and *Contrast*. Based on a *Detection Difficulty* value denoted as L_{DD} and a *Recognition Importance* level RI, the authors give a *Detection Importance* rate DI to each GT object G_i and computed as:

$$DI(G_i) = L_{DD}(G_i) \cdot RI(G_i) \tag{2.37}$$

Then, for each GT-detection pair (G_i, D_j), with c representing their overlap area and $E(x)$ denoting the number of Sobel edge points of a text box x, the authors define a *DD-independent text box detection quality* Q_{DD} as:

$$Q_{DD}(G_i) = Q_0(c)^{1/\sqrt{L_{DD}(G_i)}}, \text{ where } Q_0(c) = 1 - \frac{E(D_j - c)}{E(D_j)} \tag{2.38}$$

Two detection qualities, *Basic quality* ($Q_b(G_i)$) and *Fragmentation Quality* ($Q_{fr}(f)$), are then used to compute the total quality rate $Q(G_i)$:

$$Q(G_i) = Q_b(G_i) Q_{fr}(G_i), \tag{2.39}$$

with

$$Q_b(G_i) = \frac{\sum_{D_k \in \mathscr{D}_{G_i}} (Q_{DD}(D_k \cap G_i) E(D_k \cap G_i))}{\max(E(G_i), \sum_{D_k \in \mathscr{D}_{G_i}} E(D_k \cap G_i))} \tag{2.40}$$

$$Q_{fr}(G_i) = \frac{\sqrt{\sum_{D_k \in \mathscr{D}_{G_i}} E(D_k \cap G_i)^2}}{\sum_{D_k \in \mathscr{D}_{G_i}} E(D_k \cap G_i)}, \tag{2.41}$$

where $D_k \in \mathscr{D}_{G_i}$ corresponds to the set of detection objects that matched the GT box G_i. Finally, the overall detection rate DR is:

$$DR = \frac{\sum_{i \in N_G} Q(G_i) DI(G_i)}{\sum_{i \in N_G} DI(G_i)} \tag{2.42}$$

HUA'S evaluation protocol **advantages**(↗) and **disadvantages** (↙)

- ↗ takes into account the detection difficulty of text objects;
- ↗ non-binary local evaluation;
- ↙ matching strategies not clearly exposed;

- ↙ *many-to-one* matchings not treated;
- ↙ final score is highly dependent on the subjectivity of the annotators;
- ↙ too many subjective parameters.

2.5.7 ICDAR'03 evaluation protocol

The ICDAR'03 protocol [Lucas et al., 2003] was used to evaluate the text localization performance during the RRC. This evaluation framework is based on the best match approach $m(r,R)$ which assigns for each rectangle r in a set of rectangles R the maximum matching area m_p:

$$m(r,R) = \max\{m_p(r,r')|r' \in R\}$$

Here, the matching area m_p corresponds to the Jaccard index, which computes the ration of the intersection and union of two object surfaces. The Recall $R_{ICDAR'03}$ and Precision $P_{ICDAR'03}$ are then computed over the set of GT objects $\mathscr{G}$ and detections $\mathscr{D}$ in the following manner:

$$R_{ICDAR'03} = \frac{\sum_{i=1}^{N_G} m(G_i,\mathscr{D})}{N_G}$$

$$P_{ICDAR'03} = \frac{\sum_{j=1}^{N_D} m(D_j,\mathscr{G})}{N_D}$$

The final ranking of the participants is given by the classic $F-$Score. In practice, the match score m_p is taken into consideration as long as its value is greater than 0.5.

ICDAR'03 protocol **advantages**(↗) and **disadvantages** (↙)

- ↗ non-binary local evaluation;
- ↙ highly penalizes algorithms detecting text lines (*many-to-one* matchings) due to the best match approach;
- ↙ partial matchings allowed only if the detection box does not exceed the boundaries of a GT object;
- ↙ not dealing with *one-to-many* matchings due to the best match approach.

2.5.8 Ma's evaluation protocol

In [Ma et al., 2007] a word-level evaluation is proposed, where GT texts are clustered with respect to a proximity criterion. Two matrices, RM and PM, are defined to establish the local performance between a GT object G_i and a detection D_j:

$$RM(i,j) = \frac{Area(G_i \cap D_j)}{Area(G_i)}, \qquad PM(i,j) = \frac{Area(G_i \cap D_j)}{Area(E_j)} \tag{2.43}$$

A binary matching matrix M is then computed based on a minimal coverage constraint:

$$M(i,j) = \begin{cases} 1 & \text{if } RM(i,j) \geq th \\ 0 & \text{else} \end{cases} \tag{2.44}$$

where th is a threshold to fix. For each GT object detected several times (*one-to-many* match) only the maximum overlap area is considered:

$$RG(i) = \max_j(RM(i,j)) \tag{2.45}$$

The Recall and Precision are then redefined as:

$$Area\_recall = \frac{\sum_{i=1}^{N_G} \max_j(RM(i,j))}{N_G}, \qquad Area\_precision = \frac{\sum_{j=1}^{N_D} PE(j)}{N_D}, \tag{2.46}$$

where $PE(j)$ is the sum of $PM(i,j)$ corresponding to the largest cluster that can be formed by the covered GT objects. The protocol also proposes an overall metric measuring the false positive rate defined as:

$$Area_{false} = \frac{\sum_{D_k \in \mathscr{D}_{FP}} Area(D_k)}{\sum_{j=1}^{N_D} Area(D_j)}, \tag{2.47}$$

where $\mathscr{D}_{FP}$ represents the set of FPs in $\mathscr{D}$.

Ma's evaluation protocol: **advantages**(↗) and **disadvantages** (↙)

- ↗ *many-to-one* matchings are handled;
- ↙ *one-to-many* matching poorly treated;
- ↙ *many-to-one* matchings are penalized if covered GT objects do not belong to the same cluster;
- ↙ binary local evaluation.

2.5.9 Mariano's evaluation protocol

In [Mariano et al., 2002], authors proposed three sets of evaluation metrics for video sequences: three pixel-count based metrics, two area-unconstrained object based metrics and two area-constrained object metrics. The first set of metrics (pixel-count based metrics) are *Area-Based Recall for Frame*, *Area-Based Precision for Frame* and *Average Fragmentation* computed according to the following equations:

$$OverallRec = \frac{\sum_{t=1}^{N_{frames}} Area(U_{G^{(t)}}) \times Rec(t)}{\sum_{t=1}^{N_{frames}} Area(U_{G^{(t)}})}, \qquad OverallPrec = \frac{\sum_{t=1}^{N_{frames}} Area(U_{D^{(t)}}) \times Prec(t)}{\sum_{t=1}^{N_{frames}} Area(U_{D^{(t)}})} \tag{2.48}$$

$$Frag(G_i^{(t)}) = \frac{1}{1 + log_{10}(N_{D^{(t)} \cap G_i^{(t)}})} \tag{2.49}$$

where $U_{G^{(t)}}$ and $U_{D^{(t)}}$ represent the spatial union of the text objects in the GT frame $G^{(t)}$, respectively the union of text boxes in the detection frame $D^{(t)}$. N_{frames} represents the number of frames in the GT, $Rec(t) = \frac{Area(U_{D^{(t)}} \cap \overline{U_{G^{(t)}}})}{Area(U_{G^{(t)}})}$ and $Prec(t) = 1 - \frac{Area(U_{D^{(t)}} \cap \overline{U_{G^{(t)}}})}{Area(U_{D^{(t)}})}$. The number of output boxes in the frame $D^{(t)}$ is represented by $N_{D^{(t)} \cap G_i^{(t)}}$.

The second set of metrics is composed of the Recall and Precision measurements on three levels: object, frame and set, obtained in the following manner:

$$ObjectRecall(G_i^{(t)}) = \frac{Area(G_i^{(t)} \bigcap U_{D_j^{(t)}})}{Area(G_i^{(t)})}, \quad BoxPrecision(D_j^{(t)}) = \frac{Area(D_j^{(t)} \bigcap U_{G_i^{(t)}})}{Area(D_j^{(t)})} \tag{2.50}$$

$$Recall(t) = \frac{\sum_{i=1}^{N_{G^{(t)}}} ObjectRecall(G_i^{(t)})}{N_{G^{(t)}}}, \quad Precision(t) = \frac{\sum_{j=1}^{N_{D^{(t)}}} BoxPrecision(G_i^{(t)})}{N_{D^{(t)}}} \tag{2.51}$$

$$OverallRecall = \frac{\sum_{t=1}^{N_{frames}} N_{G^{(t)}} \times Recall(t)}{\sum_{t=1}^{N_{frames}} N_{G^{(t)}}}, \quad OverallPrecision = \frac{\sum_{t=1}^{N_{frames}} N_{D^{(t)}} \times Precision(t)}{\sum_{t=1}^{N_{frames}} N_{D^{(t)}}} \tag{2.52}$$

The third set of metrics computes the Recall and Precision based on a binary matching strategy, where GT objects and detections are validated if their overlap area satisfies a threshold T. Hence, Recall and Precision are defined as:

$$Overall\_Loc\_Obj\_Recall = \frac{\sum_{f=1}^{N_{frames}} Loc\_Obj\_Recall(t)}{\sum_{f=1}^{N_f} N_{G^{(t)}}} \tag{2.53}$$

$$Overall\_Output\_Box\_Prec = \frac{\sum_{f=1}^{N_{frames}} Loc\_Box\_Count(t)}{\sum_{f=1}^{N_f} N_{D^{(t)}}}, \tag{2.54}$$

where $Loc\_Obj\_Recall(t)$ and $Loc\_Box\_Count(t)$ are computed as:

$$Loc\_Obj\_Recall(t) = \sum_{i=1}^{N_{G^{(t)}}} ObjDetect(G_i^{(t)}), \tag{2.55}$$

$$Loc\_Box\_Count(t) = \sum_{j=1}^{N_{D^{(t)}}}, BoxPrec(D_j^{(t)}) \tag{2.56}$$

The binary local scores $BoxPrec(D_j^{(t)})$ and $ObjDetect(G_i^{(t)})$ are computed based on the minimum area coverage approach (threshold th) in the following manner:

$$BoxPrec(D_j^{(t)}) = \begin{cases} 1 & \text{if } \frac{Area(D_j^{(t)} \bigcap U_{G^{(t)}})}{Area(D_j^{(t)})} > th \\ 0 & \text{otherwise} \end{cases} \tag{2.57}$$

$$ObjDetect(G_i^{(t)}) = \begin{cases} 1 & \text{if } \frac{Area(G_i^{(t)} \bigcap U_{D^{(t)}})}{Area(G_i^{(t)})} > th \\ 0 & \text{otherwise} \end{cases} \tag{2.58}$$

Mariano's evaluation protocol: **advantages**(↗) and **disadvantages** (↙)

- ↗ complex evaluation protocol;
- ↗ global Recall and Precision averaged with respect to the total number of text objects;

↙ binary evaluation for the third set of metrics;

⚠ does not mention how different matchings are handled.

2.5.10 MSRA-TD500 evaluation protocol

The MSRA-TD500 [Yao, 2012] evaluation protocol is associated to the dataset with the same name. The framework can manage oriented bounding boxes. The matching strategy is based on the overlap ratio between a GT rectangle G_i and a detection rectangle D_j. In order to compute their intersection, the two bounding boxes are axis-aligned by rotating G_i and D_j around their centers, from angles θ_1 and θ_2 respectively, and the result is then denoted by $G_i^{\theta_1}$ and $D_j^{\theta_2}$. The overlap ratio between G_i and D_j is then the Jaccard index:

$$J(G_i, D_j) = \frac{Area(G_i^{\theta_1} \bigcap D_j^{\theta_2})}{Area(G_i^{\theta_1} \bigcup D_j^{\theta_2})} \tag{2.59}$$

The detections are divided into true or false positives according to the overlap between the minimum detection area rectangle and the GT rectangles. The protocol considers a detection as correct if the angle of θ_1 and θ_2 are less than $\pi/8$ and the overlap ratio is larger than 0.5. If multiple detections match the same text line, they are considered as false positives. Overall scores are then computed using the well known Precision, Recall and F-Score metrics.

MSRA-TD500 evaluation protocol **advantages**(↗) and **disadvantages** (↙)

↗ can evaluate detections represented with oriented bounding boxes;

↗ can handle *many-to-one* cases due to the text line annotation of the associated dataset;

↙ can not handle *one-to-many* scenarios;

↙ there is no distinction between partial and total detections;

↙ binary local evaluation for *one-to-one* matchings.

2.5.11 Nascimento's evaluation protocol

Authors in [Nascimento and Marques, 2006] proposed an evaluation protocol for object detection algorithms in video surveillance tasks. It evaluates separately the percentage of different types of matchings: *correct detection*, *false alarm*, *detection failure*, *merge region*, *split region* and *split-merge region*. The different matching scenarios depend on the overlap area between the GT and the detections that should satisfy an area constraint. Finally, the evaluation framework produces six scores representing the percentage of each of these matchings.

Nascimento's evaluation protocol: **advantages**(↗) and **disadvantages** (↙)

↗ separation of matching types;

↗ adapted to video text detection;

- ↙ no global measurement is proposed;
- ↙ binary local evaluation.

2.5.12 PASCAL metrics

The PASCAL metrics proposed for the PASCAL Visual Object Classes (VOC) Challenge [Everingham et al., 2015] consider a detection D_j, matched to a GT object G_i, correct if the corresponding overlap area between the two objects divided by their union area, denoted by a_o (and equal to the Jaccard index), exceeds the value 0.5:

$$a_o = \frac{Area(G_i \cap D_j)}{Area(G_i \cup D_j)} \tag{2.60}$$

The global performance of a detector is given by the average precision metric (AP), computed from the average precision over a set of eleven recall levels $[0, 0.1, ..., 1]$ [Everingham et al., 2015]:

$$AP = \frac{1}{11} \sum_{r \in \{0, 0.1, ..., 1\}} p_{interp}(r), \tag{2.61}$$

where $p_{interp}(r)$ is the maximum precision for which the corresponding recall $\tilde{r}$ exceeds r:

$$p_{interp} = \max_{\tilde{r}:\tilde{r}>r} p(\tilde{r}), \tag{2.62}$$

with $p(\tilde{r})$ representing the precision at recall $\tilde{r}$.

PASCAL evaluation protocol: **advantages**(↗) and **disadvantages** (↙)

- ↗ non-binary local evaluation;
- ↗ provides a global performance score that captures the quality-quantity aspects of a detection.
- ↙ penalizes partial *one-to-one* matchings;
- ↙ single global performance score.
- ⚠ provides no information on how *many-to-one* or *many-to-many* matchings are being handled.

2.5.13 Shivakumara's evaluation protocol

In [Shivakumara et al., 2009a], [Shivakumara et al., 2013], [Shivakumara et al., 2009b], [Shivakumara et al., 2011] the authors proposed an evaluation framework in which the matching strategy consists of classifying the text objects into the following categories:

Truly Detected Block (TDB): a detection that contains at least one valid character;

Falsely Detected Block (FDB): a false positive;

Text Block with Missing Data (MDB): a detection that covers less than 80% of the characters in a text line.

The global performance scores are the Recall R, the Precision P, the false positive rate FP and the missdetection rate MDR computed as:

$$R = \frac{TDB}{ATB}, \qquad P = \frac{TDB}{TDB + FDB}, \qquad FPR = \frac{FDB}{TDB + FDB}, \qquad MDR = \frac{MDB}{TDB},$$

where ATB represents the number of actual text blocks. The F-Score is finally used to combine R and P.

Shivakumara's evaluation protocol: **advantages**(↗) and **disadvantages** (↙)

- ↗ allows both fully and partially detected text lines;
- ↗ provides a complex set of metrics.
- ↙ no separation between partial and perfect detections;
- ⚠ provides no information on how *one-to-many* or *many-to-one* matchings are being handled.

2.5.14 VACE Metrics

In [Kasturi et al., 2009] a *Frame Detection Accuracy* (FDA) overall performance measurement was introduced to evaluate a set of GT-detection matchings based on the best spatial overlap approach:

$$FDA = \frac{Overlap\_Ratio}{\frac{N_G + N_D}{2}}, \tag{2.63}$$

where $Overlap\_Ratio$ is the sum of all Jaccard indices between the GT objects G_i and their corresponding detections D_j defined as:

$$Overlap\_Ratio = \sum_{i=1}^{N_{mapped}} \frac{Area(G_i \cap D_j)}{Area(G_i \cup D_j)}, \tag{2.64}$$

and N_{mapped} represents the number of matched text object pairs between the GT and the detection set. Small matching inconsistencies are avoided by thresholding this overlap ratio. The proposed thresholded overlap ratio is computed as:

$$\text{Thresholded } Overlap\_Ratio = \sum_{i=1}^{N_{mapped}} \frac{FDA\_T(i)}{Area(G_i \cup D_j)}, \tag{2.65}$$

where $FDA\_T(i)$ is computed with respect to a threshold value th in the following way:

$$FDA\_T(i) = \begin{cases} Area(G_i \cap D_j), & \text{if } \frac{Area(G_i \cap D_j)}{Area(G_i \cup D_j)} \geq th \\ Area(G_i \cup D_j), & \text{if } \frac{Area(G_i \cap D_j)}{Area(G_i \cup D_j)} < th \text{ and } th \in]0,1[\\ 0, & \text{if } \frac{Area(G_i \cap D_j)}{Area(G_i \cup D_j)} < th \text{ and } th \in \{0,1\} \end{cases} \tag{2.66}$$

VACE metric **advantages**(↗) and **disadvantages** (↙)

- ↗ relaxation of localization errors;
- ↗ non-binary local evaluation.
- ↙ no clear separation between the recall and precision;
- ↙ *one-to-many* or *many-to-one* mapping are not considered;
- ↙ normalization of all FDA to the number of frames in the database.

2.5.15 Yi's evaluation protocol

In [Yi and Tian, 2011b], the authors proposed an evaluation protocol that deals with inclined text lines. It consists in computing the precision of a detected text line $D_j^{\theta_1}$ with respect to a GT text line $G_i^{\theta_2}$, with θ_1 and θ_2 denoting the slant angles corresponding to the two objects:

$$P_{\theta_1} = \frac{Area(G_i^{\theta_2} \bigcap D_j^{\theta_1})}{Area(D_j^{\theta_1})} \tag{2.67}$$

Yi's evaluation protocol: **advantages**(↗) and **disadvantages** (↙)

- ↗ more accurate evaluation due to an adaptation to inclined texts;
- ↗ non-binary evaluation;
- ↙ no Recall value;
- ⚠ different matching strategies not explained.

2.5.16 ZoneMap metric

The ZoneMap metric proposed in [Galibert et al., 2014] is a generalization of the metric proposed in [Mao and Kanungo, 2002] and the DETEVAL framework [Wolf and Jolion, 2006] used for evaluating page segmentation and area classification in documents. It computes different error rates based on two coverage rates C_{D_j,G_i} and C_{G_i,D_j} between a GT object G_i and a detection D_j as:

$$C_{D_j,G_i} = \frac{Area(D_j \bigcap G_i)}{Area(D_j)} \tag{2.68}$$

$$C_{G_i,D_j} = \frac{Area(D_j \bigcap G_i)}{Area(G_i)} \tag{2.69}$$

For each match, a force link, f_{G_i,D_j}, is computed as a combination of the two coverage rate values:

$$f_{G_i,D_j} = C^2_{G_i,D_j} + C^2_{D_j,G_i} \tag{2.70}$$

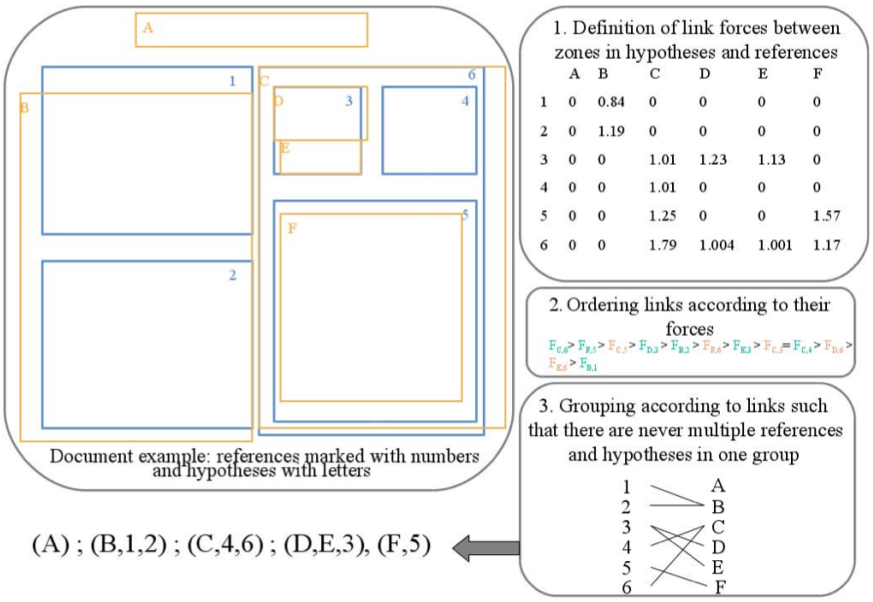


Fig. 2.8: Zonemap framework [Galibert et al., 2014].

The final score is the error rate, $E_{ZoneMap}$:

$$E_{ZoneMap} = \frac{\sum_{i=1}^{N} E_i}{Area(R)}, \tag{2.71}$$

where E_i is a linear interpolation of a surface error rate E_S and a classification error E_C defined as:

$$E = (1 - \alpha_c)E_S + \alpha_c E_C, \tag{2.72}$$

where $\alpha \in [0,1]$ is the weight assigned to the surface rates. E_S and E_C are given depending on the matching type.

ZONEMAP metric **advantages**(↗) and **disadvantages** (↙)

- ↗ *one-to-many* mappings handled;
- ↙ *many-to-many* matchings are not allowed;
- ↙ *one-to-many* and *many-to-one* matches are treated in the same way.

2.6 Text detectors and used evaluation protocols

Despite the diversity of protocols used for text detection purposes, most of them still lack accuracy or are not sufficiently elaborate to deal with the complexity of the detection scenarios that can occur. Table 2.4 summarizes a series of recent text detection methods and their experimental details. For each text detector, we mention the used datasets and evaluation protocol. Existing inconsistencies, related to the used evaluation procedures, are signaled in the last column of this table. Based on Table 2.4 we can make several conclusions:

1. The majority of text detectors are evaluated on the ICDAR databases.
2. The ICDAR′03 protocol is still used despite its well-known drawbacks (see Section 2.5.7).
3. The DETEVAL protocol is by far the most used framework for evaluation purposes. However, only a few number of text detection methods specify the chosen configuration when evaluating their results.
4. A significant number of text detection methods use as evaluation procedure "traditional" Recall, Precision and $F-$Score, without providing any information on how they were obtained or on the implied matching strategy.
5. A number of works need to manually change their level of detection to cope with the granularity imposed by the majority of protocols, mainly passing from a line level detection to a word level one. Most of the times, the way this is done is not even explained.
6. The comparison between detectors is often wrong because it is based on the results obtained not with the same evaluation protocol, but with different ones.

Based on these statements, we can conclude that there is no unified evaluation protocol used by the text detection community. While only few works question the reliability of the existing protocols, most of the time they propose different solutions to avoid the restrictions imposed by these protocols instead of directly tackling the encountered problems. While the biggest interest is given to text detection approaches, a reflexion on the validity of the used evaluation protocols seems to be neglected. Hence, we believe that a more significant importance should be given to the manner in which text detectors are being evaluated. The objective of the following chapter is to propose an evaluation framework that solves many of the existing problems discussed herein and that can deal with the diversity of outputs produced by detection methods.

Tab. 2.4: Recent text detection methods, the used datasets and evaluation protocols.

Text detectors	Dataset	Evaluation protocol	Remarks
[Chen et al., 2004a]	-	[Shivakumara et al., 2011]	
[Chen et al., 2011]	RRC'03 RRC'05	ICDAR′05 protocol	
[Du et al., 2012]	RRC'03 EPSHTEIN'S DATASET	ICDAR′03 protocol	In order to compare with other methods on ICDAR′03 dataset text line level detections were cut into separate words, but not mentioned how.
[Fraz et al., 2015]	RRC'11-SI SVT	DETEVAL	Inconclusive precision value due to incomplete annotation of SVT dataset.
[Gao et al., 2013]	RRC'11-SI	ICDAR′11 protocol	
[González and Bergasa, 2013]	RRC-'03 RRC'05 RRC'11-SI	ICDAR′03 protocol DETEVAL	
[Hua et al., 2001]	HUA'S DATASET	[Hua et al., 2001]	Video text detection.
[Huang and Ma, 2010]	"*a number of broadcast TV videos*"	Detection rate/ False alarm rate	No evaluation details given.
[Huang et al., 2013]	RRC'05 RRC'11-SI	ICDAR′03 protocol	
[Huang et al., 2014]	RRC'11-SI	ICDAR′11 protocol	Comparison with methods (SFT-TCD [Huang et al., 2013]) that follow different evaluation schemes than ICDAR′11.
[Jaderberg et al., 2014a]	RRC'03 RRC'11-SI RRC'13-SI SVT IIIT 5K IIT STR IIT SPORTS	DETEVAL	Inconclusive precision value due to incomplete annotation of SVT dataset.

Tab. 2.4: Recent text detection methods, the used datasets and evaluation protocols.

Text detectors	Dataset	Evaluation protocol	Remarks
[Jain et al., 2014]	TRECVID[29]	Pixel based evaluation [Peng et al., 2011]	
[Jameson and Abdullah, 2014]	"*109 natural scene images with colored curved text*"	[Shivakumara et al., 2013]	
[Kang et al., 2014]	MSRA-TD500 OSTD	Recall/Precision/$F-$Scorel	No evaluation details given.
[Khare et al., 2015]	RRC'13-V own dataset (1000 videos)	DETEVAL	Combination of the word level GTs into text lines for computing the measures proposed by the DETEVAL protocol.
[Lee et al., 2010]	RRC'05-SI	ICDAR′05 protocol	
[Lee et al., 2011]	RRC'03 RRC'05	Standard Recall, Precision and $F-$Score	
[Lee and Kim, 2013]	RRC'05 RRC'11-SI	DETEVAL with default parameters	Recall, Precision and $F-$Score are calculated for each image and then averaged by the total number of images.
[Li et al., 2013]	RRC'03 RRC'11-SI	Standard Recall, Precision and $F-$Score	
[Liu et al., 2012]	RRC'03	Pixel level evaluation	
[Liu et al., 2014a]	RRC'05 RRC'11-SI RRC'13-SI	Recall/Precision/$F-$Score	No evaluation details given.
[Liu et al., 2014b]	RRC'11-SI	ICDAR′03 protocol	Unfair comparison of this method with ICDAR 2011 RRC participants that were evaluated using DETEVAL protocol and not ICDAR′03.
[Liu et al., 2015]	RRC'05 RRC'13-SI	Recall, Precision and $F-$Score using the best match approach	

[29] http://trecvid.nist.gov/trecvid.data.html

Tab. 2.4: Recent text detection methods, the used datasets and evaluation protocols.

Text detectors	Dataset	Evaluation protocol	Remarks
[Lu et al., 2015]	RRC'13-SI SVT	DetEval	
[Mekhalfi et al., 2015]	2 own indoor scene databases	Sensitivity and Specificity rates	
[Meng and Song, 2012] [Meng et al., 2013]	RRC'03 RRC'11-SI SVT	ICDAR′05 protocol ICDAR′11 protocol	
[Merino-Gracia et al., 2011]	ICDAR′05	ICDAR′03 protocol	The text detection outputs are at line level. Authors changed the word level GT of ICDAR′03 dataset to line level in order to make the evaluation with other algorithms fair.
[Milyaev et al., 2015]	RRC'13-SI	ICDAR′13 protocol DetEval	
[Neumann and Matas, 2013]	RRC'11-SI	ICDAR′11 protocol	
[Neumann and Matas, 2012]	RRC'11-SI SVT	ICDAR′11 protocol	Low precision due to an incomplete annotation of the GT in SVT database. Comparison with the method in [Wang et al., 2011] which uses a different evaluation protocol.
[Pan et al., 2009] [Pan et al., 2008]	ICDAR′05	ICDAR′05 protocol	The text detection outputs are at line level. Authors changed the word level GT of ICDAR′03 dataset to line level in order to make the evaluation with other algorithms fair.
[Pan et al., 2011b]	RRC'05	ICDAR′05 protocol	

Tab. 2.4: Recent text detection methods, the used datasets and evaluation protocols.

Text detectors	Dataset	Evaluation protocol	Remarks
[Pavithra and Aradhya, 2014]	101 VIDEO SOUTH INDIAN LANGUAGE DATASET RRC'03 RRC'13-BD RRC'13-SI	[Shivakumara et al., 2012] & [Phan et al., 2009] ICDAR′13 protocol	
[Peng et al., 2011]	660 broadcast video images	Pixel/block level evaluation	
[Phan et al., 2012]	RRC'03-SI	ICDAR′05 protocol	
[Pillai et al., 2013]	RRC'03	-	No evaluation protocol.
[Prakash and Ravishankar, 2013]	RRC-V?? MSRA-TD500 OSTD Own database[30]	[Yao, 2012] protocol	
[Qu et al., 2013]	RRC'05	ICDAR′05 protocol	
[Rong et al., 2014]	RRC'13-SI	Character level evaluation using standard Recall and Precision	No matching strategy explained.
[Roy et al., 2015]	HUA'S DATA Own horizontal text dataset NUS data RRC'03 RRC'11-SI RRC'11-BD SIGN EVALUATION DATA SVT MSRA-TD500	Character recognition rate Pixel level accuracy & Character recognition rate Character recognition rate	

[30] 50 video clips

Tab. 2.4: Recent text detection methods, the used datasets and evaluation protocols.

Text detectors	Dataset	Evaluation protocol	Remarks
[Sharma et al., 2012]	Own dataset (video frames) HUA'S DATASET RRC'03	[Shivakumara et al., 2011]	
[Shekar et al., 2014]	RRC'03 HUA'S DATASET		Same annotation as in [Shivakumara et al., 2009a] and [Shivakumara et al., 2009b].
[Shi et al., 2013]	RRC'11-SI	DETEVAL with default configuration	
[Shi et al., 2014]	RRC'11-SI	DETEVAL	
[Shi et al., 2014]	RRC'11-SI	DETEVAL with default configuration	
[Shivakumara et al., 2008]	RRC'05	[Shivakumara et al., 2009a, Shivakumara et al., 2013, Shivakumara et al., 2009b, Shivakumara et al., 2011]	
[Sun et al., 2014]	RRC'13-SI	ICDAR′13 protocol	Evaluation with ICDAR′13 protocol led to poor performances of Precision and Recall rates due to text line detections.
[Sun et al., 2015]	RRC'13-SI	ICDAR′13 protocol	Manual split of text line detections into words in order to use the ICDAR′13 protocol.
[Tomer and Goyal, 2013]	-	Recall/Precision/$F-$Score	No evaluation details given.
[Wang et al., 2013a]	RRC'03 FUJITSU	ICDAR′03 protocol	Only Precision measurement needed for the evaluation (Recall is always 1).
[Wang et al., 2013b]	RRC'03 RRC'11-SI	Recall/Precision/$F-$Score	No evaluation details given.
[Wang et al., 2014]	RRC'03 FUJITSU	ICDAR′03 protocol	Only a Precision criterion is used (Recall is always 1).

Tab. 2.4: Recent text detection methods, the used datasets and evaluation protocols.

Text detectors	Dataset	Evaluation protocol	Remarks
[Wang et al., 2015a]	RRC'05 RRC'11-SI RRC'13-SI	ICDAR′05 protocol ICDAR′11 protocol ICDAR′13 protocol	GT inconsistency remarks.
[Wang et al., 2015b]	RRC'03 EPSHTEIN'S DATASET RRC'11-SI RRC'13-SI	[Mariano et al., 2002] DETEVAL	
[Wu et al., 2014]	500 video images	Recall/Precision/$F-$Score according to [Shivakumara et al., 2013]	
[Wu et al., 2015]	Own video dataset[31] RRC'13-SI RRC'13-V YOUTUBE (YVT) EPSHTEIN'S DATASET MSRA-TD500	ICDAR′13 protocol	Combination of the word level GTs into text lines for computing the measures proposed by the ICDAR′13 protocol.
[Yan et al., 2014]	RRC'05-SI	ICDAR′05 protocol	
[Yang et al., 2014]	HUA'S DATASET RRC'11-BD	Pixel based evaluation & [Zhao et al., 2010] protocol ICDAR′11 protocol (DETEVAL)	
[Yao et al., 2014]	RRC'11-SI MSRA-TD500	DETEVAL	
[Ye and Doermann, 2014]	RRC'11-SI SVT	DETEVAL Recall and precision based on the overlap area between the GT and the detection set	

[31] including multi-oriented, multi-font, multi-size digital and scene texts

Tab. 2.4: Recent text detection methods, the used datasets and evaluation protocols.

Text detectors	Dataset	Evaluation protocol	Remarks
[Yi and Tian, 2011a]	ICDAR′03 EPSHTEIN'S DATASET	Standard Recall and Precision	No details on the matching.
[Yi and Tian, 2012]	RRC'03	Recall, Precision and $F-$Score based on match area	Local object evaluation based on the Jaccard index.
[Yi and Tian, 2013]	RRC'03-SI RRC'11-SI SVT	Recall/Precision/$F-$Score	
[Yin et al., 2014]	RRC'11-SI RRC'11-BD MICROSOFT EP MULTILINGUAL DATASET MSRA-TD500	ICDAR′13 protocol (DETEVAL) ICDAR′03 protocol [Yao, 2012] protocol	
[Yuan et al., 2015]	RRC'03 OSTD	ICDAR protocols ('03/'05/'11) Yi's protocol [Yi and Tian, 2011b]	Line level annotation of both datasets.
[Zagoris and Pratikakis, 2013]	RRC'11-SI	DETEVAL	No parameters given.
[Zhang et al., 2008]		[Kasturi et al., 2009] protocol	
[Zhang and Chong, 2013]	RRC'05	Recall and Precision computed at pixel level	
[Zhang and Kasturi, 2014]	SIGNS-N800	"*standard Recall and Precision*"	No details on the metrics or the matching strategy.
[Zhang et al., 2015]	RRC'13-SI Own dataset	ICDAR′13 protocol (DETEVAL) [Chen et al., 2004a] protocol	
[Zhao et al., 2015]	RRC'05 RRC'11-SI EPSHTEIN'S DATASET	ICDAR′03/05 protocols	Values computed for each image and then averaged by the total number of images. Comparison with other methods using different protocols.

Tab. 2.4: Recent text detection methods, the used datasets and evaluation protocols.

Text detectors	Dataset	Evaluation protocol	Remarks
[Zhu et al., 2015]	RRC'11-SI RRC'13-SI RRC'13-BD Own static video image dataset Own perspective distort scene image dataset	DETEVAL Not mentioned for natural scene databases	

3 EVALTEX evaluation tool

Contents

This chapter describes the full chain of our evaluation for text detection systems, called EVALTEX. *We are interested in covering all aspects: the ground truth annotation choice, the applied matching strategies, as well as the metrics used to compute the local and global scores.*

In order to evaluate the performance of a text localization algorithm, we adopt a two-level ground truth annotation for each image (see Section 3.2): first, each word is bounded by a rectangular box; then, we group several words and bound them into text regions. This two-level annotation is then used to compare the ground truth text objects with the detection results. Based on the overlap between the GT *and the detection objects we determine to which type of matching a* GT *object belongs to: one-to-one, one-to-many, many-to-one or many-to-many. Depending on the matching type, we compute a dedicated set of performance metrics for each* GT *object (see Section 3.3). Next, we compute global scores for an image or a whole dataset (see Section 3.4.2), by providing both a quality and a quantity evaluation of the detection results. Finally, we show how* EVALTEX *can be extended to any irregular text representations, such as polygonal, elliptic or even free-form ones.*

3.1 Specifications for a reliable evaluation protocol

Before detailing the evaluation protocol proposed in this manuscript, we first need to enumerate the series of constraints and assumptions that form the basis of EVALTEX. A reliable evaluation framework should :

1. deal with different ground truth annotation representations;

2. treat all four types of matching scenarios: *one-to-one*, *one-to-many*, *many-to-one* and *many-to-many*;

3. treat the different matching scenarios consistently;

4. penalize the *one-to-many* cases as the detections are splitting the granularity of the GT elements;

5. treat equally different detection granularity levels within some well-defined limits and rules;

6. provide intuitive metrics, both at object and global levels;

7. provide a visualization tool, as an alternative to metric interpretation, capable of illustrating intuitively the characteristics of a detection;

8. provide accurate evaluation results, independent of the target application;

9. offer a clear separation between the quantity aspect of a detection and its quality aspect;

3.2 Two-level ground truth annotation

In Section 2.2 we have discussed the diversity of issues related to the GT annotation. While some of those issues still remain debatable (for example concerning annotator's subjectivity), others, such as the granularity inconsistency, can be overcome, as it will be shown in this chapter. For many evaluation protocols, dealing with detection granularities different than the GT ones can lead to severe penalizations. However, in many cases, we want to treat and score equally the different detection granularities (*i.e.* word and line level). This can be done by dealing simultaneously with multiple GT annotation levels. Hence, in our approach, we propose to annotate the GT by bounding, using a rectangular box, each text object at a word level and then to manually group text boxes into regions following a predefined criteria that will be defined in the following. Given a subset W of GT objects in $\mathcal{G} = \{G_i\}_{i=1...N_G}$, we define $Reg(W)$ as their region if and only if:

$$Area(Reg(w)) < 2 \sum Area(G_i), \text{with } G_i \in W \tag{3.1}$$

In other words, the evaluation protocol considers text boxes as part of a same region as long as the text area within the region is larger than the non text area. In practice, texts that are aligned in a same row (respectively column) and having similar heights (respectively widths) can be grouped into regions. Figure 3.1 illustrates some cases of dismissed region labeling due to the violation of the constraint in Equation 3.1.

A *region* is therefore considered the box bounding one or several GT text objects. If a GT object cannot be associated to others, then it is considered as a region itself. In Figure 3.2 two objects (words "HFC" and "BANK") are annotated belonging to two separate regions, because their association violates the constraint in Equation 3.1. The reasoning behind the region labeling is based on two aspects. First, we do not want to penalize the scores for detections covering several text boxes (*many-to-one* detections), as long as the covered boxes belong to the same region. When a detection exceeds the boundaries of a GT text object, the Precision is obviously penalized. Hence, when a detection matches several GT

Fig. 3.1: Examples of invalid text region annotations (black rectangles) due to the fact that the non textual area within the region is larger than the text area.

Fig. 3.2: GT objects (labeled in red) that are also single regions (yellow rectangles).

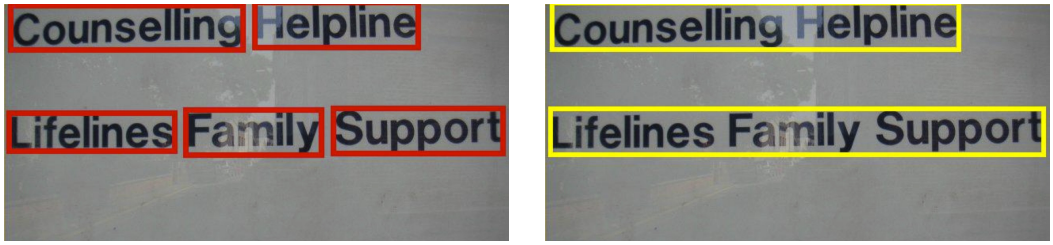
objects, the existing non-text area within the detection will contribute to the decrease of the Precision value. By region labeling the GT, all matched GT objects are treated as a single object and hence an unfair Precision penalization is avoided.

Secondly, it is essential to produce a comparable and undifferenced evaluation for algorithms that produce the same results, but at different detection levels (*i.e.* word and line level). Often, the different output levels depend on the choice of the text detection approach. Connected component based text detection methods are able to extract characters and therefore can provide a more precise character or word level. On the other hand, texture-based approaches have more difficulties in correctly defining the exact boundaries of a text region. They rely on the extraction of texture features from pre-defined size image blocks. A classifier then decides whether the blocks contain textual information or not. Therefore, the detection box is rather an estimation of the text localization than a precise set of text coordinates. However, the detection output can also be influenced by the grouping step. Once text candidates have been detected, they can be grouped into larger text areas: words, single text lines or even larger text regions (multiple lines).

Example. Figure 3.3 describes the proposed two-level annotation. In Figure 3.3a the GT annotation is done at a word level, while in Figure 3.3b we show the region level annotation. If we suppose these two figures correspond to the outputs of two detectors, an efficient protocol should evaluate equally the two sets of detections and produce equivalent scores for both methods.

In practice, the region level annotation consists in assigning to each GT object a tag, or a region number. Based on this, GT objects that have the same tag can be grouped to form a region. The practical usage of the region tag is shown on the *many-to-one* matchings, during which a detection matches multiple GT objects (see Section 3.4.1).

The arguments exhibited in this section lead to a legitimate conclusion: evaluation protocols should be more flexible and designed to deal evenly with different granularity output levels instead of constraining text detectors to conform to a specific granularity as this might slow down the research progress in the text detection field.



a b

Fig. 3.3: Two examples of GT annotation: (a) at word level; (b) at region level.

3.3 Matching strategy

The evaluation protocols designed to cope with object detections need to adopt a matching strategy to make the correspondence between a set of detections and a set of GT objects. In the following, we introduce two local detection properties: quality and quantity. The quality aspect refers to the portion of a GT object's area that has been detected or to how precise the detection is with respect to a GT object. The quantity property is focused on whether a GT has been detected, or if a detection has a correspondence in the GT.

3.3.1 Local measurements

To locally evaluate the quality of the matching between a GT object and a detection we define the *coverage* and the *accuracy* metrics, equivalent to the Recall and Precision coefficients introduced in Section 2.4.2. Let $\mathscr{G} = \{G_i\}_{i=1...N_G}$ be the set of GT text boxes and $\mathscr{D} = \{D_j\}_{j=1...N_D}$ the set of detections. N_G (resp. N_D) represents the number of objects in $\mathscr{G}$ (respectively in $\mathscr{D}$). For each G_i matched to a detected box D_j, the coverage Cov_i is computed as the ratio between the intersection area of G_i and D_j, and the area of G_i:

$$Cov_i = \frac{Area(G_i \bigcap D_j)}{Area(G_i)} \tag{3.2}$$

For each G_i matched to a detected box D_j, the accuracy Acc_i is computed as the ratio between the intersection area of G_i and D_j, and the area of D_j:

$$Acc_i = \frac{Area(G_i \bigcap D_j)}{Area(D_j)}. \tag{3.3}$$

The coverage and accuracy can be seen as local *quality* measures because they reflect the detection quality of a pair (G_i, D_j). On one hand, the coverage corresponds to the amount of the GT surface matched to a detection, while the accuracy measures the amount of the detection surface that matches a GT object. A perfect detection leads to a value of 1 for both quality coefficients; a partial detection gets a value in the interval $[0, 1]$ while the nonexistence of a matching is evaluated to 0.

Note. Commonly, in the literature, the coverage value is assigned to a GT object while the accuracy is attributed to a detection. The EvalTex framework addresses this in a different way. Both quality measurements are assigned to GT objects. This approach does not disturb in any way the evaluation truthfulness because the two detection characteristics are still counted. This adjustment was implied as

a logical response to the way of interpreting *many-to-one* matchings as multiple *one-to-one* matchings. A more detailed explanation is given in Section 3.4.1.

We have seen in Section 2.4.2 that the coverage and accuracy rates are special cases of the the Jaccard index:

$$J(G_i, D_j) = \frac{Area(G_i) \bigcap Area(D_j)}{Area(G_i) \bigcup Area(D_j)}$$

The advantage of using both the accuracy and coverage compared to the single Jaccard index is to capture two different aspects of a detection. While the Jaccard index is a convenient metric for evaluating the local complexity of a detector, the two rates are a more suitable choice if we want a better understanding of the detection results.

Additionally, for each G_i, respectively D_j, we assign a matching value, $Gmatch_i$ (respectively $Dmatch_j$), which takes a binary value, depending on the existence of an intersection between G_i and a detection, respectively between D_j and a GT box. The matching value represents a local *quantity* measure, which describes whether a GT object (respectively detection) has a correspondence in the detection set (respectively GT) or not. The local quantity metrics are used for counting the number of valid GT and detection boxes. For each object G_i in $\mathscr{G}$, $Gmatch_i$ is the metric that indicates if G_i has at least one correspondence in $\mathscr{D}$:

$$Gmatch_i = \begin{cases} 1 & \text{if } \exists\, j \in \mathscr{D} \mid Area(G_i \bigcap D_j) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

For each object D_j in $\mathscr{D}$, $Dmatch_j$ is the metric that stores whether D_j has at least one correspondence in $\mathscr{G}$:

$$Dmatch_j = \begin{cases} 1 & \text{if } \exists\, i \in \mathscr{G} \mid Area(G_i \bigcap D_j) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.5}$$

3.3.2 Ground truth - detection relationships

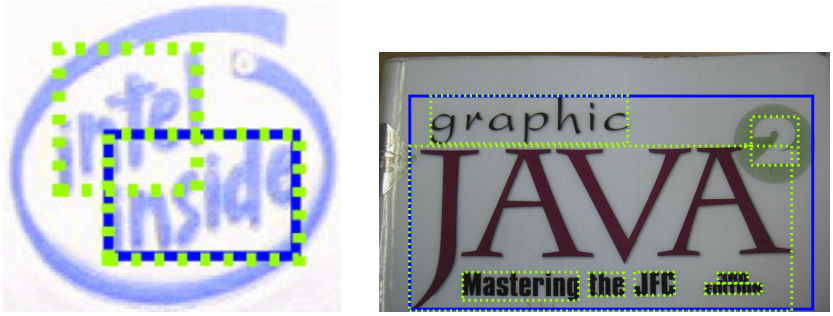
The matching process consists in establishing the relationship between the detections and the GT text boxes. Let us generally denote by $\bowtie$ the relation between the GT and its corresponding detections. We then define the matching between a set of GT text boxes $\{G_{i_1} ... G_{i_k} | \{i_1 .. i_k \in [1, N_G]\}$, and a set of detections $\{D_{j_1} ... D_{j_l} | \{j_1 .. j_l \in [1, N_D]\}$ as $(G_{i_1} ... G_{i_k} \bowtie D_{j_1} ... D_{j_l})$.

