System for caption text extraction on a hierarchical region-based image representation

UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Ekaterina Zaytseva

advisor: Ferran Marques, Miriam Leon Cristobal

Department of Signal Theory and Communications. Image and Video Processing Group

Universitat Politècnica de Catalunya. BarcelonaTECH

A thesis submitted for the degree of

the European Master of Research on Information and Communication Technologies.

Contents

List of Figures

LIST OF FIGURES

List of Tables

LIST OF TABLES

1

Introduction

1.1 Motivation

The rapid growth of personal and professional multimedia databases requires the development of automated management tools. The effort devoted by the research community to content-based image indexing is huge, but bridging the semantic gap is difficult: the low level descriptors used for indexing (e.g:interest points, texture descriptors) are not enough for an efficient manipulation of big and generic image databases. The text present in a scene is usually related to the semantic context of the image and constitutes a relevant descriptor for content-based image indexing. Database indexing and retrieval tools can enormously benefit from the automatic detection and processing of textual data in images. This is specially true for caption text which is usually synchronized with the contents in the scene. Caption text is artificially superimposed on the video at the time of editing and it usually underscores or summarizes the video content. This fact makes caption text particularly useful for building keyword indexes. Another benefit of caption text extraction is its computational cost, that is lower than the cost of extracting other semantic content, such as objects, events or their relationships, see [1].

The aim of my thesis is (i) to implement the existing state of the art caption text extraction algorithm [2], developed by Miriam Leon from the UPC's Image and Video Processing group, and (ii) improve detected drawbacks and problems of mentioned algorithm. The algorithm should be implemented using the ImagePlus library of a powerful software development platform in C++, created by members of UPC's Image

and Video Processing group. The caption text extraction algorithm is a part of the indexing and annotation system that was developed in the group. The system is based on the generic region-based hierarchical representation of images. The image model is a common for the whole indexation and annotation system, that includes faces, objects, scene annotations and indexation. This is why the caption text extraction algorithm adopts the same hierarchical region based image model.

1.2 Outline of the document

This manuscript is organized as follows. In the Chapter 2 the state of the art algorithms are reviewed. In Chapter 3, the hierarchical representation of input image is explained. Chapter 4 exposes a detailed explanation of the algorithm. Concluding with the algorithm, in Chapter 5 and 6, the improvements and obtained results are explained. Finally, in Chapter 7 the conclusions concerning the presented work are outlined.

2

State of the art

Much research work has been carried out in the area of text detection and localization in both, images and video. The main difference between a document image and an image containing text is that a document image is usually a binary image where letters are monochrome, whereas images containing text have a random background, which obviously is unknown, and letters can or not be monochrome. Two different kind of text can be found in this type of images:

- **Scene text**: This text is directly captured by the camera, that means that is present in the scene. It is more difficult to localize, in terms of image processing, because it can appear in the frame with any tilt or perspective requiring transformations. Moreover, it can be affected by illumination changes and/or partly occluded. Notice that the purpose of this text is not always be readable by the viewer, but sometimes information such as street names, trade names may be useful to know its position in the sequence.

- **Caption text**: This text is artificially added on the frame. It can be static or in movement through the frames, but in any case, it must be understandable for the viewer. Caption text is also known as artificial text [3] or graphic text [4].

Mainly algorithms are focused on detecting only caption text, but other algorithms containing tools to detect both kinds of text or just scene text can be also found [5].

As proposed in [1, 6] , algorithms can be classified in two categories, those working on the compressed domain and those working on the spatial domain. Every method

takes into account different features depending on the type of sequence (e.g. sport event or news) and its quality.

In the following sections a classification of the methods is presented. Methods have been classified in two main groups, see Table 2.1. In Section 2.1 those methods working in the compressed and semi-compressed domain are described. These are methods that work directly with macro-blocks or that work in a transformed space, such as Discrete Cosinus Transform or the Discrete Wavelet Transform, respectively. In Section 2.2 those methods working in the spatial domain are described. Most of the methods in the literature belongs to this domain. The spatial methods are classified depending on the main feature used to segment the frames, namely: region-based, textured-based, correlation-based and temporal information-based methods. Let us note that some methods could be included in more than one of the groups presented hereafter since they use different features within the different steps. However, in this work, classification is based on the most outstanding feature used by each method.

Compressed Domain	Spatial Domain
Compressed domain	Region-based : Edge-based methods Connected components-based methods
Semi-compressed domain	Texture-based methods Correlation-based methods Temporal information- based methods

Table 2.1: Classification of the methods

2.1 Compressed Domain

In this group algorithms that are both in the compressed and in the semi -compressed domain are included.

2.1.1 Compressed Domain

It is based on the localization of static characters over moving background taking into account the macro-blocks belonging to P frames (MPEG-4) [7]. Moreover it assumes that: i) text has horizontal geometry, text is constituted with more than one block and

the blocks are one next to the other horizontally, ii) it does not occupy the whole frame and, iii) it has to appear at least in three frames. These three features allow the algorithm to isolate macro-blocks and to determinate if the macroblocks are candidates to contain text. Recall and precision are high in those sequences with moving background and static text, like sports sequence (e.g. score in a football match). But it cannot be used in sequences containing moving text or static background because of the fact that none of the conditions is satisfied.

2.1.2 Semi-compressed domain

This section is called semi-compressed domain, because algorithms do not work directly with macro-blocks but analyzing the DCT (Discrete Cosinus Transform) components [7], [8] and [9]. DCT together with motion compensation are utilized in the MPEG standard video compression in order to reduce spatial redundancy in a frame and temporal redundancy in consecutive frames, respectively. DCT coefficients represent spatial and directional periodicity. Thus, low level features can be directly extracted from compressed images. AC coefficients from horizontal harmonics show horizontal intensity variations; therefore, they will be high in case of having a text line. On the other hand, AC coefficients from vertical harmonics show vertical intensity variations; they will be high in case of having more than one single text line. In [10] a detailed explanation about the DCT coefficient interpretation can be found. They show that the DCT block size, the character size and their ratio are important. For instance, if each letter is bigger than the block size, a single letter stroke intensity variation will be evaluated, instead of text intensity variation relative to the background. In the same way if the letter size is too small any texture could be similar to text and easily mistaken (e.g. grass field).

In [7] classification of semi-compressed algorithms is done into edge-based and correlation-based methods. Edge-based methods take into account contrast between text and background, this is the reason why as a first step they calculate the horizontal intensity variation in the DCT coefficients. Correlation methods [11] are only applied when a shot changes, so a shot detection must be previously done. In order to detect if the new shot contents text the intra-code blocks increment is calculated in the B- and P-frames. Once the candidate blocks are chosen some text features are searched for the localization of text within the block. For example, characters have to be made of strokes

and text has to have some color homogeneity and horizontal geometry. However, in [9] this method is analyzed and discarded due to its vulnerability to scene changes. Some other transformations could be used like the DWT (Discrete Wavelet Transform). This transformation gives more information than the DCT because spatial information is not lost with the transformation. Therefore, those areas where high frequencies have high values can be more easily related with areas in the original images.. Both [4] and [12] suggest wavelet transform (DWT) because of its capability to preserve spatial information. The text boxes are found through a hybrid: wavelet transformation and neural network. The WT output provides some relevant statistical features that can be chosen. In particular, the mean, the second and third order level calculated from the HL, LH and HH subbands of the DWT are the most discriminators. These vectors are used as input in a neural network. In [13] some other transformations such as DHT (Discrete Haar Transform), DFT and WHT (Walsh-Hadamard Transform), as well as the DWT (Discrete Wavelet Transform) are explained. As a pre-processing tool, in [14] this kind of algorithms is used in a first step to localize candidate areas.

2.2 Spatial Domain

Those methods that do not process the images or frames in a compressed domain or in a transformed space are included in this group, therefore these algorithms work directly with the real pixel values and positions. The set of algorithms included in this section is the widest. In the following subsections they are classified according to the kind of image segmentation they use:

2.2.1 Region-based

Those methods based on edge detection and connected component detection are described in this section.

Edge-based algorithms

The main goal in edge-based methods is to detect those areas that have a high contrast between text and background [7, 8, 15, 16, 17, 18, 19]. In this way, the edges from the letters are identified. Usually, an edge detector as a high-boost filter [15], a Susan corner detector [17] or a Canny filter [8] is used first, and later the selected edges are merged into regions taking into account its abundance. Once these regions are

recognized, spatial cohesion features, such as size, fill factor, aspect ratio or horizontal alignment, are applied in order to first, check if these regions are consistent with its neighborhood and second, discard false positives. Some of these methods also include a verification step using a neural network or Support Vector Machine (SVM) [8]. Figure 2.1 shows an example of local contrast computation [19]. For each pixel a neighboring pixel block is taken into account and, by means of a 2D Gaussian smoothing filter, pixels close to the central pixel have more weight than pixels away from the block center. As a result, pixels with low contrast take the value 0, whereas pixels with higher contrast take the value 1. The size of the neighborhood varies depending on the character size, which has to be previously recognized. After the contrast analysis, a morphological filter has been applied to enhance regions (fig. 2.1c) with high contrast and to be able to obtain some candidate regions (fig.2.1 d). As a conclusion, in all these methods the detection of false positives is due to the presence in the image of rich textured areas, therefore many edges are present in the images, and the detection of false negatives is due to the presence of a big letters or blurred text.

