Object count/Area Graphs for the Evaluation of Object Detection and Segmentation Algorithms

Christian Wolf, Jean-Michel Jolion

Object count/Area Graphs for the Evaluation of Object Detection and Segmentation Algorithms [1]

Christian Wolf Jean-Michel Jolion

Technical Report LIRIS

September 28th 2005

LIRIS - INSA de Lyon
Bât. Jules Verne
20, Avenue Albert Einstein
69621 Villeurbanne cedex, France
wolf@rfv.insa-lyon.fr
jean-michel.jolion@liris.cnrs.fr

Abstract

Evaluation of object detection algorithms is a non-trivial task: a detection result is usually evaluated by comparing the bounding box of the detected object with the bounding box of the ground truth object. The commonly used precision and recall measures are computed from the overlap area of these two rectangles. However, these measures have several drawbacks: they don't give intuitive information about the proportion of the correctly detected objects and the number of false alarms, and they cannot be accumulated across multiple images without creating ambiguity in their interpretation. Furthermore, quantitative and qualitative evaluation is often mixed resulting in ambiguous measures.

In this paper we propose a new approach which tackles these problems. The performance of a detection algorithm is illustrated intuitively by performance graphs which present object level precision and recall depending on constraints on detection quality. In order to compare different detection algorithms, a representative single performance value is computed from the graphs. The influence of the test database on the detection performance is illustrated by performance/generality graphs. The evaluation method can be applied to different types of object detection algorithms. It has been tested on different text detection algorithms, among which are the participants of the ICDAR 2003 text detection competition.

Keywords

Evaluation, object detection, text detection

[1] The work presented in this article has been conceived in the framework of two industrial contracts with France Télécom in the framework of the projects ECAV I and ECAV II with respective numbers 001B575 and 0011BA66.

1 Introduction

In the past, computer vision (CV) as a research domain has frequently been criticized for a lack of experimental culture [10] [17] [8] [4], which has been explained by the young age of the discipline. However, experimental evaluation of the theoretical advances is indispensable in all scientific work. We are currently trying very hard to establish a real experimental culture, and the need of strict experimental procedures in applying and evaluating algorithms is widely recognized [17] [16].

An important obstacle is the lack of common test databases and ground truth, which makes the comparison of different algorithms difficult. In some areas common test databases did emerge, as for instance the Brodatz test database for texture analysis, the NIST database for character recognition etc. However, the tuning of image processing algorithms to a small set of test databases is not undisputed. As Bowyer *et al.* put it [4], "*the world is rich enough to provide infinitely interesting imagery*".

For this reason, and because of their success in other disciplines, scientific competitions made their appearance during the last years. We may cite for example the TREC Video Track[2], a competition in the field of content based video indexing organized by NIST and held annually. The goal of the conference series is to encourage research in information retrieval from large amounts of text and video sequences by providing a large test collection, uniform scoring procedures, and a forum for organizations interested in comparing their results. The test collections are changed each year in order to avoid specialization to a single test database.

In the field of document image analysis, the ICDAR page segmentation competitions [3], the ICDAR text detection competitions [13] and the GREC competition for line and arc detection [21] should be mentioned (see section 3).

The introduction of the evaluation problem coincides largely with the emergence of the field of visual information retrieval. As a consequence, the first techniques have been naturally inspired by tools from this domain, as for instance precision/recall graphs which are frequently used in information retrieval. However, visual information has its own specificities, which need to be taken into account. This is the goal of this work.

In this paper we concentrate on the evaluation process, more specifically on the design of evaluation measures. Evaluation is a process which is often neglected by scientists, who spend most of their valuable time conceiving theories and designing solutions. However, in computer vision, a successful evaluation algorithm is rarely simple to design. Often it is necessary to conceive non-trivial algorithms in order to ensure an evaluation satisfying scientific requirements:

- A simple and intuitive interpretation of the obtained measures.
- An objective comparison between the different algorithms to evaluate.
- A good correspondence between the obtained measures and the objective performance of the algorithm to evaluate, taking into account its goal.

The latter point is particularly important. Aloimonos and Rosenfeld emphasize the purpose in CV [1]: "*If we consider biological organisms that possess vision, we find that the visual system tends to be well matched to the environment of the organism and to the*

[2]http://www-nlpir.nist.gov/projects/trecvid

tasks that the organism performs. The paradigm purposive vision *suggests that purpose should be a guiding principle in our study of vision*". If we design CV systems according to a specific purpose, then it should be natural that we evaluate their performance according to this same purpose. This is the objective of "goal oriented evaluation".

A particular problem in computer vision, which has already given birth to a multitude of solutions is the problem of detecting objects in images. In this document, we introduce a new performance measure designed for the evaluation of object detection algorithms. In this context, by detection we also mean localization, thus tackling a two-part problem. We keep the general evaluation framework independent of the object type, defining an object as a visual entity with a spatial reality, and illustrate the concepts with experiments and examples from the field of text detection.

In the context of document image analysis, a similar problem is the one of document page segmentation. As in object detection, or more specifically in text detection, lists of rectangles need to be compared in order to evaluate these algorithms. However, although the two evaluation problems may be similar from a theoretical viewpoint, practically we need to emphasize some differences between page segmentation and text/object detection:

- The density of relevant information ("generality", see section 5) is higher for page segmentation problems. In text detection, on the other hand, text areas are not so much "classified" as "detected", i.e. that there can be and will be large areas which do not contain relevant material. This difference results in different evaluation techniques, which differ for instance in the way how the algorithms treats detection quality and detection quantity.

- In the page segmentation context, regions are possibly non-rectangular. The proposed evaluation algorithm, based on a rectangle representation of object reasons, is not applicable in this case.

The second point restricts the proposed evaluation systems to objects which are well represented by rectangles, which is the case for text, faces, people, generally speaking, compound objects. We therefore focus on these kind of problems, which are mostly encountered when evaluating systems working on natural scenes and video, but also systems which extract text from complex journals.

However, this is not the case for some other problems encountered in document image analysis, notably curves as lines and arcs. These objects may overlap, therefore a single rectangle may contain several objects. While the general philosophy of the proposed system is applicable, i.e. the separation of detection quality and quantity and its representation as graphs, the object matching part itself is restricted to rectangle based representations.

The main contribution of this paper concerns the following issues:

- The separation of detection quality and detection quantity. New performance graphs allow us to easily perceive the detection quantity ("how many objects have been detected?" and "how many false alarms have been detected?") as well as detection quality ("how accurate is the detection of the objects?").

- The influence of the data base is evaluated, i.e. the relationship between the performance of the detection algorithms and the structure of the image test database

is put forward. This makes it easier to grasp the advantage an object detection algorithm might have when it is tested on an image collection which a larger percentage of relevant information.

- The derivation of a single performance value which does not depend on quality related thresholds. Although this performance value, by definition, does not allow us to fully comprehend the behavior of a detection algorithm, it makes it easier to create a ranking of the algorithms to evaluate.

The reminder of this document is organized as follows:

Section 2 gives an introduction to the problem and presents different evaluation modes on a hierarchy of different levels, which is formed by the different possible result representations.

Section 3 presents a survey on the previous work on the evaluation of object detection algorithms.

Section 4 introduces new performance graphs for an easy and intuitive interpretation of the detection performance as well as a new performance measure.

Section 5 demonstrates the dependence of evaluation algorithms on the structure of the test database and introduces a new evaluation graph which illustrates this dependence.

Section 6 applies the evaluation measure to two different text detection algorithms and illustrates its intuitive usage.

Finally, section 7 concludes.

2 Evaluation levels

Traditionally, object detection algorithms are evaluated using techniques developed for information retrieval systems. More specifically, the measures of precision and recall are widely used, since they intuitively convey the quality of the results:

$$\begin{aligned} \mathrm{R}_{IR} &= \frac{\text{N.o. correctly retrieved items}}{\text{N.o. relevant items in the database}} \\ \mathrm{P}_{IR} &= \frac{\text{N.o. correctly retrieved items}}{\text{Total n.o. retrieved items}} \end{aligned} \tag{1}$$

In order to have a single performance value for the ranking of methods, the two measures are often linearly combined. The harmonic mean of precision and recall has been introduced by the information retrieval community [19]. Its advantage is that the minimum of the two performance values is emphasized:

$$\mathrm{Perf}_{IR} = 2\,\frac{\mathrm{P}_{IR}\ \cdot\ \mathrm{R}_{IR}}{\mathrm{P}_{IR}\ +\ \mathrm{R}_{IR}} \tag{2}$$

For the object detection problem, the measures of recall and precision are not directly applicable, since the decision whether an object has been detected or not is not a binary one. Object detection algorithms may be evaluated at different levels w.r.t. the representation of the detection results, corresponding to different phases of the detection algorithms (see figure 1). The evaluation measures of the different levels differ in their relevance to the goal of the application and in their coverage, *i.e.* in the detection phases which are evaluated by the measure:

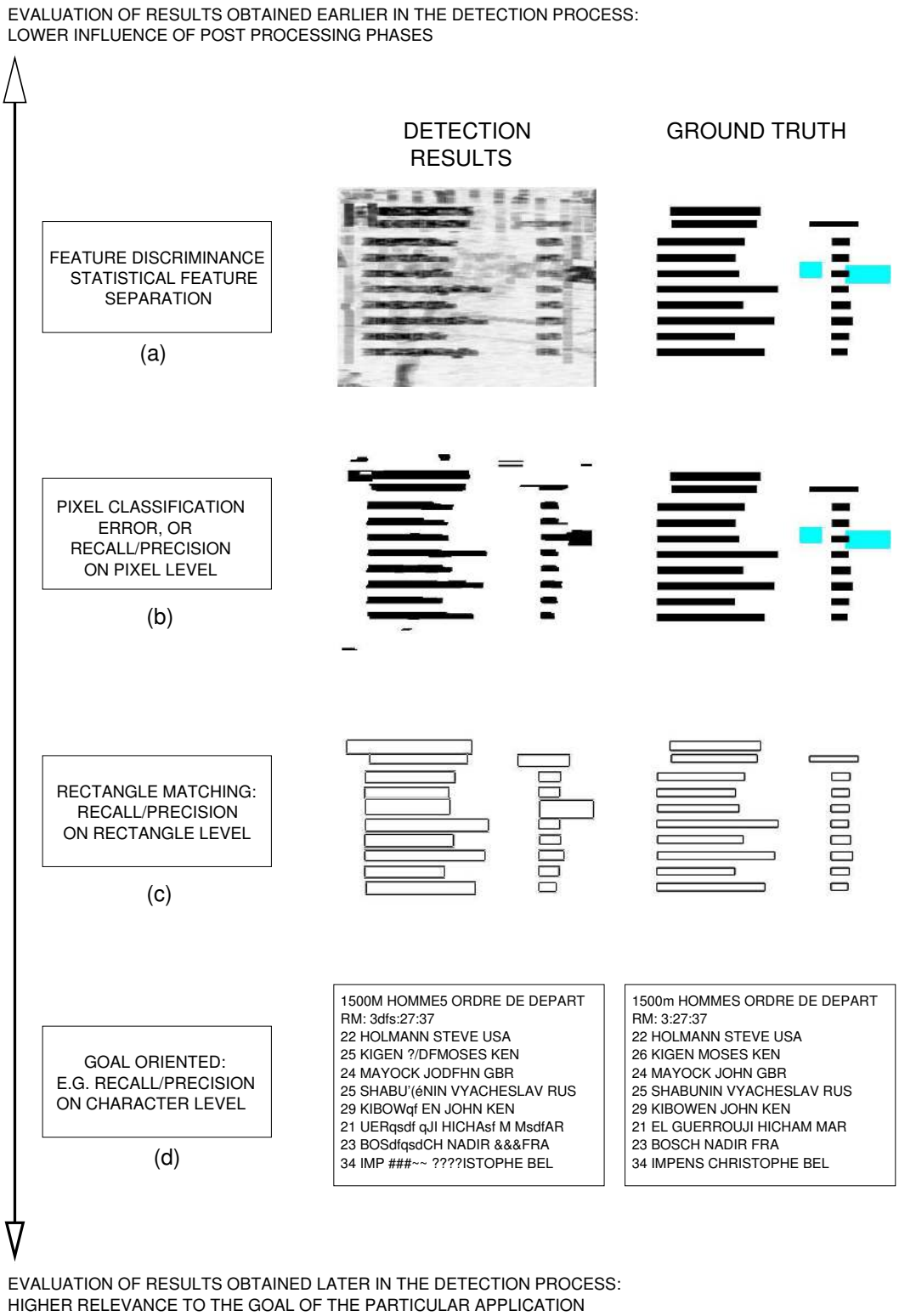


Figure 1: The different levels of evaluation for the example of text detection: (a) evaluation of the non-thresholded filter results (b) evaluation of the pixel classification results (c) evaluation on object level (d) goal oriented evaluation (depends on the application).

Feature discriminance at pixel level At this level, the quality of the chosen features is evaluated without taking into account the classification decision taken in a later phase. Therefore, the result evaluated for each pixel p is not a binary decision but a feature vector $\boldsymbol{x}_p$. Splitting the pixels into two populations, where the first population consists of the pixels labeled as "object" according to the ground truth, and the second population consists of the "non-object" pixels, the goal of the evaluation measure at this level is to assess whether the features are well separated between the two populations.

Assuming Gaussian distributions in both cases, an example of such a statistical separation measure is the Bhattacharyya distance [6]:

$$\begin{aligned} B &= \frac{1}{8}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^T \left(\frac{\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2}{2} \right)^{-1} (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) \\ &+ \frac{1}{2} \ln \frac{\left| \frac{\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2}{2} \right|}{\sqrt{|\boldsymbol{\Sigma}_1|}\sqrt{|\boldsymbol{\Sigma}_2|}} \end{aligned} \tag{3}$$

where the $\boldsymbol{\mu}_j, j = 1..2$ are the mean vectors of the two distributions and the $\boldsymbol{\Sigma}_j$ are the covariance matrices.

Note, that the Bhattacharyya distance measures the separability of the features under the assumption of a linear decision function. If a non-linear decision function is used, e.g. by employing a MLP or kernel based classifier, then other distance measures are necessary. However, this is beyond the scope of this article.

Classification at pixel level Once the classification decision for each pixel is available, *i.e.* we know for each pixel whether it belongs to the object or not, the measures of recall and precision may be applied on pixel level:

$$\begin{aligned} \mathrm{R}_{PX} &= \frac{\text{N.o. correctly detected object pixels}}{\text{N.o. object pixels}} \\ \mathrm{P}_{PX} &= \frac{\text{N.o. correctly detected object pixels}}{\text{Total n.o. pixels classified as "object"}} \end{aligned} \tag{4}$$

Alternatively, the classification error might be used for evaluation.

We note, that if the performance is evaluated at pixel level, then the ground truth must be very precise in order to get robust measures. This is rarely the case as ground truth is mainly obtained through interaction between the images and a human observer, which can easily detect an object but can rarely locate it with 1-pixel precision.

Detection at rectangle level From the end user's point of view, a more natural way is to ask the question whether an object has been detected correctly or not. On this level we still ignore domain specific knowledge from processing steps following the detection step, but we nevertheless evaluate the detection on a per object/rectangle basis. This assumes objects of compact shape, for which the rectangle approach makes sense. This is not appropriate for textures, or objects like snow, falling

water, shadows, but does make sense for objects like humans, faces, text, tools etc. The reminder of this document deals with this evaluation level.

Goal oriented evaluation In many applications, object detection is performed for a specific reason which is beyond the pure localization of the object. For instance, face detection might be a preliminary step for face recognition, text detection might be a preliminary step for text recognition, etc.

In this case, in order to take into account the specific goal, the evaluation algorithm should resort to the results of the application specific processing. In the context of text detection, a goal oriented evaluation scheme for a system which exploits the text content (as opposed to its position) should penalize lost text characters as well as additional characters which are not present in the ground truth. Possibilities are recall and precision on character level, or the string edit distance [20].

In the case of text detection for indexing video broadcasts, one might consider evaluation on an even higher level by weighting words according to their usefulness for the indexing process [11].

The evaluation level to choose depends on the application and the purpose of the evaluation. The pixel based evaluation measures are easy to calculate and easy to interpret. However, they lack relevance to the goal of the process and are not very accurate. Very often they are used to guide the choice of features used for detection, since they are not influenced by later steps of the detection algorithm.

The goal directed approaches are natural methods to employ for the final evaluation of the algorithm's performance. They directly measure the success which can be expected by the algorithm. However, very often the localization of the object is the final goal of the application. For instance, in the case of face detection or text detection, recognition of the object might be impossible because of low data quality. In image and video indexing applications, the presence of a face or of text is valuable information which can be exploited. In this context, goal directed evaluation is equivalent with evaluation on rectangle level (figure 1c).

Evaluation levels (a), (b) and (d) are easy to calculate and easy to interpret, since they treat "items" which are directly comparable (pixels and characters, respectively). On the other hand, rectangle based evaluation (level (c)) is a non-trivial task: as the detection result is rarely exactly equivalent to the object as specified in the ground truth, we cannot easily say whether an object has been correctly detected or not. In the reminder of this work, we concentrate on the problem of evaluation on rectangle level.

3 Previous work

The goal of a rectangle based object detection evaluation scheme is to take a list G of ground truth object rectangles $G_i, i = 1..|G|$ and a list D of detected object rectangles $D_j, j = 1..|D|$ and to measure the quality of the match between the two lists. The quality measure should penalize information loss, which occurs if objects or parts of objects have not been detected, and it should penalize information clutter, *i.e.* false alarms or detections which are larger than necessary[3].

[3]We should emphasize, that a comparison of the rectangles representing objects is not the same as comparing the objects themselves, since the rectangle based algorithm assumes that the object is

Most algorithms are based on an extension of the recall and precision measures which are calculated on the area of two rectangles G_i and D_i and on the area of the overlapping region:

$$\begin{aligned} \mathrm{R}_{AR}(G_i, D_i) &= \frac{Area(G_i \cap D_i)}{Area(G_i)} \\ \mathrm{P}_{AR}(G_i, D_i) &= \frac{Area(G_i \cap D_i)}{Area(D_i)} \end{aligned} \tag{5}$$

Recall illustrates the proportion of the ground truth rectangle which has been correctly detected, and precision decreases if the amount of additional incorrectly detected area increases. In the reminder of this work, we call these measures "area recall" and "area precision", respectively.

Whereas calculating these figures for a single pair of result and ground truth rectangles is straightforward, the extension to the realistic case of two lists of rectangles is not as easy. The existing evaluation methods differ in the way they treat the correspondence problem between the two rectangle lists, *i.e.* whether they consider single matches only or multiple matches, and in the way they combine the figures in order to generate a single measure for multiple rectangles and multiple images.

Doermann *et al.* present a configurable ground-truthing and evaluation system with a graphical java interface [5] for video segmentation. Their system also takes into account temporal matching of objects in videos and provides different temporal matching levels. However, the spatial matching algorithms supported by the tool are rather simple.

In [15], Mariano et al. propose a set of evaluation measures, among which are the area measures on rectangle bases given in equation (5) as well as measures on pixel level. Several extensions to multiple rectangles are suggested: summing up thresholded values of these measures, which introduces a dependence on a threshold, and directly calculating the measures on sets of rectangles by combining the rectangles to larger surfaces, which gives rise to ambiguity problems (see section 4).