The EvaLTex protocol handles the four types of matchings previously introduced in Section 2.4.2:

(a) *one-to-one* — one text object D_j in $\mathscr{D}$ matches exactly one text object G_i in $\mathscr{G}$, denoted by $(G_i \bowtie D_j)$;

(b) *one-to-many* — multiple text objects in $\mathscr{D}$ match one text object G_i in $\mathscr{G}$, denoted by $(G_i \bowtie D_{j_1} ... D_{j_l})$, with $\{j_1 ... j_l\} \in [1..N_D]$;

(c) *many-to-one* — one text object D_l in $\mathscr{D}$ matches multiple text objects in $\mathscr{G}$, denoted by $(G_{i_1} ... G_{i_k} \bowtie D_j)$, with $\{i_1 .. i_k\} \in [1, N_G]$;



a b

Fig. 3.4: Filtering procedure: matching detected boxes (blue) with GT boxes (dashed green); (a) the tilted text causes an overlap between GT text boxes, (b) the character height variation (see the letter "J") causes the inclusions of GT text boxes.

(d) *many-to-many* conditions (b) and (c) are simultaneously satisfied; this case is denoted by $(G_{i_1}...G_{i_k} \bowtie D_{j_1}...D_{j_l})$, with $\{i_1..i_k\} \in [1, N_G]$ and $\{j_1...j_l\} \in [1..N_D]$.

The FPs are denoted by $\{\emptyset \bowtie D_1...D_j\}$, with $j \leq N_D$, while the missed detections are denoted by $\{G_1...G_i \bowtie \emptyset\}$, with $i \leq N_G$. The FPs decrease the precision of a text detector, while the missed detections decrease the overall recall rate of a detector.

3.3.3 Filtering procedure

The rectangular text representation is easy to use for both an annotator and a detector, as it only requires four coordinates to be defined. However, in natural scenes and digital images we often deal with overlapping GT text boxes which can distort the matching results. This is usually caused by inclined texts that is not well fitted by a rectangular box. Namely, when two GT objects overlap, a detection that should be matched with only one of them could automatically be attributed to the other GT object. To avoid such confusions, we apply a filtering procedure to determine if all GT text boxes associated to a detection really corresponds to that detection. The filtering procedure mainly targets two scenarios: partially overlapping GT text objects and total inclusions between GT text objects. These cases are illustrated in Figure 3.4.

GT partial overlapping. The filtering process occurs when a detection matches a set of GT objects that overlap. Hence, in the case of a *many-to-one* match, we check if there is an intersection between two or more GT objects. Let D_j be a detection box covering two overlapping GT boxes G_{i_1} and G_{i_2}. We then assign D_j to G_{i_1} and not to G_{i_2} if the following area constraint is satisfied:

$$Area(G_{i_2} \bigcap D_j) - Area(G_{i_1} \bigcap G_{i_2}) \leq T_{overlapping} \cdot Area(G_{i_2}), \tag{3.6}$$

where $T_{overlapping}$ is a threshold that regulates the amount of overlap area between two GT objects. $T_{overlapping}$ was set to 0.1 in our experiments, which assures the filtering of objects that have a small overlap area in the GT. By increasing $T_{overlapping}$, we could reject valid GT objects that are part of a *many-to-one* matching.

Example. Figure 3.4a illustrates the case of two overlapping GT boxes (in dashed green) because they contain tilted text. In the proposed approach, the filtering procedure ensures that only the word "inside" is matched to the blue detected box, while "intel" is discarded from the detection.

GT total inclusion. Another situation that can perturb the matching process concerns GT boxes inclusions: one GT box contains another or several GT boxes. If a detection covers a GT object that includes one or more GT objects, two scenarios can be adopted:

1. consider all GT objects as matched;

2. consider only the bounding GT object as matched.

The first scenario favors text detectors that group detections into larger regions. However, following this approach, one risks to over-evaluate an "abusive" detection, such as outputting the whole image or to score GT objects that were never supposed to be detected. Consequently, the second choice remains a better and more straightforward alternative. Similarly, let us consider D_j a detection box covering two overlapping GT boxes G_{i_1} and G_{i_2}. We then assign D_j to G_{i_1} and not to G_{i_2} if the following area constraints are verified:

$$G_{i_1} \subset G_{i_2}; \tag{3.7}$$

$$Cov(G_{i_1}) \leq Cov(G_{i_2}); \tag{3.8}$$

$$Area(G_{i_1} \bigcap G_{i_2} \bigcap D_j) \leq Area(G_{i_1} \bigcap G_{i_2}). \tag{3.9}$$

Moreover, if the *many-to-one* match corresponds to only two GT boxes, $\{G_{i_1,i_2} \bowtie D_j\}$, then the following constraint also needs to be checked:

$$Acc(G_{i_1}) \leq Acc(G_{i_2}), \tag{3.10}$$

If more than two GT objects are part of the *many-to-one* match, $\{G_{i_1 \ldots i_l} \bowtie D_j\}$, then the following constraint needs to be satisfied:

$$Cov(G_{i_1}) \cdot Cov(G_{i_2}) \geq T_{coverage} \tag{3.11}$$

Note. The constraints in Equations 3.7, 3.8 and 3.9 ensure that the object G_{i_1} is totally included in G_{i_2}, its overlap area (with the detection) is smaller than that of G_{i_2} and the intersection surface of all three objects (G_{i_1},G_{i_2} and D_j) is smaller than the intersection surface of G_{i_1} and G_{i_2}. Once these three constraints are fulfilled, an additional verification is done based on the total number of GT objects detected by D_j. If G_{i_1} and G_{i_2} are the only objects to be matched with D_j, then we exclude G_{i_1} if and only if its accuracy is lower than the accuracy of G_{i_2}, as seen in the constraint of Equation 3.10. If more than two GT objects are part of the matching with D_j we then ensure that their both coverages are high enough (*i.e.* the product of coverages higher than a threshold $T_{coverage}$ has been set experimentally to 0.8).

Example. Figure 3.4b illustrates the case of inclusion: the bounding box of the word "JAVA" contains the bounding boxes of all words below it: "Mastering", "the", "JFC", "3RD", "EDITION". In this situation, the only matched GT text boxes are the words "JAVA", "graphic", "TM" and "2", while the other words are discarded from the detection. In order to be considered as matched, the removed GT objects should be detected with individual bounding boxes and not part of a *many-to-one* scenario.

3.4 Performance evaluation

In this section we will describe the performance evaluation of a detector, based on the different matching types described in the previous section. Firstly, a local evaluation is done, during which to each GT object will get a coverage and accuracy value. Next, based on all local scores we will derive a set of global scores to have a full characterization of a detector's efficiency.

3.4.1 Local (*object-level*) evaluation

The local evaluation refers to the attribution of scores to each GT object independently. The local measurements introduced in Section 3.3.1 are divided into two quality metrics (coverage and accuracy) and two quantity metrics (GT and detection matching values). While the local quantity measurements strictly depend on the existence of a match, the local quality measurements also depend on the type of matching. Intuitively, a truthful evaluation protocol should interpret differently a *one-to-one* match and a *one-to-many* match: for example, a detection that matches a GT object with an overlap area s should be scored higher than the case where two detections match the same GT object with the total overlap surface s. Hence, based on the matching type, we adapt the coverage and accuracy local metrics such that:

- the quality scores of GT objects belonging to a *one-to-one* match are not penalized as the detections correspond to the exact GT reference granularity;
- the quality scores of GT objects corresponding to *one-to-many* matches are penalized as the detections split the minimum granularity reference;
- the quality scores of GT objects part of a *many-to-one* match are penalized as long as the involved GT objects do not have the same text region tag;
- the quality scores of GT objects part of a *many-to-many* match are evaluated based on the combination of quality scores corresponding to *many-to-one* and *one-to-many* matches.

One-to-one match

A *one-to-one* scenario consists in an exclusive match between a detection and a GT object. If a detection box does not perfectly cover a GT box we refer to it as a *partial match*. In some cases, especially in natural scene images, the content of GT objects can be hard to read, either because they contain very small text characters or because of a blur effect. Assuming that a perfect match is rarely achieved, we

want to not penalize the local quality scores in case of very small offsets between the position of a GT text box and the position of its matched detection box. To do so, we vary the size of the GT text box G_i, by expanding or shrinking it with respect to a margin error.

Let G_i be a GT text object and T_{margin} a regularization parameter. We then define a margin error m_e for G_i by:

$$m_e = \begin{cases} T_{margin} \cdot \frac{Area(G_i)}{height(G_i)} & \text{if } height(G_i) \geq width(G_i) \\ T_{margin} \cdot \frac{Area(G_i)}{width(G_i)} & \text{otherwise} \end{cases} \tag{3.12}$$

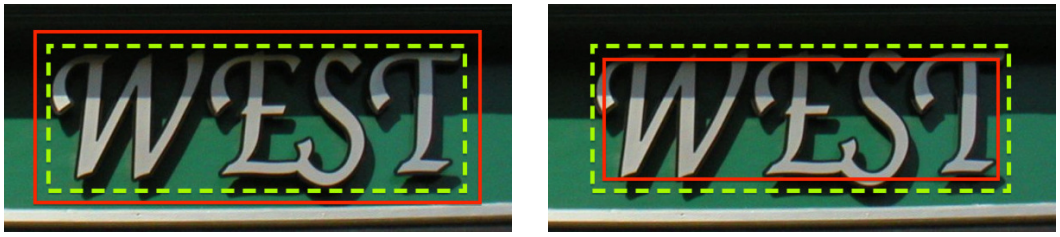
Note. T_{margin} is the parameter that regulates the thickness of the margin error with respect to the size (in pixels) of a text box. This parameter was set during our experiments to 0.1 to make a compromise for equally larger and smaller GT objects. However, very small GT objects should get very low margin values. Since these objects are the hardest to detect due to their size, we proposed to set the margin error to 3 in cases of $m_e < 3$.

Let $[x_{G_i}, y_{G_i}, w_{G_i}, h_{G_i}]$ characterize the GT text box G_i, where x_{G_i} and y_{G_i} are its left upper corner coordinates, and w_{G_i} and h_{G_i} its width and height respectively. Let us now define Ge_i and Gr_i as the extended and the reduced text boxes of G_i:

$$Ge_i : [x_{G_i} - m_e, y_{G_i} - m_e, w_{G_i} + 2 \cdot m_e, h_{G_i} + 2 \cdot m_e] \tag{3.13}$$

$$Gr_i : [x_{G_i} + m_e, y_{G_i} + m_e, w_{G_i} - 2 \cdot m_e, h_{G_i} - 2 \cdot m_e] \tag{3.14}$$

Example. Figure 3.5 shows the reduced and enlarged boxes for a GT box. As it can be observed, the margin error does not influence severely the GT box size. The reduced box slightly "cuts" the borders of text, but it still remains readable.



a: Enlarged text box G_{e_i} **b:** Reduced text box G_{r_i}

Fig. 3.5: Illustration of the extended (left) and reduced (right) boxes, in red, obtained from a GT box (dashed green).

In order to evaluate *one-to-one* detections, we use the defined coverage (Equation (3.2)) and accuracy (Equation (3.3)) rates. Let us consider the *one-to-one* matching ($G_i \bowtie D_j$) and Ge_i the enlarged GT box corresponding to G_i. Then, the accuracy measurement is computed as:

$$Acc_i = \frac{Area(Ge_i \bigcap D_j)}{Area(D_j)}, \tag{3.15}$$

Let us consider the *one-to-one* matching ($G_i \bowtie D_j$) and Gr_i the enlarged GT box corresponding to G_i. Then, the coverage measurement is computed as:

$$Cov_i = \frac{Area(Gr_i \bigcap D_j)}{Area(Gr_i)} \tag{3.16}$$

For any *one-to-one* match between a detection box D_j and a GT box G_i, the accuracy measurement is computed by considering the enlarged text box Ge_i. This allows detections that are slightly larger than the GT text box not be penalized by the extra detection area. Following the same reasoning, we compute the coverage measurement based on the reduced text box G_{r_i}. This allows detections that are slightly smaller than the GT text box not be penalized by the missing coverage area.

Impact of margin parameter. By increasing the value of the parameter T_{margin} the coverage and accuracy values will increase equally. Hence, it is not recommended to give a very high value to this parameter as it might degrade the detection evaluation. The experimental value of 0.1 allows only small imprecisions for detections.

Example. Figure 3.6 illustrates the impact of T_{margin} on the coverage scores obtained from matching the GT object "4B.532" with a detection box that is smaller than the GT one. The coverage value of the GT text when the margin option is disabled ($T_{margin} = 0$, see Figure 3.6a) is 0.68. By increasing T_{margin}, the margin error increases as well. As a consequence the size of the GT box is gradually reduced which leads to a higher matching surface between the two objects and hence an increase of the coverage values. This example illustrates the good choice of setting T_{margin} to 0.1. When higher values of T_{margin} are used, the GT object becomes too much shrinked (see Figure 3.6f) and the reduced box does not represent anymore accurately the GT text.

The accuracy scores vary in a similar way when T_{margin} increases. The only difference is that instead of reducing the box, we enlarge it and then compute the matching area between the two objects.

a: $T_{margin} = 0$, $m_e = 0$, $Cov = 0.68$ **b:** $T_{margin} = 0.1$, $m_e = 23$, $Cov = 0.72$ **c:** $T_{margin} = 0.2$, $m_e = 46$, $Cov = 0.76$

d: $T_{margin} = 0.3$, $m_e = 69$, $Cov = 0.80$ **e:** $T_{margin} = 0.4$, $m_e = 93$, $Cov = 0.85$ **f:** $T_{margin} = 0.5$, $m_e = 116$, $Cov = 0.88$

Fig. 3.6: The impact of the T_{margin} value on the coverage scores of a GT text box (red) matched to a detection (in green): (a) original GT box size; (b)-(f) shrinkage variations of the GT box.

One-to-many match

The *one-to-many* case consists in attributing multiple detections to a singular GT text box. We refer to this as a fragmented matching that, as already discussed in Section 3.4.1, will contribute to the quality measurement penalization. Figure 3.8 illustrates three different *one-to-many* cases involving one GT object and two detections.

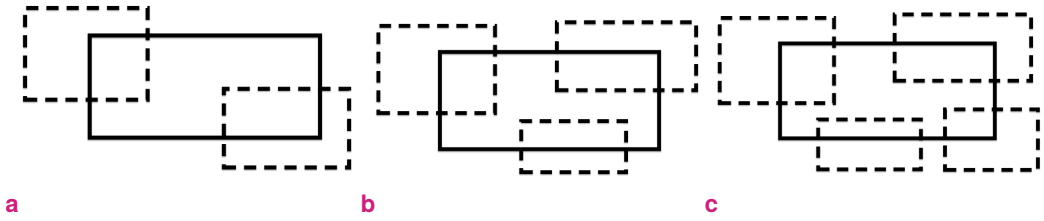


Fig. 3.7: Different *one-to-many* scenarios (detections are illustrated with dashed rectangles, the GT is depicted with plain rectangle) with different fragmentations: (a) $s_i = 2$; (b) $s_i = 3$; (c) $s_i = 4$.

We have an ideal matching if each GT object is detected only once. But in some cases, a GT object can be matched to multiple detections: we penalize such cases by applying a fragmentation rule. Let F_i be a fragmentation penalty with values in $[0,1]$, and s_i the split level corresponding to the GT text box G_i (number of detections that intersect G_i). Examples with different fragmentations are illustrated in Figure 3.7. Let us consider the *one-to-many* match $(G_i \bowtie D_{j_1} ... D_{j_{s_i}})$that intersects G_i. Then, the coverage of G_i is obtained as follows:

$$Cov_i = Cov_i^u \cdot F_i, \tag{3.17}$$

where Cov_i^u represents the union of all intersection areas between Gr_i and all detections D_j, with $j \in [j_1, j_{s_i}]$, divided by the GT area, and defined as:

$$Cov_i^u = \frac{\bigcup_{j=j_1}^{j_{s_i}} Area(Gr_i \bigcap D_j)}{Area(Gr_i)} \tag{3.18}$$

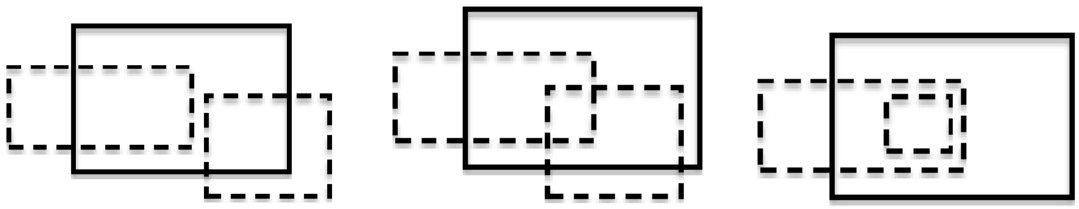
Let us now consider the corresponding accuracy for the text box G_i, defined as the union of all intersection areas between Ge_i and all detections D_j, with $j \in [j_1, j_{s_i}]$, divided by the union of all detection areas:

$$Acc_i = \frac{\bigcup_{j=j_1}^{j_{s_i}} Area(Ge_i \bigcap D_j)}{\bigcup_{j=1}^{s_i} Area(D_j)} \tag{3.19}$$

Equations 3.18 and 3.19 are usable only if the detections are disjoint (Figure 3.8a). We denote I_D as the intersection of the detections $(D_{j_1} ... D_{j_{s_i}})$:

$$I_D = \bigcap_{j=j_1}^{j_{s_i}} D_j \tag{3.20}$$

If the detections intersect, either partially or totally as illustrated in Figures 3.8b and 3.8c, the two equations would count the intersection surface of the detections I_D twice. Then, in order to avoid summing



a: Disjoint detections. **b:** Partially intersected detections. **c:** Included detections.

Fig. 3.8: Different *one-to-many* scenarios in which two detections (dashed rectangles) correspond to one GT object (plain rectangle); here, $s_i = 2$.

the surface of I_D multiple times, we recompute the coverage and the accuracy rates by subtracting it from the union of the matching surfaces:

$$Cov_i^u = \frac{\bigcup_{j=j_1}^{j_{s_i}} Area(Gr_i \bigcap D_j) - I_D}{Area(Gr_i)} \tag{3.21}$$

$$Acc_i = \frac{\bigcup_{j=j_1}^{j_{s_i}} Area(Ge_i \bigcap D_j) - I_D}{\bigcup_{j=1}^{s_i} Area(D_j)} \tag{3.22}$$

Note. The fragmentation is applied once during the computation of local quality measures to only penalize the coverage (see Equation 3.17). In the literature, the fragmentation scenarios are generally treated in three ways. First, the protocols that rely on a best match approach select the detection with the largest matching area. When detections are included into others, the best match approach completely forgives the *one-to-many* case by not considering the included detection (see Figure 3.8c) and only validating the detection with the largest matching area. Here, the matching area is not penalized due to the inclusion. In other situations, namely when the detections matched to a GT object are disjoint or partially overlap (Figures 3.8a and 3.8b), this approach is even more penalizing, as it excludes valid detection areas from the local measurement computation. Secondly, there are protocols that count all the matching surfaces between the multiple detections and the corresponding GT object but apply no penalization. A third category of methods that considers all detections, and applies a fragmentation penalization. Such a technique, introduced in [Mariano et al., 2002] and later used in [Wolf and Jolion, 2006], proposed the fragmentation penalty $F_i = \frac{1}{1+\ln(s_i)}$, with s_i the number of detections matched with a GT text G_i. For example, using this metrics in Equation 3.17 to evaluate a perfect *one-to-many* detection ($Cov_i^u = 1$) with two detections ($s_i = 2$) leads to $Cov_i = 0.59$. If the same surface is detected three times ($s_i = 3$) $Cov_i = 0.48$, while four detections lead to $Cov_i = 0.42$. One can observe that, used as a fragmentation penalization (and not as an individual metric), F_i can be rather penalizing, especially when dealing with two detections. The advantage of this index is however the linear growth property of the logarithmic function which ensures a consistent penalization with respect to the fragmentation level. Other penalization metrics can also be used, as the EVALTEX protocol evaluation is designed to allow adapting this fragmentation index with respect to the targeted application. In this manuscript all experiments are conducted using the fragmentation index proposed in [Mariano et al., 2002].

Fig. 3.9: Example of a *one-to-many* case ("Yarmouth" word detected two times): one text box in $\mathscr{G}$ (dashed green) is matched to multiple boxes in $\mathscr{D}$ (blue).

Example. The *one-to-many* case, illustrated in Figure 3.9 which shows the word "Yarmouth" being matched to two different detection boxes. The coverage value computed without the fragmentation penalization is $Cov_i^u = 0.82$. Since the word has been matched two times ($F = 0.59$) the final coverage value is $Cov_i = 0.48$. However, if only the detection matched with "Yarm" would take place, the coverage would be $Cov_i = 0.44$. Then the score difference between having only one detection and having two detection would be 0.04. Hence, this fragmentation penalty is still not ideal, as it can sometimes punish too much a *one-to-many* matching. An alternative fragmentation penalty, that provides a smoother transition between the scores obtained with different number of detections, could be $F_i' = \frac{1}{1+\ln(s_i)\cdot\ln(s_i)} \times 0.6 + 0.4$ (see Figure 3.10). By applying F_i' to the text object "Yarmouth" we obtain $Cov_i = 0.73$, which is less penalizing than the score obtained with F_i.

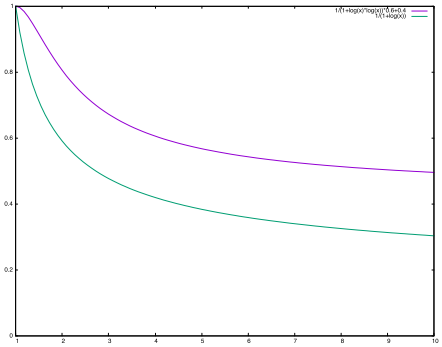


Fig. 3.10: The fragmentation penalty proposed in [Mariano et al., 2002] in green, $F_i = \frac{1}{1+\ln(s_i)}$, and our proposed fragmentation penalty in purple, $F_i' = \frac{1}{1+\ln(s_i)\cdot ln(s_i)} \cdot 0.6 + 0.4$.

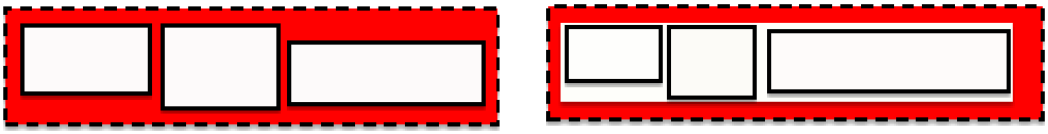
Many-to-one match

The *many-to-one* case implies merging several GT text objects into one single detection. Our protocol treats a *many-to-one* match as "many" *one-to-one* cases. This consists in partitioning the detection surface into multiple areas and attributing them to each GT object.

Let us consider the *many-to-one* match $(G_{i_1}...G_{i_{m_j}} \bowtie D_j)$, with m_j representing the merge level (number of GT objects associated to D_j) matched to the detection box D_j. Then, the coverage of each GT box G_{i_k} in $\{G_{i_1}...G_{i_{m_j}}\}$ is computed as:

$$Cov_i = \frac{Area(Gr_i \bigcap D_j)}{Area(Gr_i)} \tag{3.23}$$

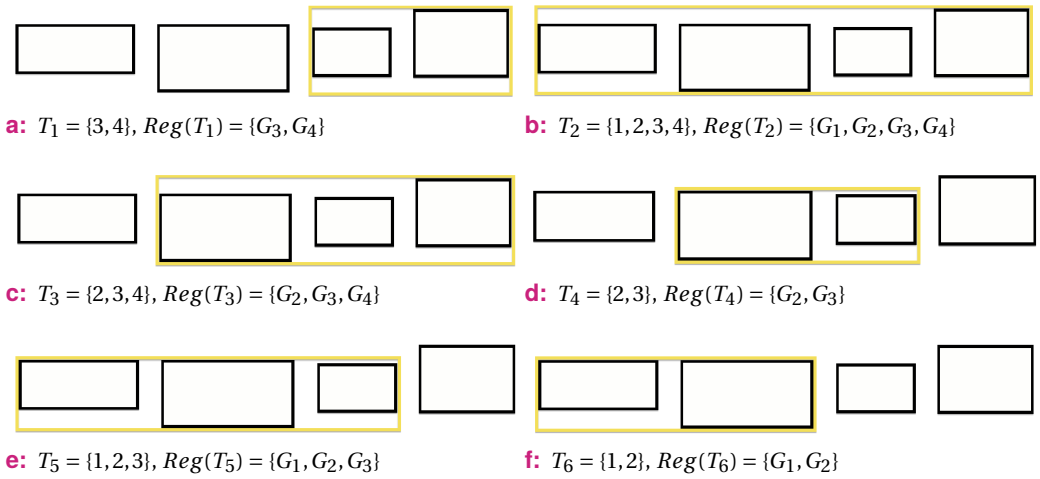
While the coverage only focuses on the amount of valid matched text areas, the detection accuracy also takes into account how much non textual areas (areas outside the GT box), illustrated in Figure 3.11 with



a: No region grouping. **b:** Region grouping.

Fig. 3.11: *Many-to-one* matchings: one detection (dashed rectangle) matches multiple GT objects (plain rectangles); the red surface corresponds to the non-textual surface; the white surface corresponds to the valid GT area.

red, has been detected. Consequently, when a detection matches several GT objects, the non textual area derived from the inter-object spacing should contribute to the penalization of the accuracy measurement. Then, a truthful comparison between a word level detection and, for example, a line level detection would not be possible. Following this reasoning, the *one-to-one* detections would always outperform a *many-to-one* detection. However, in many cases, if not most of them, detecting at object level or detecting at region level should be scored equally. The proposed solution, described in Section 3.2, of assigning a two-level GT annotation solves this problem and allows a better comparison between different detection outputs. This is achieved by assuming that the area of a text region does not contain any non textual area. We now consider the spacing area between GT objects belonging to a same region as a valid text surface. This is illustrated in Figure 3.11b where the white surface denoting the text region visibly exceeds the boundaries of the three GT objects contained into the region. In order to compute the



a: $T_1 = \{3,4\}$, $Reg(T_1) = \{G_3, G_4\}$ **b:** $T_2 = \{1,2,3,4\}$, $Reg(T_2) = \{G_1, G_2, G_3, G_4\}$

c: $T_3 = \{2,3,4\}$, $Reg(T_3) = \{G_2, G_3, G_4\}$ **d:** $T_4 = \{2,3\}$, $Reg(T_4) = \{G_2, G_3\}$

e: $T_5 = \{1,2,3\}$, $Reg(T_5) = \{G_1, G_2, G_3\}$ **f:** $T_6 = \{1,2\}$, $Reg(T_6) = \{G_1, G_2\}$

Fig. 3.12: Different valid region configurations (yellow) denoted with $Reg(T_l)$ for a set of four GT objects illustrated with black rectangles and denoted, from left to right, with $G_1, G_2, G_3$and G_4 that are labeled with the same region tag.

accuracy rate for each GT G_i, we first need to assign them a detection area. We recall that our protocol computes both the coverage and accuracy for each GT object, while traditional approaches assign the coverage to GTs and accuracy to detections. Therefore, the detection area is split between the targeted m_j GT objects. Let us define $TextArea_{D_j}$ as the *valid detection* area obtained from the union of all GT text areas within the detection box D_j.

We first suppose that the GT is as a set of text boxes exclusively at word level, as seen in Figure 3.11a. Then, $TextArea_{D_j}$ is computed as follows:

$$TextArea_{D_j} = Area(\bigcup_{i=i_1}^{i_{m_j}} (Ge_i \bigcap D_j)) \tag{3.24}$$

Let us now also consider $RT(G_i)$ the region tag associated to a GT object G_i. Now, based on all m_j GT objects involved in the *many-to-one* mapping, we generate the corresponding set of $r \leq m_j$ regions $\{Reg(T_1)...Reg(T_r) \mid \sum_{l=1}^{r} |T_l| = m_j\}$, such that $T_{l\in[1,r]}$ is a tuple of GT indices defined as $T_l = \langle t_1, ..., t_{k\in[1,m_j]} \rangle$. Then, a region $Reg(T_l)$ is composed of either:

a single GT object G_{t_1} if $\forall i \in [1, m_j] \mid i \neq t_1, RT(G_{t_1}) \neq RT(G_i)$ or

a set of GT objects $G_{t_1}...G_{t_k}$ such that $\forall b \in [1, k-1]$, $RT(G_{t_b}) = RT(G_{t_{b+1}})$ and $G_{t_b} \preceq G_{t_{b+1}}$, where $\preceq$ is a neighborhood function defined between two GT objects. Figure 3.12 illustrates an example of all possible region configurations for a set of four GT objects with the same region tag.

Then, $TextArea_{D_j}$ is computed as the union of all text regions Reg_k within a detection box:

$$TextArea_{D_j} = Area(\bigcup_{l=1}^{r} (Reg(T_l) \bigcap D_j)), \tag{3.25}$$

Consequently, the *non-valid detection* area, $nonTextArea_{D_j}$, corresponds to the total detection area of D_j excluding $TextArea_{D_j}$:

$$nonTextArea_{D_j} = Area(D_j) - TextArea_{D_j}. \tag{3.26}$$

We can now define $Area(D_{j,i})$ as the corresponding detection area for each G_i involved in the $(G_{i_1}...G_{i_{m_j}} \bowtie D_j)$ match:

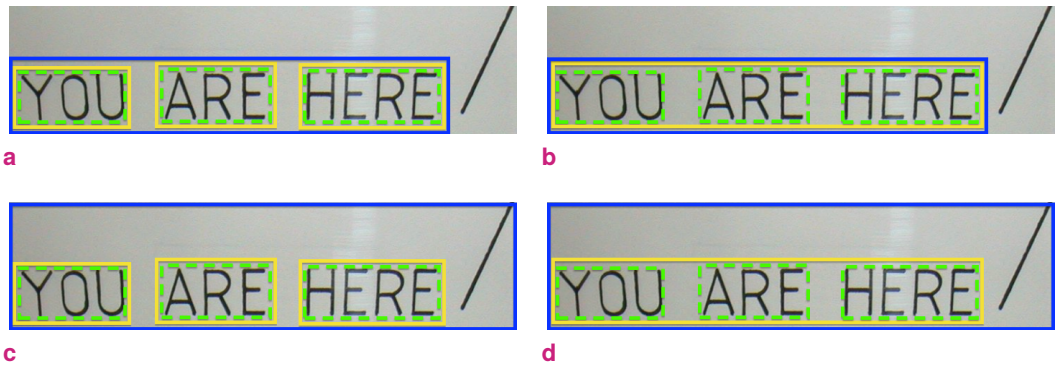
$$Area(D_{j,i}) = \frac{Area(Ge_i)}{TextArea_{D_j}} \cdot nonTextArea_{D_j}. \tag{3.27}$$

Let us consider the *many-to-one* match $(G_{i_1...i_{m_j}} \bowtie D_j)$, with m_j representing the merge level of the detection box D_j and $Area(D_{j,i})$ as the detection surface allocated to each GT text box G_i. Then, the accuracy associated to each matched G_i is:

$$Acc_i = \frac{Area(Ge_i \bigcap D_j)}{Area(D_{j,i})}, \tag{3.28}$$

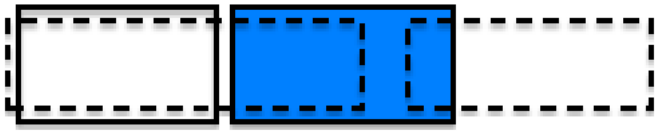
Example. Figure 3.13 shows examples of *many-to-one* scenarios. The image illustrates three GT objects ("YOU", "ARE" and "HERE") with different configurations based on the region grouping and the detection accuracy. The coverage value for all cases remains constant and equal to 1 because the entire GT surface is detected.

Let us consider the cases of the strict detections (nearby the coordinates of the GT objects). In the first example, Figure 3.13a, each text is treated as an individual region and get the accuracy values $Acc_{[YOU]} = 0.8363$, $Acc_{[ARE]} = 0.8328$ and $Acc_{[HERE]} = 0.8374$. The accuracy values for the the second case (Figure 3.13b), where the three objects have the same region tag, are $Acc_{[YOU]} = Acc_{[ARE]} = Acc_{[HERE]} = 1$ because the protocol treats this scenario as a *one-to-one* mapping between the detections (in blue) and one GT object, namely the region (depicted in yellow).

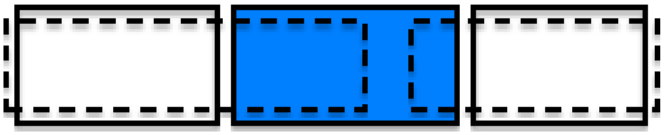


a b c d

Fig. 3.13: *Many-to-one* mapping examples: boxes in D (blue) match several boxes in G (dashed green); (a) a detection box close the GT objects; (b) a detection box close the GT objects grouped into a region (yellow); (c) a coarser detection of the GT objects; (d) a coarser detection of the GT objects grouped into a region (yellow).



a: One *many-to-one* detection and one partial *one-to-one* detection.



b: Two *many-to-one* detections.

Fig. 3.14: *Many-to-many* scenario types: detections are depicted with dashed rectangles; the plain blue rectangle corresponds to a GT object part of a *one-to-many* match.

We now focus on the examples in which the detection area is significantly larger than the GT object area. In the example of Figure 3.13c the accuracy values are much lower than in the previous two cases: $Acc_{[YOU]} = 0.3067$, $Acc_{[ARE]} = 0.2988$ and $Acc_{[HERE]} = 0.3020$. When enabling the region annotation (Figure 3.13d) the accuracy increases (due to the larger GT area and consequently decrease of the non-textual area): $Acc_{[YOU]} = Acc_{[ARE]} = Acc_{[HERE]} = 0.33318$.

Many-to-many match

The *many-to-many* case is invoked when multiple GT objects (at least three) are involved simultaneously in a *one-to-many* and *many-to-one* match. Depending on the *one-to-many* mapping, we can have two *many-to-many* configurations, as shown in Figure 3.14: it can be derived from one *many-to-one* and one *one-to-one* matching (Figure 3.14a) or it can also be generated by two *many-to-one* matchings (Figure 3.14b).

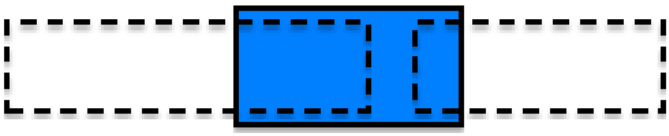


Fig. 3.15: *One-to-many* classic scenario: a GT object (depicted with plain blue rectangle) is matched by two detections.

When such scenarios occur, the coverage and accuracy rates of the involved GT objects can be computed based on two cases of *one-to-many* and *many-to-one*. The GT objects that exclusively take part of a *many-to-one* mapping (illustrated with empty plain rectangles in Figure 3.14) are computed using the standard Equations 3.23 and 3.28 seen in Section 3.4.1. For objects involved in *one-to-many* cases (depicted with plain blue rectangles in Figure 3.15), the coverage rate is computed as in Section 3.4.1 (Equation 3.17):

$$Cov_i = Cov_i^u \cdot F_i \tag{3.29}$$

The particularity of a *many-to-many* mapping consists of the adaptation of the computation of the accuracy for the *one-to-many* objects. To do so, we combine the accuracy equations used during *one-to-one* and *many-to-one* scenarios.

When a GT object is part of *one-to-many* mapping due to multiple *one-to-one* detections (see Figure 3.15), the accuracy rate is as in Equation 3.22:

$$Acc_i = \frac{\bigcup_{j=j_1}^{j_{s_i}} Area(Ge_i \bigcap D_j) - \bigcap_{j=j_1}^{J_{s_i}} D_j}{\bigcup_{j=j_1}^{j_{s_i}} Area(D_j) - \bigcap_{j=j_1}^{j_{s_i}} D_j}$$

We also use Equation 3.28 to evaluate the accuracy for the *many-to-one* case:

$$Acc_i = \frac{Area(Ge_i \bigcap D_j)}{Area(D_{j,i})}$$

Hence, we compute the accuracy as the ratio between the union of all intersection areas between the GT object G_i and the union of all k_i detection surfaces that are generated from the *many-to-one* mappings as well as all s_i detections generated from the *one-to-one* mappings:

$$Acc_i = \frac{\bigcup_{j=j_1}^{j_{s_i+k_i}} Area(Ge_i \bigcap D_j) - \bigcap_{j=j_1}^{j_{s_i}} D_j}{(\bigcup_{j=j_1}^{j_{s_i}} Area(D_j) - \bigcap_{j=j_1}^{j_{s_i}} D_j) \ \bigcup \ (\bigcup_{j=j_1}^{j_{k_i}} Area(D_{k,i}))} \tag{3.30}$$

Fig. 3.16: A *many-to-many* mapping example: a mix of *one-to-many* and *many-to-one* cases.

Example. Figure 3.16 illustrates a *many-to-many* match which is treated as a sequence of *one-to-many* and *many-to-one* cases. Particularly, two *many-to-one* matches ("HEALTHY COLC") and ("ESTER 2000") determined the *one-to-many* match of the word "COLCHESTER". All three GT objects are associated with the same region label.

The word "HEALTHY", part of a *many-to-one* match, is evaluated using the coverage and accuracy in Equations (3.23) and (3.28) respectively. This word, perfectly covered by the detection, gets a coverage $Cov_{[HEALTHY]} = 1$. Due to the region labeling the accuracy of the GT object is also evaluated to $Acc_{[HEALTHY]} = 1$. Similarly, the word "2000" also gets a perfect local evaluation: $Cov_{[2000]} = 1$ and $Acc_{[2000]} = 1$.

The word "COLCHESTER" is involved in a *one-to-many* match derived from two *many-to-one* matches. Hence, the obtained local measures are: $Cov_{[COLCHESTER]} = 0.5212$, obtained with Equation 3.29 and is due to the mismatch of the letter "H" and the two-level fragmentation; the accuracy is maximal $Acc_{[COLCHESTER]} = 1$ (Equation (3.30)) because both *many-to-one* mappings are accurate with respect to the GT region.

3.4.2 Global (*dataset*) evaluation

The performance measurements presented in the previous sections for each type of matching describe how well an individual GT text box has been detected and quantify the accuracy of its detection. Moreover, the local quantity metrics serve to mark, on one hand if a GT object was detected, and, on the other hand, if a detection has a correspondence in the GT. However, dealing with an individual evaluation of a GT-detection pair is different than dealing with a set of images (and obviously a set of text objects). One needs more complex metrics that, similarly to the local rates, can measure both the quality aspect and the quantity natures of the detections.

Let $\mathscr{G} = (G_1, G_2, ..., G_{N_G})$ be the set of GT text boxes within a database, where N_G represents the total number of GT text boxes within the set of multiple images. Let TP be the number of true positives (GT objects that were detected), computed as the sum of all matched objects in $\mathscr{G}$:

$$TP = \sum_{i=1}^{N_G} (Gmatch_i = 1), \tag{3.31}$$

Similarly, we count the number of false positives, FP, computed as the sum of objects in $\mathscr{D}$ that have no correspondence in $\mathscr{G}$:

$$FP = \sum_{j=1}^{N_D} (Dmatch_j = 0) \tag{3.32}$$

If we deal with a *many-to-one* detection, as detailed in Section 3.4.1, we split the detection box area in several detections that are assigned to the involved GT objects. Hence, an accuracy value is computed for each matched GT object (i.e. to each of the TP GT objects). The number of total detections is then the sum of true positives and false positives $TP + FP$. Based on the coverage and accuracy rates we derive two global scores. We define the global recall value R_G as the sum of all coverage values normalized by the total number of GT objects N_G:

$$R_G = \frac{\sum_{i=1}^{N_G} Cov_i}{N_G} \tag{3.33}$$

a: Two of the four objects fully detected.

b: All objects detected half.

c: One of the three objects fully detected. Two false positives.

d: All objects detected one third.

Fig. 3.17: Four examples illustrating GT objects with red rectangles and detections with green plain rectangles: (a)-(b) two examples for which recall $R_G = 0.5$; (c)-(d) two examples for which precision $P_G = 0.33$.

The global precision value P_G is computed as the ratio between the sum of accuracy measures and the total number of detections $TP + FP$:

$$P_G = \frac{\sum_{i=1}^{N_G} Acc_i}{TP + FP} \tag{3.34}$$

Although these two indicators give an overview on the performance of a set of detections, individually, they still do not provide sufficient information. As first stated by the authors in [Wolf and Jolion, 2006], it is important to differentiate the quantity aspect of a detection (*"how many GT objects/false alarms have been detected?"*) from its quality aspect (*"how accurate is the detection of the objects?"*). Figure 3.17 illustrates the importance of this distinction. One can observe that the same Recall (Figures 3.17a and 3.17b) and Precision (Figures 3.17c and 3.17d) scores can correspond to different detection outputs. Intuitively, it is then hard to correctly evaluate a detection characteristic through one value, hence we need to separately evaluate the quantity and quality properties.

Example. Figures 3.17a and 3.17b show how two different sets of detections can lead to the same global Recall $R_G = 0.5$. In the upper-left image only two out of four GT text objects are matched: $Cov_{[FIRE]} = Cov_{[EXIT]} = 1$ and $Cov_{[CLEAN]} = Cov_{[Keep]} = 0$. In the upper-right image on the other hand, the four GT objects have half of their surface covered: $Cov_{[FIRE]} = Cov_{[EXIT]} = Cov_{[CLEAN]} = Cov_{[Keep]} = 0.5$.

In the same way, Figures 3.17c and 3.17d illustrate that two distinct detection configurations can produce the same Precision score $P_G = 0.33$. The lower-left example shows a case where one out

of three detections has a correspondence in the GT, namely one perfect object level detection and two false positives: $Acc_{[NATURAL]} = 1$, $Acc_{[HISTORY]} = Acc_{[MUSEUM]} = 0$ and $FP = 2$. In the lower-right image, the three GT text objects are matched with the accuracy values: $Acc_{[NATURAL]} = Acc_{[HISTORY]} = Acc_{[MUSEUM]} = 0.33$.

We will further show that we can decompose each global metric into two separate quality and quantity components. Let us rewrite the global Recall R_G as the product of two terms:

$$R_G = \frac{\sum_{i=1}^{N_G} Cov_i}{N_G} = \frac{TP}{N_G} \cdot \frac{\sum_{i=1}^{N_G} Cov_i}{TP} \tag{3.35}$$

The left term of the product represents the ratio between the number of true positives and the total number of GT objects. We interpret this ratio as the quantity Recall R_{quant}, as it accurately describes the percentage of detected GT objects, regardless of their coverage:

$$R_{quant} = \frac{TP}{N_G}, \tag{3.36}$$

The second term is get by averaging all coverage rates of the detected GT objects. Intuitively, we can denote this proportion as the quality Recall, R_{qual}, as it characterizes the mean of covered surface of the GT:

$$R_{qual} = \frac{\sum_{i=1}^{N_G} Cov_i}{TP} \tag{3.37}$$

By applying the same reasoning, we obtain the following decomposition of the global Precision P_G:

$$P_G = \frac{\sum_{i=1}^{N_G} Acc_i}{TP + FP} = \frac{TP}{TP + FP} \cdot \frac{\sum_{i=1}^{N_G} Acc_i}{TP} \tag{3.38}$$

Here again, the left term of the product provides an insight on the percentage of detections that have a correspondence in the GT. Consequently, we call this measure the quantity precision P_{quant}:

$$P_{quant} = \frac{TP}{TP + FP}, \tag{3.39}$$

Inversely, the right term computes the accuracy average obtained from the matching of the detection set and the GT. This ratio will then be referred to as the Precision quality P_{qual}:

$$P_{qual} = \frac{\sum_{i=1}^{N_G} Acc_i}{TP}, \tag{3.40}$$

The majority of evaluation protocols in the literature propose an overall single metric used to judge the entire performance of a detector. The need of ranking different text detectors justifies the importance of such a measurement. In this work, we use the F-Score metric as the overall performance of a system, defined as the harmonic mean of recall and precision values:

$$F_G = \frac{2 \cdot R_G \cdot P_G}{R_G + P_G} \tag{3.41}$$

Fig. 3.18: A set of four images; GT objects are bounded by a red rectangle, green rectangles represent the detections.

Tab. 3.1: Quantity, quality and global scores for each individual image, as well as for the entire image set in Figure 3.18.