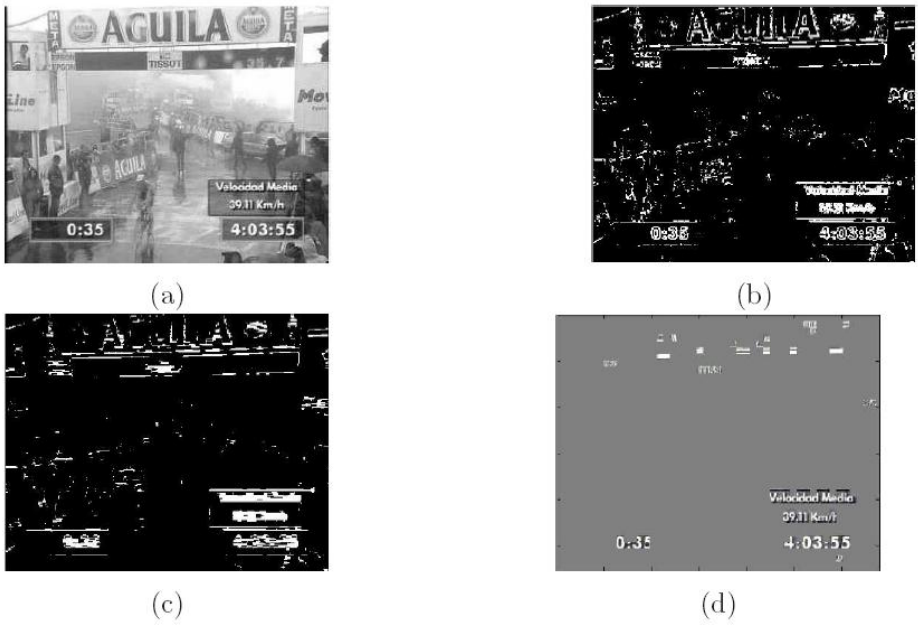


Figure 2.1: Computation of local contrast [19] - a) Input image. b) Contrast analysis output image. c) A morphological filter has been applied. d) Output image containing candidate region. Image extracted from [20]

Connected Components (CC)-based algorithms

These methods use a bottom-up approach by iteratively merge sets of connected pixels using a homogeneity criterion leading to the creation of flat-zones or Connected Components [21, 22, 23]. At the end of the iterative procedure all the flat-zones are identified. Also in this case spatial cohesion features are applied. In particular, in [23] a color input image is decomposed into multiple images by decomposing in order to generate multiple foreground images, the CC algorithm is applied in every image and the result is merged in a simple one. It does not work well when the histogram of the original image is sparse. In [21] and [22] it is also assumed that letters should be color homogeneous and similar in size to find CC. They use a split and merge algorithm. To improve the result temporal redundancy is applied: a block matching algorithm using the mean absolute media criterion allows them to discard those regions which can not be tracked. In [22] an OCR software has been also developed, but they concluded that other existing OCR in the market work better and are more robust.

2.2.2 Texture-based

A wide range of methods can be included in this group since texture-based algorithms use the property that text in images has distinct textural characteristics that distinguish them from the background [4, 24, 25, 26, 27]. For example, those methods which use the Gabor filter [24], Gaussian filter [27] or those based on the color and shape of the regions [26] and [25]. In [27] and [26], for example, the segmentation is done by means of a multiscale texture segmentation scheme. Nine second order Gaussian derivatives are applied over each subimage to find candidate text regions, then a non-linear transformation is applied to calculate at each pixel an estimation of the local energy, which is used to cluster them using the K-means algorithm. Finally, spatial cohesion is used to verify the election of candidates. The recognition rate is quite high around 94% and the false positives 5.4%. In [4] the pixel value for every channel is inserted in an arbitrary neural network, thus they do not use any specific texture segmentation. A drawback of these methods is their high computational cost and complexity due to the exhaustive search for text localization. The methods based on wavelet or FFT transform would satisfy these textural properties and could be also included in this section.

2.2.3 Correlation-based

These methods are those that use any kind of correlation in order to decide whether a pixel belongs to a character or not. In both, [28] and [29], high contrast image areas are chosen as candidate regions. Once an horizontal luminance gradient is computed with the mask [-1, 1], the Maximum gradient Difference (MGD) is calculated, see figure 2.2. This difference is calculated between the maximum and minimum value within a nx1

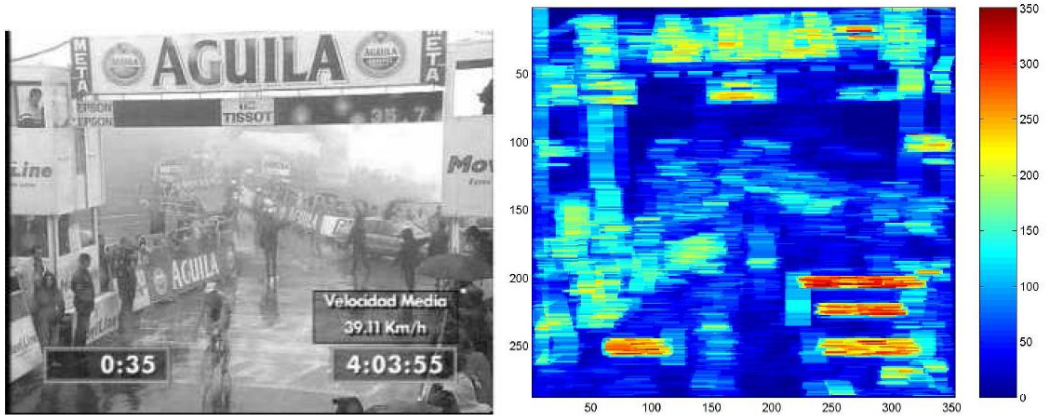


Figure 2.2: Figure of Maximum gradient difference extract from [20] -

window, which is centered in the desired pixel. In the pictures below a window size of 21 pixels has been used. In this case, a threshold should be fixed in order to discard those pixels whose MGD is lower than this threshold. Moreover, other two restrictions must be fulfilled. On one hand the number of transitions backward-to-forward and forward-to-backward must be higher than a threshold. On the other hand the mean and the variance of the horizontal distances between backward-to-forward and forward-to-backward transitions on the gradient profile must be within a certain range. This method gives rates of recognition about 88% and only 4% of false positives.

2.2.4 Temporal information-based

All the methods that have been previously mentioned in this section either do not use temporal information or use it only as a complementary tool in order to check if the selected regions or blocks are time consistent. The use of temporal information can be understood as a kind of video segmentation more than a image segmentation. Pixels

are group or classified into categories if certain features such as color homogeneity are satisfied. In [30], the use of temporal information is the main feature. After applying a shot detection technique, for each pixel in the frame, a vector is constructed, which collects the temporal value of the pixel along a fixed number of frames. The authors prove that, computing the PCA for each of these vectors, feature vectors related to the background can be separated from those related to text. The first three components of the PCA are enough to discard those pixel belonging to background. But, the main constraint of this method is that it can only be applied when sequences have static text and a moving background, otherwise pixels will be misclassified.

3

Framework

This chapter refers to the framework in terms of which the caption text detection algorithm is implemented.

Caption text detection and extraction is one of the objectives of more general task - object extraction. Due to the complexity and diversity of the problem of object extraction from images there were developed different approaches to simplify problem solving. In general object extraction is based on the image segmentation, which is a very long standing and fundamental problem in computer vision.

The segmentation refers to the partition of the digital image in the multiple segments. Based on the different forms of image representation, three main segmentation approaches could be determined: feature-based, contour-based and region-based. Feature-based approach relies on the image representation as a set of pixels, where features can be defined by the similarity of the pixel measures(color, intensity, etc..). Contour-based approach relies on the representation of the image as a set of objects, separated by their boundaries. And the region-based approaches relies on the image representation as a union of the regions, which totally covers the image. Regions correspond to the surfaces, objects or natural parts of the objects. We would like consider the caption text in the image as an unique object, but it could be segmented in more detailed entities, e.g. characters, words, text background. So it looks logic to choose the region-based approach for text bar extraction of the image.

Region-based image analysis represents a reduced space of entities which is more convenient for solving the object extraction problem. The pyramid representations in

images via tree structures are recognized methods for region-based analysis. One of the pyramid representation is the Binary Partition Tree image representation, see [31].

As it is defined in [32], the Binary Partition Tree (BPT) is a structured representation of the regions that can be obtained from an initial partition. In other words, it is a structured representation of a set of hierarchical partitions in which the finest level of detail is given by the initial partition. The leaves of the tree represent regions that belong to this initial partition. The remaining nodes of the tree are associated to regions that represent the union of two children regions. The root node usually represents the entire image support. This representation should be considered as a compromise between representation accuracy and processing efficiency. Indeed, all possible mergings of regions belonging to the initial partition (described by the RAG of the initial partition) are not represented in the tree. Only the most likely or useful merging steps are represented in the BPT. The connectivity encoded in the tree structure is binary in the sense that a region is explicitly connected to its sibling (since their union is a connected component represented by the parent), but the remaining connections between regions of the original partition are not represented in the tree. Therefore, the tree encodes only part of the neighborhood relationships between the regions of the initial partition.