Antonacopoulos et al. propose an algorithm capable of comparing lists of rectangles [2] in the context of document page segmentation. Each ground truth rectangle or polygon is extended up to the borders of the surrounding rectangles or the page border and checks whether segmented rectangles fall into these "maximized ground truth polygons". "partial misses", "misses" and "merges" are considered. However, this approach may pose problems in the case of text/object detection, where there are not always surrounding text/object rectangles. Furthermore, the evaluation algorithm focuses on reporting the accuracy the detection/classification of each rectangle, the authors do not provide performance measures for a whole document.

A simple evaluation scheme has been used to evaluate the systems participating at the text locating competition in the framework of the 7^{th} International Conference on Document Analysis and Recognition (ICDAR) 2003 [13]. Each rectangle in one list is matched with the best match in the opposing list:

identical to its bounding rectangle. In reality, a missed part of G_i may not contain object pixels, or a part of a false alarm in D_i may not contain detected pixels.

$$\begin{aligned} \mathrm{R}_{ICD}(G,D) &= \frac{\sum_{i=1}^{|G|} BestMatch_G(G_i)}{|G|} \\ \mathrm{P}_{ICD}(G,D) &= \frac{\sum_{j=1}^{|D|} BestMatch_D(D_j)}{|D|} \end{aligned} \tag{6}$$

where $BestMatch_G$ and $BestMatch_D$ are functions which deliver the quality of the closest match of a rectangle in the opposing list:

$$\begin{aligned} BestMatch_G(G_i) &= \max_{j=1...|D|} \frac{2 \cdot Area(G_i \cap D_j)}{Area(G_i) + Area(D_j)} \\ BestMatch_D(D_j) &= \max_{i=1...|G|} \frac{2\ Area(D_j \cap G_i)}{Area(D_j) + Area(G_i)} \end{aligned} \tag{7}$$

If a rectangle is matched perfectly by another rectangle in the opposing list, then the match functions evaluate to 1, else they evaluate to a value < 1. Therefore, the original measures taken from the information retrieval community, given by (1), are upper bounds for the new measures given by (6). Both, precision and recall given by (6), are low if the overlap region of the corresponding rectangles is small.

A disadvantage of the ICDAR evaluation scheme is that only one-to-one matches are considered. However, in reality sometimes one ground truth rectangle is "split" into several object rectangles or several ground truth rectangles are "merged" into a single detected object rectangle. This is a problem the authors themselves report in [13]. The problem is generally encountered in detection evaluation frameworks, which is due to the fact that we are interested in evaluating the solution of a detection problem but the ground truth is specified as the "correct" solution of a segmentation problem. However, an over- or under segmented solution may very well be a correct detection.

Liang *et al.* present a method for the evaluation of document structure extraction algorithms [12]. From the two lists G and D of detected rectangles and ground truth rectangles, they create two overlap matrices σ and τ. The lines $i = 1..|G|$ of the matrices correspond to the ground truth rectangles and the columns $j = 1..|D|$ correspond to the detected rectangles. The values of these matrices correspond, respectively, to area recall and area precision between the row rectangle G_i and the column rectangle D_j:

$$\begin{aligned} \sigma_{ij} &= \mathrm{R}_{AR}(G_i, D_j) \\ \tau_{ij} &= \mathrm{P}_{AR}(G_i, D_j) \end{aligned} \tag{8}$$

Matching rectangles is done by thresholding the values in the two matrices and clustering them into groups. Different match types are supported: one-to-one matches, one-to-many matches (splits) and many-to-one matches (merges). See figure 2 for an illustration of these concepts.

Hua *et al.* [7] also take into account splits and merges. They introduce two measures: "detection quality", which relates to recall, and "false alarm rate" which relates to (1 - precision). However, each measure is calculated as product of two factors: a factor which depends on the surface ratios — similar to the ICDAR solution — and a factor which measures the rectangle fragmentation. The latter factor decreases in the case of splits and merges.

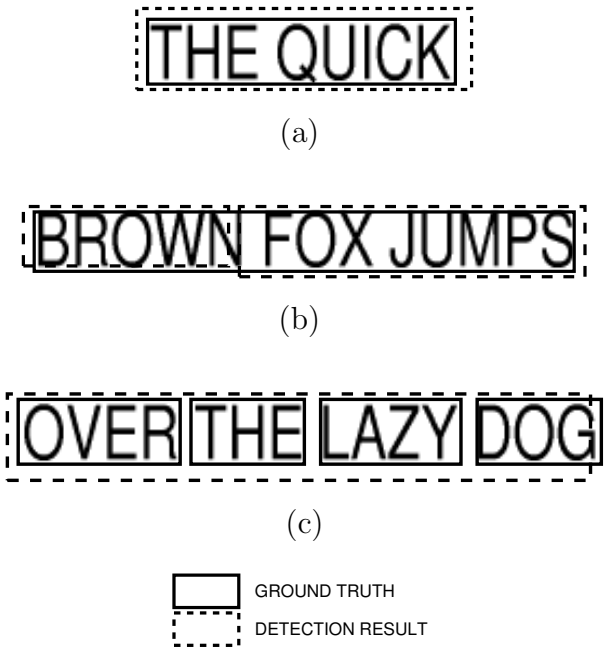


Figure 2: Different match types between ground truth rectangles and detected rectangles: (a) one-to-one match; (b) a split: a one-to-many match with one ground truth rectangle; (c) a merge: a one-to-many match with one detected rectangle.

The measures are normalized according to a detection difficulty value, which is estimated from the ground truth image. It takes into account the rectangle size and the variance of the character height. The overall detection performance is weighted by a detection importance value, which is part of the ground truth.

The evaluation protocol used for the ICDAR 2003 Page segmentation contest [3] is based on the same principles as Liang's method. The overlap matrices (they call them "MatchScore tables") are used to match ground truth entities to detected entities, where an entity (i.e., a region) may contain text, graphics, line-art, a separator or noise, which makes an adaptation of the overlap matrices necessary in order to evaluate the classification of each region. Splits and merges are supported. For each match, a performance value is calculated as the harmonic mean of a recall type measure and a precision type measure. The global performance value for all entities is a computed as a weighted sum of the individual scores.

This page segmentation protocol is very similar to the other rectangle methods described above, in particular to Liang's method, the difference being the evaluation of the region type classifier and some details in the computation of the recall and precision measures. However, it suffers from the same drawbacks: the lack of intuitivity and the ambiguity of the response due to the mixture of detection quality and detection quantity.

Landais *et al.* propose an evaluation measure which is not based on the overlap information [11]: they consider a pair of detected/groundtruth rectangles as matching if and only if the centroid of one rectangle is contained in the other rectangle. Although this solution is tempting since it avoids the usage of parameters, it tends to accept matches with very low area recall and/or precision and it does not give an information on the quality of the detection.

In the context of the Graphics Recognition Workshop (GREC) competitions, algo-

rithms for the detection of lines and arcs are evaluated. Although these graphics objects are different from rectangles, the proposed evaluation algorithms do share common features with the algorithms designed for rectangle matching. In [21], the GREC organizers describe two evaluation types, one on pixel level and one on vector level. The latter matches lines and arcs by comparing their endpoints and placing thresholds on the distances between these endpoints and the curves. For each matching pair of line/arc segments, a complex quality measure is proposed, which combines measures of endpoint distance, line overlap, line with quality, line style quality and line shape quality. For the ensemble of lines and arcs these measures are combined in order to form two classical measures: vector detection rate, which corresponds to a sort of recall measure, and vector false alarm rate, which relates to a sort of precision measure.

Like the classical rectangle based protocols, this algorithm combines detection quality and detection quantity in a single measure, which makes it hard to understand the behavior of the algorithm to evaluate. Furthermore, the complexity of the quality measure is at the same time its main drawback: the performance values are difficult to understand.

4 Object count/Area graphs

Area recall and area precision are easy to interpret as long as there are only two rectangles involved: a single ground truth rectangle and a single detection result rectangle. However, in the case of multiple images or a single image with multiple text rectangles, a combination of the measures is not straightforward.

This is the main drawback of the existing evaluation schemes described in the previous section: the way the overlap information is accumulated during the calculation of the evaluation measures leaves room for ambiguity. For instance, a recall of 50% could mean that 50% of the ground truth rectangles have been matched perfectly, or that all ground truth rectangles have been found but only with an overlap of 50%, or anything in between these two extremes. As a consequence, these recall and precision measures are not very intuitive: it is impossible to determine, how many text rectangles have been detected. Similarly, the quality of the detection is not apparent.

4.1 Requirements of an evaluation algorithm

We developed an evaluation scheme which addresses these problems. Its design has been guided by the following goals:

1. The approach should provide a quantitative evaluation: the evaluation measure should intuitively tell how many text rectangles have been detected correctly, and how many false alarms have been created.

2. The approach should provide a qualitative evaluation: it should give an easy interpretation of the detection quality.

3. It should support one-to-one matches, one-to-many matches and many-to-one matches (splits and merges).

4. The measure must scale up to multiple images without losing its power and ease of interpretation.

The most important constraint of our design goals is the contradiction between goal (1), to be able to count the number of detected rectangles, and goal (2), to be able to measure detection quality. Indeed, the two goals are related: the number of rectangles we consider as detected depends on the quality requirements which we impose for a single rectangle in order to be considered as detected. For this reason we propose a natural way to combine these two measures: two-dimensional plots which illustrate their dependence. More precisely, on the y-axis we plot the two measures which are the most interesting for us: object counts, *i.e.* the measures related to goal (1):

$$\begin{aligned} \mathrm{R}_{OB} &= \frac{\text{N.o. correctly detected rectangles}}{\text{N.o. rectangles in the database}} \\ \mathrm{P}_{OB} &= \frac{\text{N.o. correctly detected rectangles}}{\text{Total n.o. detected rectangles}} \end{aligned} \tag{9}$$

As stated above, these two measures depend on the quality requirements, which are imposed using two measures: area recall and area precision. In other words, the detection performance is illustrated using two diagrams, where the first shows the dependence on area recall and the second shows the dependence on area precision. Each diagram, on the other hand, contains two graphs: one plots object recall, the other one object precision (see figure 5 in the results section for an example).