Figure	N_G/TP/FP	R_{quant}	P_{quant}	R_{qual}	P_{qual}	R_G	P_G	F_G
3.18a	2/2/0	1	1	0.66	0.74	0.66	0.74	0.69
3.18b	15/11/0	0.73	1	0.86	0.92	0.63	0.92	0.75
3.18c	4/2/5	0.5	0.28	1	1	0.5	0.28	0.36
3.18d	1/1/2	1	0.33	1	1	1	0.33	0.5
Set	22/16/7	0.72	0.69	0.86	0.91	0.63	0.64	0.63

Example. Figure 3.18 illustrates a set of four images with their corresponding GT and detection boxes. The evaluation of these examples using the metrics proposed above, is summarized in Table 3.1: on one hand, we show the evaluation results for each image individually; on the other hand, we compute the same metrics for the whole set. In Figure 3.18a both GT objects were detected, which is reflected in R_{quant} and P_{quant} values. In Figure 3.18b only 73% of the 15 GT objects were detected with a coverage mean indicated by the quality metric $R_{qual} = 0.86$, whose value suggests that the matched GT objects were not all perfectly covered. Figure 3.18c is a good example of the precision complexity of the detection set. Only 28% of the detection boxes (2 out of 7) are valid detections, while the rest are false positives. The valid detections are however within the bounds of the targeted GT objects which justifies the precision quality value $P_{qual} = 1$.

a: with rectangles **b:** with masks

Fig. 3.19: Different shapes for GT annotation (red)

3.5 Extension to *any-form* text annotation evaluation

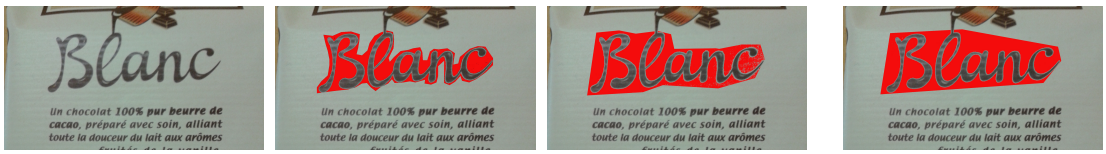
In this section we show how the EVALTEX framework can be extended to the evaluation of detection results having an irregular text annotation representation. The objective here is, first, to point out the disadvantages of using the bounding box annotation for text objects, and then to show what are the adjustments needed to evaluate text represented by irregular shapes. One of the advantages of EVALTEX consists in its ability of managing rectangular text objects even when dealing with "difficult" text strings, such as tilted, curved, circular *etc.* (see Figure 5.13 in Section 5.2 for examples) through the filtering process that can discard "unwanted" matched GT objects. Nonetheless, the rectangular representation still presents several limitations:

- tilted and curved text strings cannot be precisely located by rectangular bounding boxes;
- overlapping GT objects may mislead the matching process; in such cases, the filtering process invoked by EVALTEX cannot always assure a complete removal of invalid GT objects from the matching;
- the surface of a rectangular text box does not necessarily correspond to the true surface of the enclosing text string, which can severely distort the quality scores.

In order to avoid the situations mentioned above, we propose modifying the EVALTEX evaluation framework so that it can handle an irregular text representation of the GT. In order to use this annotation, detectors should also be able to produce precise estimations of text boundaries. Using a more accurate text annotation many detection ambiguities can be removed. In the following, we will refer to any of these irregular representations as *masks*.

3.5.1 GT annotation and representation

The interest of using masks rather than rectangles is to represent text strings, not only in horizontal or vertical configurations, but also tilted, circular, curved or in perspective. In such cases, the rectangular representation might disturb the matching process: a detection can involuntary match a GT object due to its varying direction (inclined, curved, circular). Such a situation is depicted in Figure 3.19a. Whenever the word "Pago" is matched by a precise detection, a truthful protocol that considers *many-to-one* mappings, could also match involuntary all text strings that intersect the object in the GT, namely the words "SINCE", "1889", "PREMIUM" and "FRUIT". Our method proposes a procedure, described in



a: Original **b** **c** **d**

Fig. 3.20: Different mask annotations (pixels within the red contour) for the word “Blanc”.

Section 3.3.3, to discard “unlikely” matched rectangular GT objects. However, for texts that are neither horizontal, nor vertical, typically text of urban scenes, a representation using free-form masks is a more convenient choice than rectangular ones. Figure 3.19b shows an example of such a mask-based annotations, which clearly prevents any intersections or inclusions between the GT objects.

The EvaLTex protocol considers a mask as a set of pixels situated inside the contour of a text object. Annotating text with irregular masks is a more laborious task than the bounding box labeling as it requires more than four points to represent the text areas. It also implies a higher level of subjectivity during the annotation process. Depending on the desired level of contour faithfulness of a text, different mask annotations can be formed for the same GT object. This is illustrated for the word “Blanc” in Figure 3.20.

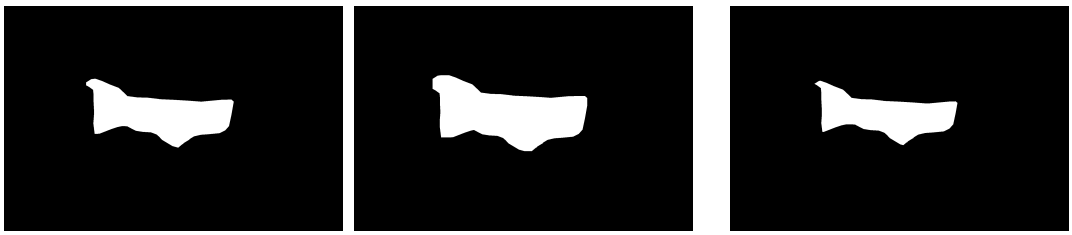
The irregular mask annotation disables the use of the region tag as it was introduced in Section 3.2 and further explained in Section 3.4.1. When dealing with rectangular boxes, the regions are generated automatically based on the coordinates of the GT objects. Consequently, a region is the bounding box of several “smaller” boxes. Thus, when masks are annotated irregularly, regions cannot be generated automatically. One possible solution would be, for a given set of GT objects labeled with the same region tag, to provide a region mask for each possible configuration (similar to the cases depicted in Figure 3.12). In practice, implementing this approach would require a laborious work, especially when dealing with larger sets of GT objects. However, if text areas are annotated following some well-defined rules, then the text region features can be used as described in this manuscript. Other examples of well-defined annotation representations besides the bounding boxes, can be inclined boxes, various polygons, ellipses *etc.*

3.5.2 Performance evaluation using masks

The evaluation principle using masks behind EvaLTex remains close, regardless the text representation. Hence, by using masks, we only need to adopt the following changes:

- additionally to the text representation using the bounding boxes (four coordinates) we add a mask representation to each GT object;
- the extension and reduction of a GT object (Equations (3.13) and (3.14)), are computed using dilation and erosion morphological operations on the text masks;
- *many-to-one* scenarios are handled without the region property.

The mask annotation does not change the computation of local quality measurements. However, the enlargement and reduction of the GT object discussed in Section 3.4.1 needs to be changed. Hence, in order to overcome the subjectivity of the GT annotation and the numerous ways algorithms can produce



a: true mask **b:** dilated mask **c:** eroded mask

Fig. 3.21: Masks for the word "Pago" illustrated in Figure 3.19

detection masks, we vary the mask area size by using dilations to produce accuracy scores and erosions to compute coverage scores. Therefore, when a detection does not perfectly match the GT mask, it is compared to the dilated and eroded GT mask. This ensures that small annotation variations do not affect the recall and precision scores. The two morphological operations correspond to the extended and reduced GT boxes computed according to Equations (3.13) and (3.14). The dilated and eroded masks corresponding to the GT object "Pago" are depicted in Figure 3.21.

Let $\mathcal{M}(G_i)$ be the GT text mask of G_i and H a $(2m_e) \times (2m_e)$ square structuring element, where m_e is the margin error defined in Equation 3.12. The margin error is computed based on the size of the rectangular box bounding the mask. We then define $\mathcal{M}_e(G_i)$ and $\mathcal{M}_r(G_i)$ as the extended and the reduced text masks of $\mathcal{M}(G_i)$ respectively and given by:

$$\mathcal{M}_e(G_i) = \mathcal{M}(G_i) \oplus H \tag{3.42}$$

$$\mathcal{M}_r(G_i) = \mathcal{M}(G_i) \ominus H, \tag{3.43}$$

where $\oplus$ and $\ominus$ represent the dilation and erosion morphological operations. The equations used to compute *one-to-one*, *one-to-many*, *many-to-one* and *many-to-many* detections correspond to those presented in Section 3.4.1. Finally, the performance measurements computed over a dataset are computed accordingly to the quality, quantity and global metrics described in Section 3.4.2. Our protocol can then be straightforwardly adapted to any kind of shapes for annotating the GT and representing the detections.

Note. Computing the margin error with respect to the rectangular bounding box of a curved text is not always a good approximation of the size of the structuring element used for the morphological operations. A text with a small character height which follows an arc form will produce a large margin error. Then, when eroding the text mask it may happen that the latter becomes completely erased (or fragmented) because the margin error is larger than the characters height. A possible solution would then be to set the margin error with respect to the distance from the contour of the mask to its skeleton.

Figure 3.19 illustrates a set of detection examples where text is annotated with masks. Five detections are attributed to the seven GT objects: one detection is matched to the words "SINCE" and "1888" (*many-to-one*); one detection is matched to the word "Pago" (*one-to-one*); one detection corresponds to the words "PREMIUM" and "FRUIT" (*many-to-one*); two more detections are matched individually to two GT objects, respectively "POIRE" and "PERA" (*one-to-one*). The local evaluation of the seven GT objects leads to the following coverage and accuracy values: $Cov_{[SINCE]} = 0.96$, $Acc_{[SINCE]} = 0.90$; $Cov_{[1888]} = 0.99$, $Acc_{[1888]} = 0.89$; $Cov_{[Pago]} = 0.94$, $Acc_{[Pago]} = 0.99$; $Cov_{[PREMIUM]} = 0.99$,

$Acc_{[PREMIUM]} = 0.92$; $Cov_{[FRUIT]} = 0.39$, $Acc_{[FRUIT]} = 0.93$; $Cov_{[POIRE]} = 0.99$, $Acc_{[POIRE]} = 0.95$; $Cov_{[PERA]} = 0.97$, $Acc_{[PERA]} = 1$. We can observe that due to the nonuse of the region tag, the two *many-to-one* mappings contribute to a diminution of accuracy values for the words "SINCE", "1888", "PREMIUM" and "POIRE". The final scores, corresponding to the quality, quantity and global recall and precision scores are: $R_{quant} = 1$, $P_{quant} = 1$; $R_{qual} = 0.89$, $P_{qual} = 0.94$; $R_G = 0.89$, $P_G = 0.94$.

Fig. 3.22: Example of a mask detection: GT objects are shown in plain red masks and detections with green contour line.

3.6 Conclusion

In this chapter we have presented a new evaluation protocol, EvaLTex, designed to estimate the performance of a text detection method. This protocol comes as smart solution to many of the existing problems that current evaluation protocols cannot deal with. First of all, EvaLTex solves the GT annotation issues, as it can handle both well-defined and irregular text representation. Hence, the protocol matchings strategy and metrics can be used in the same manner when text is bounded with rectangular boxes but also with texts that have a free-form representation (see Section 3.5), that we generically referred to as masks. In order to solve the granularity problems, we proposed in Section 3.2 the use an additional annotation level, the region tag, that allows detectors that provide for example, word-level detections and those providing line-level detections to be scores equally, without penalizing any of the detectors.

When rectangular GT annotations are used, the EvaLTex invokes a filtering procedure that filters and validates the matched GT objects, presented in Section 3.3.3. Rectangular boxes are not the best way of bounding texts that are inclined or curved because often a detection targeting a specific GT object can be erroneously matched to all GT objects that have an intersection with that GT object. Hence, the filtering procedure predicts the intention of detections.

In Section 3.4.1 we show that the proposed framework identifies and deals with all matching scenarios, including *one-to-one*, *one-to-many*, *many-to-one* but also *many-to-many*. We adapt two local measurements, coverage and accuracy, for each of this matching type. The evaluation protocol was developed to allow *many-to-one* detections and penalize them only if the detections exceeds the region area formed by the matched GT objects. Moreover, the protocol penalizes *one-to-many* detections but accordingly to the detected surface. Finally, in Section 3.4.2 we propose, for the overall evaluation of a set of detections, a series of six measurements: two that define the quality nature of the detections, two that capture the quantity aspect of the detections and two global ones.

In Chapter 5 we conduct a series of experiments which validate our evaluation protocol and point out the advantages of using EvaLTex with respect to other protocols in the literature.

4 Visual evaluation comprehension throughout histogram representation

Contents

In the previous chapter we have described an effective tool for evaluating the performance of a text detector by providing both quality and quantity global scores. However, to fully interpret the performance of a text detector, we also need a visual tool that characterizes the whole behavior of this detector. In this chapter we introduce an alternative way of capturing the quantity-quality aspects of a detector's performance throughout histogram representation. Moreover, based on this representation, we derive a second set of global scores computed using histogram distances. To do so, we use of the well known Earth Mover's Distance.

4.1 Context

In order to design an elaborate evaluation protocol one needs, not only to develop a reliable set of scores, but also to propose a visual tool to better understand the complexity of a detector and its results. We have discussed in Section 3.4.2 the importance of differentiating the quantity nature of a detection from its quality one. For that we have proposed to separate the global Precision and Recall scores into two sets: a qualitative set composed of R_{qual} and P_{qual}, and a quantitative one consisting of R_{quant} and P_{quant}. But when comparing two detectors only based on these scores, we can only determine which of them performs better, but not why. For example, one might be interested to know which algorithm produced more false positives or which one detected more GT objects entirely, instead of only partially. This kind of information cannot then be retrieved by only looking at the global scores. Consequently, the set of four quality and quantity metrics (R_{qual}, P_{qual}, R_{quant} and P_{quant}) provides a larger view on the characteristics of a detection than R_G and P_G. This is also true for cases when we want to analyze the performance of a single detector. The authors in [Wolf and Jolion, 2006] proposed a set of performance graphs based on ROC curves to illustrate the entire behavior of a detection algorithm, including its quality and quantity natures. The method generates two graphs by varying the two quality area constraints (t_r and t_p) presented in Section 2.5.5, over a wide range of values. The graph representation illustrates then how many objects (both from the GT and the detection set) respect the overlapping area constraints imposed by a given pair of thresholds (t_r, t_p). By varying these thresholds we can then have a plot of the

performance of a detector. The problem of this approach is that it is not straightforward. First, a quality measure is computed and then thresholded to obtain a binary score. Next, the threshold is varied and all objects having a local measure equal to 1 are counted and used to form the ROC curve. The obtained plots are also ambiguous and difficult to read as they rely on two parameters (t_r and t_p). The area under the curve obtained by varying these constraints is then used to represent the overall Recall and Precision measures. This is equivalent to averaging the sum of all object level measurements computed over all possible constraint values.

In this chapter we propose an alternative to the ROC curves and the AUC metrics. To capture the complexity of a set of detections we characterize them throughout histogram representations and use as metrics the distance between histograms [Calarasanu et al., 2015]. The proposed idea is illustrated in Figure 4.1. First, local measures, computed at object level, are quantified into quality histograms. Next, these histograms are compared to an optimal one using a distance to provide a final score. This approach is independent of the object representation and can be applied to rectangular or inclined bounding boxes or even free-form masks. The contributions of this method are two-fold. On one hand, it provides a practical and intuitive visualization of detection results. On the other hand, these histograms can also be used to compute performance scores necessary for a global evaluation or comparison between different detectors.

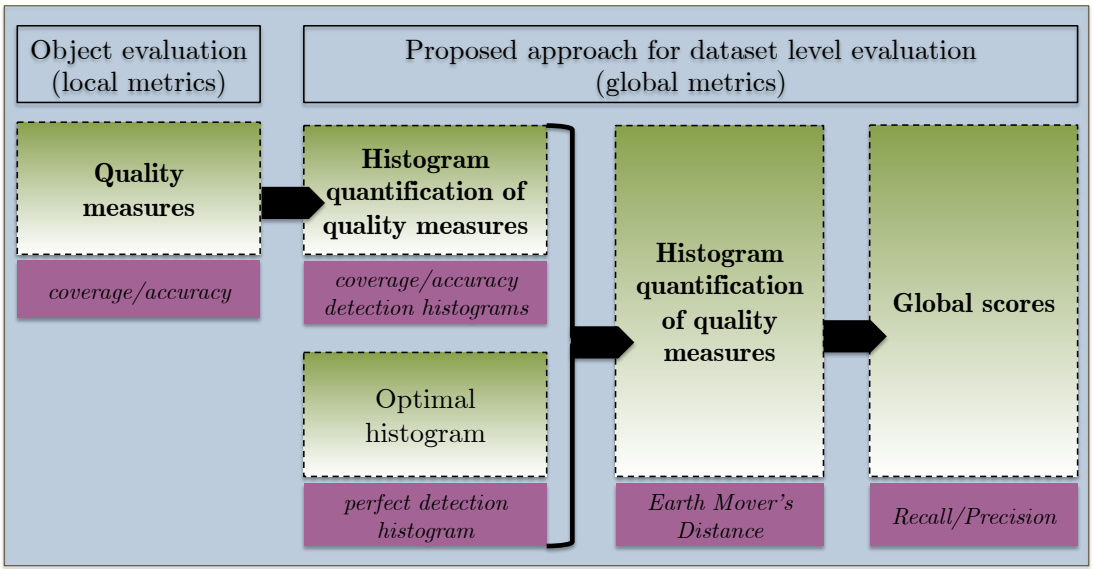


Fig. 4.1: Workflow of the histogram-based evaluation framework.

4.2 Histogram representation

Because histograms are graphical representations of frequency distributions over a set of data, they can be also seen as convenient tools to represent simultaneously the quality and quantity aspects of a set of detections:

- the quality aspect can be described by the histogram's bin: each bin corresponds to a coverage (or accuracy) interval;

- the detection quantity feature can be represented by the bin values: for example, the bin value counts how many GT objects have a coverage (or accuracy) value that belongs to that bin's interval.

This is illustrated in Figure 4.2. Let us consider a $1D$ finite valued function f that contains values

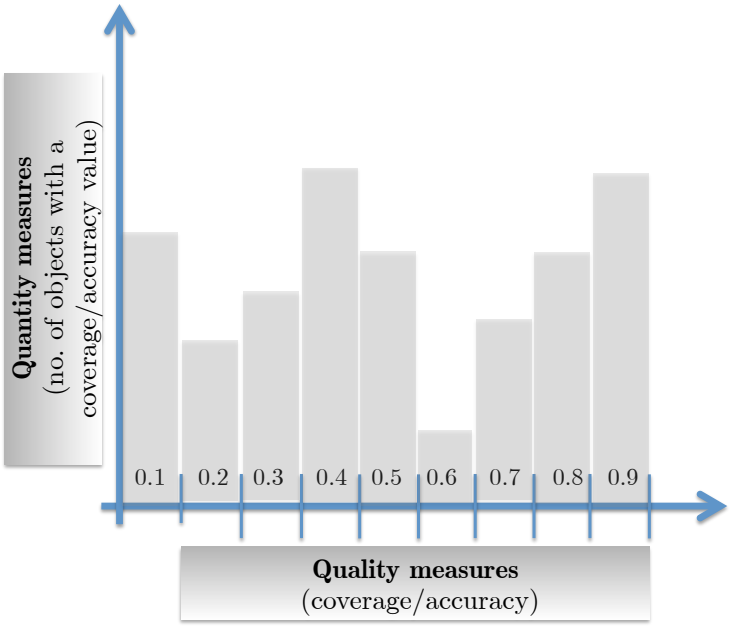


Fig. 4.2: Histogram representation of a detection set.

$f(j) \in [0,1]$, $j = 1,\ldots,n$. Its quantified histogram into B intervals (bins) is a $1D$ numerical function h defined as:

$$h(b) = \begin{cases} \sum_{j=1}^{n} \left\{ f(j) \in \left[\frac{b}{B}, \frac{b+1}{B} \right[\right\} & \text{if } b = 0,\ldots,B-2 \\ \sum_{j=1}^{n} \left\{ f(j) \in \left[\frac{b}{B}, \frac{b+1}{B} \right] \right\} & \text{if } b = B-1 \end{cases} \tag{4.1}$$

The EVALTEX evaluation protocol described in Chapter 3 provides two sets of local quality scores, namely for coverage and accuracy values. These sets can then be quantified by using two detection histograms as it will be explained in the following. Let us consider f_{Cov} the set of local coverage scores, computed, for example, using the EVALTEX protocol. We then derive the coverage histogram h_{Cov} as:

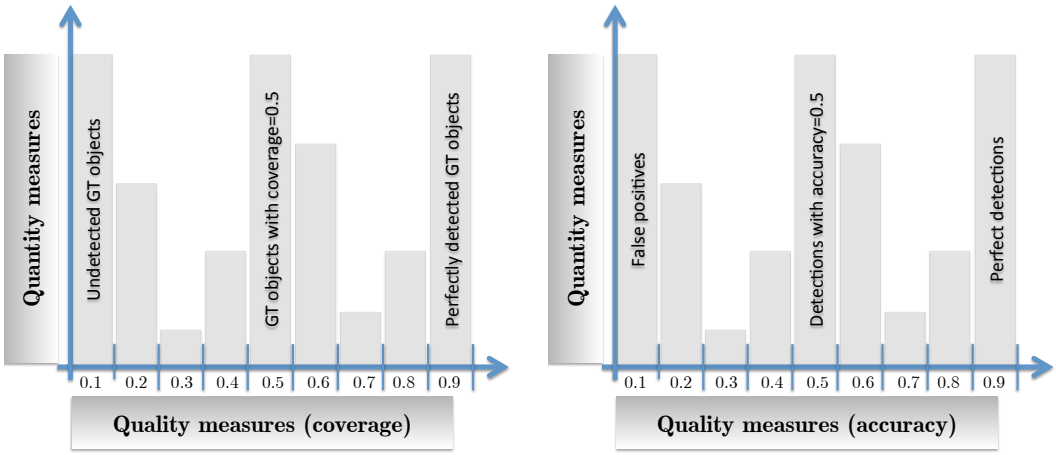
$$h_{Cov}(b) = \begin{cases} \sum_{i=1}^{N_G} \left\{ f_{Cov}(i) \in \left[\frac{b}{B}, \frac{b+1}{B} \right[\right\} & \text{if } b = 0,\ldots,B-2 \\ \sum_{i=1}^{N_G} \left\{ f_{Cov}(i) \in \left[\frac{b}{B}, \frac{b+1}{B} \right] \right\} & \text{if } b = B-1 \end{cases} \tag{4.2}$$

Similarly, let us consider f_{Acc} the set of local accuracy scores. We then derive the accuracy histogram h_{Acc} as:

$$h_{Acc}(b) = \begin{cases} \sum_{j=1}^{TP+FP} \left\{ f_{Acc}(j) \in \left[\frac{b}{B}, \frac{b+1}{B} \right[\right\} & \text{if } b = 0,\ldots,B-2 \\ \sum_{j=1}^{TP+FP} \left\{ f_{Acc}(j) \in \left[\frac{b}{B}, \frac{b+1}{B} \right] \right\} & \text{if } b = B-1 \end{cases} \tag{4.3}$$

Figure 4.3 illustrates the advantage of using histograms to represent a set of quality measures. In the case of coverage values (Figure 4.3a), the histogram provides at a glance the following rates of:

- *undetected objects* (GT objects with coverage score equal to 0) captured in the first bin,



a: Coverage histogram **b:** Accuracy histogram

Fig. 4.3: Quality histograms

- *partial matches* (GT objects with coverage values belonging to the interval $]0,1[$) are found in all bins but the first and the last ones,
- *perfect matches* (GT objects with coverage values equal to 1) captured in the last bin of the histogram.

Similarly, the accuracy histogram (Figure 4.3b) intuitively provides the following proportions of:

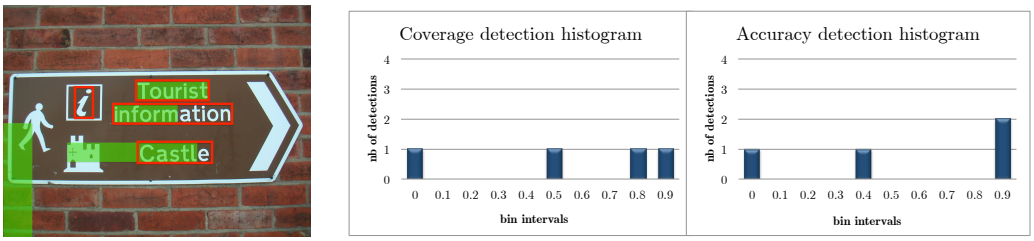
- *false positives* (detections with no correspondence in the GT) included in the first bin,
- *partial detections* (detections with accuracy values in the interval $]0,1[$) captured in all bins but the first and the last ones,
- *perfect detections* (detections with an accuracy equal to 1) found in the last bin of the histogram.

Note. The histograms in Figure 4.3 contain 10 bins which can misrepresent the exactness of the quality values. In this case, for example, all accuracy values in the interval $[0,0.1[$ will be counted as 0. This means that even if a detection has an accuracy equal to 0.05 it will still be counted as a false positive. We emphasize here the fact that, for a precise representation of local scores, the number of bins needs to be sufficiently high. However, for visual purposes exclusively, 10 bin histograms are used to illustrate different examples throughout this manuscript.

Example. The detection example in Figure 4.4a illustrates the case of four GT objects ("*i*", "Tourist", "information" and "Castle") and four detections, among which, one is a false positive. In this example, using the EvALTEx protocol we get the coverage scores $\{0.0, 0.55, 0.8, 1.0\}$ and the accuracy scores $\{0.0, 0.45, 1.0, 1.0\}$. Their representation using histograms with $B = 10$ bins is given in Figures 4.4b and 4.4c.

4.3 Histogram distances for performance evaluation

A histogram is not only a powerful tool for characterizing the whole nature of a detection, but also an instrument for computing performance scores as it will be described in the following. Until now, we have



a b: h_{Cov} c: h_{Acc}

Fig. 4.4: Detections in an image and the corresponding (b) coverage histogram and (c) accuracy histogram.

shown that we can populate a histogram with quality measures (coverage and accuracy values). We now want to compare two quality histograms and quantify their difference into a score. To simply evaluate a detection algorithm, this comparison can be made by computing the distance between its quality (or quantity) histogram and a reference one. The advantage of using the histogram distance is that the lower it is, the higher the similarity between the histograms, which finally leads to a performance score.

Let us from now on consider the normalized quality histogram $\widetilde{h}_{qual}$ of h_{qual} so that:

$$\sum_{b=0}^{B-1} \widetilde{h}_{qual}(b) = 1 \tag{4.4}$$

Consequently, let $\widetilde{h}_{Cov}$ and $\widetilde{h}_{Acc}$ be the normalized coverage and accuracy histograms (containing B bins), such that:

$$\sum_{b=0}^{B-1} \widetilde{h}_{Cov}(b) = 1 \tag{4.5}$$

$$\sum_{b=0}^{B-1} \widetilde{h}_{Acc}(b) = 1 \tag{4.6}$$

The histogram representation provides both a quantitative (*i.e.* values of bins) and a qualitative (*i.e.* number of bins) representation of the detection. A perfect algorithm should get maximal accuracy and coverage values for all detections, *e.g.* their corresponding histogram representation should have only one populated bin, the last one (for example, for $B = 10$, with all values belonging to $[0.9, 1]$). This histogram is referred to as the *optimal histogram*.

Let $\widetilde{h}_O$ be the normalized optimal histogram (containing B bins), whose all bins except the last one are empty, defined as:

$$\forall b \in [0, B-1], \widetilde{h_O}(b) = \begin{cases} 1 & \text{if } b = B-1 \\ 0 & \text{otherwise} \end{cases} \tag{4.7}$$

We then propose to measure a detector's performance as the distance between $\widetilde{h}_{Cov}$ (and $\widetilde{h}_{Acc}$) and the optimal histogram $\widetilde{h}_O$: the lower the distance, the higher the similarity between the histograms. Hence, we get two global detection performance measures corresponding to Recall and Precision.

Histogram distances. There are two main families of distances between histograms [Dubuisson, 2011]:

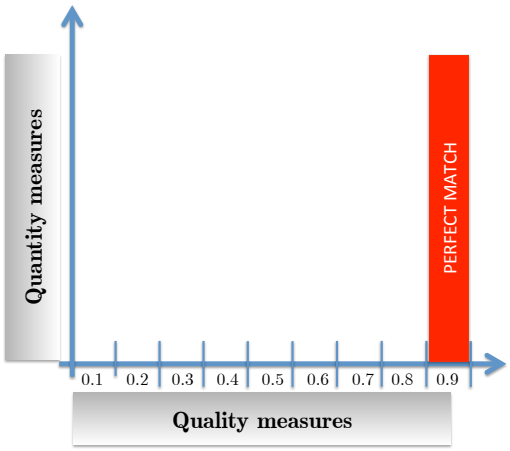


Fig. 4.5: Optimal histogram.

BIN-TO-BIN — Bin-to-bin distances only consider bin content (or size) and often make a linear combination of similarities measured between same bins of the two considered histograms (for example, the Euclidean distance). This assumes that histograms are aligned and have the same size;

CROSS-BIN — Cross-bin distances also consider the topology of histograms by integrating into the computation the distance between bins.

In any case, the topology of histograms is very important. For example, if we consider the case where all bins of $\widetilde{h}_{Cov}$ but one are empty (same reasoning for $\widetilde{h}_{Acc}$), then the Euclidean distance between $\widetilde{h}_{Cov}$ and $\widetilde{h}_{O}$ will give the value 0 if bin $\widetilde{h}_{Cov}(B-1) = 1$ (case of a perfect match), 1 otherwise (any case where $\widetilde{h}_{Cov}(b) = 1$, $b \neq B-1$). However, we would like the distance to be lower when the only populated bin of $\widetilde{h}_{Cov}$ is close to the last bin $B-1$, because this corresponds to better Recall scores on all the database. This is the reason why it is required to both consider the bin content and the distance between bins (as a kind of relationship between bins). Hence, a cross-bin distance is a better choice for computing the histogram dissimilarity in our context. The EMD has been chosen to compute the dissimilarity between a quality histogram and the optimal one for two main reasons [Rubner et al., 2000]:

1. it captures the perceptual dissimilarity better than other cross-bin distances;
2. it can be used as a true metric.

Two other cross-bin distances (see [Yan et al., 2007] for a review) have been proposed in the literature: the Quadratic-form (QF) distance [Pele and Werman, 2010] and the Diffusion distance [Ling and Okada, 2006a]. Let us consider h_1 and h_2 two histograms to compare. The Quadratic Form histogram distance between h_1 and h_2 is defined as:

$$QF^A(h_1, h_2) = \sqrt{(h_1 - h_2)^T A (h_1 - h_2)}, \tag{4.8}$$

where A is the bin-similarity matrix.

The Diffusion distance between two histograms was defined as a temperature field which uses the Gaussian pyramid to discretize the continuous diffusion process to make h_1 perfectly match h_2 and defined as:

$$K(h_1, h_2) = \sum_{l=0}^{L} \mid d_l(b) \mid, \tag{4.9}$$

where

$$d_0(x) = h_1(x) - h_2(x) \tag{4.10}$$

$$d_l(x) = [d_{l-1}(x) * \phi(x,\sigma)] \downarrow_2 \tag{4.11}$$

L represents the number of pyramid layers and σ the constant standard deviation for the Gaussian filter ϕ, while "$\downarrow_2$" defines a half size downsampling.

4.3.1 Earth Mover's Distance

The EMD, first introduced by Rubner *et al.* [Rubner et al., 2000], is a cross-bin distance function, based on the solution to the transportation problem that computes the dissimilarity between two signatures. In other words, this distance can be seen as the cost needed to transport piles of earth into a set of holes. Let $P = \{(p_i, w_{p_i})\}_{i=1}^{m}$ and $Q = \{(q_j, w_{q_j})\}_{j=1}^{n}$ be two signatures of sizes m and n, where p_i and q_j represent the position of the ith, respectively the jth, element and w_{p_i} and w_{q_j} their respective weight. The EMD searches for a flow $F = [f_{ij}]$ between p_i and q_j, that minimizes the cost to transform P into Q, so that:

$$COST(P,Q,F) = \sum_{i=1}^{m}\sum_{j=1}^{n} d_{ij} f_{ij}, \tag{4.12}$$

where d_{ij} is the ground distance between clusters p_i and q_j, while f_{ij} is the amount transported from one cluster to the other one. The cost minimization is done under the following flow constraints:

$$f_{ij} \geq 0, \quad i \in [1,m], \ j \in [1,n] \tag{4.13}$$

$$\sum_{j=1}^{n} f_{ij} \leq w_{p_i}, \quad i \in [1,m], \ j \in [1,n] \tag{4.14}$$

$$\sum_{i=1}^{m} f_{ij} \leq w_{q_j}, \quad i \in [1,m], \ j \in [1,n] \tag{4.15}$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} = \min(\sum_{i=1}^{m} w_{p_i}, \sum_{j=1}^{n} w_{q_j}), \quad i \in [1,m], \quad j \in [1,n] \tag{4.16}$$

Equation 4.13 ensures that the moved quantity is done in a precise order, namely from P to Q. Equation 4.14 ensures that the quantity sent by the clusters in P does not exceed their weights. Similarly, Equation 4.15 ensures that the quantity received by the clusters in Q does not exceed their capacities. Equation 4.16 requires to move the maximum quantity possible. After solving the transportation problem and retrieving the optimal flow F, the final EMD distance between the two signatures P and Q is computed as the cost function divided by the total flow:

$$EMD(P,Q) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} d_{ij} f_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}} \tag{4.17}$$

EMD between histograms. Histograms can be considered as a special kind of signatures where each bin of the histogram is treated as the signature's cluster, while the value of the bin (frequency) can be viewed as the signature's weight [Ling and Okada, 2006b]. For example, we can consider S_{Cov} and S_{Acc} two signatures of the same size B derived from the two normalized histograms $\widetilde{h}_{Cov}$ and $\widetilde{h}_{Acc}$, where

each bin b_i corresponds to a same cluster (intervals of the coverage/accuracy values between $[0,1]$), while the value of the bin, namely the frequency $\widetilde{h}_{qual}$, is the signature's weight :

$$S_{Cov} = \{(b_i, \widetilde{h}_{Cov}(b_i))\}_{i=1}^{B-1} \tag{4.18}$$

$$S_{Acc} = \{(b_i, \widetilde{h}_{Acc}(b_i))\}_{i=1}^{B-1} \tag{4.19}$$

Knowing that the two histograms $\widetilde{h}_{Cov}$ and $\widetilde{h}_{Acc}$ are normalized, we can then simplify the constraints that need to be satisfied for finding the optimal flow needed for the computation of the EMD between the two quality histograms and the optimal histogram, $EMD(\widetilde{h}_{Cov}, \widetilde{h}_O)$ and $EMD(\widetilde{h}_{Acc}, \widetilde{h}_O)$ respectively, in the following manner:

$$f_{ij} \geq 0, \quad i,j \in [1, B-1]$$

$$\sum_{j=1}^{B-1} f_{ij} = \widetilde{h}_{Cov}(b_i), \quad \sum_{j=1}^{B-1} f_{ij} = \widetilde{h}_{Acc}(b_i), \quad i,j \in [1, B-1],$$

$$\sum_{i=1}^{B-1} f_{ij} = \widetilde{h}_{O(b_j)}, \quad i,j \in [1, B-1]$$

EMD as a metric. In [Rubner et al., 2000], the authors proved that when the ground distance is a metric and the total weights of the two signatures are equal, the EMD is a true metric. Let us then consider d as the Euclidean distance between two clusters b_i and b_j, with $i,j \in [1,B]$, where the cluster b_i belongs to a quality histogram ($\widetilde{h}_{Cov}$ or $\widetilde{h}_{Acc}$) and b_j to the optimal histogram $\widetilde{h}_O$:

$$d = \sqrt{(b_i - b_j)^2} \tag{4.20}$$

Moreover, because the quality detection histograms were normalized, the total weights of the two derived signatures are 1 and hence equal (Equations 4.5 – 4.7). Consequently, we can use the EMD to compute global coverage and accuracy scores, that will be referred to as global Recall (R_{EMD}) and Precision (P_{EMD}). Since the EMD is a dissimilarity function (the closer the histograms, the lower the distance) the global scores, R_{EMD} and P_{EMD}, are computed as in [Wan, 2007]:

$$R_{EMD} = 1 - EMD(\widetilde{h}_{Cov}, \widetilde{h}_O) \tag{4.21}$$

$$P_{EMD} = 1 - EMD(\widetilde{h}_{Acc}, \widetilde{h}_O) \tag{4.22}$$

Example. The histogram representation of the text detection results in Figure 4.4 can be further used to compute the corresponding global Recall and Precision scores using Equations 4.21 and 4.22 which leads to the following values: $R_{EMD} = 0.625$ and $P_{EMD} = 0.6$, when the number of bins $B = 10$. These two scores can be compared with the ones obtained using the Equations 3.33 and 3.34 in Section 3.4.2: $R_G = 0.5875$, $P_G = 0.6$. We can see that the EMD is a predictable metric. Both approaches provide the same precision results. The Recall, however, gives a score difference of 0.038. When using histograms with a small number of bins (10 in this example), the EMD can overestimate the performance values if the quality values are situated within the bin intervals. In this example the difference is exclusively due to the coverage value of 0.55 attributed to the word "information", which is counted into the bin 0.6.

4.4 Conclusion

In this chapter we have introduced a novel and intuitive approach to represent text detection results and compute global performance scores. It is based on the histogram quantification of local quality scores, computed at object-level (coverage and accuracy). This approach visually captures at a glance the behavior of a detector: the rate of undetected GT objects and false positives, the percentage of partial matchings or the proportion of perfect matches and detections. Besides this straightforward interpretation of a set of detections, we have also proposed the derivation of global Recall and Precision scores using histogram distances. To do so, we introduced the notion of optimal histogram, which can describe quality detection scores. Thus, a distance between histograms distance is computed between the coverage and accuracy histograms and the optimal histogram. To do so, the EMD cross-bin distance was chosen due to its ability of handling histograms and due to its true metric property. The obtained distance is then used to derive two similarity measures: Recall and Precision. The primer interest of using histograms as an evaluation tool lies on the fact that they offer a complex visualization of the detection results. Moreover they can be used as an efficient tool to compare different text detectors by providing at a glance useful characteristics that could not be interpreted from global scores. The predictability of the obtained scores and the efficiency of using the histogram representation will be further discussed in Section 5.3.

5 Experimental tests

Contents

This chapter is dedicated to the experimental results that show the efficiency of the evaluation protocol proposed in this manuscript. In a first stage, we explore the advantages of EvaLTex *when using both a rectangular text representation or a mask annotation, described in Chapter 3. A comparison with current evaluation protocols such as* ICDAR'03, ICDAR'13 *and* DetEval *is done on the rectangular representation results, while the mask annotation results are evaluated on a smaller home-made database. A second set of tests is proposed to prove the usability of the histogram representation of text detection results. Moreover, we will show that the scores obtained using the* EMD *distance are similar to those computed with* EvaLTex.

The EvaLTex protocol was designed to evaluate text detection results in both natural and digital environments. The ICDAR'13 dataset has been chosen to conduct a series of experiments for two main reasons: the dataset is a reference in the text detection community and, due to the RRC online framework[1], numerous teams can evaluate and rank their detector performances in an acknowledge setting with all results made publicly available. Since EvaLTex uses a two-level GT, in our experiments, we use the same annotation as the one proposed by the ICDAR'13 RRC Challenge 2 dataset and in addition, we manually add the region tags (Section 3.2), to each GT object. The labeling is then checked for any region constraint violations. If the annotator grouped text objects that do not fulfill constraint in the Equation 3.1 (the sum of the text areas is larger than the non-textual area within the region formed by the GT objects labeled with the same tag), the annotator has to revise the region labeling. The experimental results using the rectangular representation are presented in Section 5.1. All ICDAR datasets have primarily focused on horizontal texts. To highlight the efficiency of EvaLTex when dealing with mask representation we need image samples containing more challenging texts. We then propose to evaluate different mask detections containing text which is tiled, curved, circular or even perspective deformed on a sample database detailed in Section 5.2. Finally, we prove the interest of using the histogram representation through a series of experiments conducted on the ICDAR'13 datasets in Section 5.3.

[1] http://rrc.cvc.uab.es/

5.1 Experimental results using the rectangular representation

To illustrate the advantages of EVALTEX, in this section we compare it to the most commonly used evaluation protocols in the text detection field that use a rectangular representation. A first set of score comparisons with ICDAR'03 (Section 5.1.1) and DETEVAL (Section 5.1.2) frameworks will be conducted based on a series of detection types with different scenarios depicted in Figures 5.1, 5.2 and 5.3. A more complex comparison is conducted with the ICDAR'13 evaluation protocol, consisting of analyzing both the matching strategy, for which the results were released, and the final performance scores.

a b c

Fig. 5.1: Examples of *one-to-one* detections: the GT (red rectangles) and the detection (solid green rectangles).

a b c

Fig. 5.2: Examples of *one-to-many* detections: the GT (red rectangles) and the detections (solid green rectangles).

a b c

Fig. 5.3: Examples of *many-to-one* detections: the GT (red rectangles) and the detections (solid green rectangles).

In Section 5.1.3, a quantitative comparison is done based on the results of several text detection methods that participated at ICDAR'13 competitions. For better differentiating these protocols we also compute

the final scores of all participants. Finally, in Section 5.1.4 we illustrate the principle of the two-level annotation with a set of examples and its impact on the performance scores.

5.1.1 Comparison to ICDAR'03/'05 evaluation protocol

The ICDAR'03 protocol, presented in Section 2.5.7, was one of the first to be extensively used for analyzing the performances of text detection system. It has been first introduced for the RRC during ICDAR 2003, and then re-used during ICDAR 2005. Although many protocols have been proposed since, this one is still used nowadays despite its drawbacks exposed in Section 2.5.7. In the following we will focus on the performance results obtained using the ICDAR'03 and EvaLTeX frameworks. When using the ICDAR'03 protocol, the *one-to-one* matches are scored accordingly to the true ratio between the intersection and the union surface of a detection-GT pair. When dealing with detection boxes that are inside the GT boundaries, the matching score should be close to the EvaLTeX coverage one. However, in some cases the two methods lead to significant score differences. In Figure 5.1a the coverage area, which is also equal to the Recall as the image only contains one GT object, gets a score of 0.74 by ICDAR'03 protocol, whereas EvaLTeX gives a value of 0.61. The 13% gap can be explained since the two methods apply different principles for the local measurements needed for computing the Recall: the ICDAR'03 protocol uses the Jaccard index, whereas EvaLTeX uses the coverage measure. A similar situation, with a larger score variation, is shown in Figure 5.1c.

The precision computed with ICDAR'03 protocol is not measured with respect to the detection surface, but with respect to the same matching score used for computing the Recall. Therefor, when evaluating a single *one-to-one* mapping the Precision is always equal to the Recall rate, as it can be seen in Table 5.1. This behavior can easily over or under penalize the performance of a detector.

Tab. 5.1: Score comparison between ICDAR'03 and EvaLTeX protocols based on the detection results (*one-to-one* matchings) of Figure 5.1.

	Figure 5.1a		Figure 5.1b		Figure 5.1c	
	ICDAR'03	EvaLTeX	ICDAR'03	EvaLTeX	ICDAR'03	EvaLTeX
Recall	0.74	0.61	0.64	0.63	0.7	0.55
Precision	0.74	1	0.64	1	0.7	1
$F-$Score	0.74	0.75	0.64	0.77	0.7	0.7

With ICDAR'03 protocol the Recall scores obtained on *one-to-many* matchings (see Figure 5.2) are 0.79, 0.73 and 0.69, as shown in Table 5.2. These scores however do not reflect, neither the union of the intersections between the GT objects and the detection, nor the coverage rate of the largest detection that matches the GT object. Moreover, the Recall scores do not take in account the fragmentation of the GT objects, as done by EvaLTeX. Our framework penalizes each of the three cases in Figure 5.2. In Figures 5.2a and 5.2b the true coverage value is 0.85 but is decreased to 0.50 due to the fragmentation penalty.

Another drawback of ICDAR'03 protocol is due to the use of the best match approach for the *many-to-one* cases, illustrated in Figure 5.3. It can be easily seen that when a detection covers more than one object the mapping procedure of ICDAR'03 is done with respect to a single GT object, by rejecting all other matched text boxes, making this protocol suitable only for detectors that are able to provide

Tab. 5.2: Score comparison between ICDAR′03 and EVALTEX protocols based on the detection results (*one-to-many* matchings) of Figure 5.2.

	Figure 5.2a		Figure 5.2b		Figure 5.2c	
	ICDAR′03	EVALTEX	ICDAR′03	EVALTEX	ICDAR′03	EVALTEX
Recall	0.79	0.50	0.73	0.50	0.69	0.54
Precision	0.51	1	0.64	1	0.62	1
$F-$Score	0.62	0.67	0.68	0.67	0.65	0.71

"unitary" (word) level results. EVALTEX protocol was designed such that all detected GT objects should be taken into consideration and hence scored. The logic behind this simply relies on the fact that having two GT objects detected should normally weight more than just having one. In this sense, EVALTEX is clearly a better choice for evaluating *many-to-one* matchings because none of the GT objects is dismissed, but all are counted and scored, unlike what is done with ICDAR′03 protocol.

Tab. 5.3: Score comparison between ICDAR′03 and EVALTEX protocols based on the detection results (*many-to-one* matchings) of Figure 5.3.

	Figure 5.3a		Figure 5.3b		Figure 5.3c	
	ICDAR′03	EVALTEX	ICDAR′03	EVALTEX	ICDAR′03	EVALTEX
Recall	0.35	1	0.54	1	0.61	1
Precision	0.39	1	0.77	1	0.65	1
$F-$Score	0.36	1	0.63	1	0.62	1

Based on all the aspects discussed above, we can conclude that ICDAR′03 protocol presents drawbacks that severely affect the evaluation accuracy of a detector. We mention here the equality of Recall and Precision obtained on *one-to-one* and *one-to-many* matchings or the best match approach used to evaluate *many-to-one* mappings. This comparison emphasizes the fact that our proposed evaluation method better characterizes the efficiency of a detector by, conversely to ICDAR′03 protocol, clearly differentiates Recall and Precision scores for all matchings, discriminates partial *one-to-one* from *one-to-many* mappings and provides a precise count of all GT objects detected in a *many-to-one* case.