The Binary Partition Tree should be created in such a way that the most interesting or useful regions are represented. This issue can be application dependent. However, a possible solution, suitable for a large number of cases, is to create the tree by keeping track of the merging steps performed by a segmentation algorithm based on region merging. This information is called the *merging sequence*. Starting from an initial partition which can be the partition of at zones or any other pre-computed partition, the algorithm merges neighboring regions following a homogeneity criterion until a single region is obtained.

An example is shown in Figure 3.1. The original partition involves four regions. The algorithm merges the four regions in three steps. In the first step, the pair of most similar regions, 1 and 2, are merged to create region 5. This is indicated in the Binary Partition Tree with a node whose label is 5 and that has to children nodes, 1 and 2. Then, region 5 is merged with region 3 to create region 6. Finally, region 6 is merged with region 4 and this crates region 7 corresponding to the region of support of the whole image. In this example, the merging sequence is: $(1, 2) \parallel (5, 3) \parallel (6, 4)$. This merging sequence progressively denes the Binary Partition Tree as shown in Figure 3.1.

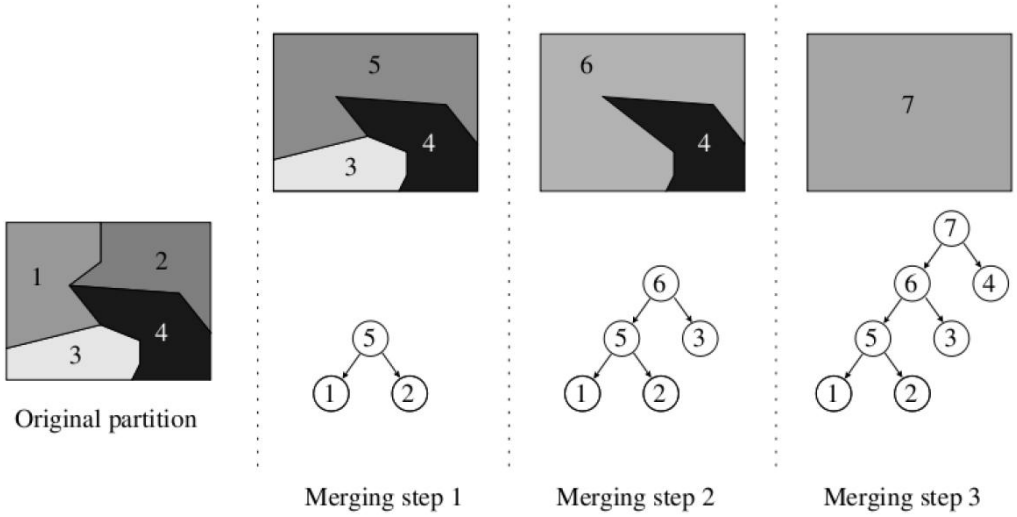


Figure 3.1: Example of Binary Partition Tree creation extracted from [32] -

In this case the initial partition is made up of 4 regions and thus, the number of nodes of the tree is $4 + (4 - 1) = 7$.

In a more general case, we may start creating the tree from an initial partition P made of N_P regions. The number of mergings that are needed to obtain one region is $N_P - 1$. Therefore, the number of nodes of the Binary Partition Tree is thus $2N_P - 1$.

As it said in [2], the BPT represents a set of regions at different scales of resolution and its nodes provide good estimates of the objects in the scene. Using the BPT representation in object detection, the image has to be analyzed only at the positions and scales that are proposed by the BPT nodes. Therefore, the BPT can be considered as a means of reducing the search space in object detection tasks. In object detection applications, the use as initial partition of a very accurate partition with a fairly high number of regions is appropriate. Since this partition is used to ensure an accurate object representation, it is called the accuracy partition (see Figure3.2). Moreover, in the context of object detection, it is useless to analyze very small regions because they cannot represent meaningful objects. As a result, two zones are differentiated in the BPT: the accuracy space providing preciseness to the description (lower scales) and the search space for the object detection task (higher scales). A way to define these two zones is to specify a point of the merging sequence starting from which the regions that are created are considered as belonging to the search space. The partition that is

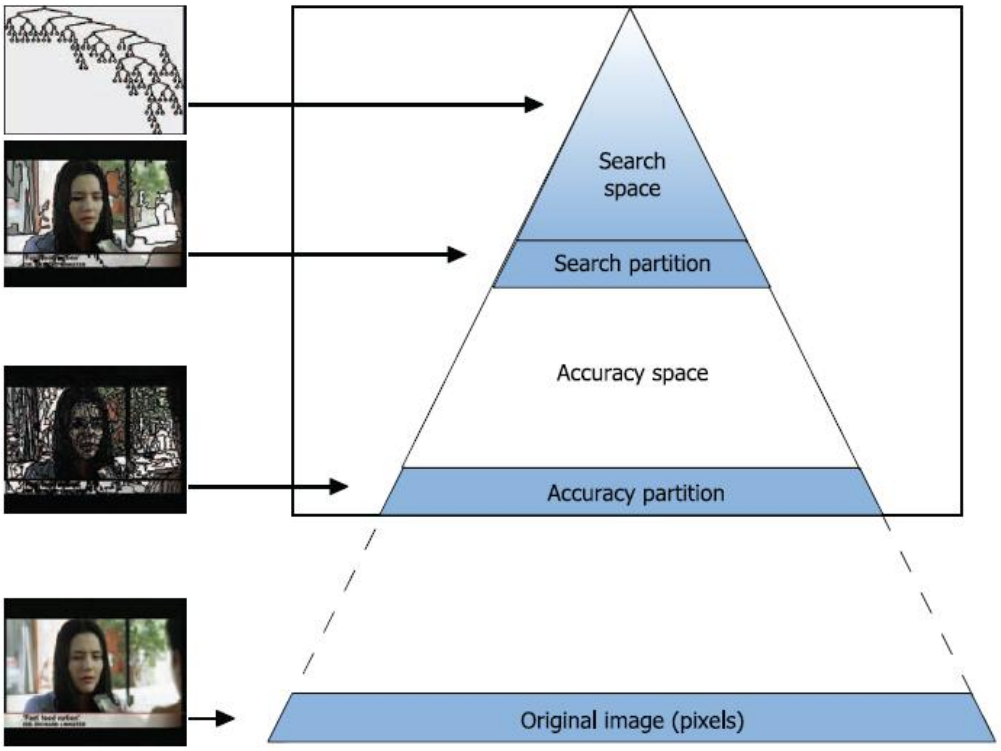


Figure 3.2: Example of Region-based hierarchical representation extracted from [2] -

obtained at this point of the merging process is called the search partition (see Figure 3.2).

In the case of caption text detection, text bars are assumed to be the objects to be detected, and they are extracted by the analysis of the search space. Due to the structure of the BPT, object extraction can be done by direct examination the region properties. This is known as tree browsing and can be done automatically by evaluating tree nodes according to some criterion.

The goal of the caption text detection algorithm is the search of criteria for detection the regions which correspond to the text.

4

Caption Text Detection

4.1 Overview

First of all we define the term "caption text" and then give some overview of the current chapter.

Caption text can be defined as a text, artificially added to the image, that is highly contrasted, textured and horizontally aligned. Normally captions are added to duplicate and/or expand information transmitted with image, also it enables those who are deaf and hard of hearing to have full access to media materials (in case of video with audio content).

This chapter refers to the caption text detection and extraction algorithm, that contains several parts:

- text candidate spotting;
- text characteristics verification;
- consistency analysis for output.

Text candidate spotting and text characteristic verification are done in terms of region-based descriptors, that are trained to model a caption, see [33]. In turn the consistency analysis for output could be subdivided in:

- extracting of caption text objects (CTOs);
- binarization of caption text objects;

- post binarization processing.

The scheme of the algorithm is in the Figure 4.1.

Each part of the algorithm is explained in a separate section. The detected problems are discussed at the end of the chapter.

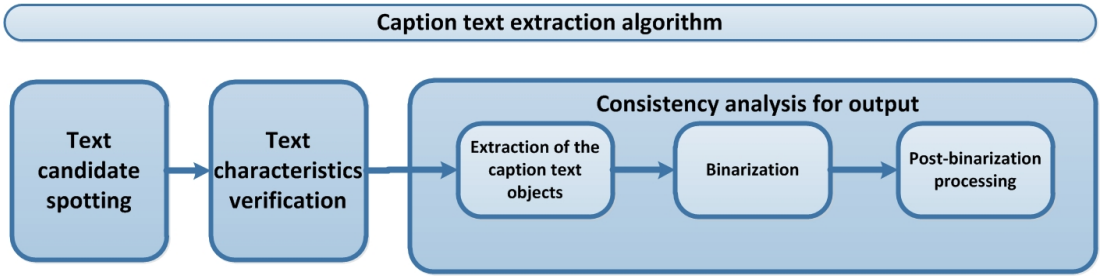


Figure 4.1: Main blocks of caption text extraction algorithm -

4.2 Text candidate spotting

Working with the BPT representation of images leads to work with a big amount of regions, most of which do not contain text. That is why it is very desirable to reduce the search zone from the whole image just to some image parts, which with high probability contain text.

Based on caption text properties (e.g. texture), it is possible to apply some preliminary process to indicate which parts of the image correspond to the text candidates. As a preliminary texture measurement, the energy of Haar wavelet decomposition of the image is used.