The reminder of this section describes in detail how object recall and object precision are calculated given fixed constraints on area recall and area precision.

4.2 Rectangle matching

The computation of the measures given in (9) requires for each ground truth rectangle G_i the determination whether it has been detected or not, and for each rectangle D_i in the detection result the determination whether its detection is correct or not. These decisions are taken based on constraints imposed on the detection quality, *i.e.* the overlap between detection result and ground truth. In order to take into account one-to-one as well as one-to-many matches (splits) and many-to-one matches (merges), we calculate the overlap matrices σ and τ introduced by Liang *et al.* in [12], as described in section 3.

The matrices are analyzed in order to determine the correspondences between the two rectangle lists. In general, a non zero value in an element with indices (i, j) indicates, that ground truth rectangle G_i overlaps with result rectangle D_j. However, the two rectangles are matched only if the overlap satisfies the quality constraints, *i.e.* if area recall and area precision are higher than the respective constraint:

$$\begin{array}{lll} (a) & \sigma_{ij} > t_r \\ (b) & \tau_{ij} > t_p \end{array} \tag{10}$$

where $t_r \in [0, 1]$ is the constraint on area recall and $t_p \in [0, 1]$ is the constraint on area precision. In detail, the different matches are determined as follows:

one-to-one matches: one ground truth rectangle G_i matches with a result rectangle D_j if row i of both matrices contains only one element satisfying (10) and column j of both matrices contains only one element satisfying (10). This situation is shown in figure 2a.

one-to-many matches (splits): one ground truth rectangle G_i matches against a set S_o of result rectangles $D_j, j \in S_o$ if

- a sufficiently large proportion of the ground truth rectangle has been detected (condition (10a) in a "scattered" version): $\sum_{j \in S_o} \sigma_{ij} \geq t_r$, and
- each contributing result rectangle overlaps enough with the ground truth rectangle to be considered a part of it (condition (10b) in a "scattered" version): $\forall j \in S_o : \tau_{ij} \geq t_p$.

Figure 2b illustrates this match type.

many-to-one matches (merges): one result rectangle D_j matches against a set S_m of ground truth rectangles if

- A sufficiently large portion of each ground truth rectangle is detected (condition (10a) in a "scattered" version): $\forall i \in S_m : \sigma_{ij} \geq t_r$, and
- Each ground truth rectangle has been detected with enough area precision (condition (10b) in a "scattered" version): $\sum_{i \in S_m} \tau_{ij} \geq t_p$

Figure 2c illustrates this situation.

many-to-many matches (splits and merges): this match type is currently not supported by our algorithm. Our experiments showed, that this situation does not occur very often in the case of text detection.

If a situation occurs which requires simultaneous splits and merges, then the algorithm translates this situation into several splits or a set of splits and one-to-one matches: each ground truth rectangle in the matching set is either part of a split if it is matched against several detected rectangles, or it is part of a one-to-one match if it is matched against a single detected rectangle. The drawback of this implementation is a slight unjustified punishment of combined splits and merges, since detected rectangles may be part of several sets of splits. In each set, the part of the detected rectangle which covers a ground truth rectangle of another set, is falsely reported as "missing" in the original set.

Based on this matching strategy, the recall and precision measures which we intuitively described in (9), can be finally defined as follows:

$$\begin{aligned} \mathrm{R}_{OB}(G, D, t_r, t_p) &= \frac{\sum_i Match_G(G_i, D, t_r, t_p)}{|G|} \\ \mathrm{P}_{OB}(G, D, t_r, t_p) &= \frac{\sum_j Match_D(D_j, G, t_r, t_p)}{|D|} \end{aligned} \tag{11}$$

where $Match_G$ and $Match_D$ are functions which take into account the different types of matches described above and which evaluate to the quality of the match:

$$
\begin{aligned}
Match_G \quad & (G_i, D, t_r, t_p) = \\
& = \begin{cases} 1 & \text{if } G_i \text{ matches against} \\ & \text{a single detected rectangle} \\ 0 & \text{if } G_i \text{ does not match against} \\ & \text{any detected rectangle} \\ f_{sc}(k) & \text{if } G_i \text{ matches against} \\ & \text{several } (\rightarrow k) \text{ detected rectangles} \end{cases}
\end{aligned}
$$

$$
\begin{aligned}
Match_D \quad & (D_j, G, t_r, t_p) = \\
& = \begin{cases} 1 & \text{if } D_j \text{ matches against} \\ & \text{a single detected rectangle} \\ 0 & \text{if } D_j \text{ does not match against} \\ & \text{any detected rectangle} \\ f_{sc}(k) & \text{if } D_j \text{ matches against} \\ & \text{several } (\rightarrow k) \text{ detected rectangles} \end{cases}
\end{aligned}
$$

where $f_{sc}(k)$ is a parameter function of the evaluation scheme which controls the amount of punishment which is inflicted in case of scattering, *i.e.* splits or merges. If it evaluates to 1, then no punishment is given, lower values punish more. In our experiments we set it to a constant value of 0.8.

Another possibility could be to use two different functions in the expressions $Match_G$ and $Match_D$ in order to punish over segmentation differently than under segmentation. This might be useful if text detection is followed by text recognition. Furthermore, more scattering might be punished more severely by adding a dependence to the number of rectangles k, for instance by setting $f_sc(k) = \frac{1}{1+\ln(k)}$, which corresponds to the fragmentation index suggested by Mariano et al. [15].

As a final remark, please note, that text which is only partly detected and therefore not matched against a ground truth rectangle, will correctly decrease the precision measure, in contrast to the ICDAR evaluation scheme described in section 3.

4.3 Multiple images

In the case of N images, we compare several lists $G^k \in \overline{\boldsymbol{G}}, k = 1..N$ of ground truth rectangles with several lists $D^k \in \overline{\boldsymbol{D}}, k = 1..N$ of result rectangles. As in information retrieval, the results on multiple images may not be accumulated by summing the recall or precision values. Instead, object recall and object precision are defined as follows:

$$
\begin{aligned}
\mathrm{R}_{OB}(\overline{\boldsymbol{G}}, \overline{\boldsymbol{D}}, t_r, t_p) &= \frac{\sum_k \sum_i Match_G(G_i^k, D^k, t_r, t_p)}{\sum_k |G^k|} \\
\mathrm{P}_{OB}(\overline{\boldsymbol{G}}, \overline{\boldsymbol{D}}, t_r, t_p) &= \frac{\sum_k \sum_j Match_D(D_j^k, G^k, t_r, t_p)}{\sum_k |D^k|}
\end{aligned}
\tag{12}
$$

4.4 Constructing the graphs

As explained before, the object related measures introduced in equation (12) depend on two constraints t_r and t_p which impose constraints on the detection quality. The

Figure 3: An example rectangle detected with area recall = 100% and area precision = 50%.

performance diagrams are produced by fixing one constraint to a set value, varying the second one (assigned to the x-axis) and plotting object recall and object precision on the y-axis of two graphs.

Figure 5 in the experimental section shows an example of the two diagrams obtained this way. The diagram shown in figure 5a is generated by varying the constraint on area recall, t_r, while constraint t_p is held to a fixed value. The diagram is composed of three graphs: object recall, object precision and the harmonic mean of the two measures. Similarly, figure 5b is created varying constraint t_p while constraint t_r is fixed.

The diagrams are easily interpreted by looking at the dynamics of the graphs: in this particular example, the fact that object recall never drops to zero when area recall approaches 1 means, that most of the text rectangles are detected with an area coverage of 100%, *i.e.* the detection rarely cuts parts of the ground truth rectangle. On the other hand, the fact that object recall does drop to zero when area precision approaches 1, means that all result rectangles exceed the ground truth boundaries. The particular amount of area which is detected additionally can be seen by the point/range where the object recall dramatically drops when area precision increases.

As stated above, during the creation of the graphs one of the two constraints is held fixed. The particular values assigned to the fixed constraints have been chosen empirically. However, we decided to pick different values for the two different constraints: while t_r is fixed to 0.8, we chose the lower value of 0.4 for constraint t_p. This decision is motivated by the fact that a detection result which cuts parts of the text rectangle is more disturbing than a detection which results in a too large rectangle. The value of 0.4 might seem very low, but keep in mind that the area of a rectangle grows with the square of the its side lengths. This fact is illustrated in figure 3, which shows a detection result with 50% area precision. The detected rectangle is twice as large as the ground truth rectangle, although the difference in the corner coordinates is quite small. Please refer to the discussion section for some remarks on the implications of this situation to text detection algorithms.

4.5 Three-dimensional graphs

An alternative presentation of the performance measures are three-dimensional plots of the three object related measures (recall, precision and the harmonic mean), respectively, on the z-axis, whereas t_r is assigned to the x-axis and t_p is assigned to the y-axis. Figure 6 shows an example of such a set of plots.

The advantage of a 3D plot is a gain in information: for each combination of thresholds t_r and t_p, *i.e.* for each conceivable combination of quality constraints, we are able to read the performance of the detection algorithm. However, this advantage is bought with several drawbacks, which severely hamper the usability of the plot:

- The 3D plots are more difficult and less intuitive to read. In particular, the actual performance value on one point of the performance is difficult to read.
- The different object related performance measures cannot be displayed in a single diagram comprising several plots, as in the 2D case, since the surfaces would be unreadable. Therefore, several diagrams need to be created, resulting in unnecessary need of space. This is illustrated in figure 6, which shows the plots for two detection algorithms, one column corresponding to one algorithm.
- The interpretation of a 3D graph is only possible if the function is smooth enough against changes of the quality parameters. This might not always be the case, depending on the behavior of the evaluated detection algorithm.
- The complexity of the calculations needed for the 3D plots is much higher. More precisely, complexity rises from $O(N)$ to $O(N^2)$.