5.1.2 Comparison to DETEVAL evaluation protocol

The DETEVAL tool, presented in Section 2.5.5, uses the object detection evaluation method proposed in [Wolf and Jolion, 2006]. This tool can be configured in different ways, depending on the chosen area thresholds. The framework can be configurable through eight parameters:

six parameters — representing the minimum recall (t_r) and precision (t_p) overlap areas between detection results and the GT for *one-to-one*, *one-to-many* and *many-to-one* cases;

one parameter — that permits or not the use of an additional border verification;

one parameter used as a threshold on the difference of the centers of two matching bounding boxes.

Depending on the parameter configuration, different results can be obtained using the DETEVAL framework. So, in order to cover as many comparisons as possible of DETEVAL and EVALTEX we imply the following three DETEVAL configurations that will be explained in the following: "relaxed", AUC metrics and default.

Tab. 5.4: *One-to-one* detection scores corresponding to Figure 5.1 using the "relaxed" DETEVAL (DETEVAL$_{rel}$), the AUC metrics of DETEVAL (DETEVAL$_{AUC}$) and EVALTEX.

	Figure 5.1a			Figure 5.1b			Figure 5.1c		
	DETEVAL$_{rel}$	DETEVAL$_{AUC}$	EVALTEX	DETEVAL$_{rel}$	DETEVAL$_{AUC}$	EVALTEX	DETEVAL$_{rel}$	DETEVAL$_{AUC}$	EVALTEX
Recall	1	0.78	0.61	1	0.76	0.63	1	0.72	0.55
Precision	1	0.78	1	1	0.76	1	1	0.72	1
$F-$Score	1	0.78	0.75	1	0.76	0.77	1	0.72	0.7

Tab. 5.5: *One-to-many* detection scores corresponding to Figure 5.2 using the "relaxed" DETEVAL (DETEVAL$_{rel}$), the AUC metrics of DETEVAL (DETEVAL$_{AUC}$) and EVALTEX.

	Figure 5.2a			Figure 5.2b			Figure 5.2c		
	DETEVAL$_{rel}$	DETEVAL$_{AUC}$	EVALTEX	DETEVAL$_{rel}$	DETEVAL$_{AUC}$	EVALTEX	DETEVAL$_{rel}$	DETEVAL$_{AUC}$	EVALTEX
Recall	0.8	0.71	0.50	0.8	0.78	0.50	0.8	0.76	0.54
Precision	0.8	0.69	1	0.8	0.76	1	0.8	0.74	1
$F-$Score	0.8	0.70	0.67	0.8	0.77	0.67	0.8	0.75	0.71

Tab. 5.6: *Many-to-one* detection scores corresponding to Figure 5.3 using the "relaxed" DETEVAL (DETEVAL$_{rel}$), the AUC metrics of DETEVAL (DETEVAL$_{AUC}$) and EVALTEX.

	Figure 5.3a			Figure 5.3b			Figure 5.3c		
	DETEVAL$_{rel}$	DETEVAL$_{AUC}$	EVALTEX	DETEVAL$_{rel}$	DETEVAL$_{AUC}$	EVALTEX	DETEVAL$_{rel}$	DETEVAL$_{AUC}$	EVALTEX
Recall	0.8	0.65	1	0.8	0.74	1	0.8	0.72	1
Precision	0.8	0.66	1	0.8	0.57	1	0.8	0.74	1
$F-$Score	0.8	0.65	1	0.8	0.64	1	0.8	0.73	1

"Relaxed" DETEVAL

In our experiments, we first evaluate the text detection results using a "relaxed" version of DETEVAL by disabling the minimum area coverage constraints. In such a case, we attempt a more fair comparison with EVALTEX, which by definition is less-restrictive than the common evaluation protocols and does not use any area thresholds. So, in order to bring closer the two evaluation approaches, we tune the DETEVAL system to obtain its "relaxed" version, such that:

- the recall and precision area thresholds are both set to $t_r = t_p = 0$: by this, similarly to EVALTEX, all matchings between the GT and a detection are considered as valid, regardless of their intersection surface.
- the center difference threshold is set to 1.

The command line used to produce the "relaxed" evaluation of DETEVAL is:

```
> evaldetection -p 0,0,0,0,0,0,0,1 det.xml gt.xml > res.xml
> readdeteval res.xml
```

We illustrate the behavior of "relaxed" DETEVAL protocol during *one-to-one*, *one-to-many* and *many-to-one* scenarios as this gives a good insight of this method's shortcomings and justifies once again our evaluation choices.

We rely on Figure 5.1 which shows some examples of partial *one-to-one* detections which, evaluated with "relaxed" DETEVAL, are granted with maximum recall values, as seen in Table 5.4. This raises again the questionability of having a fair comparison between detectors providing perfectly accurate detections and detectors that output partial ones. With our method, partial *one-to-one* matchings are scored according to their true intersection area (between the GT and detection boxes) which, in the current example, correctly penalizes the final recall scores. By doing so we defend once again our idea that a partial detection is still better than no detection especially if the text detections are used for a recognition stage.

All *many-to-one* cases, illustrated in Figure 5.2, obtain the same Recall, Precision and $F-$Scores equal to 0.8 when using the "relaxed" DETEVAL. The recurrence of this value suggests that all *one-to-many* detections are scored identically regardless of the matching areas between the GT and the detection set. On the opposite, EVALTEX keeps the Precision value constant to 1 as all detections are within the boundaries of the GT objects and applies to the true coverage area a fragmentation penalty. The EVALTEX scores are more representative because they include the count of all the detected surfaces of a GT but in the same time they reflect that the matching is not ideal, but fragmented.

Although the *one-to-many* detections depicted in Figure 5.3 match entirely all GT text objects, they are penalized by the "relaxed" DETEVAL, as seen in Table 5.6. Moreover, the penalty is applied to both Recall and Precision metrics which are set to the same constant value of 0.8 as in the case of *one-to-many* scenarios, independently of the number of matched GT text boxes. and does not penalize the *many-to-one* cases. Hence, even when using the most permissive configuration of DETEVAL, our method is able to evaluate more accurate detection results because the provided Recall score truly reflects the fact that the whole GT surface has been detected, contrary to "relaxed" DETEVAL for which this score could easily be interpreted by the fact that only 80% of the GT area has been matched, which is false.

By implying this set of comparisons, we showed that EVALTEX better describes the detection efficiency than DETEVAL protocol, even when the later was used with the configuration with the rules closest to EVALTEX ones.

AUC metrics

DETEVAL also integrates a set of new metrics to capture the complexity of the result given by a detection algorithm, by characterizing both its quality and quantity nature. Recall and Precision are computed over a range of 20 different area threshold values and then averaged to provide two overall metrics. These metrics correspond to the AUC graph obtained by ranging the area threshold. To produce the AUC scores we have relaxed the two area constraint by setting $t_r = t_p = 0$ and use the following command line:

```
> ./evalplots --tr-fix=0 --tp-fix=0 det.xml gt.xml
```

While these metrics solve the binary behavior of partial matchings of the "relaxed" DETEVAL, the Precision still tends to be equal with the Recall values when dealing with *one-to-one* cases (see Table 5.4) This is a problem because the two metrics, Recall and Precision, should characterize different aspects of a detection, as successfully shown by the scores provided by EVALTEX.

The same score similarity characteristics between Recall and Precision is produced when evaluating *one-to-many* scenarios (see Table 5.5).

The scores for the *many-to-one* matches (see Table 5.6) also present the same problem: the small difference between the Recall and Precision values corresponding to Figures 5.3a and 5.3c does not give a full understanding of how much of the GT boxes were detected *versus* how well the detection box covered the GT boxes. A higher difference between the two scores is obtained for detections of Figure 5.3b. Nonetheless, this still remains difficult to interpret.

The difference between the evaluation accuracy produced by AUC metrics and by EVALTEX is clear. The drawback of the AUC metrics, handled by EVALTEX, is the inability of separating the properties described by the Recall and Precision scores.

Default DETEVAL (ICDAR'11/13/15 evaluation protocols)

The default configuration of DETEVAL assumes that the Recall and Precision thresholds are set to $t_r = 0.8$ and $t_p = 0.4$ respectively. Hence, any detection having an accuracy value higher than 0.4 is considered correct, while any GT object for which the mapped surface is larger than 0.8 is considered as matched. This is the most used configuration of DETEVAL as it does not necessitate further tuning. Moreover, it has been already used during ICDAR 2011, 2013 and 2015 RRCs.

The evaluation method used during ICDAR 2013 RRC (*Challenge 1 & 2 - Text Localization*) is a re-implementation of the DETEVAL framework. As the organizers mentioned[2], there are "slight differences"

[2] http://rrc.cvc.uab.es/?ch=2&com=evaluation

between the scores obtained with ICDAR'13 and the ones obtained with DETEVAL due to an incomplete documentation of some heuristics in [Wolf and Jolion, 2006].

Comparison to ICDAR'13 protocol. The choice of comparing EVALTEX system to that of ICDAR'13 stands on two reasons. First, it is up-to-date and represents what is commonly done and admitted in text detection evaluation. Secondly, all results are publicly available through an interface[3] which provides for each image the final scores and matchings. The protocol uses the two area Precision and Recall thresholds, t_r and t_p, which are set to 0.8 and 0.4 respectively and which control the matching between the GT and the detections.

Its matching protocol assigns a lower weight to *one-to-many* matches, since the expected output is at the word level, while text-line level detections (*many-to-one* matches) are said not be penalized [Karatzas et al., 2013]. However, we will show later that this last "claim" is frequently violated and consequently causes misleading scores. In the following, we analyze the differences between EVALTEX and ICDAR'13 protocols based on the detection results of the *TextDetection* detector [Fabrizio et al., 2013]. These differences come from, on one hand, the matching strategy, as shown in Figures 5.4–5.7, and on the other hand, their corresponding global scores, presented in Tables 5.7–5.10. The comparison is illustrated for each type of matching.

Figures 5.4a and 5.4b illustrate a *one-to-one* case for which the Recall and the Precision scores are over-estimated by ICDAR metrics. First, although the detection misses the first letter of the word "AUSTRALIA", the Recall rate is set to 1 (Fig. 5.4a). Similarly, even if the area of the detected box for the word "moto" is considerably larger than the GT one, its Precision rate is 1. The ICDAR 2013 approach scores a GT text box with a binary Recall (1 or 0), depending on whether the area match ratio respects or not a threshold. This is not a correct evaluation, since it does not provide a good comparison between algorithms. For example, if an algorithm detects the whole word "AUSTRALIA", it would get the same score as the detection shown in Figure 5.4a leading to an unfair evaluation. Conversely, our metrics give a more precise and realistic evaluation because they take into account the real overlap match area, that provides a better comparison between different system outputs.

Tab. 5.7: Detection scores corresponding to Figure 5.4.

	Figure 5.4a		Figure 5.4b	
	ICDAR'13	EVALTEX	ICDAR'13	EVALTEX
Recall	1	0.9186	0.5	0.5
Precision	1	1	1	0.5919
$F-$Score	1	0.9575	0.6667	0.5421

[3] http://rrc.cvc.uab.es/?com=introduction

a

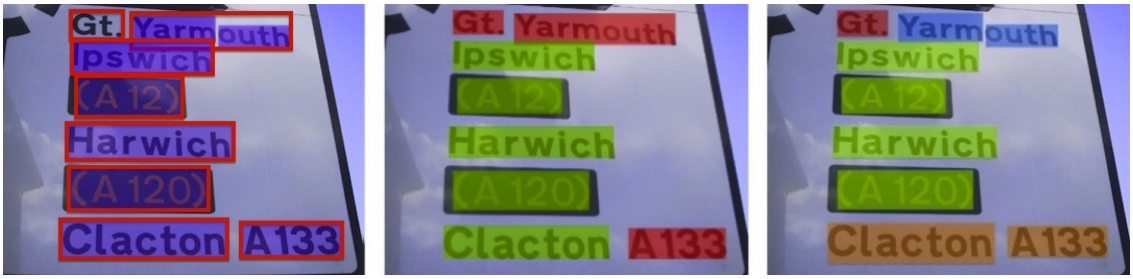
b

Fig. 5.4: *One-to-one* matching examples; left: GT (red rectangle) and the detection (solid purple rectangle); center (ICDAR 2013) and right (EvaLTex): mismatched GT objects (solid red rectangles), *one-to-one* matched GT areas (solid green rectangles), many-to-one matched GT areas (solid yellow rectangles), one-to-many matched GT areas (solid blue rectangles).

As shown in Figures 5.5a and 5.5b, the *one-to-many* case is not treated the same way for all images by ICDAR metrics. In Figure 5.5b, the word "POSTPAK" is detected by two boxes, both considered as correct matchings. In Figure 5.5a the same scenario occurs for the word "Yarmouth", but the two detected boxes are not considered as valid matches because in both cases the overlap matching area is too small. Moreover, the two detected boxes are considered as false positives, that unfairly penalizes the final scores. Firstly, it decreases the Recall rate by not matching the two detected boxes to the GT, and secondly, it decreases significantly the Precision rate due to the two detected boxes which are erroneously counted as false positives. On the contrary, our method correctly recognizes the *one-to-many* cases and matches the two detected boxes in both examples, but punishes the fragmented detection by penalizing the Recall, as seen in Section 3.3.

Tab. 5.8: Detection scores corresponding to Figure 5.5.

	Figure 5.5a		Figure 5.5b	
	ICDAR'13	EvaLTex	ICDAR'13	EvaLTex
Recall	0.625	0.8102	0.90	0.7806
Precision	0.7143	1	0.8667	1
$F-$Score	0.6667	0.8952	0.883	0.8768



a

b

Fig. 5.5: *One-to-many* matching examples; left: GT (red rectangle) and the detection (solid purple rectangle); center (ICDAR 2013) and right (EvaLTex): mismatched GT objects (solid red rectangles), *one-to-one* matched GT areas (solid green rectangles), many-to-one matched GT areas (solid yellow rectangles), one-to-many matched GT areas (solid blue rectangles).

Figures 5.6a and 5.6b show the problem of inconsistency during the *many-to-one* matching. In Figure 5.6a, the detection is at a line level. Only the second and last lines are correctly matched, while the other detected text lines are associated with the GT text box having the largest surface within that line ("unauthorized" in the first line, "Permit" in the third one and "operation" in the fourth one). The unmatched GT text boxes are considered as false positives ("No", "to", "work", "system", "in"). ICDAR metrics over punish the *many-to-one* matches and frequently considers them as *one-to-one*. On the contrary, our protocol correctly matches all text lines and leads to a Recall equal to 1. We have a similar problem when detection boxes cover a multi text-line (Figure 5.6b). The word "Roland" is matched by ICDAR protocol, while the two other words are discarded. Hence, their Recall is penalized, while their Precision is not. Our method considers all words detected, hence the Recall rate is set to 1. Nevertheless, we assign a low Precision rate, due to the presence of the logo in the left part of the detected box.

Tab. 5.9: Detection scores corresponding to Figure 5.6.

	Figure 5.6a		Figure 5.6b	
	ICDAR'13	EvaLTex	ICDAR'13	EvaLTex
Recall	0.6667	1	0.3333	1
Precision	1	1	1	0.6245
$F-$Score	0.8	1	0.5	0.7688

a

b

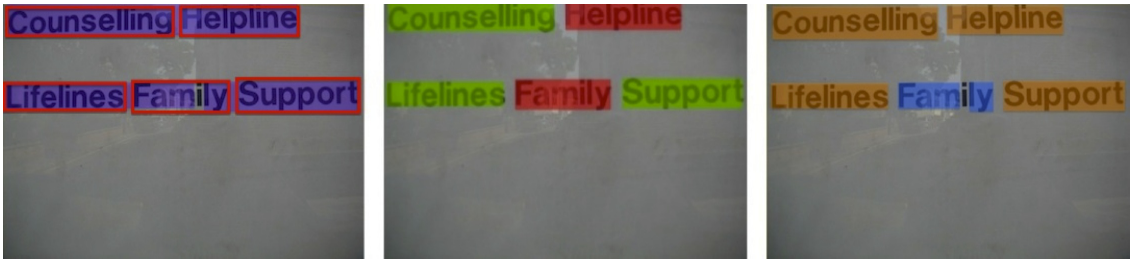
Fig. 5.6: *Many-to-one* matching examples; left: GT (red rectangle) and the detection (solid purple rectangle); center (ICDAR 2013) and right (EvaLTex): mismatched GT objects (solid red rectangles), *one-to-one* matched GT areas (solid green rectangles), many-to-one matched GT areas (solid yellow rectangles), one-to-many matched GT areas (solid blue rectangles).

Finally, the *many-to-many* case is illustrated in Figures 5.7a and 5.7b. The word "COLCHESTER" in Figure 5.7a corresponds to a *many-to-one* and a *one-to-many* match. Nevertheless, the ICDAR matching protocol rejects it and matches only the word "HEALTHY", whereas our algorithm validates both text boxes, but penalizes the Recall due to its split detection. If we look at the second line in Figure 5.7b we observe that the word "Family" is matched by two detections (*one-to-many*). Both detections involve a *many-to-one* case, the first one corresponding to words "Lifelines" and "Family", while the second one to words "Family" and "Support". The ICDAR matching algorithm considers as matched GT text boxes those containing the words "Lifelines" and "Support", and classifies the word "Family" as missed. This provides again an unfair comparison: if another localization algorithm would have completely missed the word "Family", then, both algorithms would have got the same scores, although the first detected 87% of the area of the "Family" GT text box.

Tab. 5.10: Detection scores corresponding to Figure 5.7 using the ICDAR′13 and EvaLTex protocols.

	Figure 5.7a		Figure 5.7b	
	ICDAR′13	EvaLTex	ICDAR′13	EvaLTex
Recall	0.6667	0.8404	0.6	0.9032
Precision	0.6667	1	1	1
$F-$Score	0.6667	0.9132	0.75	0.9491

a



b

Fig. 5.7: *Many-to-many* matching examples with scores; left: GT (red rectangle) and the detection (solid purple rectangle); center (ICDAR′13) and right (EVALTEX): mismatched GT objects (solid red rectangles), *one-to-one* matched GT areas (solid green rectangles), *many-to-one* matched GT areas (solid yellow rectangles), *one-to-many* matched GT areas (solid blue rectangles).

5.1.3 Quantitative results

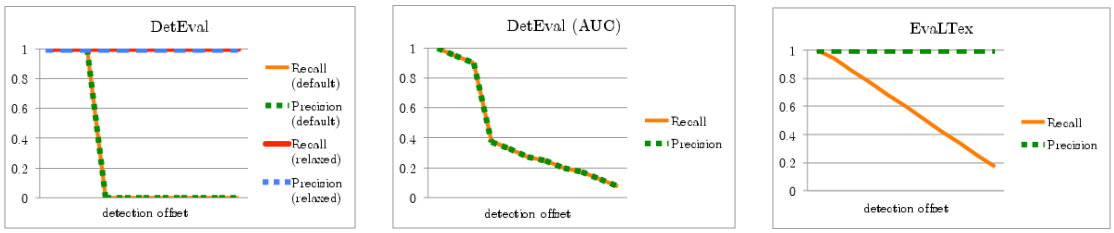
This new set of experiments will show the variety and hence the imbalance between all these protocols discussed in the previous section. A first experiment will simply target the comparison of evaluation results on single *one-to-one* matchings as they capture the best the weaknesses of each protocol. The second comparison focuses on the opposite: the examination of detection scores on a whole dataset of images.

Evaluation of partial detections

As already referred in the manuscript, the way *one-to-one* detections are being treated differs from one method to another one. To highlight these differences, we propose a simple experiment which consists in gradually decreasing the quality (coverage area) of a *one-to-one* detection with a GT text object (see Figure 5.8a), while maintaining a perfect accuracy level (equal to 1) and analyze the Recall and Precision measurements at each stage.

Figure 5.8b depicts the evolution of Recall and Precision scores given by the default and "relaxed" configurations of DETEVAL when dealing with partial *one-to-one* matchings. The "relaxed" DETEVAL maintains the two metrics at 1 despite the diminuation of the coverage area. This is the result of setting the area thresholds to 0. On the other hand, for the default DETEVAL, we can observe the steep drop of both metrics from one to zero. This describes a binary evaluation approach of *one-to-one* matchings, which depends on the Recall and Precision area parameters. Unfortunately, such behavior cannot correctly differentiate total from partial detections, regardless of the area threshold values.

a: A series of *one-to-one* detections.



b **c** **d**

Fig. 5.8: Recall and Precision plots for a series of *one-to-one* detections using different evaluation protocols; (a) the detection area is gradually reduced by an offset; (b) default DETEVAL (ICDAR'13) and relaxed DETEVAL; (c) DETEVAL (AUC); (d) EVALTEX.

In Figure 5.8c we illustrate the evaluation behavior when using the AUC metrics of DETEVAL. One can observe that the Recall and Precision plots consistently overlap during the whole set of the partial *one-to-one* detections. Since the detection coverage area always remains within the boundaries of the GT text, a good set of measurements should discriminate between the Recall and Precision values. In such a case, our method (Figure 5.8d), evaluates the Precision to 1, regardless of the partial detection. This is logical because the detection never exceeds the valid text area. On the contrary, the Recall score decreases linearly as a result of the progressive diminution of the coverage area between the detection and the GT boxes.

ICDAR 2013 Robust Reading Competition results

In this section we evaluate the detection results submitted by the ten participants at the ICDAR 2013 RRC (*Challenge 1* and *Challenge 2*) [Karatzas et al., 2013]. The results of these methods, originally published on the RRC 2013 competition website page [ICDAR, 2013], were later made available on the RRC 2015 website[4]. In Tables 5.11, 5.12 and 5.15 we provide a complete comparison of the performance results of all participants to *Challenge 2* on the RRC'13-SI dataset, while Tables 5.13, 5.14 and 5.16 provide the detection scores of the participants to *Challenge 1* on the RRC'13-BD database.

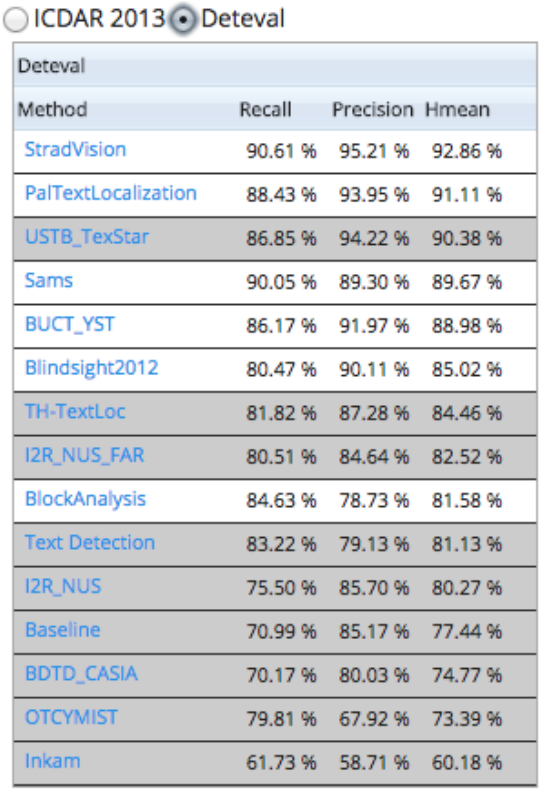
Note. Initially, the ICDAR'13 metrics were presented as being computed following the default configuration of DETEVAL protocol. However, there were "*slight differences*" between the performance results of ICDAR'13 and those obtained by DETEVAL due to a series of undocumented heuristics in [Wolf and Jolion, 2006], such as the penalization attributed to *many-to-one* cases or the order of evaluating certain matching scenarios. To fix this, the organizers "*implemented an alternative evaluation protocol which is consistent to the* DETEVAL *tool and takes into account all undocumented heuristics*". We will refer to this last protocol as ICDAR'13$_{\text{DetEval}}$. Figure 5.9 gives a screenshot taken from the ICDAR webpage showing the option of switching between these two protocols to provide the final performance results.

[4] http://rrc.cvc.uab.es/?com=introduction

The scores obtained with ICDAR'13 and ICDAR'13$_{\text{DetEval}}$ could not be reproduced with the DETEVAL tool. Hence, due to this score difference, we decided to provide the scores obtained by the following configurations of DETEVAL by directly running the following command lines:

1. Default DETEVAL (DETEVAL$_{default}$) using the framework's command line[5]
   ```
   > evaldetection det.xml gt.xml > res.xml
   ```
2. AUC metrics of DETEVAL (DETEVAL$_{\text{AUC}}$) using framework's command line
   ```
   > evalplots det.xml gt.xml > res.xml
   ```

We compare these four configurations (ICDAR'13 ,ICDAR'13$_{\text{DetEval}}$, DETEVAL$_{default}$ and DETEVAL$_{\text{AUC}}$), with the scores of ICDAR'03 and EVALTEX.


ICDAR 2013 Deteval

Deteval

Method	Recall	Precision	Hmean
StradVision	90.61 %	95.21 %	92.86 %
PalTextLocalization	88.43 %	93.95 %	91.11 %
USTB_TexStar	86.85 %	94.22 %	90.38 %
Sams	90.05 %	89.30 %	89.67 %
BUCT_YST	86.17 %	91.97 %	88.98 %
Blindsight2012	80.47 %	90.11 %	85.02 %
TH-TextLoc	81.82 %	87.28 %	84.46 %
I2R_NUS_FAR	80.51 %	84.64 %	82.52 %
BlockAnalysis	84.63 %	78.73 %	81.58 %
Text Detection	83.22 %	79.13 %	81.13 %
I2R_NUS	75.50 %	85.70 %	80.27 %
Baseline	70.99 %	85.17 %	77.44 %
BDTD_CASIA	70.17 %	80.03 %	74.77 %
OTCYMIST	79.81 %	67.92 %	73.39 %
Inkam	61.73 %	58.71 %	60.18 %

Fig. 5.9: ICDAR interface

Discussion on RR'13-SI dataset. The first remark we can make by looking at Table 5.11 is that the DETEVAL$_{\text{AUC}}$ protocol is the strictest one when computing the Recall score. On the opposite, EVALTEX protocol seems to be the most permissive one producing the highest Recall values. This is because EVALTEX validates all types of matchings and hence all coverage area are taken into account. We can notice that the difference between the scores computed with ICDAR'03 and EVALTEX reaches even 26%, in the case of the **TextDetection** method. We can also notice that the values with ICDAR'13(DETEVAL) are better than those computed with ICDAR'13 which re-enforces the motivation of organizers to correct some aspects that ICDAR'13 couldn't deal with, namely the rejection of many *many-to-one* detections. The DETEVAL$_{default}$ seems to be more penalizing than the ICDAR'13(DETEVAL) but more permissive than the AUC metrics.

[5] http://liris.cnrs.fr/christian.wolf/software/deteval/

There are some differences concerning the Precision scores, shown in Table 5.12. First of all, DETEVAL$_{AUC}$ is the most penalizing protocol. The score difference between ICDAR'13 and ICDAR'13(DETEVAL) is negligible which confirms the fact that, when *many-to-one* matchings are not considered by ICDAR'13, the only impact is on the Recall values and less on the Precision. Also, EVALTEX has the tendency to relax the Precision penalties applied by the other two methods. On the contrary, it punishes algorithms that produce detection areas significantly larger than the GT boxes (See Figure 5.10), as in the case of **TextSpotter** participant, which obtains a precision 9% smaller than the one produced by ICDAR'13$_{DETEVAL}$ and 10% lower than with ICDAR'13.

Table 5.15 shows the $F-$Scores associated to each text detection method and their ranking obtained with each evaluation protocol. This table provides the best proof that nowadays detections are not being evaluated accurately. The only two rankings that match are ICDAR'13(DETEVAL) and DETEVAL$_{default}$. On the contrary, ICDAR'03, ICDAR'13, DETEVAL$_{AUC}$ and EVALTEX provide very different rankings. For example, detector **CASIA_NLPR** is ranked second by DETEVAL$_{default}$, third by ICDAR'13, fourth by ICDAR'13(DETEVAL) and fifth by DETEVAL$_{AUC}$. ICDAR'03 and EVALTEX provide a similar ranking except for the **TextDetection** method.

Fig. 5.10: *TextSpotter* detection examples.

On the other hand, our protocol increases the Precision scores for algorithms such as **I2R_NUS_FAR**, I2R_NUS and **Inkam** which have a high number of partial *one-to-one* detections that are mismatched by DETEVAL and ICDAR'13 protocols, but correctly matched by our method. The ICDAR and DETEVAL ranking are relatively similar.

Discussion on RR'13-BD dataset. On the RR'13-BD dataset, the ICDAR'03 is the most penalizing protocol because ICDAR'03 only deals with *one-to-one* matchings, while all other mappings are not considered and hence not scored. The Recall difference on this dataset between EVALTEX and ICDAR'13, ICDAR'13(DETEVAL) and DETEVAL$_{default}$ is smaller than for RR'13-SI. We can observe that ICDAR'13 and ICDAR'13(DETEVAL) provide very close scores.

In Tables 5.11 – 5.16 the evaluation with the EVALTEX framework is done using the two-level GT annotation option, discussed in Section 3.2. In order to see the impact of this region tag on the final Precision scores, in the following section we discuss and analyze the detection results obtained when enabling or disabling the region option.

Tab. 5.11: **Recall scores** of all participants during the ICDAR 2013 RRC (Challenge 2) using the ICDAR′03, ICDAR′13, ICDAR′13(DETEVAL) DETEVAL$_{default}$, DETEVAL$_{AUC}$ and EVALTEX evaluation protocols on the **RR'13-SI** dataset.

Method	ICDAR′03	ICDAR′13	ICDAR′13(DETEVAL)	DETEVAL$_{default}$	DETEVAL$_{AUC}$	EVALTEX (R_G)	EVALTEX (R_{quant})	EVALTEX (R_{qual})
USTB_TexStar	0.62	0.66	0.69	0.61	0.58	0.72	0.76	0.95
TextSpotter	0.52	0.65	0.65	0.59	0.49	0.66	0.73	0.90
CASIA_NLPR	0.61	0.68	0.69	0.62	0.55	0.73	0.83	0.88
Text_detector_CASIA	0.57	0.63	0.67	0.58	0.54	0.72	0.76	0.94
I2R_NUS_FAR	**0.70**	**0.69**	**0.71**	**0.64**	**0.62**	**0.76**	0.82	0.92
I2R_NUS	0.67	0.66	0.70	0.62	0.61	0.75	0.81	0.92
TH-TextLoc	0.53	0.65	0.70	0.60	0.53	0.74	0.80	0.93
Text Detection	0.46	0.53	0.66	0.52	0.50	0.72	0.77	0.94
Baseline	0.33	0.35	0.35	0.32	0.30	0.36	0.38	0.94
Inkam	0.40	0.35	0.43	0.37	0.37	0.55	0.66	0.83

Tab. 5.12: **Precision scores** of all participants during the ICDAR 2013 RRC (Challenge 2) using the ICDAR′03, ICDAR′13, ICDAR′13(DETEVAL) DETEVAL$_{default}$, DETEVAL$_{AUC}$ and EVALTEX evaluation protocols on the **RR'13-SI** dataset.

Method	ICDAR′03	ICDAR′13	ICDAR′13(DETEVAL)	DETEVAL$_{default}$	DETEVAL$_{AUC}$	EVALTEX (P_G)	EVALTEX (P_{quant})	EVALTEX (P_{qual})
USTB_TexStar	**0.83**	**0.88**	**0.89**	**0.84**	**0.80**	**0.93**	0.97	0.96
TextSpotter	0.69	**0.88**	0.87	0.83	0.69	0.74	0.96	0.77
CASIA_NLPR	0.70	0.79	0.79	0.76	0.67	0.83	0.97	0.86
Text_detector_CASIA	0.79	0.85	0.85	0.80	0.75	0.90	0.96	0.93
I2R_NUS_FAR	0.75	0.75	0.76	0.73	0.71	0.87	0.89	0.97
I2R_NUS	0.73	0.73	0.73	0.71	0.69	0.85	0.88	0.97
TH-TextLoc	0.59	0.70	0.70	0.66	0.58	0.71	0.82	0.87
Text Detection	0.65	0.74	0.74	0.64	0.62	0.87	0.91	0.95
Baseline	0.56	0.61	0.61	0.60	56	0.61	0.67	0.91
Inkam	0.36	0.31	0.32	0.30	0.32	0.56	0.60	0.92

Tab. 5.13: **Recall scores** of all participants during the ICDAR 2013 RRC (Challenge 1) using the ICDAR′03, ICDAR′13, ICDAR′13(DETEVAL) DETEVAL$_{default}$, DETEVAL$_{AUC}$ and EVALTEX evaluation protocols on the **RR'13-BD** dataset.

Method	ICDAR′03	ICDAR′13	ICDAR′13(DETEVAL)	DETEVAL$_{default}$	DETEVAL$_{AUC}$	EVALTEX (R_G)	EVALTEX (R_{quant})	EVALTEX (R_{qual})
USTB_TexStar	0.58	**0.82**	**0.87**	**0.80**	0.65	0.88	0.91	0.98
TH-TextLoc	0.55	0.76	0.82	0.73	0.62	0.87	0.90	0.96
I2R_NUS_FAR	**0.62**	0.71	0.81	0.69	**0.71**	0.86	0.90	0.97
Text Detection	0.31	0.73	0.83	0.67	0.56	**0.91**	0.93	0.98
I2R_NUS	0.42	0.68	0.76	0.65	0.56	0.81	0.84	0.97
Baseline	0.59	0.69	0.71	0.70	0.60	0.75	0.77	0.98
BDTD_CASIA	0.54	0.67	0.70	0.66	0.56	0.76	0.9	0.96
OTCYMIST	0.60	0.75	0.80	0.75	0.64	0.85	0.90	0.94
Inkam	0.36	0.52	0.62	0.52	0.46	0.71	0.78	0.91

Tab. 5.14: **Precision scores** of all participants during the ICDAR 2013 RRC (Challenge 2) using the ICDAR′03, ICDAR′13, ICDAR′13(DETEVAL) DETEVAL$_{default}$, DETEVAL$_{AUC}$ and EVALTEX evaluation protocols on the **RR'13-BD** dataset.

Method	ICDAR′03	ICDAR′13	ICDAR′13(DETEVAL)	DETEVAL$_{Default}$	DETEVAL$_{AUC}$	EVALTEX (P_G)	EVALTEX (P_{quant})	EVALTEX (P_{qual})
USTB_TexStar	0.71	**0.94**	**0.94**	**0.90**	**0.73**	**0.96**	0.99	0.97
TH-TextLoc	0.70	0.87	0.87	0.82	0.70	0.91	0.97	0.94
I2R_NUS_FAR	0.68	0.84	0.85	0.78	0.70	0.94	0.98	0.96
Text Detection	0.53	0.79	0.79	0.68	0.59	0.88	0.96	0.92
I2R_NUS	0.69	0.85	0.86	0.79	0.71	0.95	0.98	0.96
Baseline	**0.72**	0.85	0.85	0.84	0.72	0.83	0.91	0.91
BDTD_CASIA	0.66	0.80	0.80	0.76	0.66	0.85	0.91	0.93
OTCYMIST	0.57	0.68	0.68	0.66	0.57	0.77	0.81	0.96
Inkam	0.50	0.58	0.59	0.51	0.49	0.86	0.88	0.98

Tab. 5.15: **F-Score (Ranking) scores** of all participants during the ICDAR 2013 RRC (Challenge 2) using the ICDAR'03, ICDAR'13, ICDAR'13(DETEVAL) $\text{DETEVAL}_{default}$, DETEVAL_{AUC} and EVALTEX evaluation protocols on the **RR'13-SI** dataset.

Method	ICDAR'03	ICDAR'13	ICDAR'13(DETEVAL)	$\text{DETEVAL}_{default}$	DETEVAL_{AUC}	EVALTEX
USTB_TexStar	0.71 (2)	0.76 (1)	0.78 (1)	0.71 (1)	0.67 (1)	0.82 (1)
TextSpotter	0.59 (6)	0.74 (2)	0.75 (2)	0.69 (2)	0.57 (6)	0.70 (8)
CASIA_NLPR	0.65 (5)	0.73 (3)	0.74 (4)	0.69 (2)	0.61 (5)	0.78 (6)
Text_detector_CASIA	0.66 (4)	0.72 (4)	0.75 (2)	0.67 (5)	0.63 (4)	0.80 (3)
I2R_NUS_FAR	0.72 (1)	0.72 (4)	0.73 (5)	0.68 (4)	0.66 (2)	0.81 (2)
I2R_NUS	0.70 (3)	0.69 (6)	0.72 (6)	0.66 (6)	0.65 (3)	0.80 (3)
TH-TextLoc	0.56 (7)	0.67 (7)	0.70 (7)	0.63 (7)	0.55 (7)	0.73 (7)
Text Detection	0.54 (8)	0.62(8)	0.70 (7)	0.57 (8)	0.55 (7)	0.79 (5)
Baseline	0.42 (9)	0.44 (9)	0.45 (9)	0.42 (9)	0.34 (9)	0.45 (10)
Inkam	0.38 (10)	0.33 (10)	0.36 (10)	0.33 (10)	0.34 (9)	0.55 (9)

Tab. 5.16: **F-Score (Ranking) scores** of all participants during the ICDAR 2013 RRC (Challenge 2) using the ICDAR'03, ICDAR'13, ICDAR'13(DETEVAL) $\text{DETEVAL}_{default}$, DETEVAL_{AUC} and EVALTEX evaluation protocols on the **RR'13-BD** dataset.

Method	ICDAR'03	ICDAR'13	ICDAR'13(DETEVAL)	$\text{DETEVAL}_{Default}$	DETEVAL_{AUC}	EVALTEX
USTB_TexStar	0.64 (2)	0.88 (1)	0.90 (1)	0.84 (1)	0.69 (2)	0.92 (1)
TH-TextLoc	0.62 (3)	0.81 (2)	0.84 (2)	0.77 (2)	0.66 (3)	0.89 (3)
I2R_NUS_FAR	0.53 (6)	0.77 (3)	0.83 (3)	0.74 (4)	0.74 (1)	0.65 (9)
Text Detection	0.39 (9)	0.76 (4)	0.81 (4)	0.67 (8)	0.57 (8)	0.90 (2)
I2R_NUS	0.52 (7)	0.75 (6)	0.80 (5)	0.71 (5)	0.63 (5)	0.87 (4)
Baseline	0.65 (1)	0.76 (4)	0.77 (6)	0.76 (3)	0.65 (4)	0.79 (7)
BDTD_CASIA	0.59 (4)	0.73 (7)	0.75 (7)	0.71 (5)	0.60 (6)	0.80 (5)
OTCYMIST	0.58 (5)	0.71 (8)	0.73 (8)	0.70 (7)	0.60 (6)	0.80 (5)
Inkam	0.42 (8)	0.55 (9)	0.60 (9)	0.52 (10)	0.48 (9)	0.78 (8)

5.1.4 Region annotation impact on global scores

In this section we explain the contribution of using the region tag. We first compare the scores obtained when the region tag is enabled or disabled. Next, we show the global impact of this option on the RR'13-BD and RR'13-SI datasets.

Region impact on single images. Figure 5.11 shows the impact on the Precision value (computed in Section 3.4.2) using different region annotations for three GT objects matched by one detection. One can observe that the Precision value increases proportionally to the surface of the text region. The logic behind this is that the more GT objects a region contains, the smaller the non textual area becomes and the less the Precision is penalized. Table 5.17 summarizes the impact of using the region level annotation on some key examples illustrated in Figure 5.12. Most of the detections correspond to *many-to-one* matchings. Here, the region labeling is done at line level. One can easily observe that if the region annotation is used, the Precision scores are higher than those obtained when only the object level GT annotation is considered. On the contrary, Recall scores are not influenced in any way by the region labeling.

a **b:** $R = 1, P = 0.60$ **c:** $R = 1, P = 0.62$ **d:** $R = 1, P = 0.65$

Fig. 5.11: The impact of the region GT (yellow rectangles) annotation on the Precision; (a) 3 GT objects (red rectangles), 1 detection (green filled rectangle); (b) 3 GT objects grouped into 3 text regions; (c) 3 GT objects grouped into 2 text regions; (d) 3 GT objects grouped into one text region.

We can then see that by attributing a region tag to a group of GT we can evaluate a *many-to-one* match without having to penalize the Precision due to the non-textual area assumed by this kind of matching. Consequently, the EVALTEX protocol is capable of evaluating equitably detectors that produce word or line level detections. Table 5.17 clearly show that word and line level detections get the same scores. This is a major advantage of EVALTEX as, instead of imposing a certain granularity level, it permits detectors to choose their own.

Fig. 5.12: *One-to-many* detections and the associated region annotation; left: detections (green filled rectangles); right: object GT annotation (red rectangles) and region annotation (yellow rectangles).

Region impact on a dataset. Tables 5.18 and 5.19 show the influence of the grouping GT objects into regions on the Precision scores obtained on the RRC'13-SI and RR'13-BD datasets. For each

Tab. 5.17: Global scores, Recall and Precision, when enabling and disabling the region GT annotation.

	One-level annotation		Two-level annotation	
Fig. 5.12	Recall	Precision	Recall	Precision
Top	1	0.96	1	1
Middle	1	0.89	1	1
Bottom	1	0.92	1	1

participant we give both the global Precision value P_G and the quality Precision value P_{qual} obtained when we use a one-level annotation (word) and the proposed two-level annotation (word and line). Text detection algorithms for which the Precision difference score obtained using these two annotations, produce more line-level detections that those for which this difference is lower. On the RR'13-SI dataset (see Table 5.18, three methods stand out: **Text_detector_CASIA**, **TH-TextLoc** and **Inkam** which have the highest Precision difference, equal to 0.03. A higher score variance between the two annotation levels can be seen in the scores obtained on the RR'13-BD dataset shown in Table 5.19. Here, the **TextDetection** method presents the highest difference of P_G scores equal to 0.07. This is explicable as many of its detections are at line level. Similarly, we can deduce that the text detectors for which there is no score difference between the Precision values, produce detections exclusively at word level. We mention here the **CASIA_NLPR** and **I2R_NUS_FAR** algorithms on the RR'13-SI database. Based on these experiments, we highlight that the comparison of Precision scores when enabling and disabling the region tag, can serve as an additional information on the level of detections produced by a text localization method.

Tab. 5.18: Precision scores of all participants during the ICDAR 2013 RRC (Challenge 2) on the RR'13-SI using both the one-level (only word) and two-level (word and line) annotations.

	One-level annotation		Two-level annotation	
Method	EvalTex (P_G)	EvalTex (P_{qual})	EvalTex (P_G)	EvalTex (P_{qual})
USTB_TexStar	0.91	0.94	0.93	0.96
TextSpotter	0.73	0.77	0.74	0.77
CASIA_NLPR	0.83	0.85	0.83	0.86
Text_detector_CASIA	0.86	0.90	0.89	0.93
I2R_NUS_FAR	0.87	0.97	0.87	0.97
I2R_NUS	0.84	0.95	0.85	0.97
TH-TextLoc	0.68	0.82	0.71	0.87
Text Detection	0.80	0.88	0.87	0.95
Baseline	0.60	0.90	0.61	0.91
Inkam	0.53	0.87	0.56	0.92

An example which higlights the importance of this region tag is represented by the Precision scores obtained by detector **TextDetection** and illustrated in Table 5.18. The Precision difference of 0.07 obtained by enabling or disabling the region tag can be viewed as a penalization applied to a detector whose majority of detections are at line level while the output expected by the protocol was at word-line level. These aspects together with the set of experimental results presented in this section motivate the valuable impact of assigning the region tags to GT objects, namely allowing detectors, with different granularity levels to be evaluated in the same manner.

Tab. 5.19: Precision scores of all participants during the ICDAR 2013 RRC (Challenge 2) on the RR'13-BD using both the one-level (only word) and two-level annotations (word and line).

	One-level annotation		Two-level annotation	
Method	EvaLTex (P_{quant})	EvaLTex (P_{qual})	EvaLTex (P_{quant})	EvaLTex (P_{qual})
USTB_TexStar	0.93	0.94	0.96	0.97
TH-TextLoc	0.87	0.90	0.91	0.94
I2R_NUS_FAR	0.89	0.90	0.94	0.96
Text Detection	0.80	0.88	0.87	0.95
I2R_NUS	0.89	0.90	0.95	0.96
Baseline	0.82	0.90	0.83	0.91
BDTD_CASIA	0.84	0.91	0.85	0.93
OTCYMIST	0.75	0.93	0.77	0.96
Inkam	0.82	0.93	0.86	0.98

5.2 Experimental results using the mask representation

In this section we present the evaluation results given by EvaLTex on some examples of detections corresponding to text objects that could not well represented by rectangular boxes. Figure 5.13 shows the advantage of using a mask based GT representation rather than a rectangular one when dealing with any of the following categories of text objects: perspectively deformed, tilted, curved, wavy or circular. Moreover, for each of these examples we illustrate possible detection masks for which the performance results are given in Table 5.20. Figure 5.13 shows some of the problems posed by the rectangular GT annotation:

Problem 1 intersection of rectangular boxes in the GT: "ENTIER" with "ALBACORE" (Figure 5.13f), "FLORANIS" with "FRERES" (Figure 5.13f), "COMPUTATIONAL" with "COMPLEXITY" (Figure 5.13f);

Problem 2 GT rectangles contain considerably more non-textual areas than textual ones: "KEMAKEUR" and "H03VV-F" (Figure 5.13b), "Enjoy" and "Coffee" (Figure 5.13c), "ENTIER" and "NATURAL" (Figure 5.13f), "ALAINAFFLELOU" (Figure 5.13e), "COMPUTATIONAL" and "COMPLEXITY" (Figure 5.13g);

Problem 3 inclusions of GT boxes: "GRAS" into "ANISETTE" (Figure 5.13f).

Note. The intersection of rectangular boxes (*Problem 1*) in the GT is a problem because a detection that matches only one of the GT objects can easily be interpreted as all GT objects were equally detected. The fact that GT boxes contain more non-text surface than text surface (*Problem 2*) can produce imprecise coverage rates (e.g. for example the mapping between a text with a capital letter at the beginning and a detection that covers all letters but the first one). Finally, using rectangular boxes, a GT object can be included into another one and hence the evaluation ambiguity that could either consider them both detected or only one of them.