The discrete wavelet transform is a very useful tool for signal analysis and image processing, especially in multi-resolution representation. Signal can be decomposed into different components in the frequency domain. One-dimensional discrete wavelet transform (1-D DWT) decomposes an input sequence into two components (the average component and the detail component) by calculations with a low-pass filter and a high-pass filter. Two-dimensional discrete wavelet transform (2-D DWT) decomposes an input image into four sub-bands, one average component (LL) and three detail components (LH, HL, HH) as shown in Figure 4.2. The decomposition procedure can be applied iteratively until each sub-image contains only one pixel. Generally, the decomposition is applied only on the average component as a component that contains

more information than others. In image processing, the multi-resolution of 2-D DWT

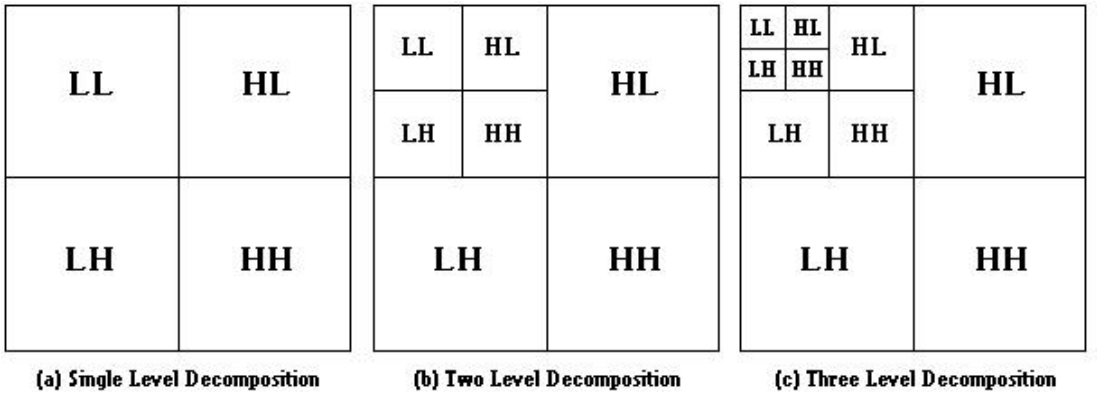


Figure 4.2: Pyramidal decomposition of an Image -

has been employed to detect edges of an original image. The traditional edge detection filters can provide similar results as well, but obtaining these results with traditional edge detection filters is slower than with 2-D DWT.

Haar wavelets were chosen since they are adequate for local detection of line segments, because one of the text features is a great amount of edges.

Taking into account that the text is characterized by significant amount of the vertical and horizontal lines, the LH and HL components of the wavelet transform will be used,these subbands contain information, which corresponds to the vertical and horizontal lines. For the determination of the textured areas of the image is calculated the power of the mentioned subbands of the Haar decomposition over a sliding window of fixed size (H, W), where W is usually greater the H due to the horizontal text alignment (typical values are $H = 6$, $W = 18$).

$$P^l_{HL}(m,n) = \frac{1}{WH}\sum_{i=1}^{W}\sum_{j=1}^{H} HL^2_l(m+i, n+j) \tag{4.1}$$

$$P^l_{LH}(m,n) = \frac{1}{WH}\sum_{i=1}^{W}\sum_{j=1}^{H} LH^2_l(m+i, n+j) \tag{4.2}$$

In this case l corresponds to the decomposition level. To detect characters of different size were analyzed 3 levels. The size of a character is classified according to its height:

- small - height is less than 40 pixels;

- medium - height is in range from 40 to 80 pixels;
- big - height is bigger than 80 pixels.

Using the power of the first level of wavelet decomposition, small-sized characters could be detected, with the power od second level - medium-sized characters, and the power of the third - big-sized characters. Sliding window is moved over the subbands with the overlapping of the half of the window size in both directions.

Both subbands LH and HL are analyzed but with different thresholds, because if the text is present in the region under analysis, it is highly probable that the vertical or horizontal edges of characters are prevailed, then the power value is at least high in one subband and relevant in the another. The pixels under the window are classified as candidate text if next condition is fullfilled:

$$((P^l_{HL} > T_1) \wedge (P^l_{LH} > T_2)) \vee ((P^l_{LH} > T_1) \wedge (P^l_{HL} > T_2)), \quad T_1 > T_2, \quad T_1 = 1200 \;\; T_2 = 400 \tag{4.3}$$

After the pixels classification, the binary mask, where text regions are marked as TRUE, is obtained.

Due to the lack of contrast in the luminance component of some images, this process of pixels classification could produce false negative results, if it is applied only to the grayscale component of the image. In order to include all possible text regions, the wavelet power analysis is applied to all components of the YUV image. In this case the final binary mask is the union of the individual binary masks of each color component. To obtain the individual masks the threshold values T_1 and T_2 were chosen to be different for each component, because the luminance in general contains much more relevant information than the chroma components. In Figure4.3 can be seen example of binary masks, that were obtained through wavelet analysis in grayscale and color components of image is in .

Based on wavelet-power mask and BPT image representation, regions from search partition that possibly contain text are pre-selected. With these regions also are pre-selected all their ancestors, including the whole image (as a root of BPT). At this point the verification, if region contains text or not, should be applied. The example of the wavelet-power mask and corresponded to it, preselected regions could be seen in Figure4.4.

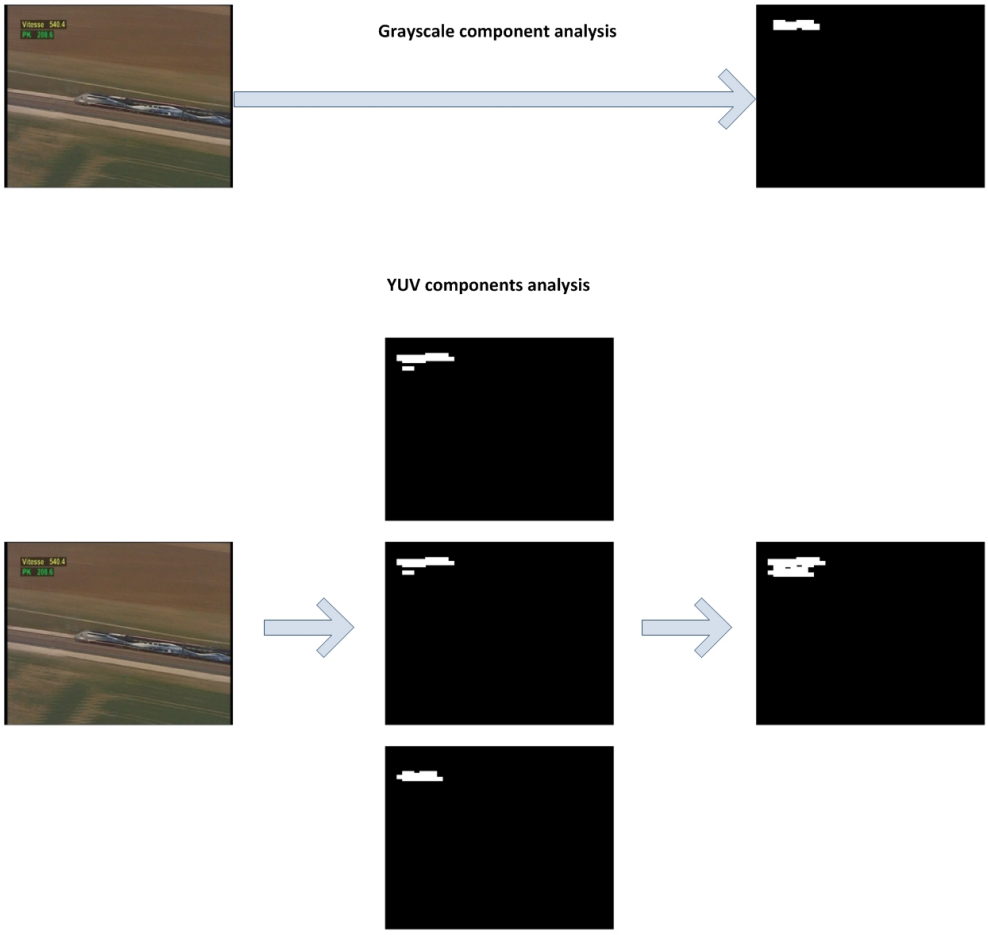


Figure 4.3: Haar Power Masks - Haar Power Mask based on the Haar wavelet decomposition of the grayscale image(upper block) and 3 channels of the YUV image (lower block)

Figure 4.4: Regions selected based on Haar power mask - (a)Original image, (b)Mask based on power of HL and LH subbands of Haar wavelet decomposition, (c)Regions of the search partition which are selected, based on mask from (b)

4.3 Text characteristics verification

Caption text is characterized by its texture, contrast and orientation, which could be translated in texture and geometrical descriptors.

The texture descriptor is recalculated as a sum over the interior pixels of region, see equations 4.1 and 4.2. Only internal pixels are used to exclude the effect of the border on the wavelet coefficients. If the values of the sum is over the empirically found threshold, then the region is not discarded because it is textured enough.