In general, we think that the gain in additional information is small compared to the drawbacks of the 3D plots.

4.6 A single performance value

The performance diagrams introduced above are an easy and intuitive way to illustrate the performance of an object detection algorithm. However, very often it is useful and desirable to determine a single performance value for an algorithm, either for direct comparison of the performances of different algorithms, or to optimize the parameters of the detection algorithm, or to control the algorithm, for instance in a reinforcement learning environment [18].

For the reasons laid out in sub section 4.1, an objective comparison of the algorithms by a single scalar value is difficult, up to impossible. A single value is hardly able to characterize the complex behavior of a detection algorithm, which makes it necessary to resort to compromises. At first sight, a simple solution might be to hold the quality constraints t_p and t_r at fixed values, calculate object recall and object precision and combine them in a harmonic mean. However, this evaluation would depend heavily on the particular chosen values. One algorithm could outperform another one for given quality constraints, while it could show a weaker performance for other constraints.

A special case of this solution would be the end points of the curves ($t_p = 1$ and $t_r = 1$, respectively). As for any other fixed value of t_p and t_r, this solution ignores the behavior of the algorithm for other detection quality constraints. It is immediately clear that this behavior is important when we look at figure 8. H.W.David's algorithm (displayed in the top row) and Todoran's algorithm (displayed in the 4^{th} row) share the same end point in the right diagram: Recall=Precision=0 for $t_r = 1$. This means, that both algorithms detect rectangles which are larger than the ground truth rectangles, since not a single rectangle is considered as found if a precision of 100% is required. However, looking at the rest of the curve, we can see the difference in the behavior of the two detection

algorithms: H.W.David's algorithm features a Recall of almost 60% across a large section of the precision quality constraint. Recall only drops rather sharply when a quality constraint of about 55% is reached. Summing it up, we might say that H.W.David's algorithm detects 60% of the rectangles with realistic assumptions on detection precision. On the other hand, Todoran's algorithm shows an almost linear dependance of Recall on detection quality. This tells us, that the differences in size between the ground truth rectangles and the detected rectangles are more equally distributed, the algorithm's behavior is therefore less predictable.

A good indicator should cover the performance of the evaluated algorithm across a whole range of quality constraints. We therefore propose the proportion of the graph area which is beneath the performance graphs as a reliable and objective measure, which is equivalent to the mean value of object measures over all possible constraint values.

More precisely, we first calculate the area proportion separately for object recall and object precision:

$$\begin{aligned} \mathrm{R}_{OV} &= \tfrac{1}{2T}\sum_{i=1}^{T} \mathrm{R}_{OB}(\overline{\boldsymbol{G}}, \overline{\boldsymbol{D}}, i/T, t_p) + \\ &+ \tfrac{1}{2T}\sum_{i=1}^{T} \mathrm{R}_{OB}(\overline{\boldsymbol{G}}, \overline{\boldsymbol{D}}, t_r, i/T) \\ \mathrm{P}_{OV} &= \tfrac{1}{2T}\sum_{i=1}^{T} \mathrm{P}_{OB}(\overline{\boldsymbol{G}}, \overline{\boldsymbol{D}}, i/T, t_p) + \\ &+ \tfrac{1}{2T}\sum_{i=1}^{T} \mathrm{P}_{OB}(\overline{\boldsymbol{G}}, \overline{\boldsymbol{D}}, t_r, i/T) \end{aligned} \tag{13}$$

The final performance value is the harmonic mean of the two measures:

$$\mathrm{Perf}_{OV} = 2\,\frac{\mathrm{P}_{OV}\ \cdot\ \mathrm{R}_{OV}}{\mathrm{P}_{OV}\ +\ \mathrm{R}_{OV}} \tag{14}$$

The parameter T is a granularity parameter which controls the trade-off between the computational complexity of the evaluation algorithm and the precision of the integration approximation. However, it is not likely that the object related measures change sharply after changing the quality constraints in very small steps. Consequently, in our experiments, we set the parameter to $T = 20$.

5 Evaluating the influence of the test database

As for information retrieval (IR) tasks, the measured performance of an object detection algorithm highly depends on the test database. It is obvious, that the nature of the images determines the performance of the algorithm. As an example we could think of the object type (different poses for face detection, artificial text or scene text for text detection), its size, the image quality, noise, compression artifacts etc. For this reason, an objective comparison between different algorithms will only be possible if the respective communities decide on shared common test databases. Alternatively, we recommend tackling this problem partly by performing different experiments for different test databases with different difficulties.

On the other hand, the nature of the images is not the only variable which determines the influence of the test database on the detection performance. The structure of the data, *i.e.* the ratio between the relevant data and the irrelevant data, is a major factor which influences the results. This simple but important fact has been overlooked by the information retrieval community for a long time.

In [9], Huijsmans *et al.* call attention to this fact and adapt the well known precision/recall graphs in order to link them to the notion of generality for an IR system, which is defined as follows:

$$\text{Generality}_{IR} = \frac{\text{N.o. relevant items in the database}}{\text{n.o. items in the database}} \tag{15}$$

Very large databases with low generality, *i.e.* much irrelevant clutter compared to the relevant material, produce results with lower precision than databases with higher generality. This makes sense, since the probability to retrieve a relevant item is lower if there is more irrelevant noise present in the database. A standard IR system presents the retrieved items to the user in a result set of predefined size. Since this size is fixed, with falling generality the amount of relevant material in the result set — thus the recall — will tend to be smaller. Thus, recall and precision depend on the generality of the database. In IR one is interested in the retrieval performance with respect to the generality as well as with respect to the size of the result set, which determines the search effort for the user. The dependence on two parameters makes three-dimensional performance graphs necessary. Alternatively, Huijsmans proposes two-dimensional graphs, which corresponds to a plane of the 3D space defined by Precision = Recall. Therefore, the graph plots Precision=Recall on the y-axis against generality on the x-axis.

However, unlike IR tasks, object detection algorithms do not work with items (images, videos or documents). Instead, images (or videos) are used as input, and object rectangles are retrieved. Nevertheless, a notion of generality can be defined as the amount of objects which are present in the images of the database. We define it to be

$$\text{Generality} = \frac{\text{N.o. object rectangles in the database}}{\text{N.o. images in the database}} \tag{16}$$

Note, that using this definition, generality may attain values $\gg 1$. This is not a problem since the value is interpreted by humans or used in plots (see section 6.1).

Another difference to IR systems is the lack of a result set window, because all detected items are returned to the user. Therefore, the generality of the database does influence precision, but *not* recall. Thus, the influence of the database structure on the system performance can be shown with simple two-dimensional precision/generality graphs. The graphs introduced by Huijsmans are displayed on a logarithmic scale, since the generality in very large IR databases may attain very low values. On the other hand, the amount of objects per image (or per video frame) should remain relatively high, therefore we decided to display the graphs on a linear scale.

A decision needs to be made concerning the generality level of the database when result tables or graphs are displayed which contain a fixed level of generality. In other words, it is necessary to decide how many images with zero ground truth (no object present) should be included in the database. The exact amount depends on the particular application. The *a priori* probability of an image to contain exotic objects, as for instance water falls or fire might be very low. Another determining factor is the type of medium. In most cases, for applications working on single images the probability

is higher than for applications working on video sequences. In this document, where experiments were performed on images containing text objects (see section 6), we chose a mixture of 50% images with relevant objects and 50% images without relevant objects.

6 Experimental results

We tested our new evaluation metric on two different sets of text detection algorithms which have been applied to different image test databases, respectively.