Table 5.20 summarizes the comparison between the local and global performance scores obtained when using a rectangular and a mask representation for text detected in images of Figure 5.13. The rectangular detections are not represented in Figure 5.13 but correspond to boxes surrounding the

detection masks given in the last column of this figure which will be matched with the GT rectangles shown in the second column of the figure. One can observe that, when using a rectangular representation, the matching procedure is disturbed by the text objects that intersect in the GT. Namely, text objects such as “ALBACORE” in Figure 5.13f, are matched two times: firstly with their corresponding detection and secondly with detections targeting objects that intersect them in the GT. Hence, the coverage scores of such GT objects are penalized by the fragmentation parameter invoked during the *one-to-many* matching, which can furthermore impact the global Recall score. Similar cases that imply text object intersections in the GT, such as “GRAS” and “ANISETTE” (Figure 5.13f), can be successfully avoided using the filtering procedure, described in Section 3.3.3. However, due to numerous examples involving GT intersections, the filtering cannot always predict correctly a detector’s choice. For this reason, using masks to represent text objects could prevent such situations generated by the rectangular text annotation.

Recall values of Figure 5.13e show another example of differences when using these two representations in the case of a tilted and perspective deformed text (“ALAINAFFLELOU”) only partially matched. The coverage ratio computed on rectangles is smaller than the coverage ratio computed on masks and consequently leads to a significant difference of recall values.

Another discrepancy between the rectangular and mask annotations comes from the Precision value variations that are more accentuated when dealing with *many-to-one* detections. Such situations can be seen in Figures 5.13b and 5.13c that illustrate *many-to-one* detections covering curved text strings (“KEMA-KEUR”, “3G0.75” and “VDE" in Figure 5.13b, respectively GT objects “Enjoy” and “yours" in Figure 5.13c). For Figure 5.13b, the precision values vary from 0.48, when using the rectangular representation to 0.81, in the case of mask annotation. Similarly, the precision scores for the two text representations in Figure 5.13c range from 0.73 to 0.98. Once again, the rectangle representation shows its limitation and that it can significantly penalize the performance evaluation of a detector.

In this section we have provided a detailed evaluation for a series of images in order to point out the interest of having a more accurate representation of curved, arc-form or circular texts, namely using an annotated with masks. The illustrated images contain eloquent examples of text regions for which the rectangular annotation is not well-adapted. We have shown that by using bounding boxes, the accuracy of some detections can be under or over evaluated and that the irregular mask annotation can successfully be used to avoid such situations.

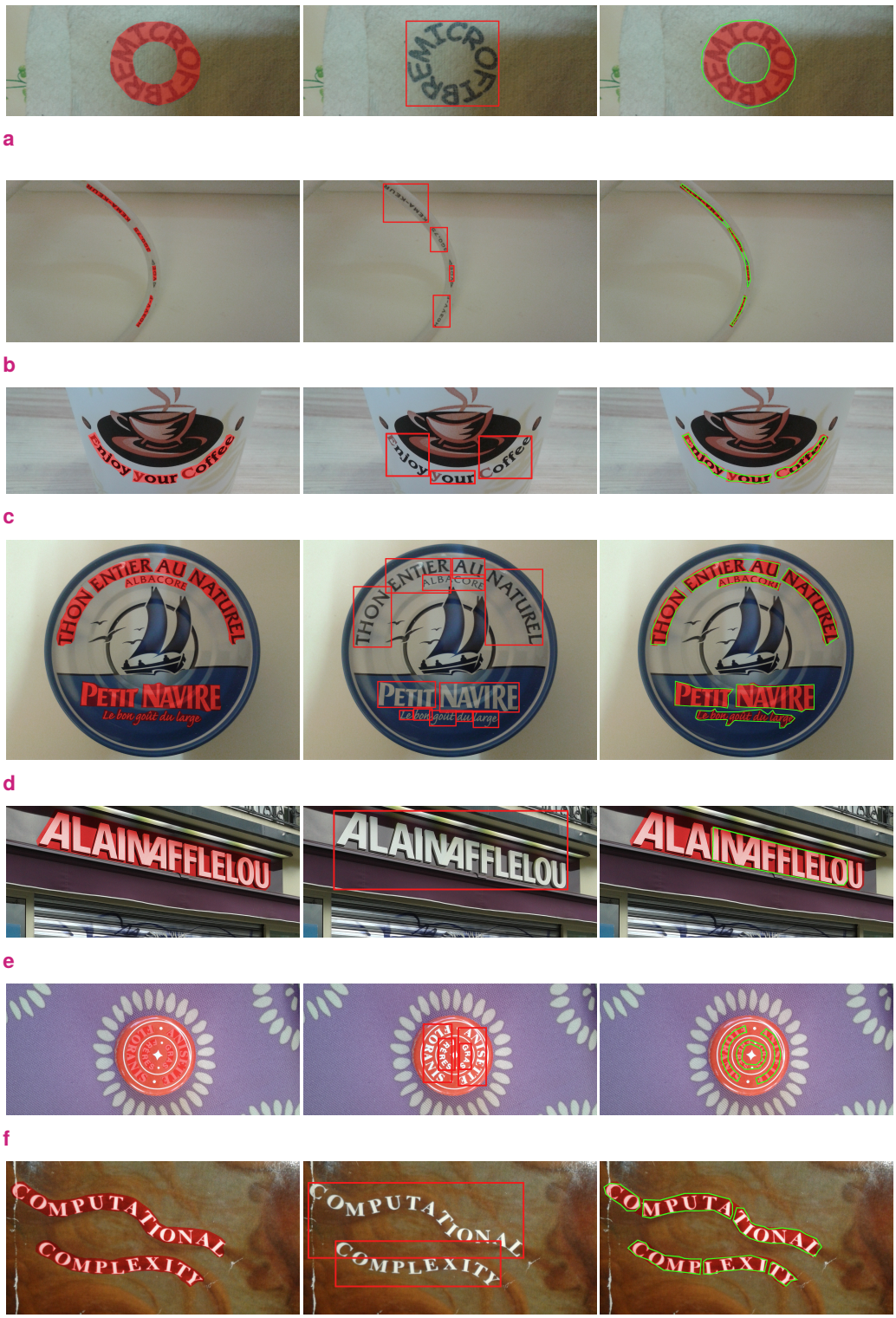


Fig. 5.13: Examples of different texts (inclined, curved, perspectively deformed, following a circular path); left: mask GT annotation (red); center: rectangular GT annotation (red); right: GT masks (red) overlapped by detection masks (green).

Tab. 5.20: Object (Coverage and Accuracy) and global (Recall, Precision and F-Score) performance scores corresponding to the detections in Figure 5.13 computed using the rectangular and the mask representations.

		RECTANGULAR REPRESENTATION					MASK REPRESENTATION				
Figure	GT text object	*Cov*	*Acc*	Recall	Precision	F-Score	*Cov*	*Acc*	Recall	Precision	F-Score
5.13a	*MICROFIBRE*	1	1	1	1	1	1	1	1	1	1
5.13b	*KEMA-KEUR*	1	0.32	1	0.48	0.65	0.97	0.76	0.96	0.81	0.88
	3G0.75	1	0.3				0.97	0.76			
	VDE	1	0.32				1	0.75			
	H03VV-F	1	1				0.9	0.98			
5.13c	*Enjoy*	1	0.6	1	0.73	0.85	0.98	0.96	0.94	0.98	0.96
	your	1	0.6				0.9	0.96			
	Coffee	1	1				0.93	1			
5.13d	*THON*	1	1	0.97	0.88	0.92	1	1	1	0.9	0.94
	ENTIER	1	0.98				1	0.95			
	AU	1	0.98				0.99	0.95			
	NATUREL	1	1				1	1			
	ALBACORE	0.59	1				1	0.92			
	PETIT	1	1				1	1			
	NAVIRE	1	1				1	1			
	Le	1	0.72				1	0.78			
	bon	1	0.73				1	0.78			
	gout	1	0.72				1	0.78			
	du	1	0.71				1	0.78			
	large	1	0.73				1	0.8			
5.13e	*ALAINAFFLELOU*	0.58	1	0.58	1	0.73	0.63	1	0.63	1	0.77
5.13f	*FLORANIS*	1	1	1	0.93	0.96	0.96	1	0.97	0.91	0.94
	ANISETTE	1	1				0.97	1			
	GRAS	1	0.85				0.97	0.83			
	FRERES	1	0.86				0.96	0.83			
5.13g	*COMPUTATIONAL*	0.34	1	0.28	1	0.44	0.46	0.99	0.45	0.99	0.63
	COMPLEXITY	0.31	1				0.45	0.99			

5.3 Experimental results using the histogram representation and EMD-based evaluation

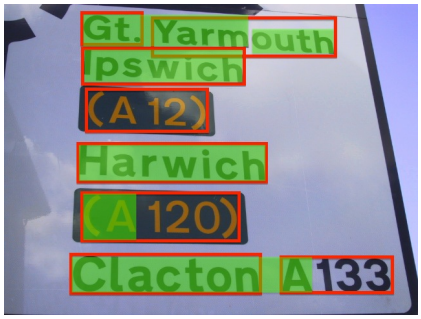
In this section we show, through a set of pertinent examples, the motivation of using histograms to represent text detection results and histogram distances as a reliable tool for computing the final scores. In a first step, we locally evaluate some detection results on single images to explicitly show how each detection accurately can be described by a bin interval and a bin value of a histogram. Next, we use the histograms to represent a set of detections with the goal of illustrating the behavior of a detector. We also show that this global representation of detections is suitable for successfully comparing different detectors. Finally, the proposed quality histograms are compared to the performance ROC plots introduced in [Wolf and Jolion, 2006].

Analysis of single detection results. We have shown in Chapter 4 that through histograms, one can easily "read" and understand the detection characteristics of a text detector. Figure 5.14 gives three examples of detections, their corresponding non-normalized coverage (depicted in blue) and accuracy (depicted in orange) histograms (see Figure 4.3) with $B = 10$ bins and the resulting global Recall and Precision scores. The interpretation of these two histograms is straightforward. The first bin of h_{Cov} (in orange) encloses the total number of non-detected (or poorly detected, $Cov \leq 0.1$) GT objects, while the first bin of h_{Acc} (blue) encloses the number of false positives (or detections with poor precision, $Acc \leq 0.1$). The last bin corresponds to very good matchings, while all intermediate bins are correlated to either partial detections (in h_{Cov}) or detection areas that are larger than the GT areas, respectively in h_{Acc}.

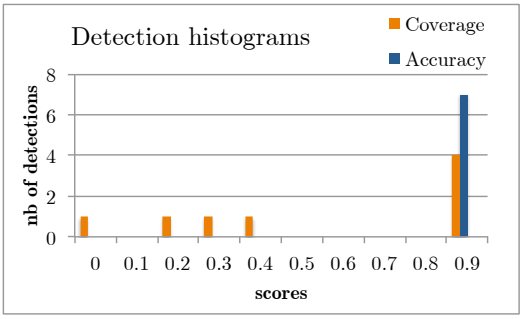
- In the top example of Figure 5.14, the scattered coverage values of h_{Cov} indicate the presence of either partial ("A120" ($[0.3, 0.4[$) and "A133" ($[0.2, 0.3[$)) or one-to-many ("Yarmouth" ($[0.4, 0.5[$)) detections. On the other hand, all accuracy values are accumulated into the last bin of h_{Acc} which means that all detections were truthful with respect to the GT.
- By analyzing the histograms in the middle example of Figure 5.14, we observe that the first bin value of h_{Cov} equals the sum of values of the other bins. This shows that only half of the GT objects were detected ("INTRODUCTION", "TO", "DATABASE", "SYSTEMS", "DATE"), while the other half was missed or poorly detected ("AN", "C.", "J.", "SIXTH", "EDITION").
- h_{Acc} associated to the detection examples in the bottom of Figure 5.14, suggests there are three possible false positives. The values 1 of bin intervals $[0.7, 0.8[$ and $[0.9, 1]$ correspond to one detection that exceeds its corresponding GT boundary object ("RIVERSIDE") and one accurate detection ("WALK") respectively.

Comparison of two algorithms. A good advantage of this representation is that, applied on a dataset, it allows to characterize and compare at a glance text detectors. In Fig. 5.15 we illustrate the overall detection behavior of two algorithms, **Inkam** and **TextSpotter**, based on the detection results submitted to ICDAR 2013 RRC [ICDAR, 2013].

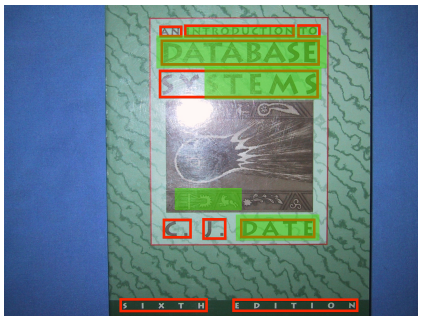
- The left plot shows coverage values ($\widetilde{h}_{Cov}$) of both algorithms. Both normalized coverage histograms illustrate a similar tendency: two high peaks on the first and last bins and a lower peak around the value 0.5. This means that, for both algorithms, most of the GT objects were either



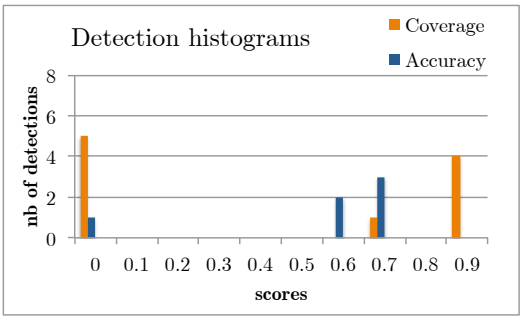
a



b: $R_G = 0.66$, $P_G = 1$

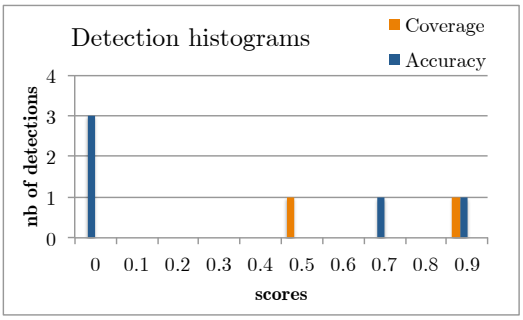


c



d: $R_G = 0.53$, $P_G = 0.65$

e



f: $R_G = 0.8$, $P_G = 0.42$

Fig. 5.14: GT (red rectangles) and detection (filled green rectangles) examples and their corresponding coverage/accuracy histograms and R_G/P_G scores.

missed, either accurately detected, while only approximately 6% of the GT objects were involved in partial or *one-to-many* detections. One can however conclude that from the coverage aspect, **TextSpotter** slightly outperforms **Inkam**: the number of missed GT objects (value of the first bin) is lower while the last bin's value is higher.

- The right plot corresponds to accuracy values ($\widetilde{h}_{Acc}$) of both algorithms. Contrary to the coverage similarity behavior discussed above, the accuracy profiles of the two detectors are very different. **Inkam** produces a significantly higher number of false positives than **TextSpotter**. The accuracy histogram of **textSpotter** has higher bin values in the quality intervals $[0.7, 0.8[$ and $[0.8, 0.9[$. This is because **TextSpotter** adds a large border to all its detections [ICDAR, 2013], which decreases the object-level accuracies. On the other hand, **Inkam** produces as many false positives as accurate detections (first and last bin values close to 0.4).

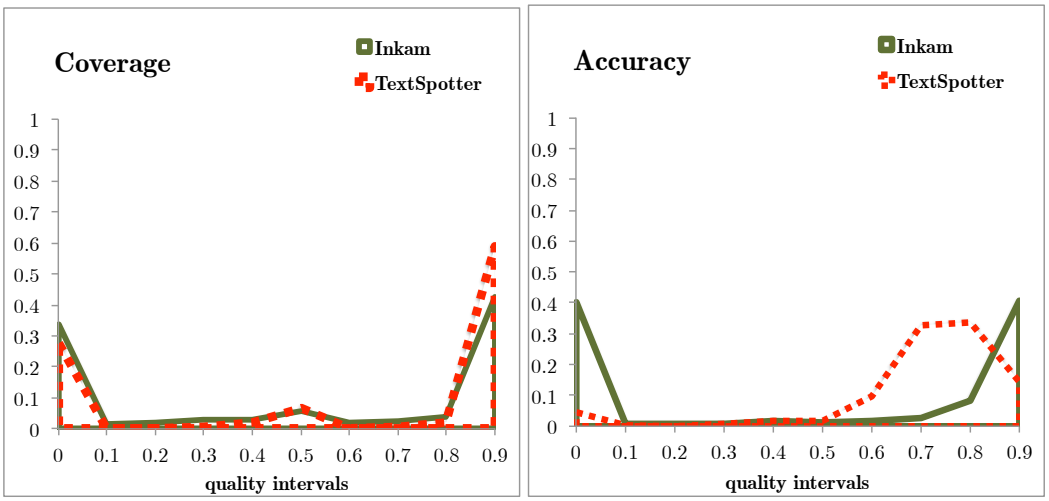
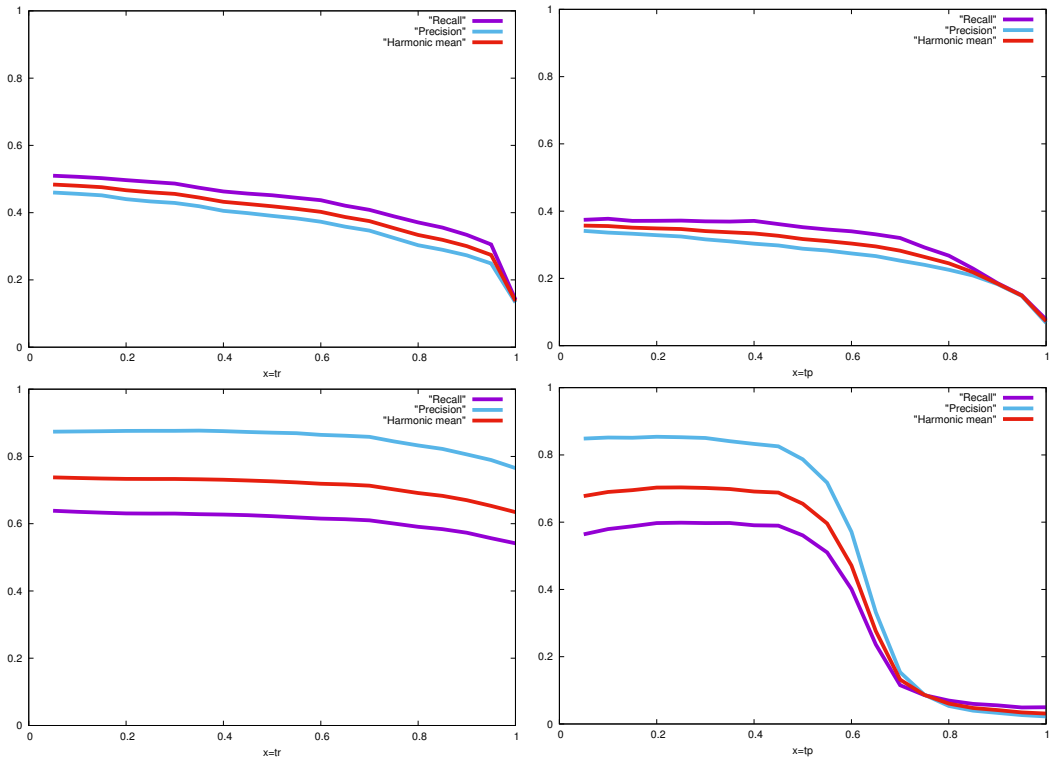


Fig. 5.15: Coverage and accuracy normalized histograms associated to detector **Inkam** ($R_G = 0.60$, $P_G = 0.58$) and detector **TextSpotter** ($R_G = 0.70$, $P_G = 0.80$).



c: varying constraint t_r **d:** varying constraint t_p

Fig. 5.16: Performance plots generated with DETEVAL tool [Wolf and Jolion, 2006] (Recall in purple, Precision in blue); top: **Inkam** ($R_{OV} = 0.37$, $P_{OV} = 0.32$); bottom: **TextSpotter** ($R_{OV} = 0.49$, $P_{OV} = 0.69$).

The quality histograms of all participants at the ICDAR 2013 RRC are illustrated in Figure 5.17 for *Challenge 1* and in Figure 18 for *Challenge 2*.

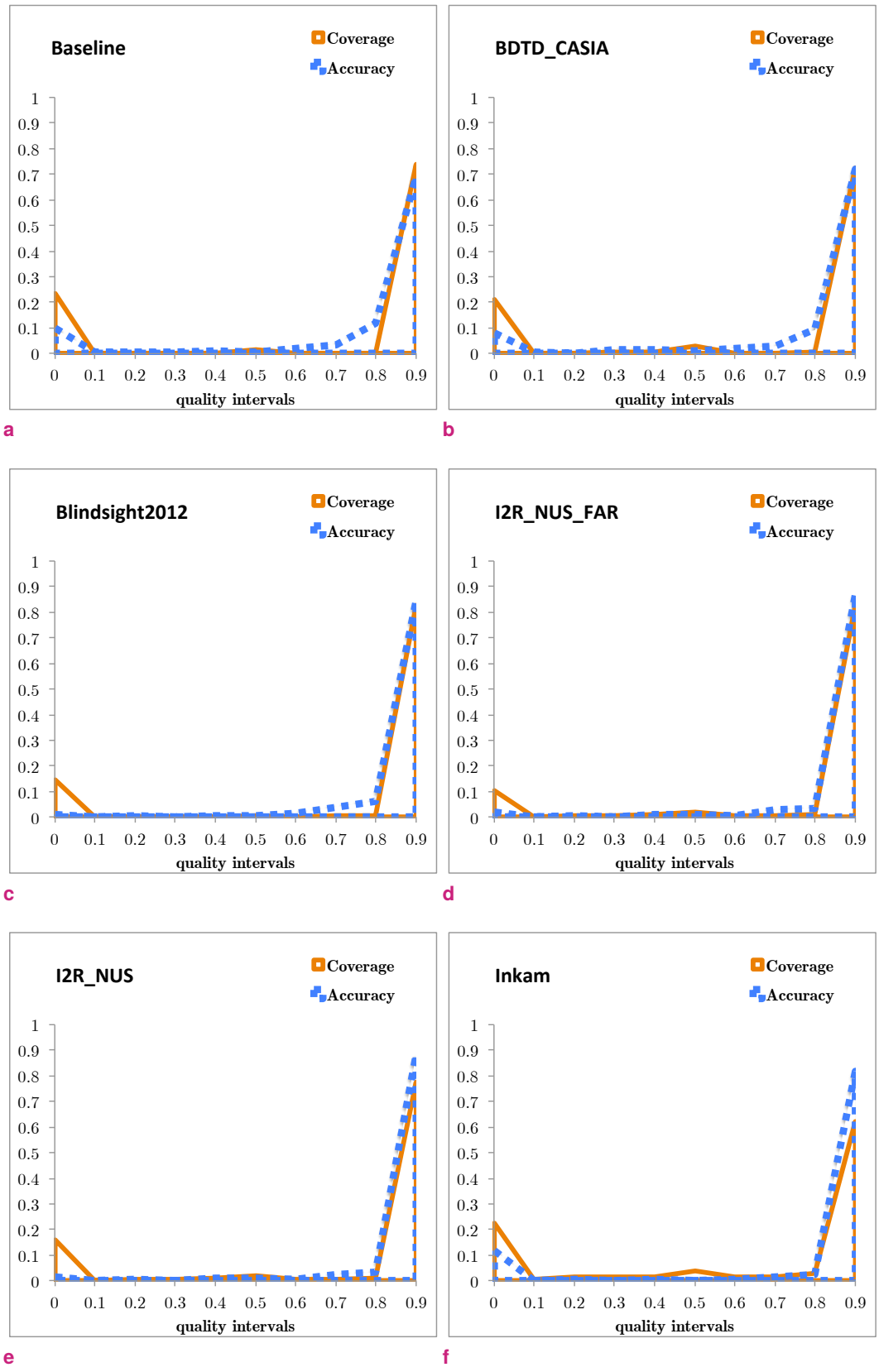


Fig. 5.17: Quality (Coverage and Accuracy) histograms of participating text detection methods at the ICDAR 2013 RRC on the born-digital image dataset (RR'13-BD).

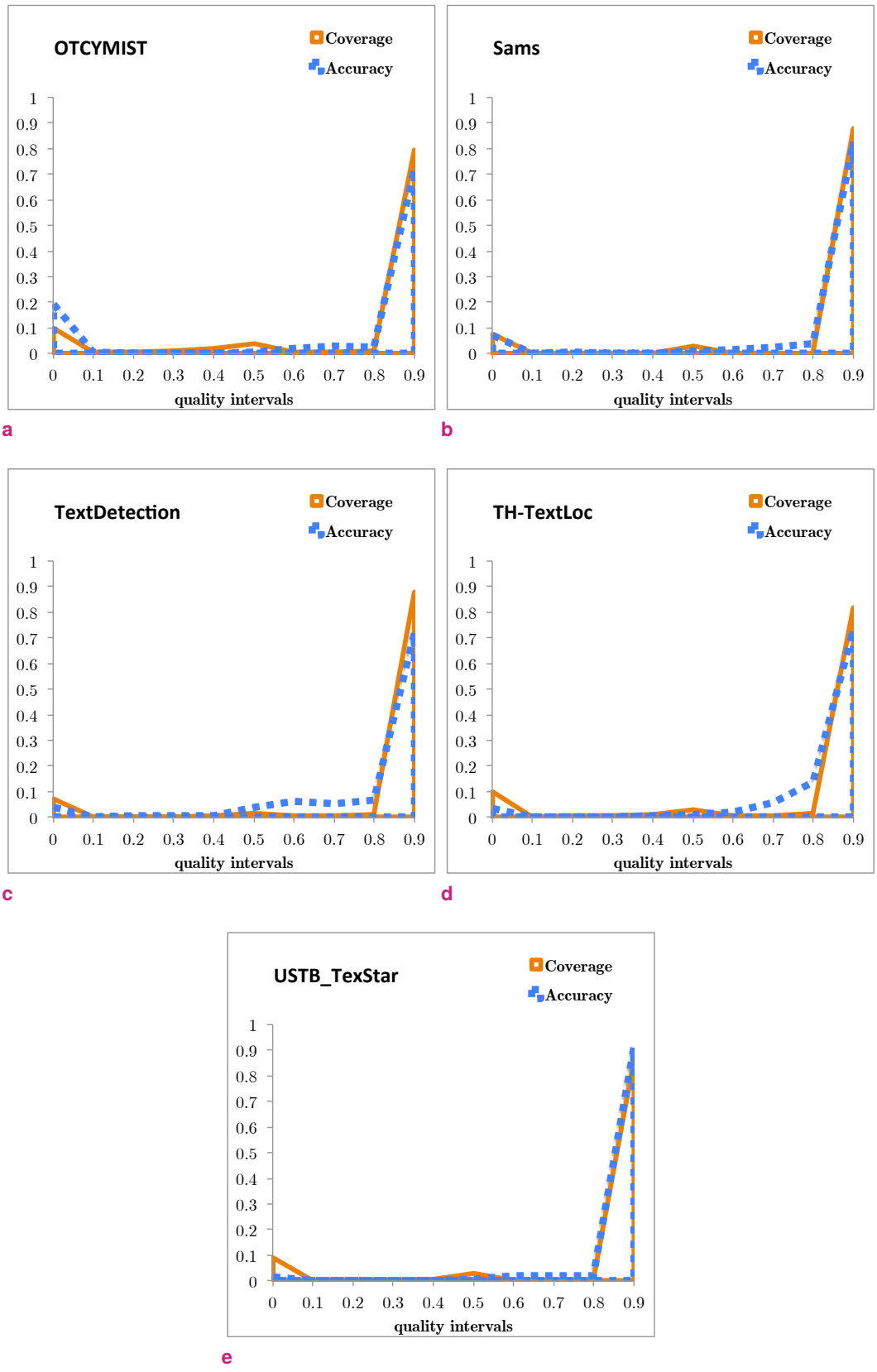


Fig. 17 (Cont.): Quality (Coverage and Accuracy) histograms of participating text detection methods at the ICDAR 2013 RRC on the born-digital image dataset (RR'13-BD).

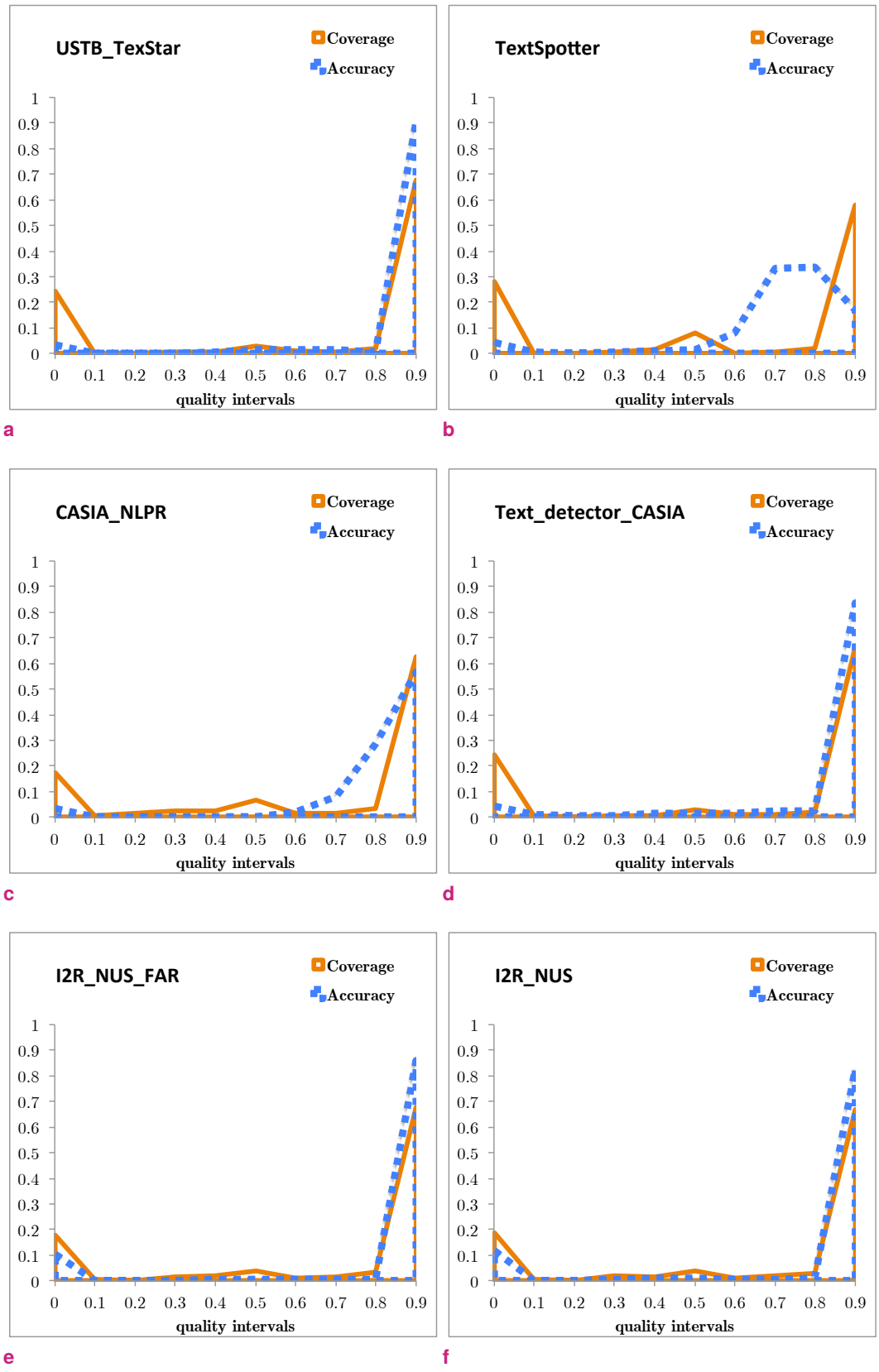


Fig. 18: Quality (Coverage and Accuracy) histograms of participating text detection methods at the ICDAR 2013 RRC on the scene image dataset (RR'13-SI).

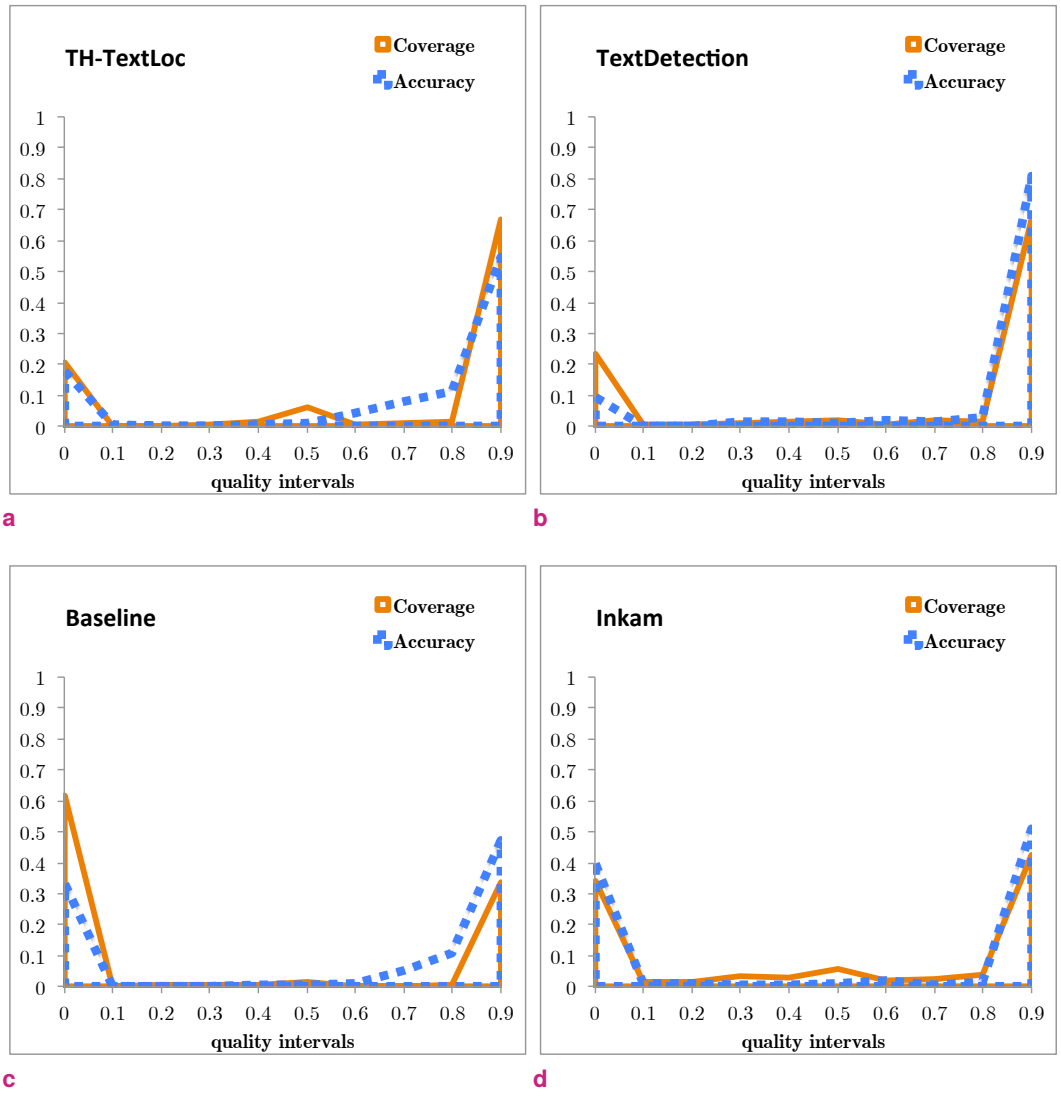


Fig. 18 (Cont.): Quality (Coverage and Accuracy) histograms of participating text detection methods at the ICDAR 2013 RRC on the scene image dataset (RR'13-SI).

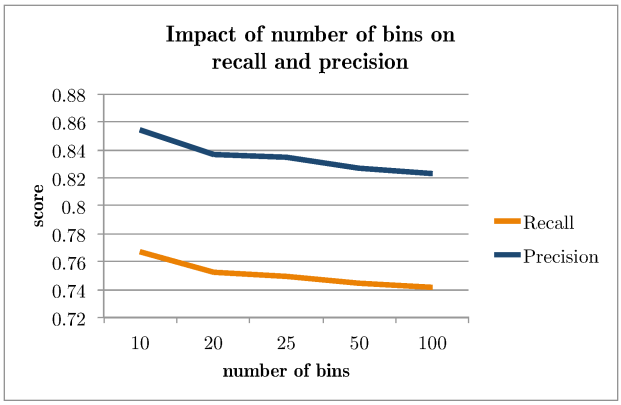


Fig. 19: Variation of R_G and P_G scores depending on the number of bins B (detection results provided by [Fabrizio et al., 2013] on the ICDAR 2013 dataset).

Comparison of the histogram representation and DETEVAL plots. We now compare our histogram representation with the performance plots generated by the DETEVAL tool [Wolf and Jolion, 2006] (see Fig. 5.16). The plot representation in [Wolf and Jolion, 2006] is obtained by varying the two quality constraints (t_r for Recall, and t_p for Precision) and counting how many objects fall into a certain interval, whereas our method implies a qualitative local evaluation from the start. Although both approaches capture the quality and quantity natures of a detection, we introduce a more compact representation using only two plots for depicting a detection (instead of generating four plots, two for Recall and two for Precision, as proposed in [Wolf and Jolion, 2006]). Secondly, histograms have the advantage of being more intuitive and easier to interpret in the given context of text detection. One can easily visualize the proportion of missed GT objects or false positives, as well as the amount of detections that fall into any other coverage or accuracy interval. Concerning the overall Recall and Precision scores obtained with the two approaches, we can observe that the results are different, which is due to the different object level evaluation used by the two methods.

Impact of tuning the number of bins. By using histograms to represent detections, the generated global scores will depend on the chosen number of bins (B). Namely, the higher the number of bins, the more accurate the scores will be. Consequently, if we increase the number of bins, the score will decrease. For example, a detection that was evaluated to a 0.52 coverage value, will be counted in the $[0.5, 0.6[$ bin interval if $B = 10$. If we use 20 bins, the same 0.52 coverage value will be quantified in the $0.5, 0.55[$ bin interval.

Tab. 5.21: Impact of the number of bins on Recall and Precision scores obtained from the detection results of the **TextDetection** method during the ICDAR 2013 RRC on the RR'13-SI dataset.

Method	Recall	Precision
EMD_{10bins}	0.7667	0.8799
EMD_{20bins}	0.7526	0.8713
EMD_{25bins}	0.7495	0.8693
EMD_{50bins}	0.7441	0.8659
$EMD_{100bins}$	0.7413	0.8642

While a value of 10 bins is mostly appropriate for graphical illustration purposes, when computing final scores, one should however choose a higher number of bins to produce a more precise evaluation result. Figure 19 illustrates the variation of R_G and P_G scores when B varies from 10 to 100 bins. As expected,

the natural tendency of these two metrics is to decrease when B increases. When B exceeds 50 intervals, one can observe the stabilization of these two global scores. Scores are reported in Table 5.21.

Links between EVALTEX and EMD scores. Both the EMD-derived scores and the global scores obtained using EVALTEX (introduced in Section 3.4.2) characterize the overall performance of a detector and are based on the same local measurements and matching rules. The experimental results conducted on the two ICDAR datasets (RR'13-SI and RR'13-BD) have shown that the scores obtained using these two approaches are very close as it can be seen in Tables 5.22 and 5.23. The slight score difference, which does not exceed 0.064 in Recall and 0.051 in Precision, is mainly due to the fact that the EMD scores were computed based on 100−bin histograms, making EVALTEX scores a little more accurate. We can conclude that the close values to scores obtained from the EMD approach corroborate the proposed overall metrics of EVALTEX. Hence, we can state that both of these two approaches are reliable and provide an accurate view of the performance of a detector.

Tab. 5.22: Comparison of performance scores of detection methods on the RR'13-SI dataset obtained using the EVALTEX global metrics and the EMD.

	RECALL		PRECISION		F-SCORE	
Method	EVALTEX	EMD	EVALTEX	EMD	EVALTEX	EMD
USTB_TexStar	0.7234	0.7264	0.9331	0.9345	0.8150	0.8010
TextSpotter	0.6610	0.6648	0.7388	0.7439	0.6977	0.8549
CASIA_NLPR	0.7339	0.7370	0.8336	0.8383	0.7806	0.8173
Text_detector_CASIA	0.7163	0.7195	0.8938	0.8960	0.7953	0.8094
I2R_NUS_FAR	0.7606	0.7633	0.8718	0.8736	0.8124	0.8054
I2R_NUS	0.7519	0.7546	0.8533	0.8553	0.7994	0.8105
TH-TexLoc	0.7387	0.7416	0.7146	0.7197	0.7264	0.8421
TextDetection	0.7210	0.7241	0.8667	0.8685	0.7872	0.8139
Baseline	0.3618	0.3682	0.6062	0.6113	0.4531	0.4596
Inkam	0.5490	0.5539	0.5553	0.5600	0.5521	0.5569

Tab. 5.23: Comparison of performance scores of detection methods on the RR'13-BD dataset obtained using the EVALTEX global metrics and the EMD.

	RECALL		PRECISION		F-SCORE	
Method	EVALTEX	EMD	EVALTEX	EMD	EVALTEX	EMD
USTB_TexStar	0.8795	0.8811	0.7593	0.7642	0.8150	0.8038
TH-TexLoc	0.8585	0.8602	0.7980	0.8027	0.8271	0.8006
I2R_NUS_FAR	0.8604	0.8620	0.8824	0.859	0.8712	0.7860
TextDetection	0.9015	0.9027	0.8252	0.8287	0.8616	0.7887
I2R_NUS	0.8066	0.8088	0.8879	0.8913	0.8453	0.7940
Baseline	0.7533	0.7558	0.8270	0.8309	0.7884	0.8150
BDTD_CASIA	0.7583	0.7608	0.8529	0.8551	0.8028	0.8094
OTCYMIST	0.8354	0.8373	0.6562	0.6620	0.7351	0.8334
Inkam	0.6993	0.7027	0.8245	0.8274	0.7567	0.7600

Computational time. Generally, the evaluation protocols are not submitted to any computational time constraints. This means that, in theory, an evaluation process could take as long time as it needs to accurately analyze the performance of a detector. In practice however, we want evaluation frameworks that are able to deal with large datasets and hence to have a fair time complexity. The computational

complexity needed for computing the EMD on $N-$bin histograms is $O(N^3 logN)$. Recent works have shown that the computational cost of EMD can reasonably reach ~ 0.03s (see [Pele and Werman, 2009]) with a complexity equal to $O(\min(t^2 N, N^2))$, where t is a distance threshold. In our experiments, the average computational time needed for evaluating a detection set using the histogram representation and the EMD is approximately $0.01s$ for an image.

In the following section we will synthesize the results obtained from the experiments presented in this chapter.

5.4 Conclusion

In this chapter we have conducted a series of experiments to validate our proposed evaluation framework. This is not an easy task, as the goal here is to evaluate an evaluation protocol. To do so we proposed a visual evaluation of the EvaLTex protocol by comparing the scores obtained from some key detection scenarios.

In a first stage, we have provided a detailed comparison of EvaLTex with commonly used evaluation protocols in the literature, namely ICDAR'03 and three different configurations of DetEval. We provided the scores obtained with these evaluation methods on individual images and an analysis of the corresponding matching strategies. We emphasized the obvious drawbacks of ICDAR'03 method and why it should not be used anymore for evaluating text detectors. First, it provides a poor matching strategy as it can only deal with *one-to-one* mappings even for the challenging scenarios in which texts are often split by detections (*one-to-many*) or merged into single detections (*many-to-one*). Secondly, the provided metrics do not differentiate clearly and accurately the different aspects of the detection and hence, in many cases the obtained scores are unrepresentative. In the same way, we compared EvaLTex with the complex DetEval evaluation protocol. Due to its numerous parameters and metric diversity, this framework can be used with different configurations that provide distinct results. The first configuration, denoted in this work "relaxed" DetEval, consists of relaxing all area constraints. In this way, we attempted a closer approach to our evaluation protocol that does not imply any area constraints to provide a fair comparison between it and DetEval. We then showed that the scores are highly permissive and consequently not representative. The second configuration relies on the AUC metrics that take into account, as stated by the authors in [Wolf and Jolion, 2006], both the quality and quantity aspects of a detection. Hence, due to the concept of quantity-quality detection relationship proposed both by DetEval and in this work, we analyzed the differences of scores obtained for different matching strategies. Although the AUC metrics solve some of the problems of ICDAR'03 or even the "relaxed" DetEval, it still fails to discriminate the characteristics of the Recall and Precision. The third set of comparisons with DetEval relies on its default configuration that was used during ICDAR 2013 RRC competitions. The detection results of the participants at these competitions allowed, not only to compare the performance scores, but also to provide a visual comparison of the matching processes used by the two protocols. In this way we have highlighted many problems of the default DetEval (or ICDAR 2013 metrics) such as the inconsistencies related to *one-to-many* and *many-to-one* matchings. To emphasize once again the inconsistencies between existing evaluation protocols we provided the overall results of all participants at the ICDAR 2013 RRC on RR'13-SI and RR'13-BD datasets using each of the protocols discussed above.