Caption text, in general, has a regular shape, normally similar to rectangle. Because of that characteristic, the BPT regions will be explored based on the next criteria:

- **height**: the height of the region is checked to be in the predetermined range of values: $[H_{min}, H_{max}]$, where H_{min} is empirically defined as 13 pixels, the height of the character that person is able to recognize, and H_{max} is a quarter of the height of the image (it is rare that the caption text occupies more than that value);

- **aspect ratio** of the bounding box of the region:

$$AR = Width/Heigth, \tag{4.4}$$

  aspect ration is also expected to be in the predetermined range of values: $[AR_{min}, AR_{max}]$, where the minimum value is 1.33 and the maximum 20. Aspect ratio reflects the horizontal orientation of the text and thresholds have been selected by the evaluation of the aspect ratio of captions in the test database,see chapter 4.6;

- **area** of the region is defined as a number of pixels of the region. Area is expected to be in the predetermined range of values: $[A_{min}, A_{max}]$, where A_{min} is the area of the node of minimum aspect ratio and with minimum height, which is exactly 225, and A_{max} is limited by one third of the whole image area. This is done to exclude from posterior analysis the image itself and nodes that contain caption text and much more information;

- **compactness**:

$$C = Perimeter^2/Area, \tag{4.5}$$

  is limited by the upper value $C_{max} = 800$ and is used to exclude thin and long regions, which normally do not contain text; The value of this threshold was empirically obtained also after processing test database, see chapter 4.6

- **rectangularity**: is used to measure how similar the region is to a perfect rectangular shape. As it is said in the [34], a rectangle is fitted to the region to obtain a precise value, and the discrepancies between the rectangle and region are measured. The rectangle is fitted using the image ellipse, which could be described as an ellipse with the same first- and second-order moments, as the region. From the semi-major and semi-minor axes α and β of the ellipse rectangles sides a_1 and b_1 are estimated as

$$a_1 = \sqrt{3}\alpha = \sqrt{\frac{6[\mu_{20} + \mu_{02} + \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}]}{\mu_{00}}} \tag{4.6}$$

$$b_1 = \sqrt{3}\beta = \sqrt{\frac{6[\mu_{20} + \mu_{02} - \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}]}{\mu_{00}}} \tag{4.7}$$

  where μ_{pq} are the central moments of the region. The center and orientation of the ellipse (also of the fitted rectangle) are also calculated using moments as shown

$$x_{center\ of\ mass} = \frac{\mu_{10}}{\mu_{00}} \tag{4.8}$$

$$y_{center\ of\ mass} = \frac{\mu_{01}}{\mu_{00}} \tag{4.9}$$

$$\theta = \frac{1}{2} tan^{-1} \frac{2\mu_{11}}{2\mu_{20} - 2\mu_{02}} \tag{4.10}$$

Whereas the bounding rectangle (general methods for rectangularity measurement) circumscribes the region, the image ellipse rectangle will pass through the region, providing a more representative fit. The fitted rectangle is used to clip the region. The following areas are then measured: A_1 - the complete region, A_2 - the clipped region, and A_3 - the rectangle. The discrepancy between the region and the fitted rectangle consists of two parts: the area of the region outside the rectangle A_1 -A_2 , and the area of the rectangle that is not filled by the region A_3 - A_2 (see figure 4.5 as example). Errors are normalized by the size of the rectangle, and subtract from one so as to peak at one, to get

$$R_d = 1 - \frac{A_1 + A_3 - 2A_2}{A_3} \tag{4.11}$$

The restrictions of the rectangularity are the next ones:

- first of all, the horizontally oriented regions should be chosen, so the fitted ellipse should also be horizontally oriented: θ is in the interval $(-0.5; 0.5)$;
- when the node width equals to the image width, then the region is expected to be as much rectangular as possible, so the value of the rectangularity is lower-bounded by $R_d \geq 0.95$;
- when the node width is less than the image width, then it is expected that the region is also less rectangular, than in the previous case, so the threshold is lower : $R_d \geq 0.85$;

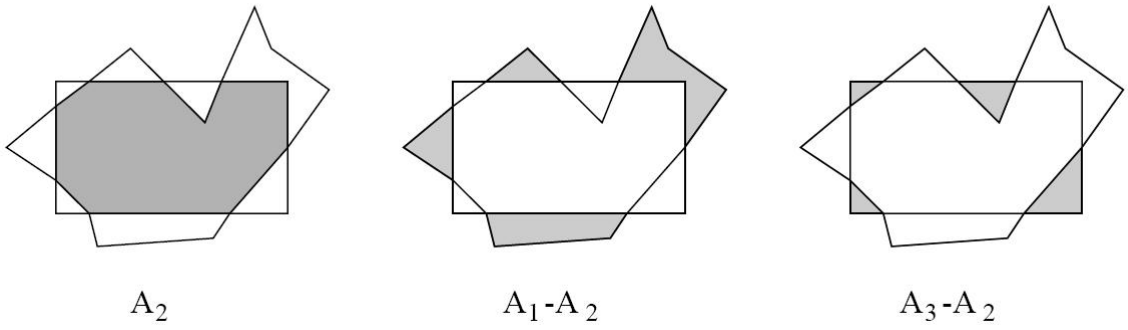


Figure 4.5: Subregions used to measure discrepancy between region and rectangle -

All above described descriptors should be extracted from previously selected text regions. But because of merge-and-split criterium used in the creation of BPT, the background of captions and characters of captions often are represented by different nodes. And BPT regions corresponded with characters are discarded based on geometrical constraints, because of region "noisy" properties: character's shapes, see Figure 4.6. That is why most of the geometrical characteristics (except compactness) are extracted not from the real region masks, but from the simplified version of the region masks. For the simplification morphological operations are used.

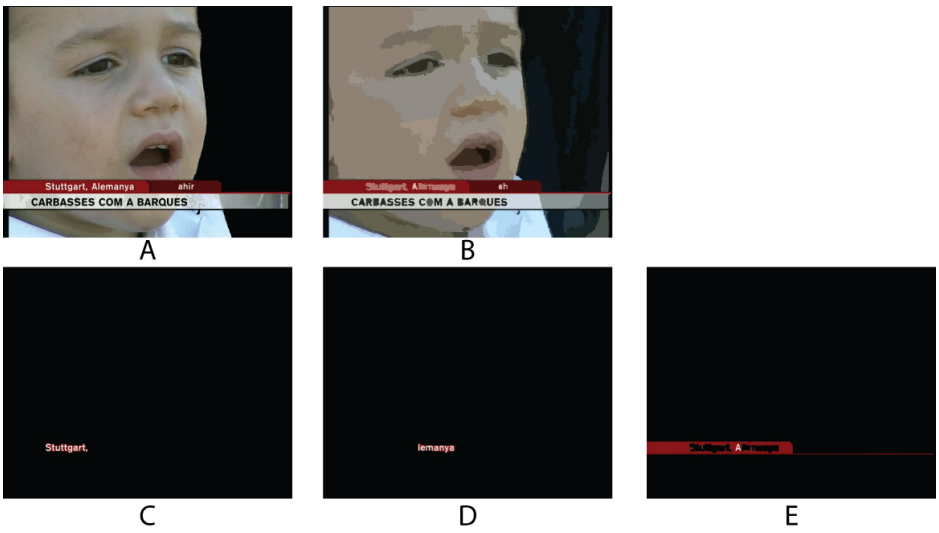


Figure 4.6: Regions where the text information is separated from the background - Original image(a); search partition of the original image(b); regions which contain text(c,d) and background(e) separately

The region mask is modified as follows:

- hole filling: the holes which are presented in the region are filled (example in the Figure 4.7(d));
- opening: in order to eliminate the "noisy" parts of the region the opening with structuring element 9 by 9 pixels is applied, see Figure 4.7(e);
- selection of the biggest component: when the opening divides the region in several parts, the biggest one will represent the region node. As a leaf nodes of BPT is a smallest regions and with opening are divided into several parts, in order to

not discard parts of region contained text, all components of leaf nodes will be analysed, see section 4.4.1.2.

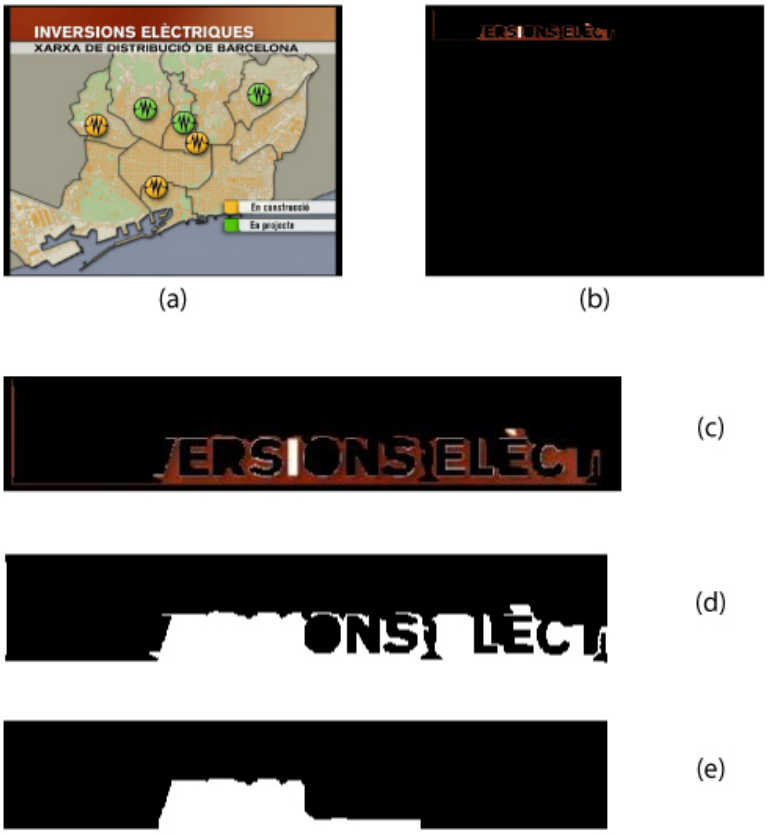


Figure 4.7: The process of region mask modification - (a)Original image, (b)selected region, (c)zoom of the selected region, (d)hole-filled mask of the region, (e)final mask of the selected region

In the subsequent analysis the region will be characterized by two different masks:

- temporary mask: mask that is obtained after hole filling and opening of the original node mask;
- final mask: mask that is obtained after hole filling, opening and selection of the biggest component.