6.1 Evaluating text detection in video frames

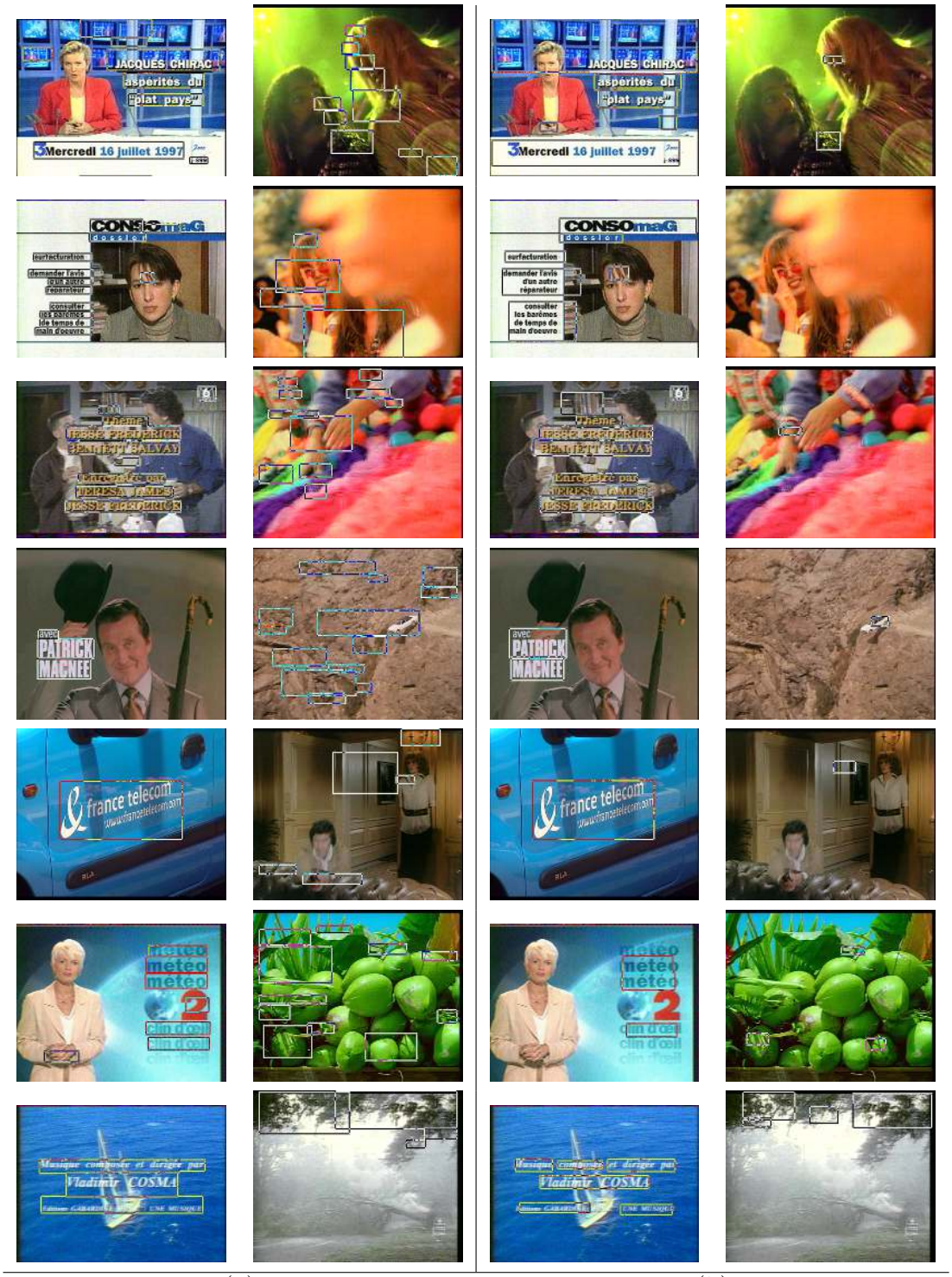
The first test dataset contains two algorithms, which have been developed by the authors. Details are given in [23] and [22], respectively. For the sake of brevity, in the reminder of this paper we call them *algorithm 1* and *algorithm 2*. The two methods have been applied to a small set of video frames in the CIF format (384×288 pixels), which have been provided by INA[4] and France Tï¿½ï¿½om. This small database contains only 14 images, which makes it possible to visually show the detection results superimposed on the images (see figure 4). Thus, a direct comparison can be made between the detected object rectangles and, respectively, the object/area performance graphs (figure 5) and the performance/generality graphs (figure 7).

The left column of figure 5 shows object recall and precision depending on the constraints imposed on area recall. Object recall and precision decrease only slowly when t_r approaches 1, which means that most of the object rectangles are detected with their entire area. Note, that the object recall graph drops faster for algorithm 2, illustrating a lack of the algorithm to detect the whole area of each rectangle. This can be confirmed looking at the superimposed results in figure 4a and figure 4b, respectively.

The right column of figure 5 shows object recall and precision depending on the constraints imposed on area precision. Object recall and precision drop to zero when t_p approaches 1, illustrating the fact that all object rectangles are larger than the corresponding ground truth rectangles. We can see that algorithm 1 is more precise, since object recall drops slower when the t_p is increased. Again, this is confirmed looking at the superimposed results in figure 4.

Figure 7 shows the dependence of the performance on the database structure. In order to create graphs falling with lower generality, inverse generality has been assigned to the x-axis. More precisely, the left most value of the graph ($1/Generality = 0.2$) corresponds to a set with 7 images containing text only, whereas the right most value of the graph ($1/Generality = 0.4$) corresponds to a set with 7 images containing text and 7 images not containing text. In order words, the left most value is calculated using only the first column of images in figure 4, and as we traverse the x-axis to left, lowering generality, more and more non-text images taken from column 2 of figure 4 are added to the dataset. As we can see, object recall stays constant, since adding non-text images does not add any new ground truth images. However, precision decreases due to false alarms. We note that the graphs for algorithm 2 are flatter, illustrating the fact that this algorithm produces less false alarms in images not containing text - confirmed by

[4]The *Institut National de l'Audiovisuel* (INA) is the French national institute in charge of the archive if the public television broadcasts. See http://www.ina.fr



(a) (b)

Figure 4: Some detection examples: (a) detection algorithm 1 [23] (b) detection algorithm 2 [22].

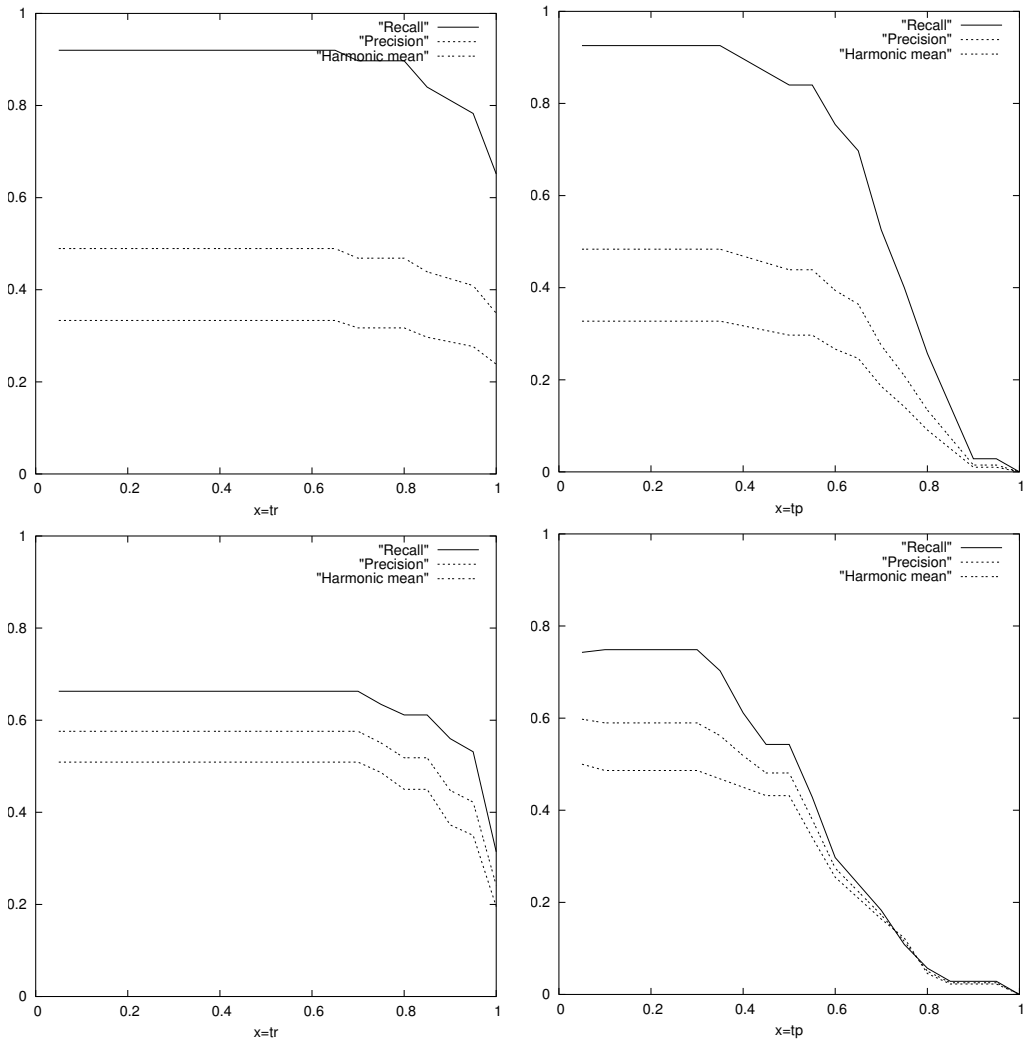


Figure 5: Results on the images shown in figure 4. Top: detection algorithm 1 [23], bottom: detection algorithm 2 [22]; Left: varying constraint t_r (area recall) while t_p is constant and equal to 0.4, right: varying constraint t_p (area precision) while t_r is constant and equal to 0.8;

the fact that algorithm 1 is based on the hypothesis that the images used do contain text [23].

Let us recall, that the values plotted on the y-axis of the generality graphs are consolidated performance values. For each value on the x-axis, *i.e.* for each generality value, and for each performance measure, i.e. precision, recall and their harmonic mean, we calculate a single value as given in equations (13) and (14).

6.2 The ICDAR 2003 text detection competition results

The second dataset consists of the text detection algorithms participating at the text detection competition organized in the framework of the 7^{th} International Conference on Document Analysis and Recognition (ICDAR), 2003 [13]. Simon Lucas, the organizer of the competition, kindly provided results of the participants in XML format. The test image database consists of various images taken with digital cameras. In contrast to the