The experiments discussed above were exclusively done on texts represented by horizontal bounding boxes. However, one of the advantages of EvaLTex consists of its ability of coping with free-form representations of texts. We have illustrated a series of images where the GT and detection objects were annotated using masks. We have shown the advantage of using a free-form labeling of texts by providing the associated scores obtained when using the rectangular and the mask annotations on key examples containing curved, inclined or deformed texts.

In Section 5.3 we focused on presenting the interest of using histograms to represent text detection results and the EMD to compute global scores. First, single images were used to explain the different detection aspects that are captured at a glance with the two quality histograms (based on the coverage and accuracy local measurements), such as the number of False Positives and True Positives, or the percentage of partial or missed detections. Next, we showed that the quality histograms are also useful tools to compare two sets of detections. To illustrate this we have compared the coverage and accuracy histograms generated from the detection results of two text localization algorithms on the RR'13-SI dataset. We have successfully demonstrated that we can capture essential information that could not be otherwise derived by simply analyzing the final scores. To prove the intuitiveness of the histogram representation, we compared it to the ROC plots generated by DetEval framework. We showed that compared to the ROC curves, our approach is less confusing and provides an immediate view of the global structure of a detection set, without relying on any area thresholds. Next, we have analyzed the variations of the EMD-derived Recall and Precision scores when increasing the number of bins. If a sufficiently large number of bins is used to represent the quality histograms, then the scores tend to stabilize and converge to the the scores obtained with EvaLTex. This last assumption was confirmed by comparing the Recall, Precision and $F-$Scores obtained with the EMD with the ones obtained with the EvaLTex protocol.

Part II

Contribution to text rectification

6 Introduction on text rectification processes

Contents

In this chapter we explain the role of a text rectification step in the global framework of a text understanding system. First, we detail the different deformations that texts present in born-digital and natural scene images are often subject to. Next, we focus on the works done in this research area and finally we draw some conclusions and list our contributions.

6.1 Introduction

Retrieving the textual information from born-digital and real-scene images can often be a challenging task due to the variety of text properties (color, size, font, orientation) but also due to external causes, such as difficult lighting conditions (shadows, specularity, reflections, *etc.*), cluttered backgrounds, possible occlusions, poor image resolution and quality, or situations where the the text plane is not parallel to the camera one. These circumstances do not only affect the text detection process but also the text recognition stage. Unfortunately, most of the current OCRs have low performances on recognizing curved, inclined, vertical or perspective distorted texts. Such text examples are illustrated in Figures 1 and 2.

Fig. 1: Examples of real scene images with deformed text from the ICDAR 2015 Competition Scene Text Rectification dataset.

In order to obtain a high recognition accuracy rate, the detected texts need to be corrected and adjusted to a front-parallel view. Text rectification methods usually target oriented, sheared or texts in perspective

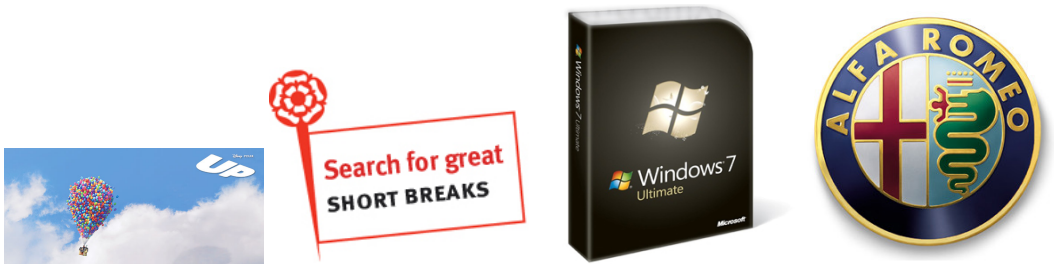


Fig. 2: Examples of born-digital images with deformed text taken from the ICDAR RRC Born-Digital dataset.

present in natural or born-digital images. Text strings can be classified with respect to their orientation in the following way:

vertical:	the characters within the text line are positioned in a vertical configuration.
inclined text:	the text line is inclined.
curved text:	the characters within the text line follow a curve.
irregular orientation	the characters within the text line follow an irregular path.

Fig. 3: An example of sheared text.

The shear transform, usually linked to italic text fonts, maps a set of coordinates such that one coordinate remains fixed, while the other ones are shifted creating a skew effect. A sheared text example is given in Figure 3. Finally, texts in perspective are usually subject to foreshortening, which is an optical illusion that makes objects appear shorter than they actually are because they are angled towards the camera view. Based on this angle, the foreshortening can be classified into horizontal or vertical foreshortening, as illustrated in Figure 4.

Fig. 4: Types of foreshortening transformations[1]: (a) horizontal foreshortening; (b) vertical foreshortening.

The recognition of such texts (subject to rotation, shearing or perspective transformations) is vital for many text understanding systems. However, due to severe distortions, traditional OCRs have difficulties in providing truthful transcriptions. Most of the OCRs require text regions to be horizontal and taken from a front-parallel view in order to be correctly recognized. The increasing popularity of natural scene acquisitions for text detection purposes has re-enforced the need of introducing an intermediate stage, to correct (or rectify) text, and then ameliorate the performance of existing OCRs.

[1] Credit http://ocrserv.ee.tsinghua.edu.cn/icdar2015_str/

6.2 Related work

In the literature, the problem of distorted text has been handled in different manners. Some works tackled this by proposing powerful recognition stages capable of managing distorted characters. On the opposite, many works first rectify the distortions, then the recognition. The first category of approaches relies on feature learning. However, when texts are severely distorted, these methods fail to provide a correct transcription. In such cases, the rectification procedure is a better alternative. Special types of text rectification target multi-oriented, italic or text in perspective.

Orientation rectification. Several approaches for correcting curved text strings have been proposed in the literature. Authors in [Vasudev et al., 2007] described a method based on an ellipse drawing algorithm that rectifies arc-form text strings. Later, in [Kasar and Ramakrishnan, 2013] a different technique has been proposed that invokes the spacial regularity properties of a text and the characteristics of its adjacent components. The authors in [Roy et al., 2008] proposed a recognition method of English characters invariant to orientation or scale. The recognition is based on the extraction of a set of features (angular information, circular ring and convex hull) from each character and on the use of a SVM classifier.

Italic rectification. Some works have proposed methods to rectify italic texts to enhance the performance of OCRs that have difficulties in providing accurate transcription of sheared texts. The authors in [Zhang et al., 2004] proposed an approach based on the statistical analysis of stroke patterns extracted from the wavelet decomposition of text images. In [Fan and Huang, 2005] authors introduced a method that rectifies italic texts using a shear transform. First, the characters are classified into three classes of angles. Then, the shear angle is determined differently for each character based on its corresponding italic class.

Perspective rectification. Perspective recovery needs to be applied when the camera axis is not perpendicular to the text plane. When a text is in perspective, the characters change their original structure. This makes OCRs perform poorly and produce low accuracy scores. However, a series of works proposed recognition modules capable of identifying oriented characters or texts in perspective. The authors in [Lu and Tan, 2006] proposed a recognition technique capable of recognizing characters in perspective by extracting perspective invariant features such as character ascenders and descenders or number of centroid intersections. Cross ratio spectrum and Dynamic Time Wrapping techniques were employed during the recognition process in [Li and Tan, 2008a, Li and Tan, 2008b, Zhou et al., 2009]. In [Phan et al., 2013] SIFT features were extracted to recognize texts in perspective in different orientations. To correct the perspective distortion, many works rely on the homography transformation [Myers et al., 2005, Ye et al., 2007, Cambra and Murillo, 2011, Kiran and Murali, 2013]. In [Ye et al., 2007], the rectification is done based on a correlation between a set of feature points and a plane-to-plane homography transformation. The extension of this work, presented in [Cambra and Murillo, 2011], consists of an optimization of parameters of the homography. The method in [Kiran and Murali, 2013] implied a first stage where text borders are captured using geometry based segmentation and then corner points are selected using the Harris corner detector. The authors in [Merino-Gracia et al., 2013] implied parallel rectification using an homography and a shearing transform. The method first proposes a horizontal foreshortening by detecting the upper and lower lines bounding the text region. Next, the vertical foreshortening and shearing are done by using a linear regression based on the variation of shear characters.

The authors in [Chen et al., 2004b] used an affine transformation to correct the perspective deformations, but the method requires the camera parameters to be known. Such an assumption was also required in the work in [Clark et al., 2001]. The borderline analysis was implied in [Ferreira et al., 2005, Liu et al., 2008]. The main problem of these approaches is that they rely on the hypothesis that text regions were previously bounded by rectangles.

Work in [Zhang et al., 2013] used the Transformed Invariant Low-rank Textures (TILT) algorithm to rectify English, Chinese and digit characters. The method presented in [Bušta et al., 2015] proposed a skew text rectification in real scene images based on five skew estimators used for character segmentation (or polygon approximation): Vertical Dominant (VD), Vertical Dominant on Convex Hull (VC), Longest Edge (LE), Thinnest Profile (TP) and Symmetric Glyph (SG). In [Myers et al., 2005], the authors use a projective transformation to correct text in perspective. The parameters used for the rectification are derived from a series of features extracted from each text line, such as top and baselines of a text or the dominant vertical direction of character strokes. In [Yonemoto, 2014] a correction method based on quadrangle estimation is proposed, which supposes that the text contains a sufficient number of horizontal and vertical strokes. Authors in [Hase et al., 2001] proposed a generic method to correct inclined, curved and distorted texts. Text is first classified with respect to the alignment and distortion of its characters, then different types of corrections are applied. A rectification approach for license plate images was proposed [Deng et al., 2014] using the Hough transform and different types of projections. The method, based on finding parallel lines, consists of two transformations: a horizontal tilt and a vertical shear transform.

Many of the approaches discussed above correct the text of individual text lines. Some works proposed rectification algorithms on whole documents. The work in [Stamatopoulos et al., 2011] targets the rectification of distorted documents. It performs a curved surface projection, a word baseline fitting and an horizontal alignment. The authors in [Liang et al., 2008] proposed a rectification method for planar and curved documents by estimating $3D$ document shapes from texture flow information.

6.3 Contributions

We have presented in this chapter a short survey of the methods to rectify texts before their transcription by an OCR. Our contributions, that will be described in the next chapters, concern a rectification method that can simultaneously correct rotation, shearing and perspective deformations. It uses an homography that maps the image coordinates onto the world coordinate system and brings the deformed texts to a front-parallel view. Contrary to other works that use the same approach and which imply an affine transformation for perspective correction followed by a shearing rectification that corrects the perspective correction, the proposed method uses a single affine transformation computed from a precise quadrangle estimation of the distorted text. The validation of this method will be done on a recent dataset, used during the ICDAR 2015 *Competition on Scene Text Rectification*. It contains a very large amount of challenging texts, from synthetic and real scenes, with different transformations. Moreover, we will show that some stages of the rectification procedure can be used as a simple and efficient approach to correct multi-oriented texts, adapted to curved, arc-form or irregularly oriented texts. It relies on the properties of the local neighborhood of each character of an oriented text. Some preliminary results are given to show the potential of rectification of our proposed method.

The entire rectification procedure is detailed in Chapter 7, while the experimental results are presented in Chapter 8.

7 Proposed text rectification method

Contents

In this chapter we describe a text rectification method dedicated to text strings in perspective and curved texts. The proposed approach relies on a well-known projective transformation that maps the coordinates of the deformed text onto the world coordinate system, which requires a very accurate approximation of the boundaries of the text. This approximation represents one of the main contributions of this chapter for which we propose a complex solution that can be used to rectify highly distorted texts. This chapter also proposes a simple and efficient method to correct some curved text strings. It consists in approximating the orientation of a character with respect to the location of its neighbors.

The rectification is a challenging stage in a text understanding system due to the diversity of text deformations. Texts can be distorted by the perspective view (fore-shortening), have different orientations (e.g. inclined, vertical or multi-oriented) or present shearing effects (e.g. italic format). Moreover, the varying direction of the characters of a text string can also affect the rectification process. A character is said to be *upright* if it is orthogonal to the horizontal direction and consequently does not need any correction. Otherwise, the character is said to be *rotated*. In some cases the orientation of the characters does not follow the direction of the text string. Figure 1 gives different text string types.

Our proposed rectification method is dedicated to two types of deformations: text strings in perspective and curved texts. The perspective correction approach, described in Section 7.1, is the main contribution of this part and concerns the correction of one-directional texts, namely texts that follow a straight line. On the other hand, we show that we can use an extension of this work to easily rectify curved texts. Final conclusions are provided in Section 7.2. In the following, we introduce some notations that will be used for the description of the proposed approach.

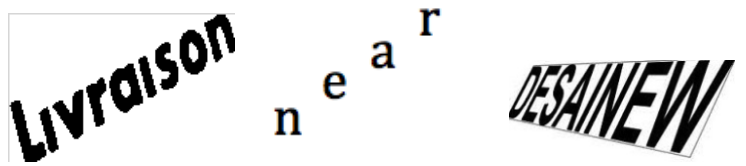
Notations. Let us consider a text string as a set of N characters defined as $\mathscr{C} = \{C_i\}_{i=1..N}$, where C_i is the individual CC corresponding to the i^{th} character. We define $\mathscr{G} = \{G_i\}_{i=1..N}$ as the set of centroids

a: horizontal text with upright characters

b: horizontal text with characters in perspective

c: vertical text with rotated and italic characters

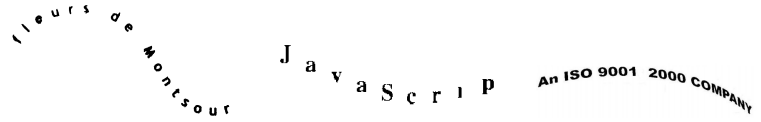
d: vertical text with upright characters



e: inclined text with rotated characters

f: inclined text with upright characters

g: inclined text with characters in perspective



h: curved text with rotated characters

i: curved text with upright characters

j: curved text with some characters in perspective ("COMPANY")

Fig. 1: Different types of texts.

corresponding to each CC in $\mathscr{C}$. Similarly, we denote by $\mathscr{W} = \{W_i\}_{i=1..N}$ the set of weighted centroids that belong to each CC in $\mathscr{C}$. We classify the CCs into two categories:

extremity CCs: CCs corresponding to the first or last characters of the text string (in the order of reading). The two extremities will be referred to as C_{e1} and C_{e2}, where $e1, e2 \in [1, N]$.

inner CCs: CCs corresponding to any of the characters that are located between the two extremity CCs.

Note. A weighted centroid is calculated by considering each pixel intensity as a weight inside the CC bounding box, whereas the traditional one is the center of the rectangular bounding box, *i.e.* the intersection of its two diagonals. Generally, for symmetrical characters, such as"O", "I", these two centroids are the same. However, when dealing with asymmetrical characters, especially ascender and descender ones, the weighted centroids provide better references for the text orientation approximation. The difference of the two centroids is illustrated in Figure 2.

7.1 Text rectification process

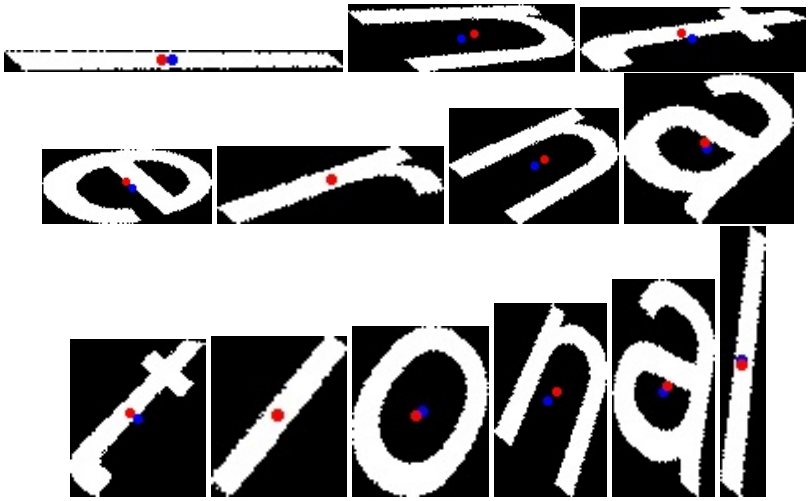


Fig. 2: Classical (blue) and weighted (red) centroids of the characters in the text string of Figure 5.

7.1.1 Overview of the text rectification process

Generally, the perspective rectification process relies on the availability of extrinsic camera parameters. If these parameters are known, they can be used to compute the homography matrix that maps the camera coordinates onto the world coordinate system. Otherwise, as in our case, we need to compute this homography differently. To find the homography matrix and produce the rectified text image we imply several stages, listed below and illustrated in Figure 3.

- The method first relies on a CC filtering, described in Section 7.1.2 during which punctuation signs and point over some characters are temporarily removed.
- The filtering is followed by an extremity CC identification procedure, discussed in Section 7.1.3, which targets the identification of the first and last characters of a text string.
- The process then estimates a precise quadrangle (see Section 7.1.4) that bounds the distorted text. The four points that define the quadrangle will be used to compute the homography matrix. Finally, this homography transformation is used to map all the points of the deformed text onto a parallel-front plane, as explained in Section 7.1.5.

Finally, we show how we can use some of the information acquired during this rectification process to propose an efficient technique to correct multi-oriented text strings. This is presented in Section 7.1.6.

The rectification approach relies on a series of hypotheses on text:

- the text needs to be upward;
- each character needs to be a separate CC;
- text needs to have a single orientation.

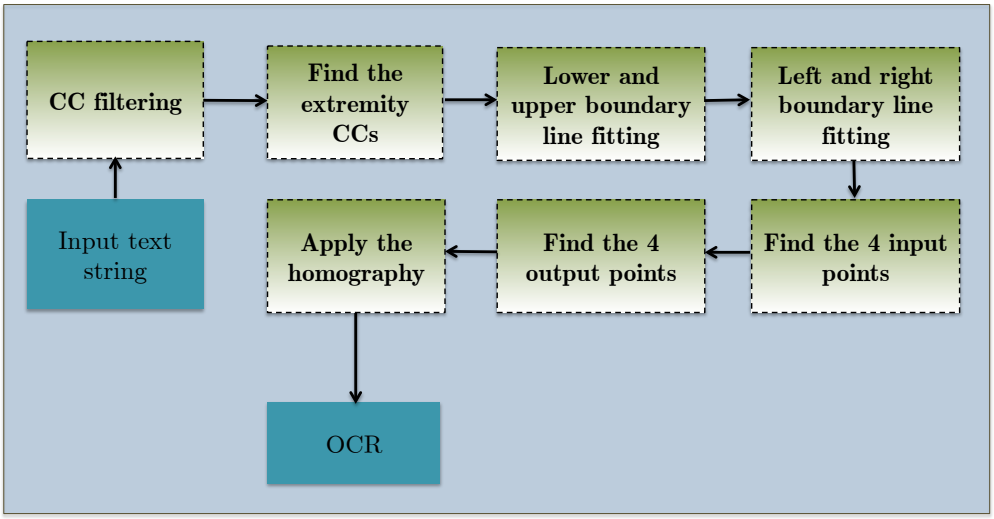


Fig. 3: Proposed rectification process.

7.1.2 Connected component filtering

Before applying the rectification, we need to filter the CCs and remove the small punctuation marks such as ".", "," or ":" or points over some characters such as "i" and "j". Such a removal is needed because the entire text correction is based on the relative position of a CC with respect to the other ones. We define l_d^i as the length of the diagonal of the box bounding of C_i computed as:

$$l_d^i = \sqrt{h_{C_i}^2 + w_{C_i}^2}, \tag{7.1}$$

where h_{C_i} and w_{C_i} are respectively the height and width of the bounding box of C_i. We also define l_d^{av} as the average of all diagonals lengths such that:

$$l_d^{av} = \frac{\sum_{i=1}^{N} l_d^i}{N} \tag{7.2}$$

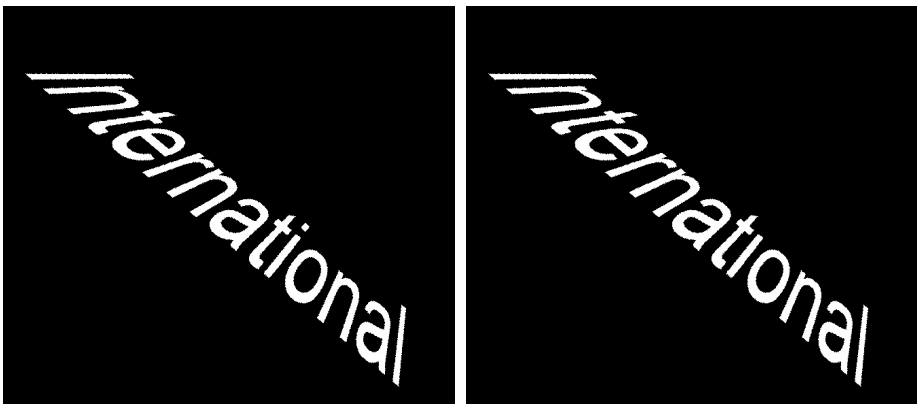
Hence, C_i is kept during the filtering procedure as long as its diagonal satisfies the following constraint:

$$l_d^i > l_d^{av} \cdot T_{pt}, \tag{7.3}$$

where T_{pt} is a threshold that was experimentally set to 0.35. This constraint removes all CCs whose diagonal is considerably smaller than the average diagonal. Figure 4 gives an example of filtering.

7.1.3 Extremity connected components

After the CC filtering, we need to find the two extremity CCs, *i.e.* corresponding to the first and to the last characters. This requires several steps. First, we compute the weighted centroids of each CC. Next, we deduct the text orientation by approximating the *reference* line that best fits all centroids. Following this, we search for the left and right neighbors of each CC. The angle between each pair of neighbors is then computed in order to obtain the extremity CCs. Finally, we decide which of the two extremities is the

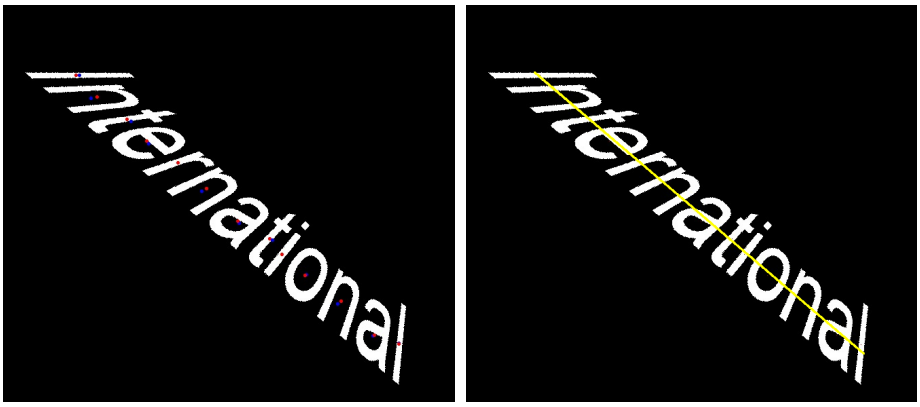


a b

Fig. 4: A distorted text string: (a) before filtering; (b) after filtering.

first and which one is the last characters based on some pre-defined assumptions. The entire procedure is illustrated in Figure 6.

Fitting the reference line. An approximation of the text orientation is obtained by using the LSM, that can fit a reference line to the set of weighted centroids $\mathcal{W}$. The slope of this line, called L_{Ref} gives an approximation of the text orientation. Figure 5 shows examples of centroids and a reference line for the text string "International".



a b

Fig. 5: Centroids and reference line fitting using LSM: (a) classical centroids are in blue, while weighted centroids are in red; (b) the reference line that best fits the weighted centroids in yellow.

Identifying the two extremity CCs. First we identify its two closest neighbors for each CC C_i, denoted as C_i^{n1} and C_i^{n2}. If C_i is the first extremity, then its two nearest neighbors will be the two following characters. If C_i is the last extremity, its two nearest neighbors will be its two preceding characters. If C_i is not an extremity, but an inner CC, then its two closest neighbors will be its predecessor and its successor. Let W_i^{n1} and W_i^{n2} be the weighted centroids of the two neighbors of C_i. We then define l_i^{n1} and l_i^{n2} the lines passing through W_i to its two neighbors:

$$l_i^{n1} = (W_i^{n1}, W_i) \tag{7.4}$$

$$l_i^{n2} = (W_i^{n2}, W_i) \tag{7.5}$$

Then, we introduce θ_i as the orientation angle of C_i computed as:

$$\theta_i = angle(l_i^{n1}, l_i^{n2}) \tag{7.6}$$

All CCs for which this angle is smaller than 45° are selected as extremity CC candidates. If more than two CCs satisfy this constraint, we compute the largest distance between each pair of candidate CCs. The pair of CCs for which the distance between their centroids is the largest are identified as the two extremities C_{e1} and C_{e2}, with $e1, e2 \in [1, N]$. This stage is illustrated in Figures 6a, 6b and 6c.

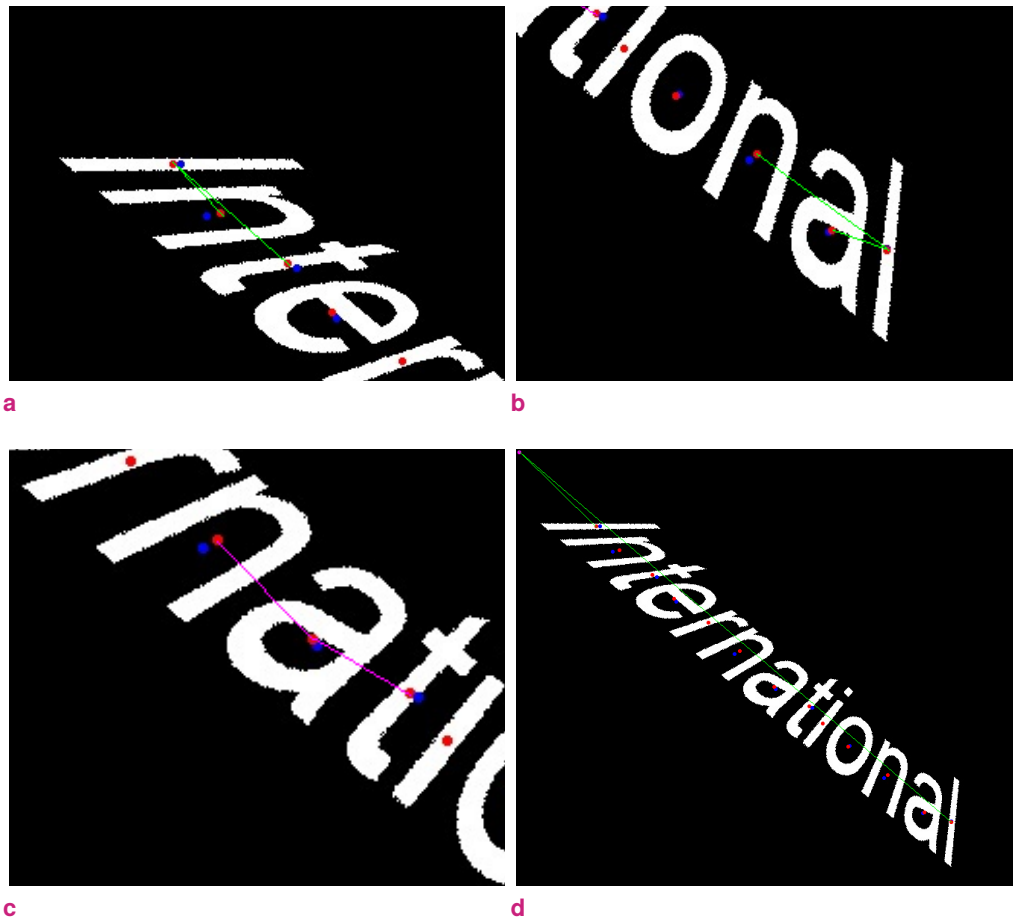


Fig. 6: The procedure for finding the extremity CCs: (a)-(b) the angles between the lines (in green) passing through the centroids of the two extremities and the centroids of their two closest neighbors; (c) the angle between the lines (in magenta) passing through the centroid of an inner CC ("a") and the centroids of its two closest neighbors ("n" and "t"); (d) the distance (in green) between the weighted centroids of the two extremities and the left upper origin in magenta.

Identifying the first and last extremities. Once the two extremity CCs have been localized, we need to identify which one is the first character and which is the last one, and then determine the order of reading of the text string. Namely, we determine which one of C_{e1} or C_{e2} corresponds to C_1 and which one corresponds to C_N. Depending on the orientation, an upward text string can have the first and last characters situated in different locations, as listed below.

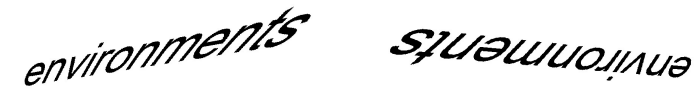
1. For an horizontal text, the first character is the left-most one, while the last character is the right-most one (see Figure 7a).
2. For an inclined text in which the first character is in the upper left corner, the last character is in the bottom right corner (see Figure 7b).
3. For an inclined text in which the first character is in the bottom left corner, the last character is in the upper right corner (see Figure 7c).

a: Left to right reading order. **b:** Top-left to bottom-right reading order. **c:** Bottom-left to top-right reading order.

Fig. 7: Reading order of a text string depending on its orientation.

If the text is vertical, we rotate the text to the horizontal and then apply the rectification procedure. To determine the verticality of the text line the angle of the reference line needs to be in the interval $[80°, 100°]$. The correct rotation angle is difficult to determine, as better explained at the end of in Section 7.1.6. In our experiments, we have however set this angle to $-90°$.

Note. These assumptions are valid only for texts containing upward characters. When a text contains downward characters, we use the opposite of the previous rules. Figure 8 shows two inclined strings with upward and downward characters.



a **b**

Fig. 8: Character orientation in a text string: (a) upward characters; (b) downward characters.

Given two points $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ belonging to reference line L_{Ref}, we denote $m(L_{Ref})$ its slope given by:

$$m(L_{Ref}) = \frac{y_2 - y_1}{x_2 - x_1} \tag{7.7}$$

Depending on the orientation of the text line, the slope can be positive, negative, zero or undefined.

positive:	the orientation of the line is from bottom-left to top-right;
negative:	the orientation of the line is from top-left to bottom-right;
zero	the line is horizontal;
undefined	P_1 and P_2 have the same x-coordinates: the line is vertical.

Based on these assumptions and on the slope $m(L_{Ref})$, we can find the two extremities C_1 and C_N. If the slope $m(r) \in [-0.1, 0.1]$, the text is considered as horizontal and hence we determine the first and last

characters depending on the y-coordinates of the weighted centroids of the two CCs. If $m(r) < -0.1$, the text is inclined following a bottom-left to top-right direction. In this case we choose the CC closer to the bottom origin point defined as $O_b = (0, y_{max})$. If $m(r) > 0.1$, the text follows a top-left to bottom-right direction and the first and last characters are chosen based on the smallest distance between the upper origin point $O_u = (0,0)$ and the two centroids W_{e1} and W_{e2}. This procedure is detailed in Algorithm 1.

Algorithm 1 Algorithm for identifying the first and last extremities.

procedure FINDFIRSTLASTEXTREMITIES(G_{e_1}, G_{e_2})
  if $|m(r)| \leq 0.1$ **then**
    if $y_{e_1} < y_{e2}$ **then**
      $C_1 = C_{e_1}$ and $C_N = C_{e_2}$
    else
      $C_1 = C_{e_2}$ and $C_N = C_{e_1}$
    end if
  else
    if $m(r) < -0.1$ **then**
      $d_1 = distance(O_b, G_{e_1})$
      $d_2 = distance(O_b, G_{e_2})$
      if $d_1 < d_2$ **then**
        $C_1 = C_{e_1}$ and $C_N = C_{e_2}$
      else
        $C_1 = C_{e_2}$ and $C_N = C_{e_1}$
      end if
    else
      $d_1 = distance(O_u, G_{e_1})$
      $d_2 = distance(O_u, G_{e_2})$
      if $d_1 < d_2$ **then**
        $C_1 = C_{e_1}$ and $C_N = C_{e_2}$
      else
        $C_1 = C_{e_2}$ and $C_N = C_{e_1}$
      end if
    end if
  end if
end procedure

7.1.4 Quadrangle approximation

In this stage we are interested in finding the quadrangle that best fits a text string. This consists in identifying the four lines that bound the text, referred here as the *bottom* (L_b), the *upper* (L_u), the *left* (L_l) and the *right* (L_r) lines.

Bottom and up boundary line fitting. Let us consider $\mathscr{P}^u = \{P_i^u\}$ and $\mathscr{P}^b = \{P_i^b\}$ the sets containing the upper and lower extremity points of C_i respectively. In order to find these points we use the slope of the reference line as a guideline in the following manner:

1. We plot lines parallel to the reference line L_{Ref} in two directions (positive and negative) corresponding to the upper and bottom points, at different distances. We denote such a line as L_s^d,

where d is the distance to L_{Ref} and s the direction sign. The procedure consists in, for each direction sign, plotting lines, parallel to L_{Ref}, by incrementing the distance by 1 until we find a line L_s^d that intersects any CC C_i and L_s^{d+1} does not. Then, we retrieve all intersection points between the line L_s^d, situated at the distance d from L_{ref}, and C_i and store them in $\mathscr{P}^u$ and $\mathscr{P}^b$ (see Figures 9a and 9b).

2. The LSM is then used on the set of points $\mathscr{P}^u$ to get an approximation of the upper line L_u and on $\mathscr{P}^b$ to get and approximations of the bottom line L_b (see Figure 9c).

3. Finally, we check if lines L_u and L_b correctly bound the text string. If L_u or L_b intersects the set of CCs $\mathscr{C}$, the lines are shifted (parallel to L_u or L_b) until they perfectly bound the text string (see Figure 9d).

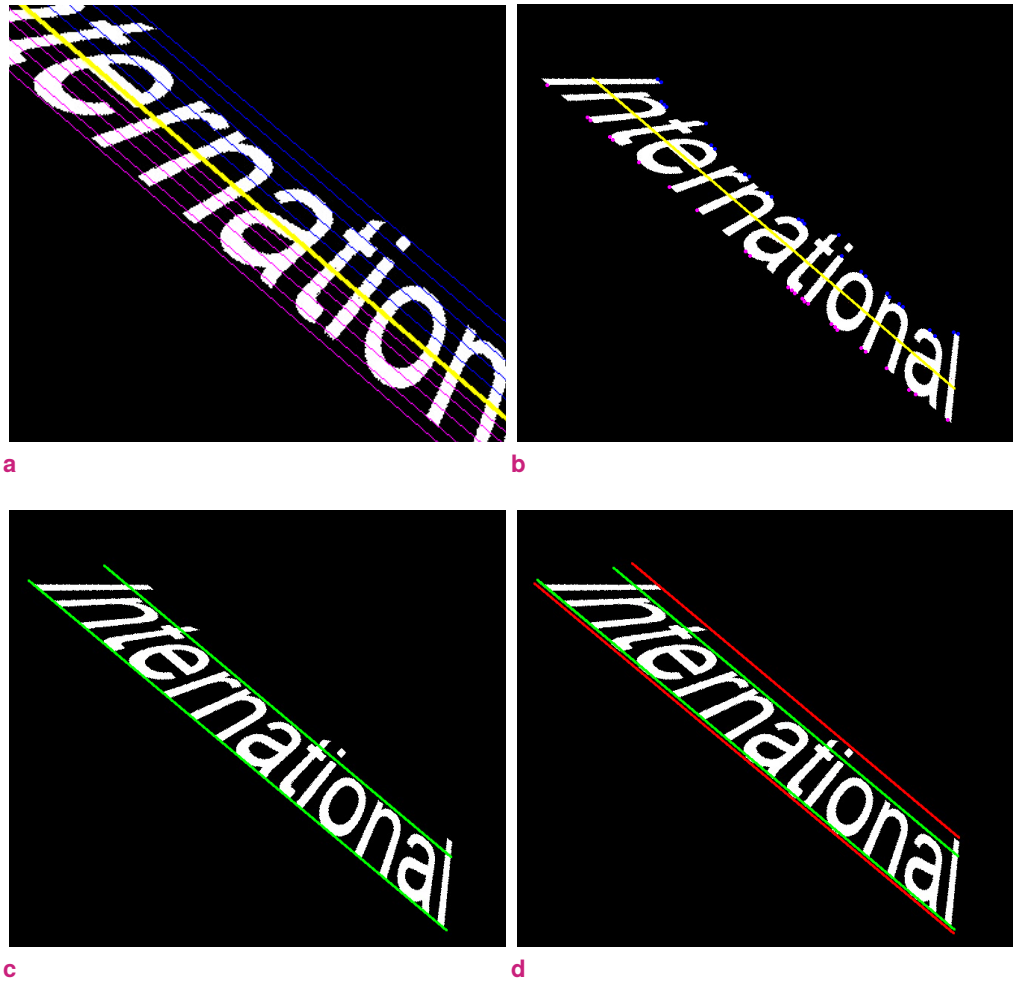


Fig. 9: Lower and upper boundary line fitting procedure: (a) parallels to the reference line in both directions: upper (blue) and bottom (magenta); (b) extremity points: upper (blue) and bottom (magenta); (c) LSM line fitting of the lower and bottom extremity points; (d) shifting of the initial upper and lower lines.

Left and right boundary line fitting. We call $\mathscr{P}^l = \{P_i^l\}$ the set containing the left extremity points and $\mathscr{P}^r = \{P_i^r\}$ the set containing the right extremity points. To obtain the left and right boundary lines, the positions of the first and last CCs are used, following the stages described below:

1. Find the left and right extremity points following the same reasoning used to find the upper and lower extremity points. The difference here is that these extremity points are detected only with respect to the extremity CCs C_1 and C_N. Let us define L_{Ref}^P the line normal to L_{Ref} in W_i. The procedure consists of tracing parallels to L_{Ref}^P with a distance of 1 until the CC extremities are not crossed anymore by the parallel lines. All border points that belong to both the last parallel line and the extremity CC are stored into the two sets $\mathscr{P}^l$ and $\mathscr{P}^r$ (see Figures 10a and 10b).

2. For each of the two sets $\mathscr{P}^l$ and $\mathscr{P}^r$ average left point P_{av}^l and right point P_{av}^r are computed:

$$P_{av}^l = \left(\frac{\sum_j^{|\mathscr{P}^l|} x_{P_j^l}}{|\mathscr{P}_l|}, \frac{\sum_j^{|\mathscr{P}^l|} y_{P_j^l}}{|\mathscr{P}_l|} \right), \quad P_{av}^r = \left(\frac{\sum_j^{|\mathscr{P}^r|} x_{P_j^r}}{|\mathscr{P}_r|}, \frac{\sum_j^{|\mathscr{P}^r|} y_{P_j^r}}{|\mathscr{P}_r|} \right) \tag{7.8}$$

3. Given the point P_{av}^l (respectively P_{av}^r) we look for the line passing through this point that best fits the extremity C_1 (respectively C_N). Let us consider L_P the line, normal to L_{Ref} in P_{av}^l. The best fitting left line is obtained by rotating the line L_P (see Figures 10c and 10c) until it covers the maximum number of border points. Algorithm 2 describes the procedure used to find the rotation angle of the line L_P that best fits the CC borders. This step is also illustrated in Figures 10e and 10f.

4. Finally, we check if the lines L_l and L_r correctly bound the text string. If L_l or L_r intersects the set of CCs $\mathscr{C}$, the lines are shifted (parallel to L_l or L_r) until they perfectly bound the text string (see Figure 10f).

7.1.5 Homography

In general, in order to rectify a perspective distorted image, one needs to rely on the extrinsic camera parameters. However, in many situations these parameters are not known and hence other correction approaches need to be used. For example, by using a perspective projection matrix, the image coordinates can be mapped onto a parallel-frontal plane. This type of transformation is also known as the homography. The relationship that maps a point (x, y) from the perspective plane onto point (x', y') from the normalized plane is defined as:

$$c \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = H \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{7.9}$$

where c is any non-zero constant and $H = \begin{pmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{pmatrix}$ is the homographhy matrix. In order to compute H we need eight points: four points from the image plane and their corresponding four points from the real world plane. The perspective distortion is then corrected by applying Equation 7.9 to all

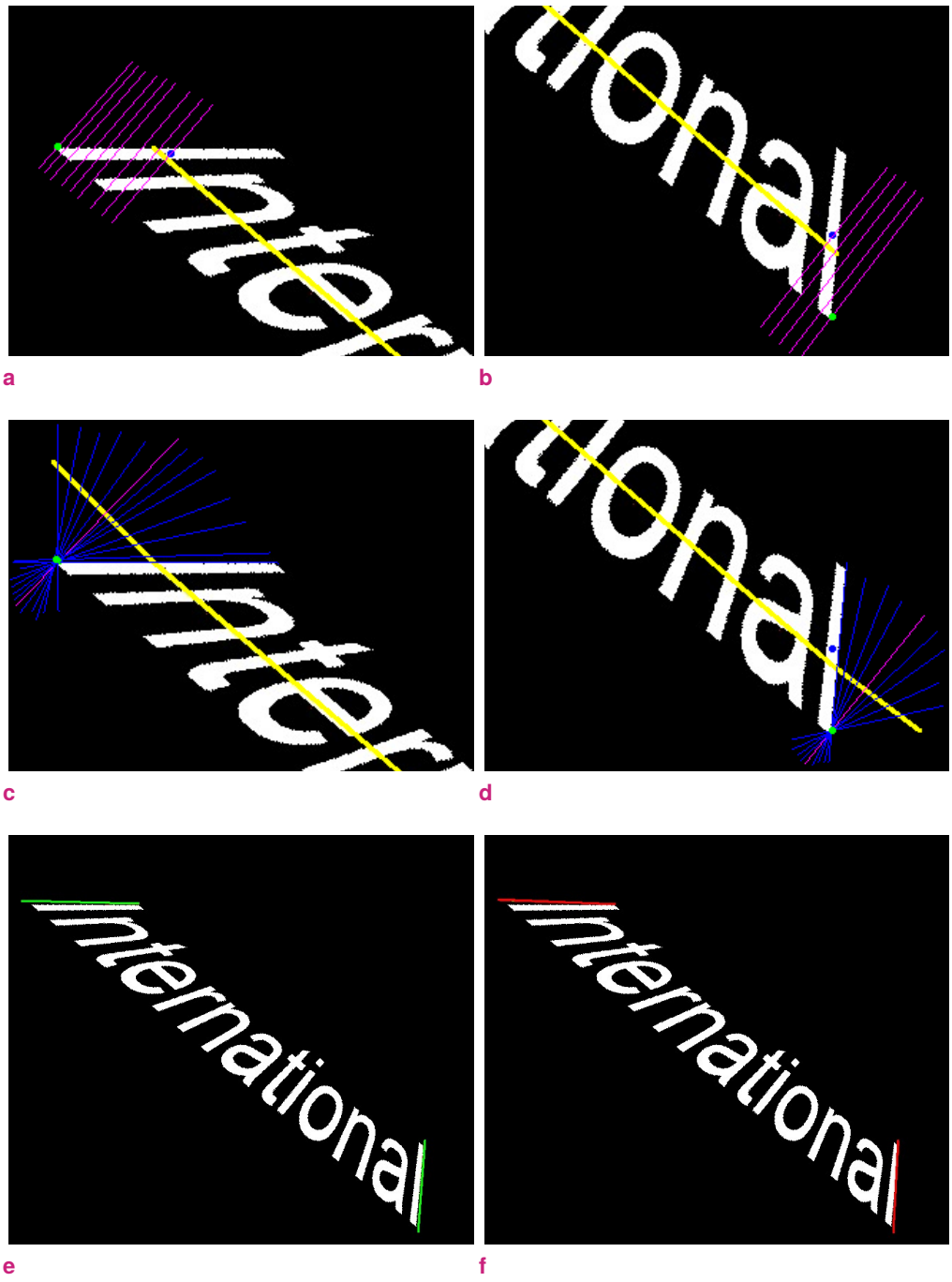


Fig. 10: Left and right boundary line fitting procedure: (a)-(b) Finding the left and right extremity points; (c)-(d) line variation to find the best fitting left and right lines; (e) left and right boundary lines; (f) the left and right boundary lines after shifting.

Algorithm 2 Algorithm for identifying the left or right line boundaries.

procedure FINDLEFTRIGHTBOUNDINGLINE(P_{av})
  for $\theta = -88°; step < 88°; step+ = 0.05°$ **do**
    $m' = m(L_P) + tan(\theta)$
    l' is the line with the slope m' passing though P_{av}
    if $m' - m_r > 0.05$ **then**
      for all points P on l' that belong to CC **do**
        l'_p is the parallel line to l' at a distance = 6
        c_T is the number of points on l'_p that belong to CC
        c_F is the number of points on l'_p that do not belong to CC
      end for
    end if
    if $c_T = 0$ **then**
      if $c_F > c_{max}$ **then**
        $c_{max} = c_F$
        $m_e = m(L_P) - tan(\theta)$
      end if
    end if
  end for
return θ
end procedure

points (x, y) in the image plane to get the real world coordinates (x', y'). In our case, we use the four corners of the quadrilateral that bounds the distorted text and provide the coordinates of the rectangular plane onto which we want to map this text string. We will refer to these four coordinate pairs as the *input* and *output* points.

Input point set detection. By using approximations of lines L_u, L_b, L_l and L_r we can get the four corners (P_1, P_2, P_3 and P_4) as their intersections. To form the input quadrilateral, we determine the order of its corners in a clockwise way, depending on the orientation of the text string as described in Algorithm 3.

Algorithm 3 Algorithm for determining the order of corners of the quadrangle that bounds the text.

procedure DECIDEORDEROFPOINTS(G_{e_1}, G_{e_2})
  if $|m(r)| \leq 0.1$ **then**
    $A = P_4$ and $B = P_3$ and $C = P_2$ and $D = P_1$
  else
    if $m(r) < -0.1$ **then**
      $A = P_4$ and $B = P_3$ and $C = P_2$ and $D = P_1$
    else
      $A = P_1$ and $B = P_2$ and $C = P_3$ and $D = P_4$
    end if
  end if
end procedure

Fig. 11: Perspective distortion rectification: (a) bounding quadrangle estimation; (b) rectified text.