Both masks contain important information that will be used in the consistency analysis of the output. As the final mask is now the representation of BPT region, the

geometrical characteristics, such as height, aspect ratio and rectangularity are calculated based on the final mask. Only the compactness is estimated from the original node mask. Also to improve the speed and efficiency of the algorithm, the geometrical descriptors are calculated sequentially, starting from the cheapest in a computational sense. First of all the compactness descriptor is calculated. All nodes that do not fulfill the compactness condition are discarded from the future analysis. That way, the temporary and final mask of regions are calculated for the smaller number of nodes. Then, in the following step, the descriptors with low computation cost, such as height and aspect ratio, are extracted from the final region mask. At the end of the geometrical features verification the rectangularity descriptor is calculated over the final mask of a relatively small number of regions.

At the end of the text verification process, regions with the highest probability to contain text are selected. Based on this selection of regions caption text objects (CTOs) will be created. The creation and modification of the CTOs are explained in the next section - consistency analysis of the output.

4.4 Consistency analysis of the output

4.4.1 Extraction of caption text objects

In this part of the algorithm text regions which have fulfilled geometrical and texture constrains are modified to represent the caption text objects in the best way. Mainly there are 2 different problems:

- caption text object could appear splitted in the BPT in several nodes, so the goal is to select a tree node which contains the whole caption text object in the best way (with minimum noise) or create union of the nodes which contains the whole CTO ;

- several caption text objects could be represented in one node, due to there's color and texture similarity. The way to process this situation is to search for nodes that will represent each CTO separately, if there are no such nodes, try to separate CTOs during the binarization;

Since CTO is represented through the nodes of BPT there are two type of masks associated with it - temporary and final. A CTO has been described as a set of geometrical features:

- center of masses of the bounding box of the final mask of CTO;
- width of the bounding box of the final mask of CTO;
- height of the bounding box of the final mask of CTO;
- rectangularity of the final mask of CTO;
- list of BPT node ids associated with the current caption text object.

Through these geometrical parameters, each CTO could be fully described, also its position in the image is identified, so different CTOs could be compared to check if they describe the same caption or part of the same caption text.

During the creation of the masks it is possible to loose some part of the text, see Figure4.7. If the search partition of BPT contains significant number of regions (normally, around 300 regions), then the leaf nodes are small and temporary and final masks of the leaf node do not contain all relevant text information. That is why the generic algorithm of the search of the best node to represent CTO is divided into 2 part: the simplified algorithm of the search of the best node, that is applied only to leaf nodes of BPT, and the normal search of the best node algorithm, that is applied to the rest of the nodes, see figure 4.8

First will be explained the generic algorithm of the search of the best node to represent CTO, and later the simplified version.

4.4.1.1 Best node search algorithm

The diagram of the algorithm is represented in Figure 4.9. Each part of the algorithm is explained more detailed below.

All previously selected regions of BPT form one or several subtrees of the BPT. That is why the search of node, which represents the CTO in best way, is done in terms of the exploring these subtrees. Each subtree of the BPT is analyzed separately. It means that CTOs associated with different subtrees are not compared. To extract the best node, subtrees are checked bottom-up. If there is no CTO associated with

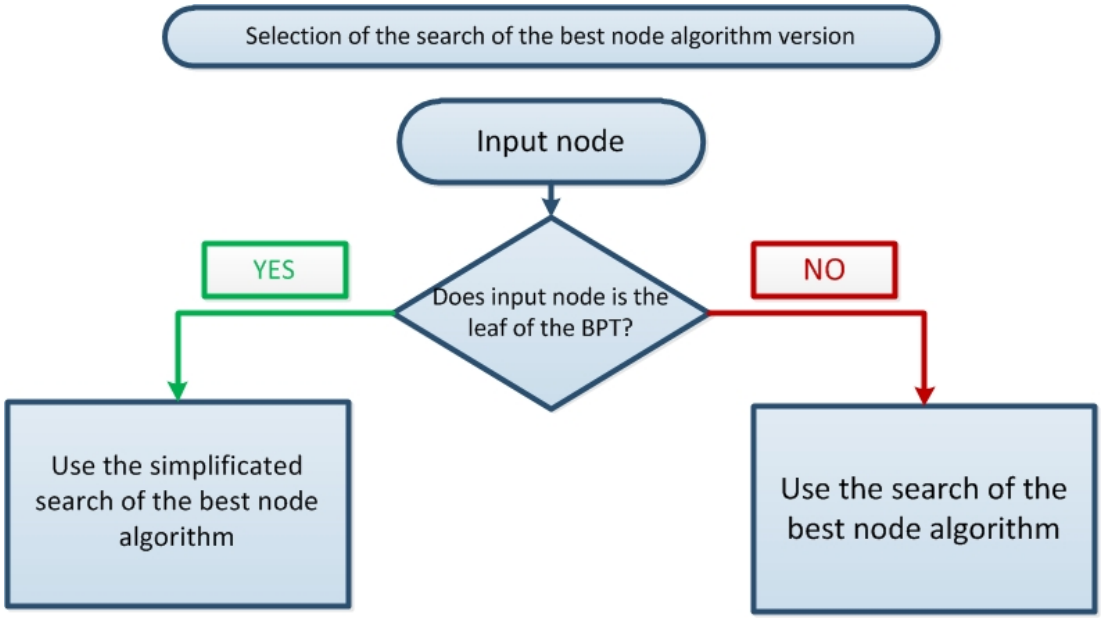


Figure 4.8: Selection of the type of the best node search algorithm - .

the analyzed subtree, then the leaf node of that subtree is used as an anchor point for creation first CTO. In this way at least one CTO is associated with each subtree. In the process of the exploring of the subtree, either CTO values are updated or new CTO is created. Each new candidate node of subtree is checked to be part of the existed CTO. The process of updating/creating CTO is done based on the geometrical descriptors, previously calculated from the final mask on the nodes.

There are three main possible situations:

- the candidate node completes an existing caption text object,
- the candidate node extends an existing caption text object,
- the candidate node does not belong to any existing caption text object, so a new one is created.

Each situation is explained more detail below.

Case 1. Node completes the existed CTO. If at least one CTO has already associated with the subtree, then all following nodes are checked to be part of the existing CTOs. Nodes are checked according to theirs order in subtree - bottom-up.

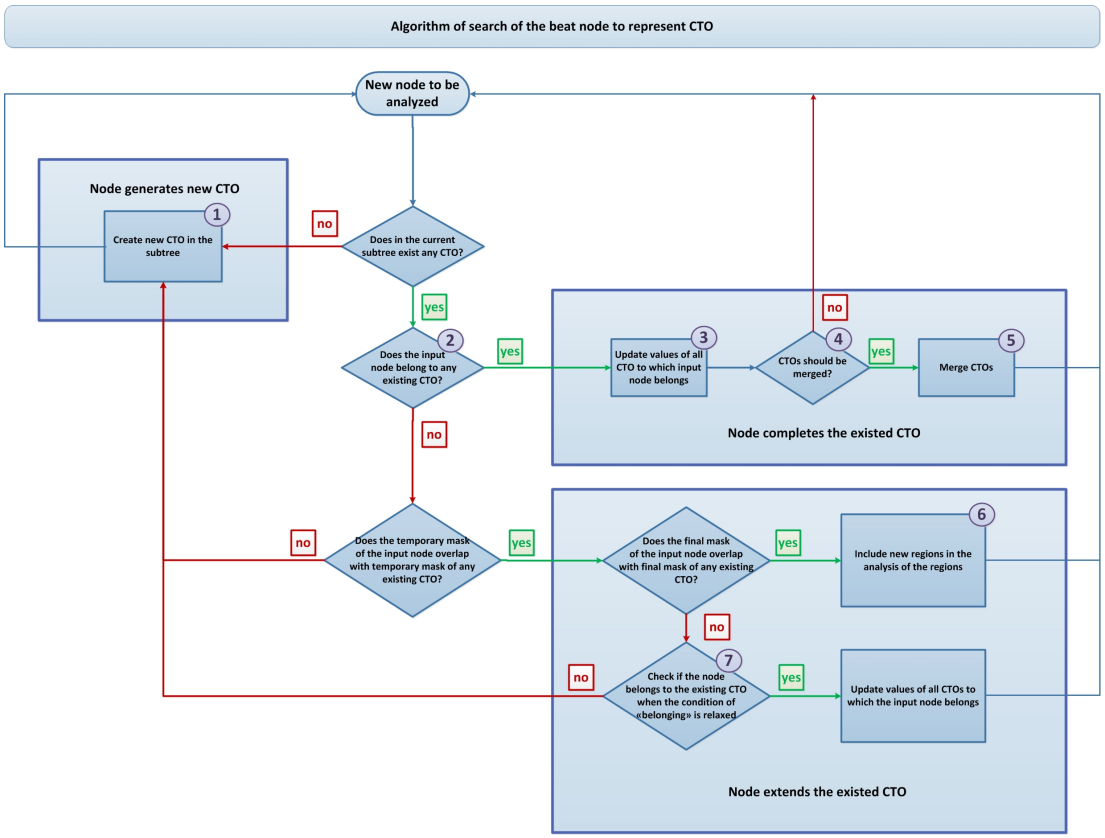


Figure 4.9: Basic scheme for search of the best node containing CTO -

There is a check if the candidate node already belongs to some existing CTO (marked as "2" in Figure 4.9). The candidate node is said to be part of CTO then one of the next conditions is fulfilled:

1. Width and height and also Y- and X-component of the center of masses of bounding box of final masks of the candidate node and CTO are in the predetermined range of values:

$$\begin{aligned} &abs|Heigth_{CTO} - Heigth_{node}| < 0.25 * Height_{CTO} \\ &abs|Width_{CTO} - Width_{node}| < 0.4 * Height_{CTO} \\ &abs|y_{cm\ of\ CTO} - y_{cm\ of\ node}| \leq 0.225 * Height_{CTO} \\ &abs|x_{cm\ of\ CTO} - x_{cm\ of\ node}| < Height_{CTO} \end{aligned} \tag{4.12}$$

   This condition indicates that the candidate node coincides with existed CTO and does not add any relevant information to it, so simply node id is added to the CTO nodelist. Figure 4.10 illustrates this condition.

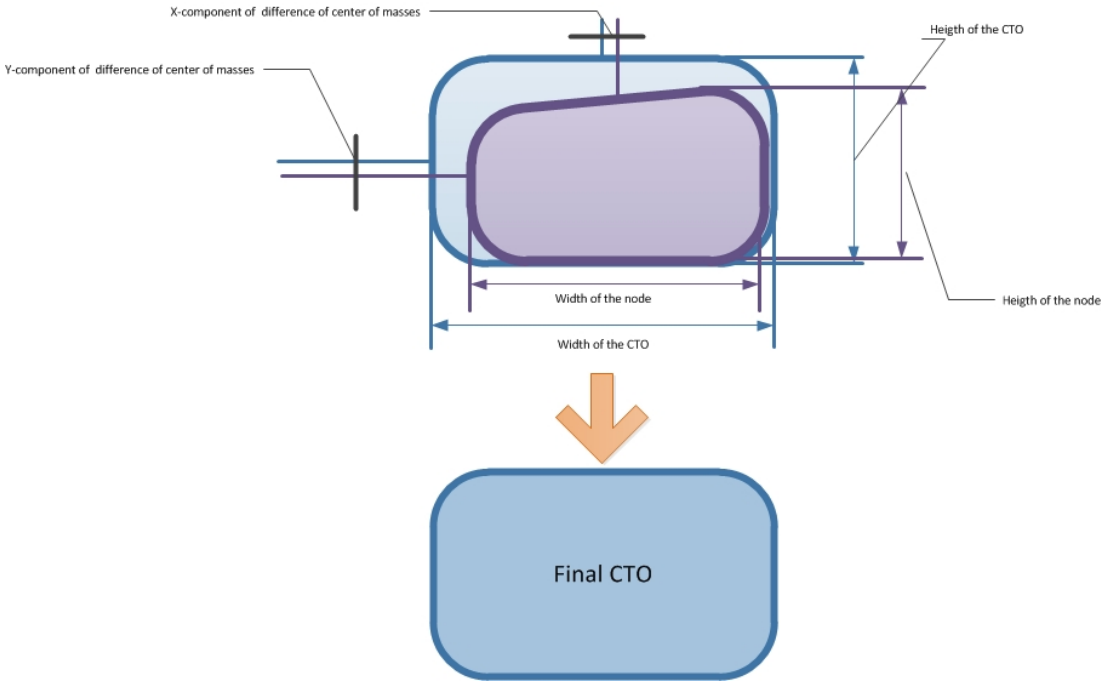


Figure 4.10: Candidate node's geometrical parameters are fully correlated with the CTO's geometrical parameters - .

2. Height and Y-component of the center of masses of BBox of the final mask of candidate node and CTO are in predermined range of values:

$$abs|Heigth_{CTO} - Heigth_{node}| < 0.25 * Height_{CTO}$$
$$abs|y_{cm\ of\ CTO} - y_{cm\ of\ node}| \leq 0.225 * Height_{CTO} \tag{4.13}$$

This condition corresponds to the case, when candidate node and CTO have different width. To complete the CTO, candidate node is added to the CTO nodelist, final and temporary mask of CTO are recalculated, based on the corresponding candidate node mask, and the width and X-component of the CTO's center of masses are updated. Figure 4.11 illustrates this condition.

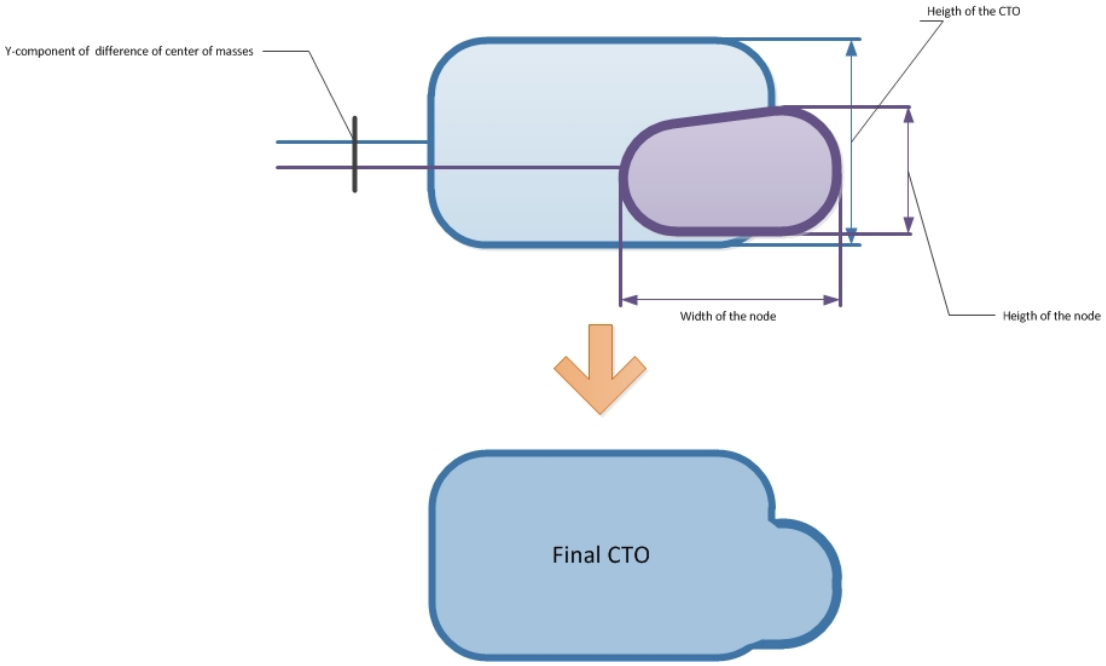


Figure 4.11: CTO width and x-component of center of masses are updated based on candidate node geometrical parameters - .

3. Y-component of the center of masses of BBox of the final mask of the candidate node and CTO is a predetermined range of values, but height of candidate node and CTO vary significatly.

$$abs|y_{cm\ of\ CTO} - y_{cm\ of\ node}| \leq 0.225 * Height_{CTO} \tag{4.14}$$

There are 2 possible situation:

- the CTO height is less then the candidate node height.

$$Heigth_{CTO} < Heigth_{node} \tag{4.15}$$

If so, then the node belongs to the current CTO only if the CTO is slightly more rectangular than the node:

$$R_{CTO} - R_{node} < 0.5 \tag{4.16}$$

If all above conditions(4.14, 4.15, 4.16) are fulfilled then the node is assigned to the current CTO and all CTO's parameters (including masks) are recalculated based on the candidate node characteristics. Figure 4.12 illustrates this condition.

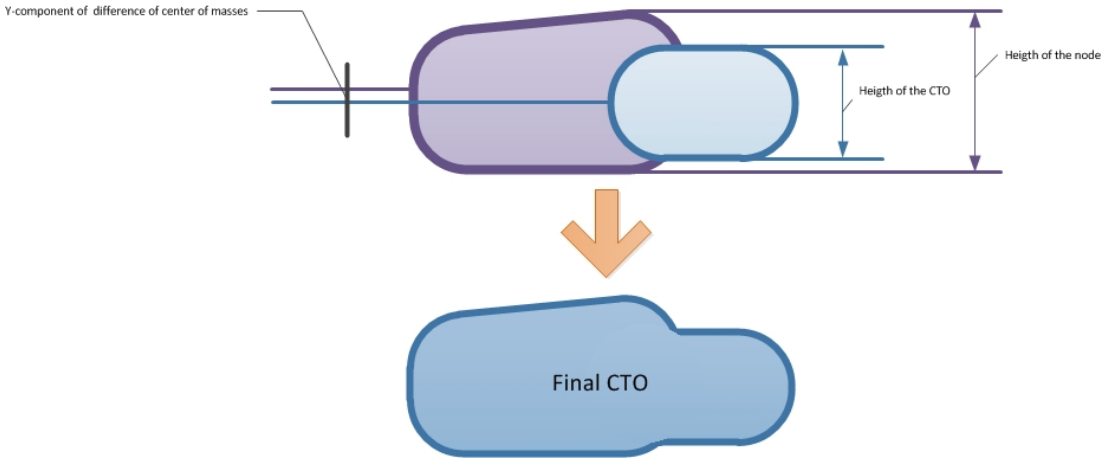


Figure 4.12: y-components of center of masses of CTO and node do not change a lot and CTO is more rectangular than the candidate node, so the CTO parameters are updated based on candidate node geometrical parameters - .