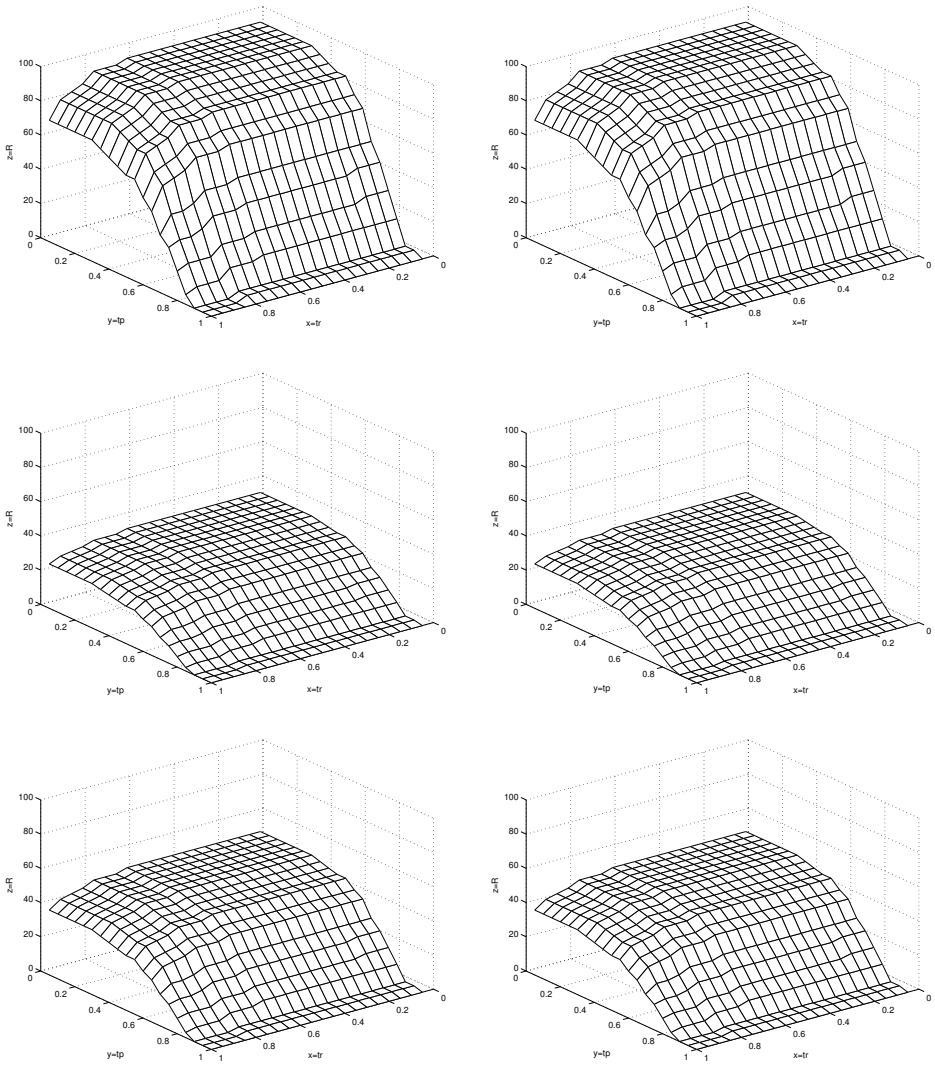


Figure 6: Results on the images shown in figure 4. From top to bottom: recall, precision, harmonic mean. Left: detection algorithm 1 [23], right: detection algorithm 2 [22];

first database, these images have been acquired in relatively high resolution: the image dimensions range between 1600×1200 pixels and 1000×800 pixels.

Four participants have been evaluated: Ashida's algorithm, H.W. David's algorithm, Wolf's algorithm, and Todoran's algorithm [14]. The third algorithm, developed by the authors of this document, corresponds to algorithm 2 evaluated in the last section. A fifth virtual participant combines the results of the other four methods using an algorithm proposed by the organizers of the competition. Descriptions of the methods can be found in [14].

Figure 8 shows the performance graphs for the five contestants. The clear winner seems to be Ashida's algorithm, which shows superior recall and precision across the whole range of quality requirements. Applying the same reasoning as in the last sub section, we clearly see the two leading algorithms differ in their detection approach: while Ashida's detected rectangles tend to be too small, H.W. David's detected rectangles tend to be too large.

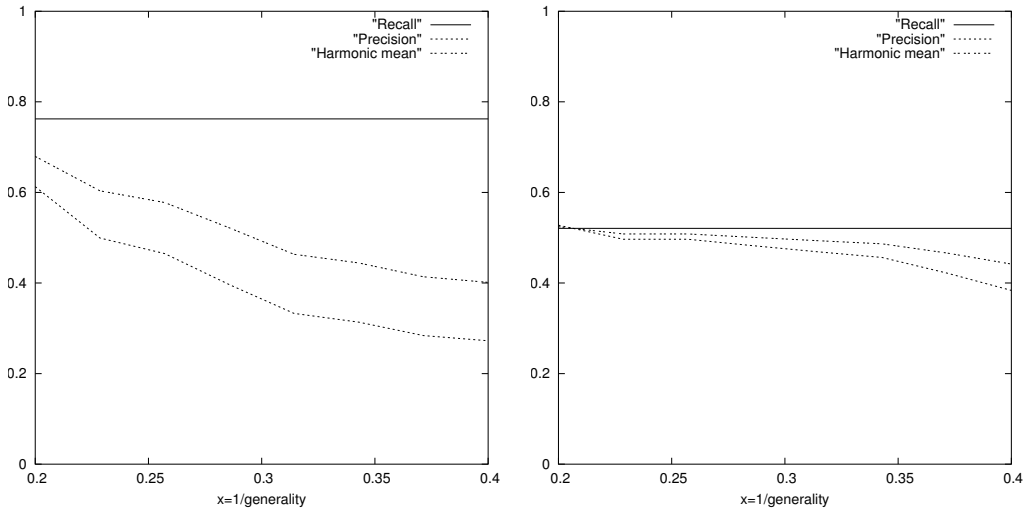


Figure 7: Results on the images shown in figure 4, with varying generality: (a) detection algorithm 1 [23] (b) detection algorithm 2 [22].

Method	ICDAR Metric (eq. (6))			New Metric (eq. (13))		
	Recall	Precision	H.Mean	Recall	Precison	H.Mean
Ashida	46.0	55.0	50.0	41.7	55.3	47.5
H.W. David	46.0	44.0	45.0	46.6	39.6	42.8
Wolf *et al.*	44.0	30.0	35.0	44.9	19.4	27.1
Todoran	18.0	19.0	18.0	17.9	14.3	15.9
All combined	N/A	N/A	N/A	50.1	53.1	51.7

Table 1: Single performance values on the ICDAR 2003 data set.

In general, the performance characteristics of the detection algorithms are well illustrated by the graphs: the proportion of "recalled" objects and the proportion of false alarms is immediately visible for the quality a user might want to impose. Inflection points in the performance curves show the precision of the detection algorithm. For instance, the inflection at point $t_r = 0.8$ of the object recall graph of Ashida's algorithm (top row, left column), illustrates the fact that most objects are detected with about 80% of the object area. If the quality constraints are further increased, the number of objects considered as detected drops.

Table 1 presents the performance values for each algorithm compared to the original metric used during the ICDAR competition, introduced in section 3. The ranking of the algorithms stayed the same, although there are differences in the different performance values. More important, the interpretation of the values changes: recall according the ICDAR metric corresponds to the area recall, averaged across all images, which results in the ambiguity described in section 4. On the other hand, the new recall value corresponds to averaged object recall and may thus be interpreted as the proportion of correctly detected objects, averaged across the whole range quality constraints a user might want to impose. Precision is interpreted in a similar manner.

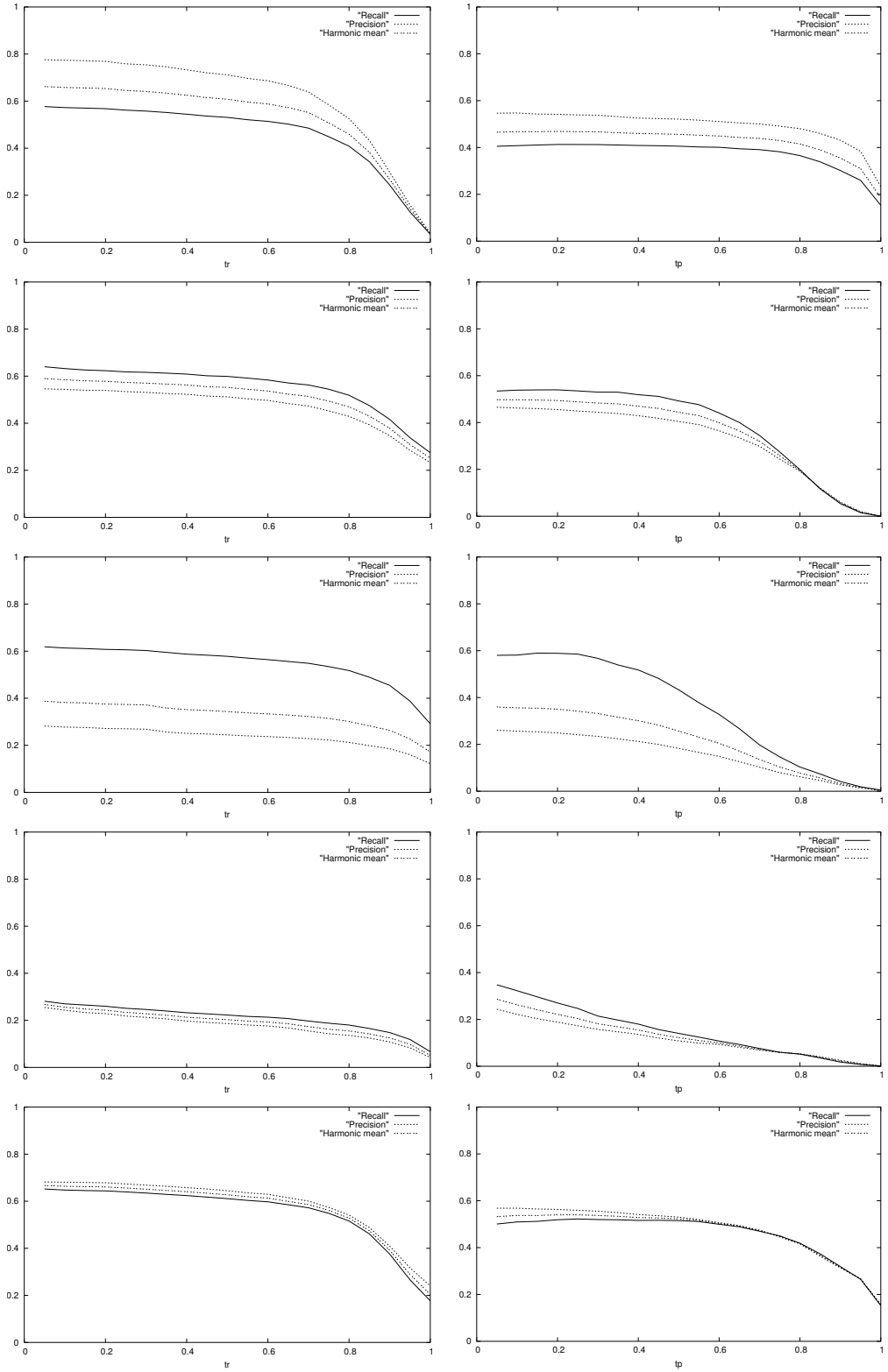


Figure 8: Results on the ICDAR 2003 data set. Top row: Ashida's algorithm, 2^{nd} row: H.W.David's algorithm; 3^{rd} row: our algorithm; 4^{th} row: Todoran's algorithm; Bottom row: combined result; Left: varying constraint t_r (area recall) while t_p is constant and equal to 0.4, right: varying constraint t_p (area precision) while t_r is constant and equal to 0.8;