Output point set detection. The output quadrangle is set such that the text proportion is preserved.

$$width_O = \max(dist(P_1, P_2), dist(P_3, P_4)) \tag{7.10}$$

$$height_O = \max(dist(P_2, P_3), dist(P_4, P_1)) \tag{7.11}$$

The output quadrangle is defined by the set of points P'_1, P'_2, P'_3 and P'_4 such that:

$$P'_1 = (0,0)$$

$$P'_2 = (2 \times width_O, 0)$$

$$P'_3 = (2 \times width_O, 2 \times height_O)$$

$$P'_4 = (0, 2 \times height_O)$$

Then, for each point P_i belonging to the input quadrilateral we get a corresponding point P'_i of the output quadrangle using Equation (7.9):

$$P'_i = HP_i \quad i = 1,..,4 \tag{7.12}$$

7.1.6 Using the orientation angle to correct irregular oriented texts

In this section we show that we can use the orientation angle of each CC, introduced in Section 7.1.3, to correct a curved text.

A text can be straight or curved. The text line type can be determined based on the relative positions of the inner CCs with respect to the extremity CCs. Namely, by plotting a line between the centroids of the two extremities C_{e1} and C_{e2}, we can compute the variance of the distance from all inner character centroids to that line. If the variance is small, then the text string follows a straight line. Otherwise, the text string follows a curved line, as shown in Figure 12.

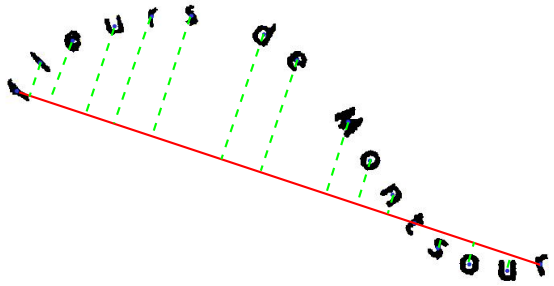


Fig. 12: Text line type estimation based on the distance from the inner characters to the line (C_{e1}, C_{e2})

For curved text our rectification can not be based on only one orientation, since each CC can have a different direction. The chosen approach estimates the orientation of a character based on its local neighborhood. Namely, each C_i is assigned a different orientation angle, θ_i, defined from its two nearest neighbors, C_i^{n1} and C_i^{n2}.

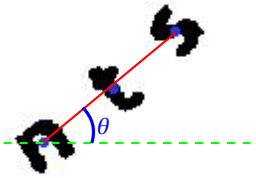


Fig. 13: Inner character orientation. The character "t" needs to be rotated by the angle θ

For inner CCs C_i the orientation is given by the slope of the line linking the centroids of its two closest neighbors, *i.e.* passing through (W_i, W_i^{n1}) and (W_i, W_i^{n2}). The value of the slope indicates the rotation angle. A positive slope implies a clockwise direction, while a negative one involves an anti-clockwise direction for the rotation. If the slope is equal to zero, the character is aligned horizontally and does not need to be rectified. We can therefor rectify the text by rotating each character such that its neighborhood line is aligned horizontally.

For extremity characters, the orientation cannot be estimated in the same way. Several directions can be taken into consideration:

1. assign to each extremity the orientation of its nearest neighbor;
2. assign to each extremity the orientation based on the slope of the line linking the centroid of the extremity with the centroid of its nearest neighbor;
3. predict for each extremity a possible orientation based on its two following neighbors.

This orientation rectification stage still remains challenging due to a number of problems related to vertical texts: the uncertainty of using a rotation or a translation transformation and the choice of the orientation angle sign that can directly affect the reading order of a text string.

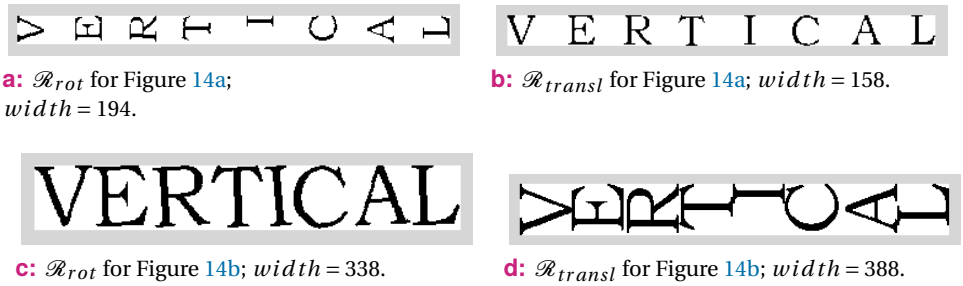
Rotation versus Translation. Figure 14a depicts the case of a vertical line text with upright characters. Figures 14b and 14c on the other hand, show two vertical text lines with characters that follow the direction of the line. If we would use the orientation estimation stage seen in Section 7.1.6, for the text strings in Figures 14a and 14b, the characters would be rotated 90° clockwise in both cases, but would

a: Vertical line text with upright characters; **b:** Vertical line text with rotated characters (90°) **c:** Vertical line text with rotated characters (−90°)

Fig. 14: Vertical text lines with different character orientations.

correctly rectify only the text in the second case. Indeed, the rectification for the first case should only consist in transposing the characters into an horizontal line, as upright characters situated on oriented text lines do not need a rotation transformation.

To determine if a text should be rectified by rotating or translating the characters a possibility would be to take into account the geometric properties of characters. The majority of characters in a text string are higher than larger. Hence, for a given text line, two rectification transformations can be applied: a rotation transformation, $\mathscr{R}_{rot}$, with respect to the line direction, and a translation transformation, $\mathscr{R}_{transl}$, which simply translates the characters onto a horizontal line. By comparing the widths of the two rectified text strings we choose the rectified text string that has the smaller width. This is illustrated in Figure 15. For Figure 14a the correct rectification is done by a translation of the characters on a horizontal line, while the text string in Figure 14b needs to undergo a rotation. However, this assumption is not always true, namely for strings containing multiple occurrences of characters such as “m” or “w”, for which the width size is larger than the height size.



a: $\mathscr{R}_{rot}$ for Figure 14a; $width = 194$.

b: $\mathscr{R}_{transl}$ for Figure 14a; $width = 158$.

c: $\mathscr{R}_{rot}$ for Figure 14b; $width = 338$.

d: $\mathscr{R}_{transl}$ for Figure 14b; $width = 388$.

Fig. 15: Rotation versus translation.

Logical order of characters (reading order). In the context of latin alphabet, the order of reading horizontal texts is from left to right. While in horizontal, inclined or even curved texts, we can suppose that the first character is located somewhere in the left part, for a vertical text string, where the order of reading depends on whether the characters are rotated or straight, the first character can be located at the bottom or at the top. Moreover, Figure 14c shows a vertical text string with characters that need to be rotated by an angle og −90°, and for which the logical order of reading is from top to bottom, the same as in Figure 14a. No method has been implemented for differentiating the two cases represented by Figures 14b and 14c. If text strings are composed of lower case letters, a possible solution could be to use the ascender and descender frequencies to estimate the rotation angle and hence to determine the

reading order. However, if we deal with capital letter strings, the rotation angle estimation remains a challenging task.

7.2 Conclusion

In this chapter we have presented a perspective rectification method that accurately corrects highly deformed text strings. Moreover, we showed that some of the stages implied during the perspective correction can be used as an efficient way to correct curved texts. The proposed perspective rectification relies on a homographic transformation that maps the camera coordinates onto a parallel-front plane. The homographic transformation is powerful as it handles both rotation and perspective projections, including shearing effects. Hence, it can correct oriented or perspective deformed texts, as well as texts that are subject simultaneously to both an orientation and a perspective deformation. The performance of the homographic transformation depends on how accurate is the estimation of the quadrangle that bounds the distorted text. We proposed to use a two-stage procedure. First, we approximate the up and bottom boundary lines based on a reference line computed using the LSM. Secondly, we provide a precise estimation of the lines bounding the extremity characters by iterating all possible lines until finding the one that best bounds the two CCs. If an accurate approximation of the bounding quadrangle is provided, only one affine transformation can be sufficient to correct the transformations that contribute to the distortion of text. The techniques used in this method imply however some limitations. Namely, the deformed text should be in latin alphabet, each character of a string should be represented by a single CC and moreover the text should be upward only (rotated or not). Both the advantages and disadvantages of the rectification method are detailed in Chapter 8 through a set of experimentations performed on a large dataset.

In this chapter we have explained how the neighborhood information can be used as a precise estimation of a character's orientation. Texts that follow arc-form or curve line paths are challenging cases for which the traditional rectification techniques often fail, due to the existence of multiple orientations within a same string. Hence, our proposition relies on the hypothesis that the orientation of each character is given by the direction of the line passing through its two closest neighbors (*i.e.* its left and right neighbors). Some preliminary results of this hypothesis are presented in Section 8.2.3 of Chapter 8. Finally, in this chapter we have raised the challenges of rectifying vertical text lines. In this sense we have discussed some future perspectives regarding the problem of discriminating texts with upright characters from those with oriented characters. Moreover, we have exhibited the difficulty of identifying the logical reading order of characters when text lines are vertical.

Rectification experimental results

8

Contents

This chapter is dedicated to the experimental results obtained using the rectification method described in Chapter 7 on two datasets proposed during the ICDAR *2015 Competition on Scene Text Rectification. The advantages and the weaknesses of the method are both quantitatively and qualitatively evaluated.*

In this chapter we give the experimental results of the proposed rectification method, tested on text in perspective. We present the two datasets used during the experiments in Section 8.1. Next, we present the rectification results in Section 8.2. First, a visual evaluation is provided to demonstrate the efficiency of the method to rectify many cases (see Section 8.2.1). The accuracy performance scores on both datasets are provided and discussed in Section 8.2.2. Finally, preliminary results on irregular text orientation correction are shown in Section 8.2.3.

8.1 Datasets

The experimental results of the perspective rectification method are conducted on the two datasets used for two tasks during the ICDAR 2015 *Competition on Scene Text Rectification.*

Task 1: Synthetic text rectification competition (STRC'15). The first dataset contains synthetic texts obtained by applying on 1000 text samples random deformation types, such as rotation, shearing, horizontal fore-shortening and vertical fore-shortening with different parameters. Multiple deformations can be applied to an individual text block. 500 images contain Times New Roman font texts, while the other 500 samples contain Arial font texts. The synthetic dataset contains 2500 English and 2500 Chinese word samples. Figure 1 gives some examples of synthetically generated deformations applied to a text string.

Task 2: Real scene text rectification competition (RSTRC'15). The second task targets the rectification of real-scene texts and proposes a dataset derived from MSRA-TD500 (see Section 2.3) which contains many texts that are subject to orientation and perspective distortions. The real-scene dataset contains a subset of 60 image samples with English texts and a subset of 60 images with Chinese

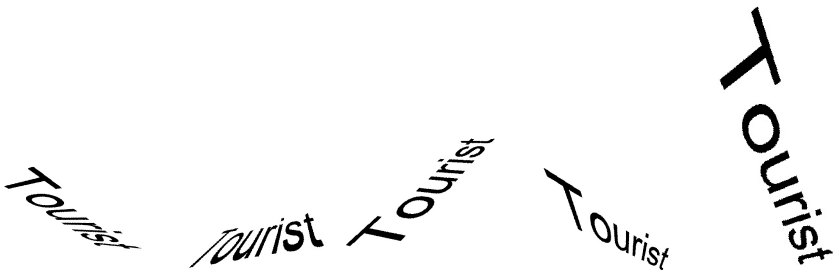


Fig. 1: Examples of synthetic deformations applied on the text string "Tourist" from the ICDAR 2015 dataset.

texts. For each image, a segmentation of text in connected components is also given, as illustrated in Figure 2.

Fig. 2: An example of image from the *Real scene text rectification competition* of ICDAR 2015 and its associated ground truth.

We have manually created the transcription GT corresponding to the two datasets as it was not provided by the competition organizers. This GT was used to compute the accuracy performance of the rectification method as seen in Section 8.2.2.

8.2 Rectification results

The rectification results are evaluated based on the text recognition accuracy obtained using the Tesseract OCR engine [Smith, 2007]. The validation of the proposed rectification approach is exclusively done based our own results, as no result of any participants at the ICDAR 2015 *Competition on Scene Text Rectification* were made public. The implementation of the rectification procedure was done in `C++` using the *Olena* image processing library [Levillain et al., 2014].

8.2.1 Qualitative results

To show the ability of the proposed rectification method, we provide in this section a qualitative evaluation, by visually exemplifying rectified synthetic and real-scene text strings together with their OCR transcription.

Synthetic dataset

The ICDAR 2015 synthetic dataset [Liu and Wang, 2015] contains different text length distributions: 7 strings of 2 characters; 29 strings of 3 characters, 67 strings of 4 characters; 67 strings of 5 characters; 82 strings of 6 characters; 86 strings of 7 characters; 57 strings of 8 characters; 41 strings of 9 characters; 34 strings of 10 characters and 30 strings of more than 10 characters. For each length distribution we provide five examples of distorted text strings together with the corresponding rectification result. Figures 3-12 illustrate text correction results for which the OCR transcription using Tesseract [Smith, 2007] is most of the time correct with regards to the GT.

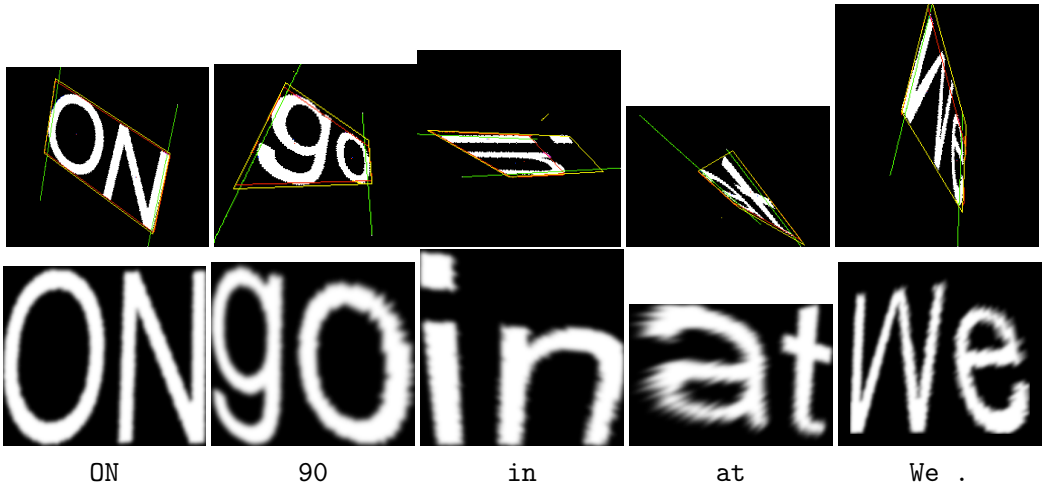


Fig. 3: Rectification results on text strings of two characters: original image (top), text rectification result (middle) and OCR transcription using Tesseract (bottom).

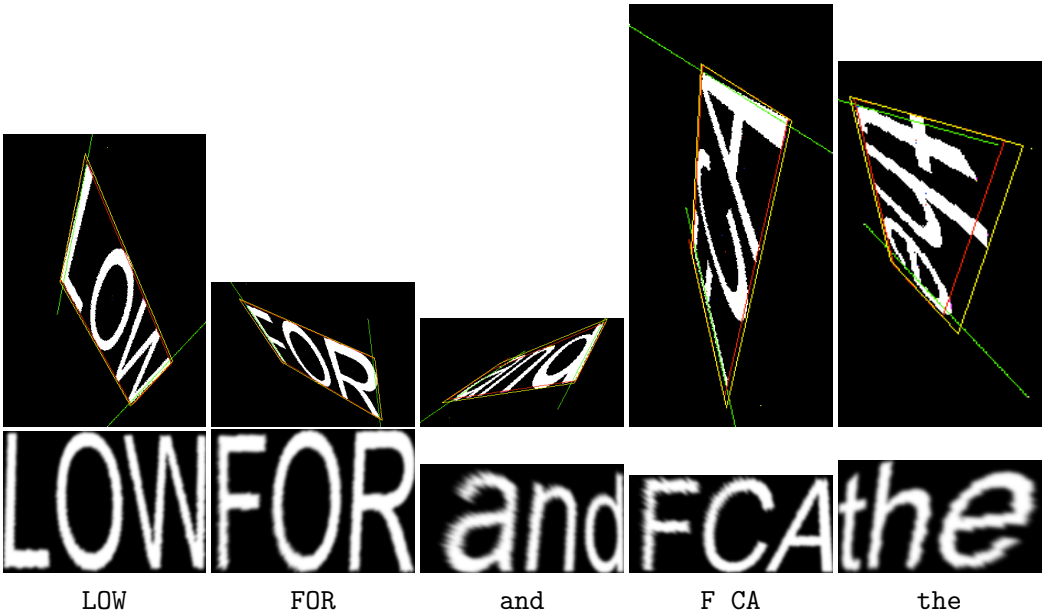


Fig. 4: Rectification results on text strings of three characters: original image (top), text rectification result (middle) and OCR transcription using Tesseract (bottom).

Fig. 5: Rectification results on text strings of four characters: original image (top), text rectification result (middle) and OCR transcription using Tesseract (bottom).

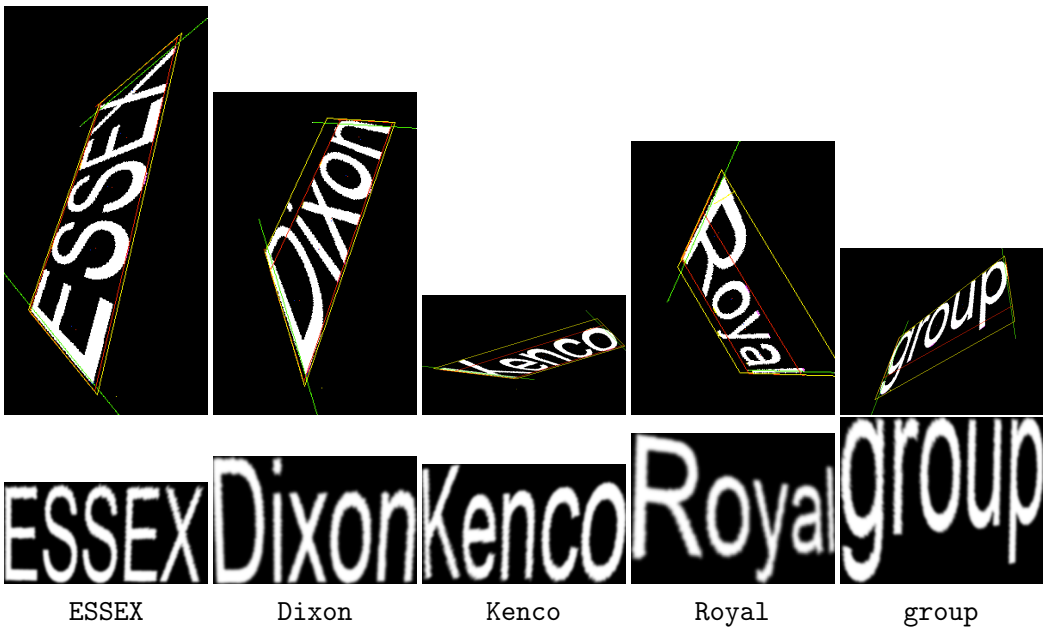


Fig. 6: Rectification results on text strings of five characters: original image (top), text rectification result (middle) and OCR transcription using Tesseract (bottom).

Fig. 7: Rectification results on text strings of six characters: original image (top), text rectification result (middle) and OCR transcription using Tesseract (bottom).

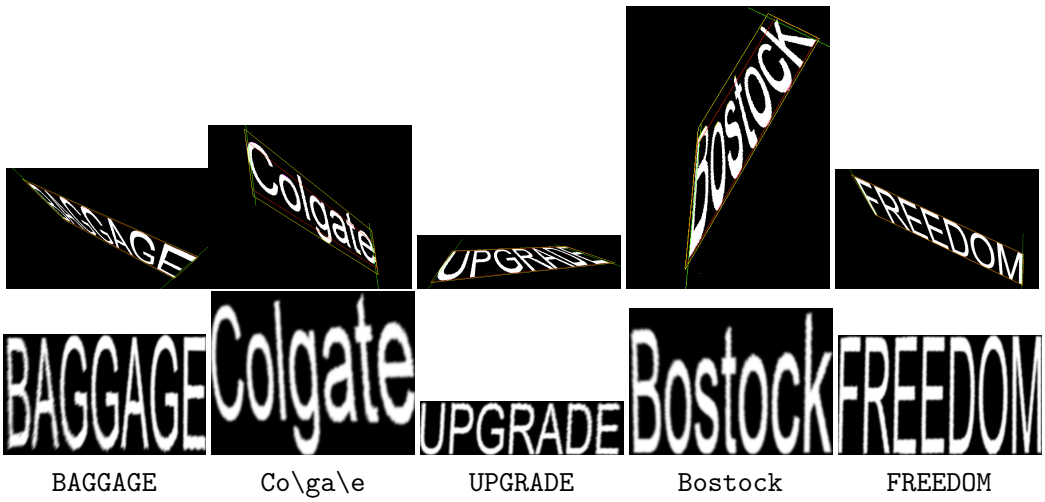


Fig. 8: Rectification results on text strings of seven characters: original image (top), text rectification result (middle) and OCR transcription using Tesseract (bottom).

Fig. 9: Rectification results on text strings of eight characters: original image (top), text rectification result (middle) and OCR transcription using Tesseract (bottom).

Fig. 10: Rectification results on text strings of nine characters: original image (top), text rectification result (middle) and OCR transcription using Tesseract (bottom).

Fig. 11: Rectification results on text strings of ten characters: original image (top), text rectification result (middle) and OCR transcription using Tesseract (bottom).

Fig. 12: Rectification results on text strings of more than ten characters: original image (top), text rectification result (middle) and OCR transcription using Tesseract (bottom).

Real-scene dataset

To show the performance of the rectification process on real-scene images, we have chosen a subset of nine examples from the RSTRC'15 dataset illustrated in Figure 13.

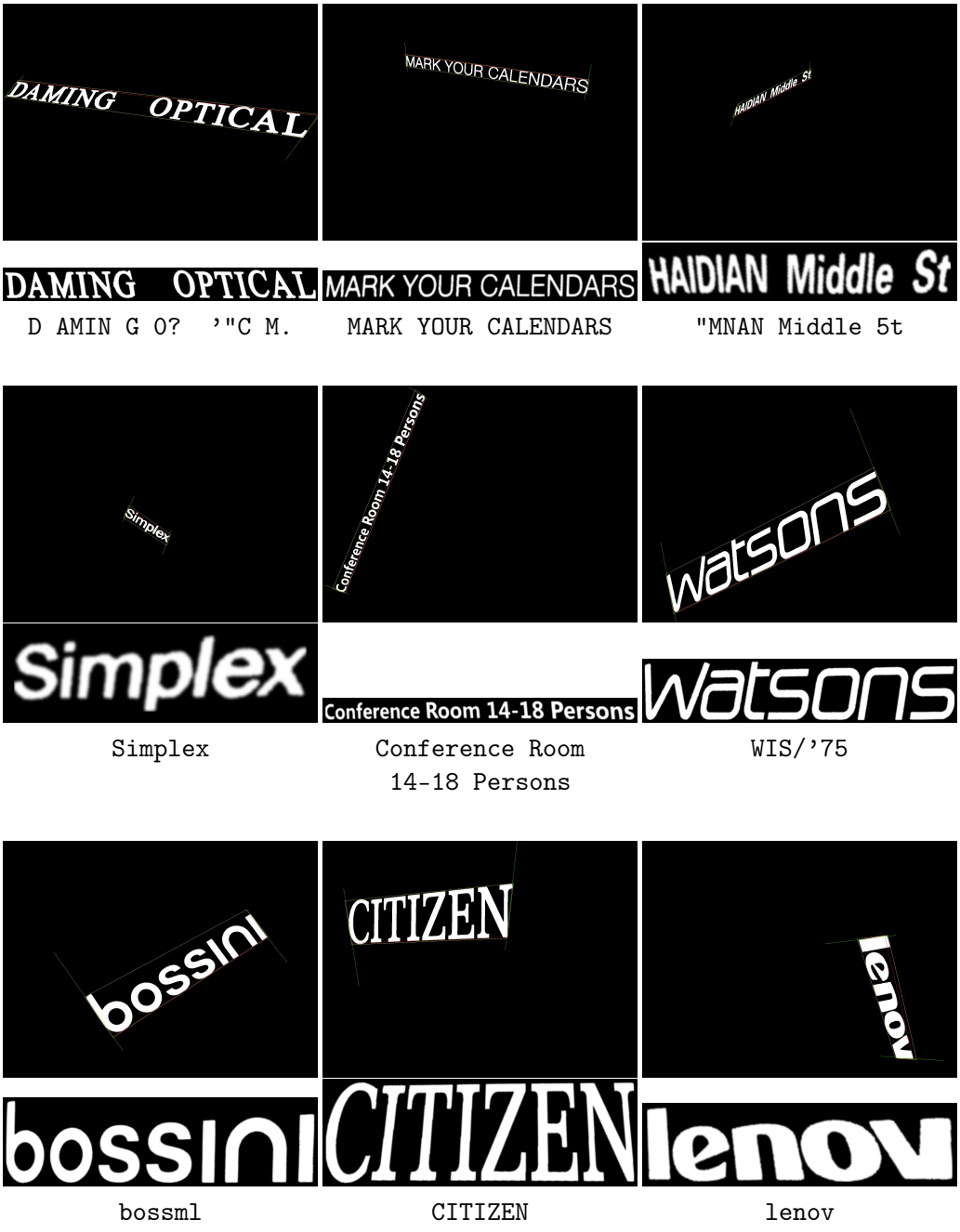


Fig. 13: Rectification example results on the real-scene dataset: original image (top), text rectification result (middle) and OCR transcription using Tesseract (bottom).

8.2.2 Performance results

Metrics. We evaluate the performance of the rectification method using the performance measurements proposed during the ICDAR 2015 *Competition on Scene Text Rectification* [Liu and Wang, 2015]. The two metrics used to compute the scores are based on the OCR results obtained from the original and rectified text images. The first metric is the OCR *accuracy* between the recognition result R of a rectified text string and its corresponding GT transcription G defined as:

$$accuracy(R,G) = \frac{1 - L(R,G)}{\max(|R|,|G|)}, \tag{8.1}$$

where $|R|$ and $|G|$ represent the length of the two strings. $L(R,G)$ is the Levenshtein distance between R and G, measuring the dissimilarity between these two text sequences.

The second metric is the *rectification performance* which considers the OCR results before and after the rectification and is defined as:

$$rectification\_performance(R,D,G) = accuracy(R,G) - accuracy(D,G), \tag{8.2}$$

where D is the recognition result of the distorted text before applying the rectification. The rectification performance reflects on one hand the impact of the rectification method on the final OCR result but also the difficulty of the text recognition process.

The performance results obtained using the two metrics rely, not only on the rectification efficiency, but also on the OCR performance. In our experiments we used the Tesseract OCR engine [Smith, 2007] to obtain the recognition results.

We now define A_b and A_a as the overall accuracy performance before and after the rectification over a dataset of N text strings, computed as:

$$A_b = \frac{\sum_i^N accuracy(D_i,G_i)}{N} \tag{8.3}$$

$$A_a = \frac{\sum_i^N accuracy(R_i,G_i)}{N} \tag{8.4}$$

Similarly, we define the overall rectification performance RP as:

$$RP = \frac{\sum_i^N rectification\_performance(R_i,D_i,G_i)}{N} \tag{8.5}$$

Tab. 8.1: Rectification evaluation results on the ICDAR 2015 *Competition on Scene Text Rectification* datasets.

Dataset	A_a	RP	A_b
SYNTHETIC	**0.721979**	**0.637037**	0.0849421
REAL-SCENE (EN)	**0.65149**	**0.187165**	0.464326

Discussion on the results obtained on the synthetic dataset. Table 8.1 contains the performance results obtained using our rectification method on the synthetic and real-scene datasets. The

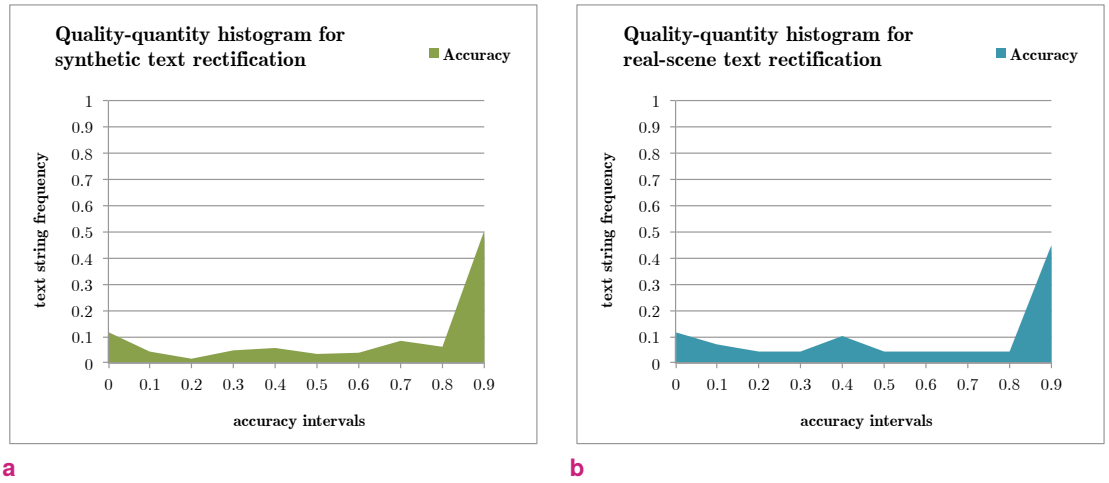


Fig. 14: Quality-quantity histograms for text rectification. Accuracy values for: (a) synthetic text; (b) real-scene text.

accuracy of the rectified method is evaluated to approximately 0.72. On the other hand, the accuracy before the rectification is very low, approximately 0.08, which indicates the difficulty to deal with text string deformations and also the efficiency of our method. Hence, the rectification performance RP is equal to 0.64. Figure 14a illustrates the quantity-quality 10-bins histograms containing the distributions of accuracy values. By looking at the frequency in the last bin of the histogram, one can notice that half of the rectified texts have obtained a nearly perfect recognition accuracy (*i.e.* accuracy values in the intervals $[0.8, 0.9[$ and $[0.9, 1]$). Approximately 10% of the texts got a low accuracy rate, belonging to interval $[0, 0.1[$: the rectification process has then probably failed.

While some problems come from the performance of the OCR, the proposed rectification procedure has also some weaknesses that directly affect the recognition accuracy. In some cases, the rectification fails if the filtering procedure does not correctly remove punctuation marks. However, most of the drawbacks mainly come from an incorrect approximation of the quadrangle that bounds the deformed text. The left and right bounding lines, introduced in Section 7.1.4, do not always find the best orientation of the extremity CC. The procedure to determine the left or right boundary lines does not well approximate the direction of characters such as "A" or "T". The procedure searches the line that maximizes the number of border points. In the case of capital letters "A" or "T" this line does not correspond to the direction of the character, as seen in Figure 15. Similarly, we can have an inaccurate quadrangle approximation when texts have as last extremity the capital letter "L" or small letter "r". Rectification examples and corresponding OCR transcriptions of texts containing these extremity characters are illustrated in Figure 16.

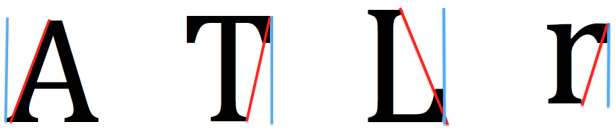


Fig. 15: Left and right bounding lines for capital letters "A", "T", "L" and small character "r": incorrect approximation (red) versus correct approximation (blue).

Fig. 16: Success and failures (in red) of OCR transcription of text strings containing extremity characters “A”, “T”, “L” and/or “r”: original image (top), text rectification result (middle) and OCR transcription using Tesseract (bottom).

Furthermore, the imprecise approximation of the upper and/or bottom bounding lines can also deteriorate the rectification process and possibly the text transcription. This usually happens in case of severe perspective deformations, when some small characters look larger than the capital letters as it can be seen in Figure 17. Despite the imperfect correction, the text is still successfully recognized by the Tesseract OCR. On the contrary, the disproportion of character sizes is even more visible for texts containing one capital and one small letter. Here, the upper and lower lines are not approximations, but the unique lines passing through the upper and bottom extremity points which produces wrong lines (*i.e.* not parallel to the real direction of the text). In Figure 18 we show three examples of texts of two characters, where one of the letters is a capital letter (“To”), a descender (“go”) and an ascender (“at”).

When the deformed text is upward, the rectification successfully transforms the text into an horizontal configuration but cannot correct the upside down orientation of the characters. Hence, the OCR produces erroneous text transcriptions. This is illustrated in Figure 19.

One of the advantages of the proposed rectification method is its ability to correct very challenging deformations, that make text strings unreadable. Although the rectification is not always very accurate,

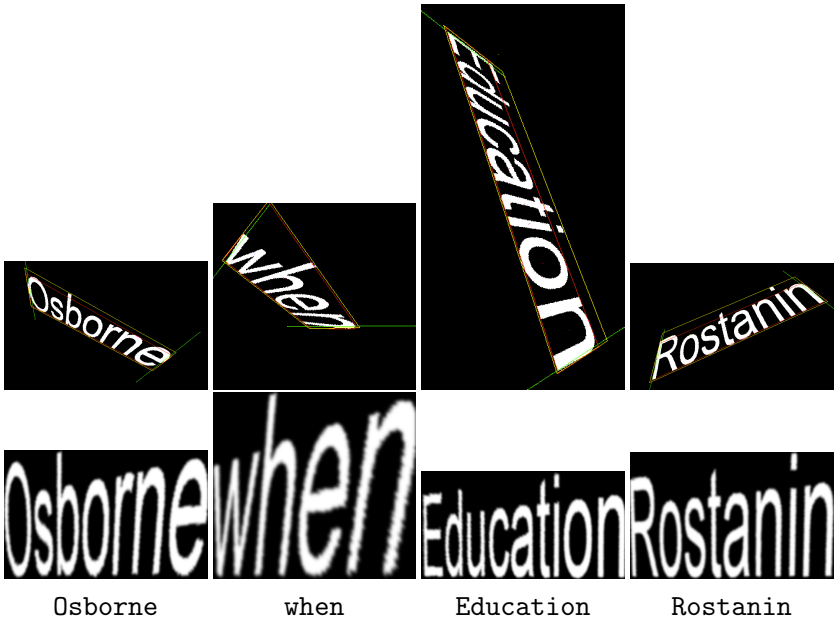


Fig. 17: Successful recognition of rectified text strings with disproportionate character sizes due to inaccurate upper and lower line approximations: original image (top), text rectification result (middle) and OCR transcription using Tesseract (bottom).

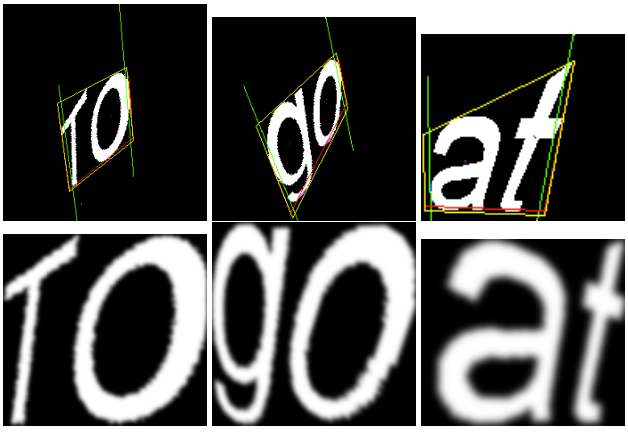


Fig. 18: Recognition failures (in red) due to inaccurate upper and lower line approximations for text strings of two characters (containing one ascender or one descender): original image (top), text rectification result (middle) and OCR transcription using Tesseract (bottom).

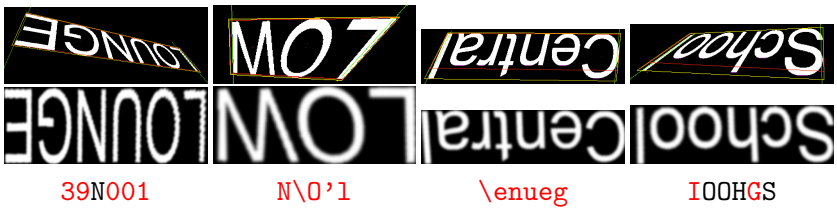


Fig. 19: Recognition failures (in red) due to upward rectified text strings: original image (top), text rectification result (middle) and OCR transcription using Tesseract (bottom).

which consequently leads to imprecise OCR transcriptions, the visual results remain notable. From a visual point of view, we succeed to transform illegible texts into readable ones. Such examples are

depicted in Figure 20. The recognition performance of Tesseract when dealing with inclined texts varies from case to case. The first part of the word “Warehouse” was missed, while its last six characters were correctly recognized. Similarly, the OCR performed better on the last part of the sequence “Gt. Yarmouth” (“mouth”) than on the apparently easier to read part “Gt. Ya”.

Fig. 20: Rectification result of challenging unreadable texts: original image (top), text rectification result (middle) and OCR transcription using Tesseract (bottom).

Discussion on the results obtained on the real-scene dataset. The accuracy score obtained on the real-scene datasets is slightly lower than the one obtained on the synthetic dataset (approximately 0.65). The rectification performance is very low, equal to 0.19. This is due to many reasons, listed below.

1. The fonts of natural scene texts are more challenging than the synthetic text ones (Times New Roman and Arial) and are then sometimes not correctly handled by the OCR (see examples in Figures 21a, 21b and 21c);
2. Natural scene texts can have complex designs in which characters are composed of multiple CCs, as in the example in Figure 21e. In such cases, the rectification method fails, as it can only handle characters represented by one CC.
3. The text distortion transformations are not as challenging as for the synthetic dataset, which explains the low value of *RP* in Table 8.1. The real-scene dataset contains mainly oriented texts and only few text strings in perspective. Such examples are provided in Figure 22.

4. Text in natural scene images can contain very small characters, which can affect the rectification process, as seen in Figure 21d.

5. The scores on the real-scene dataset are less representative because the dataset contains only 60 images, versus 2000 images in the synthetic dataset.

Fig. 21: OCR recognition failures due to (a-c) challenging fonts, (d) small text size and (e) complex design: original image (top), text rectification result (middle) and OCR transcription using Tesseract (bottom).

Fig. 22: Similarity of OCR recognition before and after the rectification process: original image (top), text rectification result (middle), OCR transcription before the rectification and OCR transcription after the rectification using Tesseract (bottom).

Moreover, by looking at the accuracy histogram in Figure 14b, we can observe that approximately the same number of deformed texts have been incorrectly rectified, as this was the case for the synthetic dataset. On the other hand, the distribution of accuracy values is compacted into the intervals

[0.3, 0.5[, [0.8, 0.9[and [0.9, 1], whereas the values computed on the synthetic dataset were more scattered. Nonetheless, both histograms in Figure 14 present a similar behavior which validates the fact that the proposed rectification method is independent of the text type, *i.e.* synthetic or natural.

8.2.3 Preliminary results on irregular text orientation correction

In Section 7.1.6, we have shown an extension of the rectification procedure that can correct multi-oriented texts. This work has not yet been validated on larger datasets designed specifically for irregular oriented texts. We show however some representative examples of different types of oriented text strings and their corresponding rectification results. Figure 23 illustrates that we can successfully rectify challenging curved (Figures[1] 23a and 23b) and arc-form (Figures 23c and 23d) text strings.

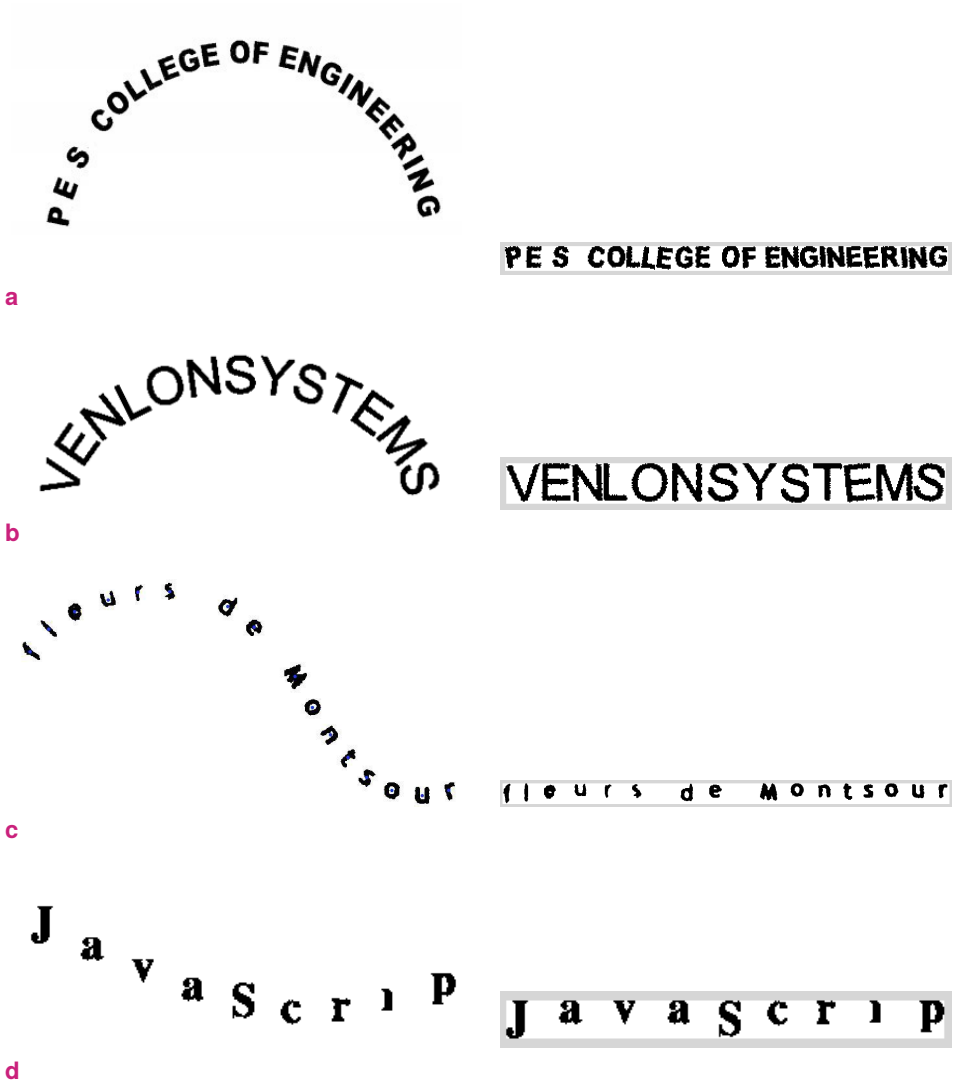


Fig. 23: Rectification results (right) of multi-oriented text string examples (left).

[1] Images taken from [Vasudev et al., 2007]

8.3 Conclusion

In this chapter we have presented the experimental results obtained using the text rectification method proposed in Chapter 7. Two series of experiments were conducted on two different datasets proposed during the ICDAR 2015 *Competition on Scene Text Rectification*. A first dataset consists of 2500 generated synthetic texts with different transformations, such as rotation, shearing, horizontal and vertical foreshortening. The second dataset is composed of 60 images containing natural scene text regions taken from the MSRA-TD500 dataset.

We have shown that the rectification procedure gives similar performances on both datasets. A slightly lower recognition accuracy was obtained on the real-scene dataset due to a number of reasons, such as the low performance of Tesseract OCR on texts with complex fonts or designs. On the other hand, many texts in this dataset present only rotation or slight perspective deformations, compared to the synthetic dataset, which contains more challenging texts that are subject to multiple transformations at the same time. The difficulty of the synthetic dataset is also proven by the high rectification performance score. We have demonstrated that the proposed rectification method can successfully correct oriented, sheared or perspective distorted texts. We have also shown that we could rectify unreadable texts and obtain satisfactory OCR accuracy scores.

The weaknesses of the proposed approach have also been identified, namely the situations where the rectification procedure fails to properly bring a text in a perfect front-parallel view. This is most of the time due to a wrong approximation of the lines bounding the distorted text. For example, the proposed quadrangle approximation approach fails when the text extremities are the capital letters "A", "L" or "T" or when dealing with two character strings containing one ascender or descender. Nevertheless, we have shown that, in many cases, although the rectification is not perfect, the OCR still provides a correct transcription of the corrected text.

The rectification procedure is evaluated based on the recognition accuracy performance. This is however influenced by the OCR. In our experiments, the Tesseract engine was used to produce the recognition results. Tesseract expects a very accurate text rectification and often fails when the characters are slightly inclined. For example, the letter "t" is often interpreted as "f", "l" as the symbol "\", "L" as "Z".

Finally, we have illustrated some preliminary rectification results on multi-oriented texts such as arc-form or curved ones, that proves that the approach can be adapted to any irregular oriented text. However, further experiments on larger datasets need to be conducted before making any further conclusions.

9 General discussion and future works

> *To reach a port we must set sail –*
> *Sail, not tie at anchor*
> *Sail, not drift.*
>
> **— Franklin D. Roosevelt**

In this chapter we draw our conclusions and present future perspectives. We review the main aspects and the contributions of this work.

In this thesis we have presented our contributions in the Document Image Analysis field. The presented work follows two directions. The first one, exposed in Part I, pointed out the improvement of the accuracy level of evaluation protocols designed for text detection tasks. This consist of proposing a complex evaluation method that handles the diversity of text detection algorithms. The second direction, presented in Part II, focuses on the improvement of the text recognition performance by proposing a rectification method capable of correcting highly distorted texts that cannot be handled by common OCRs.

Discussion on performance evaluation. We have analyzed and explained the numerous problems that text detectors are facing during the evaluation process (see Chapter 2). Unrepresentative scores due to matching failures between the detection results and the ground truth often under evaluate or over evaluate the performance of a detector. Most detection algorithms are complying to the rules imposed by different evaluation protocols and adapt their result outputs to not be penalized. Moreover, inconsistent comparisons between algorithms are very often performed as the obtained detection results are evaluated using different protocols. This is a crucial problem in the literature that has been frequently neglected. Accepting such inaccuracies can lead to a progress slowdown in the domain of Text Understanding Systems. Filtering the growing number of works proposed in this field is often difficult due to the lack of a reference evaluation protocol that could provide a reliable ranking between all these works.

Our contributions concern an alternative interpretation of how evaluation protocols should be designed. Hence, in this thesis we proposed a unified evaluation framework, EvaLTeX, that can analyze the performance of various text detectors regardless of the detection type. This was introduced in Chapter 3. Our objective was to provide a protocol with general rules that do not impose any output restrictions to text detectors but which takes into consideration different text granularities and representations. In this sense, we showed that EvaLTeX can be adapted to both well-defined and irregular text representations. Namely, it can evaluate text detectors that output detections represented by bounding boxes but also represented by irregular shapes, such as masks.

The matching protocol, which represents the core of an evaluation method, contrary to many protocols, was designed to handle all scenarios that can occur for matching the ground truth to a set of detections. An ideal matching happens when one detection is matched to a single ground truth object. However, due to the variability of texts present in natural scene and born-digital images, other scenarios, which imply more than just one ground truth object and one detection object are very frequent. There are three others matching cases. The *one-to-many* consists of matching one GT object to multiple detections, the *many-to-one* matches one detection to multiple GT objects and the *many-to-many* that matches many GT objects to many detections. The way these cases are evaluated are based on some hypothesis: a GT object should be detected only once, otherwise the detection should be penalized; a partial detection is still better than no detection which motivates our choice of a qualitative and non-binary local evaluation; detections covering multiple GT objects should not be penalized as long as they are not abusive (*i.e.* detecting a whole image).

Another novelty consists in the definition of a set of new rules and of the re-interpretation of standard metrics at object level, namely coverage and accuracy, for each of the scenarios discussed above. Hence, a *one-to-one* matching is evaluated qualitatively with respect to the true coverage area between the two objects. For a *one-to-many* case, we apply a fragmentation penalty. To robustly deal with *many-to-one* detections, we introduced a new GT granularity level, the region tag, that relaxes the precision penalization and allows a fair evaluation between text detectors having different output levels (*i.e.* word and line-level detections). *Many-to-many* scenarios are further categorized and their evaluation assumed a complex and particular adaptation of the local metrics. To provide a global evaluation we proposed to derive two quality and two quantity metrics from the well-known Recall and Precision measurements. Hence, we introduced a Recall and a Precision quality value that reflect the accuracy of the detections with regard to the GT. Moreover, we included quantitative Recall and Precision metrics to represent, respectively, the proportion of GT objects that have been correctly detected and the number of detections having a correspondence in the GT. We then showed that using these additional metrics we could obtain more information on the detection results and hence provide a detailed evaluation of a set of detections.

Based on a series of experiments, we successfully proved that the scores obtained with our evaluation protocol, not only are more representative, but also provide more realistic scores than commonly used protocols such as ICDAR'03, ICDAR'13 or DETEVAL. This statement is based on many comparisons between the scores obtained with EVALTEX and the protocols mentioned above on both single images and larger datasets. Moreover, the scores were computed for several text detectors, which allowed us to highlight the existing variations of rankings produced by the different protocols.

One of the main characteristics of this protocol is its ability to simultaneously deal with two granularity levels. In our experiments, this was illustrated on word and line level detections. However, the two-level annotation could be equally applied to character and word-line detections or to line and region-level outputs. An interesting perspective would be to add a third annotation level. Then we could handle equally, for example, word, line and paragraph level detections. This could represent a convenient choice for the evaluation of text detection in documents. Another perspective targets the region tagging, consisting of grouping GT objects, when using a mask annotation, which assumes that texts are represented with irregular shapes. Automatically generating regions for such objects is difficult because there are no simple rules as for the case of rectangular bounding boxes. Nonetheless, possible solutions exist. An idea would be the use of meta-balls to connect different masks together. Another idea, which necessitates

more time and a higher user interaction, would be to manually annotate all region configurations for a set of GT objects.

The EvaLTex protocol was designed for text detection tasks. However, its framework could also serve as a base for other tasks. If the mask annotation is applied to characters instead of words, we could then use EvaLTex to evaluate the text segmentation or binarization performance. In such a case, the GT objects are represented by characters and all matching rules and metrics would remain valid. The evaluation protocol could be used for text recognition purposes as well. The ability of the protocol of correctly identifying all matchings between the GT objects and a set of detections, could also be used to automatically match their OCR transcriptions. All these aspects represent perspectives that will be taken into consideration during our future works.

Performance scores are good indicators of a detector's performance. They can provide a ranking of different text detection algorithms but cannot justify it. This is why we enhanced our evaluation protocol by proposing a visual tool, based on quality histograms, capable of describing both the advantages and the drawbacks of a detector, features that could otherwise not be observed from the global scores. More intuitive than the ROC curves, the quality histograms provide at a glance the distribution of quality scores (for example coverage and accuracy) obtained on a whole dataset. This characterization is very powerful as it can also be used as a comparison tool between different detectors by showing in which cases one detector outperforms another one. Following this representation, we proposed an alternative set of scores obtained using the Earth Mover's Distance which can be easily applied to histograms and has the property of being a true metric. Each quality histogram was then compared to a GT histogram (called the *optimal* histogram) and their distance was used to compute two global scores (Recall and Precision). Our experiments showed that the scores obtained using the EMD are not only representative but also that they are similar to the scores obtained with EvaLTex which reinforced the fact that the proposed evaluation methods accurately describe the detection results.

In this work, the histogram representation was applied exclusively to illustrate text detection results. However, this approach could be equally extended to represent various results in the object detection domain, such as face or vehicle detection. Generally, all applications producing common detection scenarios with the ones in the text detection field could be evaluated using the histogram representation and its associated metrics. In the same way, other local measurements than the coverage or the accuracy, could be used to populate the quality histograms. For example, the fragmentation level (applied in this work to *one-to-many* matchings for the computation of the coverage) could be used to represent an independent measurement. We have shown in this manuscript that we could derive global scores by computing the EMD between a detection histogram and an optimal one. An interesting perspective would be to exploit the histogram distances to compute the difference between two detection sets and analyze if the obtained results provide any useful information for their comparison.

The implementation of the EvaLTex and histogram visualization tools was done in `C++`. In the near future, the two tools will be publicly available online such that they can be used by the community.

Discussion on text rectification. The second part of this thesis was focused on presenting the challenges of text variations in born-digital and natural scene images and their impact on the recognition stage. Since most of the OCRs expect as input horizontal texts taken from a parallel-frontal view in order to provide an accurate result, many texts, not conform to this configuration, are often bad transcript. We then have proposed a rectification process as an intermediate stage, able to enhance the OCR

performance. We have tested our method on a challenging dataset, recently proposed during the ICDAR 2015 *Competition on Scene Text Rectification*. The main characteristic of this dataset is the fact that it contains highly perspective distorted texts that has allowed a fair evaluation of our proposed approach.

The proposed rectification method mainly targets texts with perspective, orientation and shearing deformations. The process of the method relies on a homographic transformation that maps the camera coordinates onto a parallel-front plane. Since we do not know the camera parameters, the core of this rectification method relies entirely on the determination of the four coordinates that bound the deformed text. These coordinates were then used to compute a homography matrix that was later applied to each pixel of the text images to obtain its rectification. Finding the exact quadrangle formed by the four points is not an easy task. As described in Chapter 7, the quadrangle estimation procedure was divided into two stages. During the first step, the upper and lower bounding lines are approximated, following the generic Least Square Method approach. During the second stage, a series of strong hypothesis are used to rectify many challenging texts.

This work, being still recent, present some weaknesses exposed in Chapter 8, for which we here contour some future perspectives. One of the cases in which the rectification produces unsatisfactory results is the situation in which the extremity characters are capital letters "A" and "T". For these letters, the estimated border line rarely coincides with the direction line. To improve this, a deeper analysis on the shape of characters is needed, namely a study of their symmetry. Sometimes, this could be difficult, due to the severe deformation. Another possibility would be to make this analysis during an additional correction stage after the rectification.

Another future work consists in merging the orientation refinement stage and the perspective rectification process. Namely, if a text is only oriented we should apply the orientation correction procedure, if more deformations are involved, then the homographic transformation should be applied. This would assume a clear identification of the text's deformation. Once again, this is not an obvious task. While estimating a text's orientation is easy, determining its perspective or shearing transformation is more complicated, especially when a text is subject to more than one deformation. These remain open perspectives on which we will concentrate our work in the future.

Regardless of the limitations of the rectification process, we have still illustrated the fact that the used OCR does not always need a perfect correction in order to provide accurate transcriptions. However, the recognition accuracy evaluates two components: the rectification ability and the OCR recognition performance. Still, in our experiments, we have used the Tesseract OCR engine, which is known not to be the most reliable one, but has the advantage of being free. More performant OCRs, that could be tested on the rectified text images, are CuneiForm[1], ABBYY[2] or OmniPage[3].

[1] http://cognitiveforms.com/products_and_services/cuneiform
[2] http://www.abbyy.com/finereader/
[3] http://www.nuance.fr/for-individuals/by-product/omnipage/index.htm

Bibliography

[Anthimopoulos et al., 2010] Anthimopoulos, M., Gatos, B., and Pratikakis, I. (2010). A two-stage scheme for text detection in video images. *Image and Vision Computing*, 28(9):1413 – 1426.

[Bouman et al., 2011] Bouman, K., Abdollahian, G., Boutin, M., and Delp, E. (2011). A low complexity sign detection and text localization method for mobile applications. *Transactions on Multimedia*, 13(5):922–934.

[Bušta et al., 2015] Bušta, M., Drtina, T., Helekal, D., Neumann, L., and Matas, J. (2015). Efficient character skew rectification in scene text images. In *Proc. Asian Conference on Computer Vision*, volume 9009, pages 134–146.

[Calarasanu et al., 2015] Calarasanu, S., Fabrizio, J., and Dubuisson, S. (2015). Using histogram representation and earth mover's distance as an evaluation tool for text detection. In *Proc. International Conference on Document Analysis and Recognition*.

[Cambra and Murillo, 2011] Cambra, A. and Murillo, A. (2011). Towards robust and efficient text sign reading from a mobile phone. In *Proc. International Conference on Computer Vision*, pages 64–71.

[Chen et al., 2004a] Chen, D., Odobez, J.-M., and Thiran, J.-P. (2004a). A localization/verification scheme for finding text in images and video frames based on contrast independent features and machine learning methods. *Signal Processing: Image Communication*, 19(3):205 – 217.

[Chen et al., 2011] Chen, H., Tsai, S., Schroth, G., Chen, D., Grzeszczuk, R., and Girod, B. (2011). Robust text detection in natural images with edge-enhanced maximally stable extremal regions. In *Proc. International Conference on Image Processing*, pages 2609–2612.

[Chen et al., 2004b] Chen, X., Yang, J., Zhang, J., and Waibel, A. (2004b). Automatic detection and recognition of signs from natural scenes. *Transactions on Image Processing*, 13(1):87–99.

[Clark et al., 2001] Clark, P., Mirmehdi, D., and Doermann, D. (2001). Recognizing text in real scenes. *International Journal on Document Analysis and Recognition*, 4:243–257.

[Clavelli et al., 2010] Clavelli, A., Karatzas, D., and Llados, J. (2010). A framework for the assessment of text extraction algorithms on complex color images. In *Proc. Document Analysis Systems*, pages 19–26.

[David, 2011] David, M. W. (2011). Evaluation: From precision, recall and f-measure to roc, informedness, markedness & correlation. *Journal of Machine Learning Technologies*, 2:37–63.

[de Campos et al., 2009] de Campos, T. E., Babu, B. R., and Varma, M. (2009). Character recognition in natural images. In *Proc. International Conference on Computer Vision Theory and Applications*.

[Deng et al., 2014] Deng, H., Zhu, Q., Tao, J., and Feng, H. (2014). Rectification of license plate images based on hough transformation and projection. *TELKOMNIKA Indonesian Journal on Electrical Engineering*, 12(1):584–591.

[Du et al., 2012] Du, Y., Duan, G., and Ai, H. (2012). Context-based text detection in natural scenes. In *Proc. International Conference on Image Processing*, pages 1857–1860.

[Dubuisson, 2011] Dubuisson, S. (2011). Tree-structured image difference for fast histogram and distance between histograms computation. *Pattern Recognition Letters*, 32(3):411–422.

[Epshtein et al., 2010] Epshtein, B., Ofek, E., and Wexler, Y. (2010). Detecting text in natural scenes with stroke width transform. In *Proc. Computer Vision and Pattern Recognition*, pages 2963–2970.

[Everingham et al., 2015] Everingham, M., Eslami, S., Van Gool, L., Williams, C., Winn, J., and Zisserman, A. (2015). The pascal visual object classes challenge: A retrospective. *International Journal on Computer Vision*, 111(1):98–136.

[Fabrizio et al., 2013] Fabrizio, J., Marcotegui, B., and Cord, M. (2013). Text detection in street level image. *Pattern Analysis and Applications*, 16(4):519–533.

[Fan and Huang, 2005] Fan, K. C. and Huang, C. H. (2005). Italic detection and rectification*. *Journal of Information Science and Engineering*, 23:403–419.

[Ferreira et al., 2005] Ferreira, S., Garin, V., and Gosselini, B. (2005). A text detection technique applied in the framework of a mobile camera-based application. In *Proc. Camera-Based Document Analysis and Recognition*, pages 133–139.

[Fraz et al., 2015] Fraz, M., Sarfraz, M., and Edirisinghe, E. (2015). Exploiting colour information for better scene text detection and recognition. *International Journal on Document Analysis and Recognition*, 18(2):153–167.

[Galibert et al., 2014] Galibert, O., Kahn, J., and Oparin, I. (2014). The zonemap metric for page segmentation and area classification in scanned documents. In *Proc. International Conference on Image Processing*, pages 2594–2598.

[Gao et al., 2013] Gao, S., Wang, C., Xiao, B., Shi, C., Zhang, Y., Lv, Z., and Shi, Y. (2013). Adaptive scene text detection based on transferring adaboost. In *Proc. International Conference on Document Analysis and Recognition*, pages 388–392.

[González and Bergasa, 2013] González, A. and Bergasa, L. M. (2013). A text reading algorithm for natural images. *Image and Vision Computing*, 31(3):255 – 274.

[Hase et al., 2001] Hase, H., Yoneda, M., Shinokawa, T., and Suen, C. (2001). Alignment of free layout color texts for character recognition. In *Proc. International Conference on Document Analysis and Recognition*, pages 932–936.

[Hua et al., 2001] Hua, X.-S., Wenyin, L., and Zhang, H.-J. (2001). Automatic performance evaluation for video text detection. In *Proc. International Conference on Document Analysis and Recognition*, pages 545–550.

[Huang et al., 2013] Huang, W., Lin, Z., Yang, J., and Wang, J. (2013). Text localization in natural images using stroke feature transform and text covariance descriptors. In *Proc. International Conference on Computer Vision*, pages 1241–1248.

[Huang et al., 2014] Huang, W., Qiao, Y., and Tang, X. (2014). Robust scene text detection with convolution neural network induced mser trees. In *Proc. European Conference on Computer Vision*, pages 497–511.

[Huang and Ma, 2010] Huang, X. and Ma, H. (2010). Automatic detection and localization of natural scene text in video. In *Proc. International Conference on Pattern Recognition*, pages 3216–3219.

[ICDAR, 2013] ICDAR (2013). Robust reading competition results. http://dag.cvc.uab.es/icdar2013competition/.

[Jaccard, 1901] Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bulletin del la Société Vaudoise des Sciences Naturelles*, 37:547–579.

[Jaderberg et al., 2014a] Jaderberg, M., Simonyan, K., Vedaldi, A., and Zisserman, A. (2014a). Reading text in the wild with convolutional neural networks. *arXiv preprint arXiv:1412.1842*.

[Jaderberg et al., 2014b] Jaderberg, M., Simonyan, K., Vedaldi, A., and Zisserman, A. (2014b). Synthetic data and artificial neural networks for natural scene text recognition. *arXiv preprint arXiv:1406.2227*.

[Jaderberg et al., 2014c] Jaderberg, M., Vedaldi, A., and Zisserman, A. (2014c). Deep features for text spotting. In *Proc. European Conference on Computer Vision*, pages 512–528.

[Jain et al., 2014] Jain, A., Peng, X., Zhuang, X., Natarajan, P., and Cao, H. (2014). Text detection and recognition in natural scenes and consumer videos. In *Proc. International Conference on Acoustics, Speech and Signal Processing*, pages 1245–1249.

[Jameson and Abdullah, 2014] Jameson, J. and Abdullah, S. (2014). Extraction of arbitrary text in natural scene image based on stroke width transform. In *Proc. International Conference on Intelligent Systems Design and Applications*, pages 124–128.

[Kang et al., 2014] Kang, L., Li, Y., and Doermann, D. (2014). Orientation robust text line detection in natural images. In *Proc. Computer Vision and Pattern Recognition*, pages 4034–4041.

[Karatzas et al., 2015] Karatzas, D., Gomez-Bigorda, L., Nicolaou, A., Ghosh, S., Bagdanov, A., Iwamura, M., Matas, J., Neumann, L., Chandrasekhar, V. R., Lu, S., et al. (2015). Icdar 2015 competition on robust reading. In *Proc. International Conference on Document Analysis and Recognition*.

[Karatzas et al., 2013] Karatzas, D., Shafait, F., Uchida, S., Iwamura, M., Bigorda, L. G., Mestre, S. R., Mas, J., Mota, D. F., Almazan, J. A., and de las Heras, L. P. (2013). ICDAR 2013 robust reading competition. In *Proc. International Conference on Document Analysis and Recognition*, pages 1484–1493.

[Kasar and Ramakrishnan, 2013] Kasar, T. and Ramakrishnan, A. (2013). Alignment of curved text strings for enhanced ocr readability. *International Journal of Computer Vision & Signal Processing*, 3(1):1–9.

[Kasturi et al., 2009] Kasturi, R., Goldgof, D., Soundararajan, P., Manohar, V., Garofolo, J., Bowers, R., Boonstra, M., Korzhova, V., and Zhang, J. (2009). Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol. *Pattern Analysis and Machine Intelligence*, 31(2):319–336.

[Khare et al., 2015] Khare, V., Shivakumara, P., and Raveendran, P. (2015). A new histogram oriented moments descriptor for multi-oriented moving text detection in video. *Expert Systems with Applications*, 42(21):7627 – 7640.

[Kiran and Murali, 2013] Kiran, A. G. and Murali, S. (2013). Automatic rectification of perspective distortion from a single image using plane homography. *International Journal on Computational Sciences & Applications*, 3(5):47–58.

[Kumar et al., 2013] Kumar, D., Prasad, M. N. A., and Ramakrishnan, A. G. (2013). Multi-script robust reading competition in icdar 2013. In *Proc. International Workshop on Multilingual OCR*, pages 14:1–14:5.

[Lee et al., 2011] Lee, J.-J., Lee, P.-H., Lee, S.-W., Yuille, A., and Koch, C. (2011). Adaboost for text detection in natural scene. In *Proc. International Conference on Document Analysis and Recognition*, pages 429–434.

[Lee et al., 2010] Lee, S., Cho, M. S., Jung, K., and Kim, J. H. (2010). Scene text extraction with edge constraint and text collinearity. In *Proc. International Conference on Pattern Recognition*, pages 3983–3986.

[Lee and Kim, 2013] Lee, S. and Kim, J. H. (2013). Integrating multiple character proposals for robust scene text extraction. *Image and Vision Computing*, 31(11):823 – 840.

[Levillain et al., 2014] Levillain, R., Géraud, T., Najman, L., and Carlinet, E. (2014). Practical genericity: Writing image processing algorithms both reusable and efficient. In *Proc. Iberoamerican Congress on Pattern Recognition – Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, volume 8827, pages 70–79, Puerto Vallarta, Mexico.

[Li and Tan, 2008a] Li, L. and Tan, C. (2008a). Character recognition under severe perspective distortion. In *Proc. International Conference on Pattern Recognition*, pages 1–4.

[Li and Tan, 2008b] Li, L. and Tan, C. (2008b). Character recognition under severe perspective distortion. In *Proc. International Conference on Pattern Recognition*, pages 1–4.

[Li et al., 2013] Li, Y., Shen, C., Jia, W., and van den Hengel, A. (2013). Leveraging surrounding context for scene text detection. In *Proc. International Conference on Image Processing*, pages 2264–2268.

[Liang et al., 2008] Liang, J., DeMenthon, D., and Doermann, D. (2008). Geometric rectification of camera-captured document images. *Pattern Analysis and Machine Intelligence*, 30(4):591–605.

[Liang et al., 2001] Liang, J., Phillips, I. T., and Haralick, R. M. (2001). Performance evaluation of document structure extraction algorithms. *Computer Vision and Image Understanding*, 84(1):144 – 159.

[Ling and Okada, 2006a] Ling, H. and Okada, K. (2006a). Diffusion distance for histogram comparison. In *Proc. Computer Vision and Pattern Recognition*, volume 1, pages 246–253.

[Ling and Okada, 2006b] Ling, H. and Okada, K. (2006b). Emd-l1: An efficient and robust algorithm for comparing histogram-based descriptors. In *Proc. European Conference on Computer Vision*, pages 330–343.

[Liu and Wang, 2015] Liu, C. and Wang, B. (2015). Icdar2015 competition on scene text rectification. http://ocrserv.ee.tsinghua.edu.cn/icdar2015_str/.

[Liu et al., 2008] Liu, H., Wu, Q., Zha, H., and Liu, X. (2008). Skew detection for complex document images using robust borderlines in both text and non-text regions. *Pattern Recognition Letters*, 29(13):1893 – 1900.

[Liu et al., 2015] Liu, J., Su, H., Yi, Y., and Hu, W. (2015). Robust text detection via multi-degree of sharpening and blurring. *Signal Processing*, (0):–.

[Liu et al., 2014a] Liu, S., Zhou, Y., Zhang, Y., Wang, Y., and Lin, W. (2014a). Text detection in natural scene images with stroke width clustering and superpixel. In *Advances in Multimedia Information Processing – PCM*, volume 8879, pages 123–132.

[Liu et al., 2012] Liu, X., Lu, K., and Wang, W. (2012). Effectively localize text in natural scene images. In *Proc. International Conference on Pattern Recognition*, pages 1197–1200.

[Liu et al., 2014b] Liu, Y., Zhang, D., Zhang, Y., and Lin, S. (2014b). Real-time scene text detection based on stroke model. In *Proc. International Conference on Pattern Recognition*, pages 3116–3120.

[Lu et al., 2015] Lu, S., Chen, T., Tian, S., Lim, J.-H., and Tan, C.-L. (2015). Scene text extraction based on edges and support vector regression. *International Journal on Document Analysis and Recognition*, 18(2):125–135.

[Lu and Tan, 2006] Lu, S. and Tan, C. (2006). Camera text recognition based on perspective invariants. In *Proc. International Conference on Pattern Recognition*, volume 2, pages 1042–1045.

[Lucas, 2005] Lucas, S. (2005). ICDAR 2005 text locating competition results. In *Proc. International Conference on Document Analysis and Recognition*, pages 80–84.

[Lucas et al., 2003] Lucas, S. M., Panaretos, A., Sosa, L., Tang, A., Wong, S., and Young, R. (2003). ICDAR 2003 robust reading competitions. In *Proc. International Conference on Document Analysis and Recognition*, pages 682–687.

[Ma et al., 2007] Ma, Y., Wang, C., Xiao, B., and Dai, R. (2007). Usage-oriented performance evaluation for text localization algorithms. In *Proc. International Conference on Document Analysis and Recognition*, pages 1033–1037.

[Mao and Kanungo, 2002] Mao, S. and Kanungo, T. (2002). Software architecture of pset: A page segmentation evaluation toolkit. *International Journal on Document Analysis and Recognition*, 4(3):205–217.

[Mariano et al., 2002] Mariano, V., Min, J., Park, J.-H., Kasturi, R., Mihalcik, D., Li, H., Doermann, D., and Drayer, T. (2002). Performance evaluation of object detection algorithms. In *Proc. International Conference on Pattern Recognition*, pages 965–969.

[Matthews, 1975] Matthews, B. (1975). Comparison of the predicted and observed secondary structure of {T4} phage lysozyme. *Biochimica et Biophysica Acta - Protein Structure*, 405(2):442 – 451.

[Mekhalfi et al., 2015] Mekhalfi, M. L., Melgani, F., Bazi, Y., and Alajlan, N. (2015). Toward an assisted indoor scene perception for blind people with image multilabeling strategies. *Expert Systems with Applications*, 42(6):2907 – 2918.

[Meng and Song, 2012] Meng, Q. and Song, Y. (2012). Text detection in natural scenes with salient region. In *Proc. Document Analysis Systems*, pages 384–388.

[Meng et al., 2013] Meng, Q., Song, Y., Zhang, Y., and Liu, Y. (2013). Text detection in natural scene with edge analysis. In *Proc. International Conference on Image Processing*, pages 4151–4155.

[Merino-Gracia et al., 2011] Merino-Gracia, C., Lenc, K., and Mirmehdi, M. (2011). A head-mounted device for recognizing text in natural scenes. In *Proc. Camera-Based Document Analysis and Recognition*, pages 29–41.

[Merino-Gracia et al., 2013] Merino-Gracia, C., Mirmehdi, M., Sigut, J., and González-Mora, J. L. (2013). Fast perspective recovery of text in natural scenes. *Image and Vision Computing*, 31(10):714–724.

[Milyaev et al., 2015] Milyaev, S., Barinova, O., Novikova, T., Kohli, P., and Lempitsky, V. (2015). Fast and accurate scene text understanding with image binarization and off-the-shelf ocr. *International Journal on Document Analysis and Recognition*, 18(2):169–182.

[Mishra et al., 2013] Mishra, A., Alahari, K., and Jawahar, C. (2013). Image retrieval using textual cues. In *Proc. International Conference on Computer Vision*, pages 3040–3047.

[Mishra et al., 2012] Mishra, A., Alahari, K., and Jawahar, C. V. (2012). Scene text recognition using higher order language priors. In *Proc. British Machine Vision Conference*.

[Myers et al., 2005] Myers, G., Bolles, R., Luong, Q.-T., Herson, J., and Aradhye, H. (2005). Rectification and recognition of text in 3-d scenes. *International Journal on Document Analysis and Recognition*, 7(2-3):147–158.

[Nascimento and Marques, 2006] Nascimento, J. and Marques, J. (2006). Performance evaluation of object detection algorithms for video surveillance. *Transactions on Multimedia*, 8(4):761–774.

[Netzer et al., 2011] Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. (2011). Reading digits in natural images with unsupervised feature learning. In *Workshop on Deep Learning and Unsupervised Feature Learning - NIPS*.

[Neumann and Matas, 2012] Neumann, L. and Matas, J. (2012). Real-time scene text localization and recognition. In *Proc. Computer Vision and Pattern Recognition*, pages 3538–3545.

[Neumann and Matas, 2013] Neumann, L. and Matas, J. (2013). On combining multiple segmentations in scene text recognition. In *Proc. International Conference on Document Analysis and Recognition*, pages 523–527.

[Nguyen et al., 2014] Nguyen, P. X., Wang, K., and Belongie, S. (2014). Video text detection and recognition: Dataset and benchmark. In *Proc. Winter Conference on Applications of Computer Vision*, pages 776–783.

[O'Gorman, 1997] O'Gorman, L. (1997). *Document Image Analysis: An Executive Briefing*. IEEE Computer Society Press, Los Alamitos, CA, USA, 1st edition.

[Pan et al., 2008] Pan, Y.-F., Hou, X., and Liu, C.-L. (2008). A robust system to detect and localize texts in natural scene images. In *Proc. Document Analysis Systems*, pages 35–42.

[Pan et al., 2009] Pan, Y.-F., Hou, X., and Liu, C.-L. (2009). Text localization in natural scene images based on conditional random field. In *Proc. International Conference on Document Analysis and Recognition*, pages 6–10.

[Pan et al., 2011a] Pan, Y.-F., Hou, X., and Liu, C.-L. (2011a). A hybrid approach to detect and localize texts in natural scene images. *Transactions on Image Processing*, 20(3):800–813.

[Pan et al., 2011b] Pan, Y.-F., Zhu, Y., Sun, J., and Naoi, S. (2011b). Improving scene text detection by scale-adaptive segmentation and weighted crf verification. In *Proc. International Conference on Document Analysis and Recognition*, pages 759–763.

[Pavithra and Aradhya, 2014] Pavithra, M. and Aradhya, V. M. (2014). A comprehensive of transffrms, gabor filter and k-means clustering for text detection in images and video. *Applied Computing and Informatics*, pages –.

[Pele and Werman, 2009] Pele, O. and Werman, M. (2009). Fast and robust earth mover's distances. In *Proc. International Conference on Computer Vision*, pages 460–467.

[Pele and Werman, 2010] Pele, O. and Werman, M. (2010). The quadratic-chi histogram distance family. In *Proc. European Conference on Computer Vision*, volume 6312, pages 749–762.

[Peng et al., 2011] Peng, X., Cao, H., Prasad, R., and Natarajan, P. (2011). Text extraction from video using conditional random fields. In *Proc. International Conference on Document Analysis and Recognition*, pages 1029–1033.

[Phan et al., 2009] Phan, T. Q., Shivakumara, P., and Tan, C. (2009). A laplacian method for video text detection. In *Proc. International Conference on Document Analysis and Recognition*, pages 66–70.

[Phan et al., 2012] Phan, T. Q., Shivakumara, P., and Tan, C. L. (2012). Text detection in natural scenes using gradient vector flow-guided symmetry. In *Proc. International Conference on Pattern Recognition*, pages 3296–3299.

[Phan et al., 2013] Phan, T. Q., Shivakumara, P., Tian, S., and Tan, C. L. (2013). Recognizing text with perspective distortion in natural scenes. In *Proc. International Conference on Computer Vision*, pages 569–576.

[Pillai et al., 2013] Pillai, A., Balakrishnan, A., Simon, R., Johnson, R., and Padmagireesan, S. (2013). Detection and localization of texts from natural scene images using scale space and morphological operations. In *Proc. International Conference on Circuits, Power and Computing Technologies*, pages 880–885.

[Posner et al., 2010] Posner, I., Corke, P., and Newman, P. (2010). Using text-spotting to query the world. In *Proc. International Conference on Intelligent Robots and Systems*, pages 3181–3186.

[Prakash and Ravishankar, 2013] Prakash, S. and Ravishankar, M. (2013). Multi-oriented video text detection and extraction using dct feature extraction and projection based rotation calculation. In *Proc. International Conference on Advances in Computing, Communications and Informatics*, pages 714–718.

[Qu et al., 2013] Qu, Y., Liao, W., Lu, S., and Wu, S. (2013). Hierarchical text detection: From word level to character level. In *Proc. Advances in Multimedia Modeling*, volume 7733, pages 24–35.

[Rijsbergen, 1979] Rijsbergen, C. J. V. (1979). *Information Retrieval*. Butterworth-Heinemann, 2nd edition.

[Risnumawan et al., 2014] Risnumawan, A., Shivakumara, P., Chan, C. S., and Tan, C. L. (2014). A robust arbitrary text detection system for natural scene images. *Expert Systems with Applications*, 41(18):8027–8048.

[Rong et al., 2014] Rong, L., Suyu, W., and Shi, Z. (2014). A two level algorithm for text detection in natural scene images. In *Proc. Document Analysis Systems*, pages 329–333.

[Roy et al., 2008] Roy, P., Pal, U., Llados, J., and Kimura, F. (2008). Convex hull based approach for multi-oriented character recognition from graphical documents. In *Proc. International Conference on Pattern Recognition*, pages 1–4.

[Roy et al., 2015] Roy, S., Shivakumara, P., Roy, P. P., Pal, U., Tan, C. L., and Lu, T. (2015). Bayesian classifier for multi-oriented video text recognition system. *Expert Systems with Applications*, 42(13):5554 – 5566.

[Rubner et al., 2000] Rubner, Y., Tomasi, C., and Guibas, L. (2000). The earth mover's distance as a metric for image retrieval. *International Journal on Computer Vision*, 40(2):99–121.

[Shafait et al., 2008] Shafait, F., Keysers, D., and Breuel, T. (2008). Performance evaluation and benchmarking of six-page segmentation algorithms. *Pattern Analysis and Machine Intelligence*, 30(6):941–954.

[Shahab et al., 2011] Shahab, A., Shafait, F., and Dengel, A. (2011). Icdar 2011 robust reading competition challenge 2: Reading text in scene images. In *Proc. International Conference on Document Analysis and Recognition*, pages 1491–1496.

[Sharma et al., 2012] Sharma, N., Shivakumara, P., Pal, U., Blumenstein, M., and Tan, C. (2012). A new method for arbitrarily-oriented text detection in video. In *Proc. Document Analysis Systems*, pages 74–78.

[Shekar et al., 2014] Shekar, B., Smitha, M., and Shivakumara, P. (2014). Discrete wavelet transform and gradient difference based approach for text localization in videos. In *Proc. International Conference on Signal and Image Processing*, pages 280–284.

[Shi et al., 2014] Shi, C., Wang, C., Xiao, B., Gao, S., and Hu, J. (2014). End-to-end scene text recognition using tree-structured models. *Pattern Recognition*, 47(9):2853 – 2866.

[Shi et al., 2013] Shi, C., Wang, C., Xiao, B., Zhang, Y., and Gao, S. (2013). Scene text detection using graph model built upon maximally stable extremal regions. *Pattern Recognition Letters*, 34(2):107 – 116.

[Shivakumara et al., 2013] Shivakumara, P., Basavaraju, H., Guru, D., and Tan, C. (2013). Detection of curved text in video: Quad tree based method. In *Proc. International Conference on Document Analysis and Recognition*, pages 594–598.

[Shivakumara et al., 2008] Shivakumara, P., Huang, W., and Tan, C. (2008). An efficient edge based technique for text detection in video frames. In *Proc. Document Analysis Systems*, pages 307–314.

[Shivakumara et al., 2009a] Shivakumara, P., Phan, T. Q., and Tan, C. (2009a). A gradient difference based technique for video text detection. In *Proc. International Conference on Document Analysis and Recognition*, pages 156–160.

[Shivakumara et al., 2009b] Shivakumara, P., Phan, T. Q., and Tan, C. (2009b). A robust wavelet transform based technique for video text detection. In *Proc. International Conference on Document Analysis and Recognition*, pages 1285–1289.

[Shivakumara et al., 2011] Shivakumara, P., Phan, T. Q., and Tan, C. (2011). A laplacian approach to multi-oriented text detection in video. *Pattern Analysis and Machine Intelligence*, 33(2):412–419.

[Shivakumara et al., 2012] Shivakumara, P., Sreedhar, R., Phan, T. Q., Lu, S., and Tan, C. (2012). Multioriented video scene text detection through bayesian classification and boundary growing. *Transactions on Circuits and Systems for Video Technology*, 22(8):1227–1235.

[Smith, 2007] Smith, R. (2007). An overview of the tesseract ocr engine. In *Proc. International Conference on Document Analysis and Recognition*, pages 629–633.

[Stamatopoulos et al., 2011] Stamatopoulos, N., Gatos, B., Pratikakis, I., and Perantonis, S. (2011). Goal-oriented rectification of camera-based document images. *Transactions on Image Processing*, 20(4):910–920.

[Stehman, 1997] Stehman, S. V. (1997). Selecting and interpreting measures of thematic classification accuracy. *Remote Sensing of Environment*, 62(1):77 – 89.

[Sun et al., 2014] Sun, L., Huo, Q., Jia, W., and Chen, K. (2014). Robust text detection in natural scene images by generalized color-enhanced contrasting extremal region and neural networks. In *Proc. International Conference on Pattern Recognition*, pages 2715–2720.

[Sun et al., 2015] Sun, L., Huo, Q., Jia, W., and Chen, K. (2015). A robust approach for text detection from natural scene images. *Pattern Recognition*, 48(9):2906 – 2920.

[Swets, 1988] Swets, J. (1988). Measuring the accuracy of diagnostic systems. *Science*, 240(4857):1285–1293.

[Tomer and Goyal, 2013] Tomer, P. and Goyal, A. (2013). Ant clustering based text detection in natural scene images. In *Proc. International Conference on Computing, Communications and Networking Technologies*, pages 1–7.

[Vasudev et al., 2007] Vasudev, T., Hemanthkumar, G., and Nagabhushan, P. (2007). Transformation of arc-form-text to linear-form-text suitable for {OCR}. *Pattern Recognition Letters*, 28(16):2343 – 2351.

[Vondrick et al., 2013] Vondrick, C., Patterson, D., and Ramanan, D. (2013). Efficiently scaling up crowdsourced video annotation. *International Journal on Computer Vision*, 101(1):184–204.

[Wan, 2007] Wan, X. (2007). A novel document similarity measure based on earth mover's distance. *Information Sciences*, 177(18):3718 – 3730.

[Wang et al., 2011] Wang, K., Babenko, B., and Belongie, S. (2011). End-to-end scene text recognition. In *Proc. International Conference on Computer Vision*, pages 1457–1464.

[Wang and Belongie, 2010] Wang, K. and Belongie, S. (2010). Word spotting in the wild. In *Proc. European Conference on Computer Vision*, volume 6311, pages 591–604.

[Wang et al., 2014] Wang, L., Fan, W., He, Y., Sun, J., Katsuyama, Y., and Hotta, Y. (2014). Fast and accurate text detection in natural scene images with user-intention. In *Proc. International Conference on Pattern Recognition*, pages 2920–2925.

[Wang et al., 2013a] Wang, L., Katsuyama, Y., Fan, W., He, Y., Sun, J., and Hotta, Y. (2013a). Text detection in natural scene images with user-intention. In *Proc. International Conference on Image Processing*, pages 2256–2259.

[Wang et al., 2015a] Wang, R., Sang, N., and Gao, C. (2015a). Text detection approach based on confidence map and context information. *Neurocomputing*, 157:153 – 165.

[Wang et al., 2013b] Wang, X., Song, Y., and Zhang, Y. (2013b). Natural scene text detection with multi-channel connected component segmentation. In *Proc. International Conference on Document Analysis and Recognition*, pages 1375–1379.

[Wang et al., 2015b] Wang, X., Song, Y., Zhang, Y., and Xin, J. (2015b). Natural scene text detection with multi-layer segmentation and higher order conditional random field based analysis. *Pattern Recognition Letters*, 60–61(0):41 – 47.

[Weinman et al., 2009] Weinman, J., Learned-Miller, E., and Hanson, A. (2009). Scene text recognition using similarity and a lexicon with sparse belief propagation. *Pattern Analysis and Machine Intelligence*, 31(10):1733–1746.

[Wolf and Jolion, 2006] Wolf, C. and Jolion, J.-M. (2006). Object count/area graphs for the evaluation of object detection and segmentation algorithms. *International Journal on Document Analysis and Recognition*, 8(4):280–296.

[Wu et al., 2015] Wu, L., Shivakumara, P., Lu, T., and Tan, C. (2015). A new technique for multi-oriented scene text lines detection and tracking in video. *Transactions on Multimedia*, PP(99):1–1.

[Wu et al., 2014] Wu, L., Shivakumara, P., Lu, T., and Tan, C. L. (2014). Text detection using delaunay triangulation in video sequence. In *Proc. Document Analysis Systems*, pages 41–45.

[Yan et al., 2014] Yan, S.-Y., Xu, X.-X., and Liu, Q.-S. (2014). Robust text detection in natural scenes using text geometry and visual appearance. *International Journal of Automation and Computing*, 11(5):480–488.

[Yan et al., 2007] Yan, W., Wang, Q., Liu, Q., Lu, H., and Ma, S. (2007). Topology-Preserved Diffusion Distance for Histogram Comparison. *BMVC*, pages 1–10.

[Yang et al., 2014] Yang, H., Quehl, B., and Sack, H. (2014). A framework for improved video text detection and recognition. *Multimedia Tools and Applications*, 69(1):217–245.

[Yao, 2012] Yao, C. (2012). Detecting texts of arbitrary orientations in natural images. In *Proc. Computer Vision and Pattern Recognition*, pages 1083–1090.

[Yao et al., 2014] Yao, C., Bai, X., and Liu, W. (2014). A unified framework for multioriented text detection and recognition. *Transactions on Image Processing*, 23(11):4737–4749.

[Ye and Doermann, 2014] Ye, Q. and Doermann, D. (2014). Scene text detection via integrated discrimination of component appearance and consensus. In *Proc. Camera-Based Document Analysis and Recognition*, volume 8357, pages 47–59.

[Ye and Doermann, 2015] Ye, Q. and Doermann, D. (2015). Text detection and recognition in imagery: A survey. *Pattern Analysis and Machine Intelligence*, 37(7):1480–1500.

[Ye et al., 2007] Ye, Q., Jiao, J., Huang, J., and Yu, H. (2007). Text detection and restoration in natural scene images. *Journal of Visual Communication & Image Representation*, 18(6):504–513.

[Yi and Tian, 2011a] Yi, C. and Tian, Y. (2011a). Text detection in natural scene images by stroke gabor words. In *Proc. International Conference on Document Analysis and Recognition*, pages 177–181.

[Yi and Tian, 2011b] Yi, C. and Tian, Y. (2011b). Text string detection from natural scenes by structure-based partition and grouping. *Transactions on Image Processing*, 20(9):2594–2605.

[Yi and Tian, 2012] Yi, C. and Tian, Y. (2012). Assistive text reading from complex background for blind persons. In *Proc. Camera-Based Document Analysis and Recognition*, volume 7139, pages 15–28.

[Yi and Tian, 2013] Yi, C. and Tian, Y. (2013). Text extraction from scene images by character appearance and structure modeling. *Computer Vision and Image Understanding*, 117(2):182 – 194.

[Yin et al., 2014] Yin, X.-C., Yin, X., Huang, K., and Hao, H.-W. (2014). Robust text detection in natural scene images. *Pattern Analysis and Machine Intelligence*, 36(5):970–983.

[Yonemoto, 2014] Yonemoto, S. (2014). A method for text detection and rectification in real-world images. In *Proc. International Conference on Information Visualisation*, pages 374–377.

[Yuan et al., 2015] Yuan, J., Wei, B., Liu, Y., Zhang, Y., and Wang, L. (2015). A method for text line detection in natural images. *Multimedia Tools and Applications*, 74(3):859–884.

[Zagoris and Pratikakis, 2013] Zagoris, K. and Pratikakis, I. (2013). Text detection in natural images using bio-inspired models. In *Proc. International Conference on Document Analysis and Recognition*, pages 1370–1374.

[Zhang and Chong, 2013] Zhang, J. and Chong, Y. (2013). Text localization based on the discrete shearlet transform. In *Proc. International Conference on Software Engineering and Service Science*, pages 262–266.

[Zhang et al., 2008] Zhang, J., Goldgof, D., and Kasturi, R. (2008). A new edge-based text verification approach for video. In *Proc. International Conference on Pattern Recognition*, pages 1–4.

[Zhang and Kasturi, 2014] Zhang, J. and Kasturi, R. (2014). Sign detection based text localization in mobile device captured scene images. In *Proc. Camera-Based Document Analysis and Recognition*, volume 8357 of *Lecture Notes in Computer Science*, pages 71–82.

[Zhang et al., 2004] Zhang, L., Lu, Y., and Tan, C. (2004). Italic font recognition using stroke pattern analysis on wavelet decomposed word images. In *Proc. International Conference on Pattern Recognition*, volume 4, pages 835–838.

[Zhang et al., 2013] Zhang, X., Lin, Z., Sun, F., and Ma, Y. (2013). Rectification of optical characters as transform invariant low-rank textures. In *Proc. International Conference on Document Analysis and Recognition*, pages 393–397.

[Zhang et al., 2015] Zhang, Y., Lai, J., and Yuen, P. C. (2015). Text string detection for loosely constructed characters with arbitrary orientations. *Neurocomputing*, pages –.

[Zhao et al., 2010] Zhao, M., Li, S., and Kwok, J. (2010). Text detection in images using sparse representation with discriminative dictionaries. *Image and Vision Computing*, 28(12):1590 – 1599.

[Zhao et al., 2015] Zhao, Z., Fang, C., Lin, Z., and Wu, Y. (2015). A robust hybrid method for text detection in natural scenes by learning-based partial differential equations. *Neurocomputing*, 168:23–34.

[Zhou et al., 2009] Zhou, P., Li, L., and Tan, C. (2009). Character recognition under severe perspective distortion. In *Proc. International Conference on Document Analysis and Recognition*, pages 676–680.

[Zhou et al., 2015] Zhou, X., Zhou, S., Yao, C., Cao, Z., and Yin, Q. (2015). ICDAR 2015 text reading in the wild competition. *CoRR*, abs/1506.03184.

[Zhu et al., 2015] Zhu, A., Wang, G., and Dong, Y. (2015). Detecting natural scenes text via auto image partition, two-stage grouping and two-layer classification. *Pattern Recognition Letters*, (0):–.

List of Figures

List of Tables

Colophon

This thesis was typeset with LaTeX 2_ε. It uses the *Clean Thesis* style developed by Ricardo Langner. The design of the *Clean Thesis* style is inspired by user guide documents from Apple Inc.

Download the *Clean Thesis* style at http://cleanthesis.der-ric.de/.