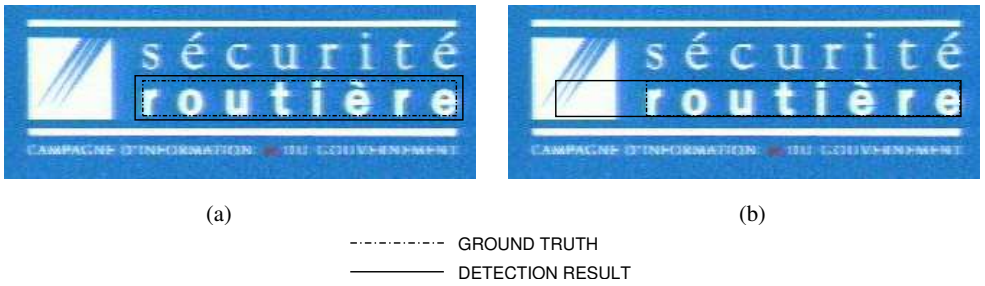


Figure 9: Ground truth rectangle and detected rectangles for an example image. Precision and recall for figures (a) and (b) are equivalent.

7 Discussion and Conclusion

In this paper we have presented a novel method to evaluate object detection algorithms. The proposed method is applicable to any kind of object, as long as the detection result may be represented by a list of rectangles.

We introduced diagrams containing two dimensional graphs which depict measures on object level depending on quality constraints, making easy a clear and intuitive interpretation. A clear distinction is made between a quantitative evaluation of the detection algorithm and a qualitative evaluation. The dynamics of the graphs illustrate the behavior of the detection algorithm against different quality constraints which might be imposed by a user, where inflection points correspond to the fundamental characteristics of the detection algorithm. The proposed evaluation method overcomes several shortcomings of the existing approaches, notably the ambiguity problem which follows from the direct accumulation of overlap proportions. Since the performance values are calculated on object level, a user can directly see the number of correctly detected objects and the amount of false alarms.

For the comparison of different detection algorithms we have proposed a single performance measure which is directly derived from the performance graphs. The integral of the object level performance across the full range of quality constraints gives an intuitive and objective measure of the detection algorithm's performance.

Additionally, a graph displays the dependence of the detection algorithm's performance on the generality of the test database, *i.e.* the amount of relevant information in the database. This often overlooked criterion significantly influences the measured performance of any object detection or information retrieval algorithm.

Our evaluation method is based on the amount of overlap between the ground truth rectangles and the result rectangles, not on the location of this overlap. In many applications, e.g. in the case of text detection, however, the amount of overlap between two rectangles is not a perceptively valid measure of quality, as can be seen in figure 9. Precision and recall are equivalent for both detection examples, but the detection shown in figure 9a might be considered as better, since the additional detected space is distributed over all sides of the ground truth rectangle.

As specified in section 4.4, in order to prevent the rejection of detection results as the one in figure 9a, the precision constraint t_p is set to a very low value. This is necessary because the error surface grows with the square of the additional rectangle length (or height). However, we still might want to reject detections as the one illustrated in figure 9b.

One possibility to check whether the error space is equally distributed could be to estimate the distribution of the angles of the error pixels against the center of the ground truth rectangle. Unfortunately, the angle distribution of a perfectly aligned detection, e.g. the detection shown in figure 9a, is not a uniform distribution but a distribution resulting after a piecewise application of a tangent function. A statistical test (e.g. a Kolmogorov-Smirnov test) against such a distribution after an estimation of its parameters would be possible but not very robust.

Furthermore, a statistical test using all error pixels would be overkill given the fact that the functional form of the error distribution is known and that it depends on 4 parameters only: the absolute differences of the left (respectively right, upper and lower) coordinates of the rectangle pair. We chose therefore a simpler yet more effective method, which directly checks these parameters: the 4 values described above are checked against thresholds, which are calculated from the size of the rectangle.

In the more specific case of text detection, we are more interested in detecting a horizontal disequilibrium. Therefore, we concentrate on two of the differences measures: the absolute differences of the left (respectively right) coordinates of the rectangles to match need to be smaller than a constraint which depends on the width of the ground truth rectangle. This constraint, which does not depend on the overlap information, makes sure that a situation depicted in figure 9b is unlikely to occur.

References

[1] Y. Aloimonos and A. Rosenfeld. REPLY: A Response to "Ignorance, Myopia and Naiveté in Computer Vision Systems" by R.C. Jain and T.O. Binford. *CVGIP: Image Understanding*, 53(1):120–124, 1991.

[2] A. Antonacopoulos and A. Brough. Methodology for Flexible and Efficient Analysis of the Performance of Page Segmentation Algorithms. In *Proceedings of the International Conference on Document Analysis and Recognition*, pages 451–454, 1999.

[3] A. Antonacopoulos, B. Gatos, and D. Karatzas. ICDAR 2003 Page Segmentation Competition. In *Proceedings of the International Conference on Document Analysis and Recognition*, pages 688–692, 2003.

[4] K.W. Bowyer and J.P. Jones. REPLY: Revolutions and Experimental Computer Vision. *CVGIP: Image Understanding*, 53(1):125–126, 1991.

[5] D. Doermann and D. Mihalcik. Tools and Techniques for Video Performance Evaluation. In *Proceedings of the International Conference on Pattern Recognition*, volume 4, pages 4167–4170, 2000.

[6] K. Fukunaga and R. R. Hayes. Effects of sample size in classifier design. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(8):873–885, 1989.

[7] X.-S. Hua, L. Wenyin, and H.-J. Zhang. An Automatic Performance Evaluation Protocol for Video Text Detection Algorithms. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(4):498–507, 2004.

[8] T.S. Huang. REPLY: Computer Vision Needs More Experiments and Applications. *CVGIP: Image Understanding*, 53(1):125–126, 1991.

[9] N. Huijsmans and N. Sebe. Extended Performance Graphs for Cluster Retrieval. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 26–31, 2001.

[10] R.C. Jain and T.O. Binford. Ignorance, Myopia and Naiveté in Computer Vision Systems. *CVGIP: Image Understanding*, 53(1):112–117, 1991.

[11] R. Landais, L. Vinet, and J.-M. Jolion. A goal directed methodology for groundtruthing and evaluating a commercial OCR. *Pattern Recognition (submitted)*, 2004.

[12] J. Liang, I.T. Phillips, and R.M. Haralick. Performance evaluation of document layout analysis algorithms on the UW data set. In *Document Recognition IV, Proceedings of the SPIE*, pages 149–160, 1997.

[13] S.M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young. ICDAR 2003 robust reading competitions. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition*, volume 2, pages 682–687, 2003.

[14] S.M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, R. Young, K. Ashida, H. Nagai, M. Okamoto, H. Yamamoto, H. Miyao, J. Zhu, W. Ou, C. Wolf, J.-M. Jolion, L. Todoran, M. Worring, and X. Lin. ICDAR 2003 Robust Reading Competitions: Entries, Results and Future Directions. *International Journal on Document Analysis and Recognition - Special Issue on Camera-based Text and Document Recognition*, 7(2-3):105–122, 2005.

[15] V.Y. Mariano, J. Min, J.-H. Park, R. Kasturi, D. Mihalcik, H. Li, D. Doermann, and T. Drayer. Performance Evaluation of Object Detection Algorithms. In *Proceedings of the International Conference on Pattern Recognition*, volume 3, pages 965–969, 2002.

[16] G. Nagy. Candide's Practical Principles of Experimental Pattern Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):199–200, 1983.

[17] M.A. Snyder. REPLY: A Commentary on the Paper by Jain and Binford. *CVGIP: Image Understanding*, 53(1):118–119, 1991.

[18] G.W. Taylor and C. Wolf. Reinforcement Learning for Parameter Control of Text Detection in Images and Video Sequences. In *Proceedings of the International Conference on Information & Communication Technologies (IEEE)*, 2004. IEEE Section France.

[19] C.J. van Rijsbergen. *Information Retrieval*. Butterworths, London, 2nd edition, 1979.

[20] R.A. Wagner and M.J.Fisher. The string to string correction problem. *Journal of Assoc. Comp. Mach.*, 21(1):168–173, 1974.

[21] L. Wenyin and D. Dori. A Protocol for Performance Evalution of Line Detection Algorithms. *Machine Vision and Applications: Special issue on performance evaluation*, 9(5-6):240–250, 1997.

[22] C. Wolf. *Text Detection in Images taken from Videos Sequences for Semantic Indexing*. PhD thesis, INSA de Lyon, 20, rue Albert Einstein, 69621 Villeurbanne Cedex, France, 2003.

[23] C. Wolf and J.-M. Jolion. Extraction and Recognition of Artificial Text in Multimedia Documents. *Pattern Analysis and Applications*, 6(4):309–326, 2003.