- the second situation is then the CTO's height is bigger than the node's height.

$$Heigth_{CTO} \geq Heigth_{node} \tag{4.17}$$

In this case tha candidate node is assigned to CTO and only CTO's masks are recalculated. Figure 4.13 illustrates this condition.

All above conditions are exclusive and are checked in order of the above sequence. It means that is one condition is fulfilled and candidate node is classified as a node that

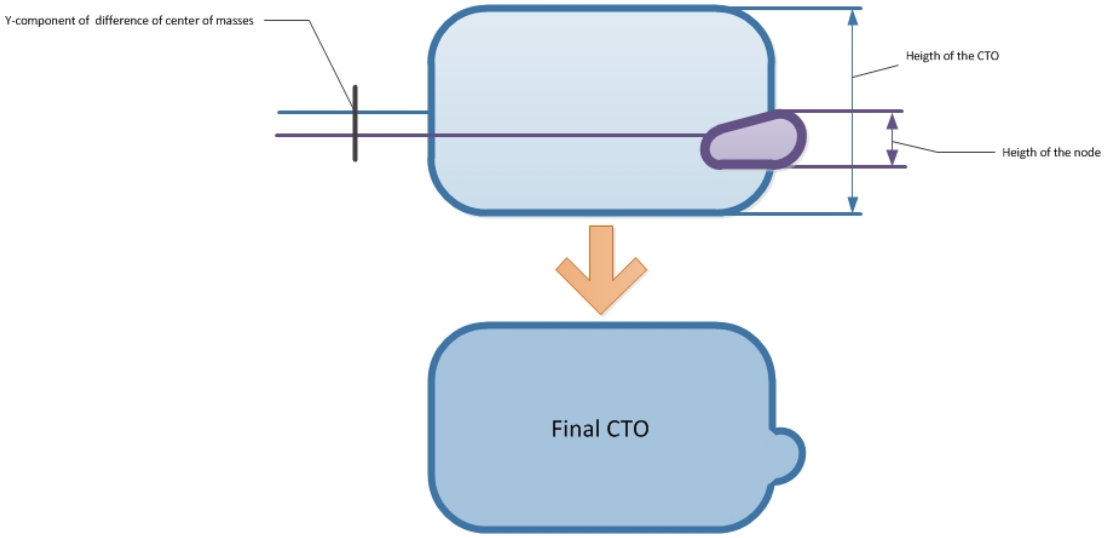


Figure 4.13: y-components of center of masses of CTO and node do not change a lot and CTO's height is bigger than the candidate node's height, so the CTO masks are updated based on candidate node masks - .

completes current CTO, no other conditions are checked (marked as "3" in Figure 4.9).

The verification if the candidate node belongs to some existing CTO is done between the candidate node and all CTOs, associated with the analyzed subtree. If the candidate node belongs to some CTO, this CTO is marked. If the candidate node belongs to more than one CTO, all marked CTOs are checked to be part of a bigger caption text object, and if this is true, all CTOs, that form part of one bigger caption text object, are merged (marked as "4" and "5" in Figure 4.9)

All marked CTOs are analyzed pairwise in sequence by next rules:

First, it's checked if there is overlapping between final masks of CTOs. If there is no overlapping, then CTOs are maintained as a separate ones and are not merged. Second, if there is overlapping between the finals masks, then it is checked the overlapping between temporary masks of CTOs. If there is overlapping in temporary and final masks, it means that CTOs form the part of the same, but bigger, caption text object,and should be merged.

If the candidate node does not complete any previously created CTO, then there is a check: does the candidate node extend any already created CTO?

Case 2. Node extends the existed CTO. In case when the candidate node extends some of already created CTO, the same geometrical constraints, as for completeness case, are checked -width, height, center of masses correspondence and mask overlapping- , but with more relaxed thresholds.

The overlap between temporary masks of candidate node and CTO are checked. If there is no overlap - the candidate node neither extends the CTO, nor completes it, so the new CTO are created from the candidate node (marked as "1" in Figure 4.9).

If there is an overlap in the temporary masks, then it is highly probable that the relevant information which is contained in the temporal mask of the candidate node will extend the CTO. To be sure that relevant information is in the parts of the temporary mask that are not included in the final mask, final masks of candidate node and CTO are checked for overlap. If there is no overlap between final masks, but there is one between temporary mask, it is supposed that the information that will extend CTO is in the temporal mask of the candidate node, to be sure that the candidate node are aligned with the CTO next condition is checked:

$$abs|y_{cm\ of\ CTO} - y_{cm\ of\ node}| < 1,5 * min(Height_{CTO}, Height_{node}) \tag{4.18}$$

If the above condition is fulfilled, the candidate node extends the CTO, so the temporal and final masks of CTO are recalculated based on the candidate node masks, and parameters of CTO (width, height, center od masses and rectangularity) are recalculated based on the new CTO's masks.

If the condition 4.18 does not fulfill the candidate node contains "noise" and does not extend the CTO. So the node is marked as processed and next candidate node analysis starts.

The most interesting case is when both temporal and finals masks of CTO and candidate node are overlapped, but the node does not fulfill conditions 4.4.1.1 to be part of that CTO. In this case (marked as "6" in Figure 4.9) it is assumed that the candidate node is an ancestor node in the tree and contains not only caption text objects, but also some background information. In this case the superior bounding box technically is CTO, that, generally, contains several smaller CTOs (see figure 4.14), but for the binarization we are interesting in extraction of the internal- small- CTOs.

To extract only relevant information from the candidate node, a union mask is created, based on the union of bounding box of final masks of all CTOs that are

Figure 4.14: Images of sport events - Captions, associated with scoreboards, are difficult cases because the scoreboards could be represented not as a unique CTO, but as a set of smaller CTOs

overlapped with candidate node in temporal and finals masks. After that, XOR mask between the union mask and temporary mask of the candidate node is created (Figure 4.15 (b)). This XOR mask includes parts of temporary mask of candidate node, which are not included in any of CTOs. To eliminate "noise" from XOR mask, such as long thin lines that is came form the temporary mask of the candidate node, the morphological opening with structured element 9x9 pixels are applied. In the further analysis every time then the XOR mask is recalculated the morphological opening is done with the same structuring element.

Because the XOR mask contains information which is corresponded to more than one CTO, it is logic to process this information in parts, corresponding to each CTO. That is why, the XOR mask is vertically divided in several segments proportionally to the CTO's heights and all segments are analyzed separately (Figure 4.15 (c)). Segment is analyzed in horizontal direction (left and right parts), then in vertical (bottom and top parts) iteratively. After each iteration from the xor mask is excluded already analyzed segment.

The analysis of the segment is done in terms of texture and color similarity. As texture measurement, the same texture descriptor as before is used (see equation 4.3),

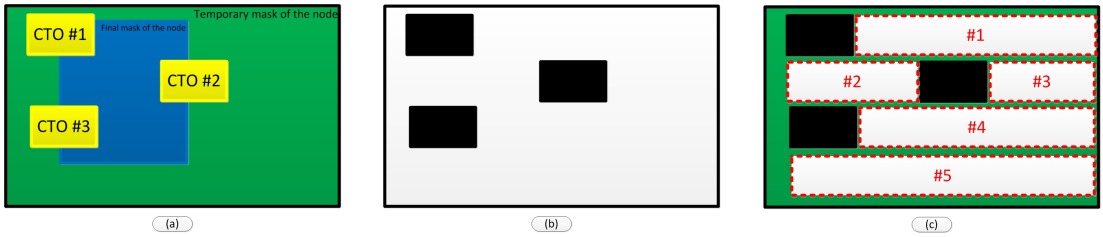


Figure 4.15: Scheme of BPT node analysis based on the color and texture information - (a) The distribution of the bounding boxes of the CTOs and final and temporary masks of the node; (b) XOR mask; (c)The order(red dashed lines) in which the segments are processed, green color corresponds to the small segments which will be deleted by the morphological opening with square structuring element 9 by 9 pixels

but with lower thresholds, 1000 instead of 1200 for the T_1 and 300 intead of 400 for T_2. The values of threshold are lower, because not the whole BPT region is analyzed, but just part of it. If the segment is textured enough, the color descriptors of the segment and the CTO are calculated. This is done to avoid the evaluation of areas corresponded to background of caption.

To measure color similarity between segment and CTO two different color descriptors are calculated. The first one is the ColorMean descriptor, which is calculated over the RGB components of the CTO and the segment. To compare how similar are ColorMeans, the Euclidean distance between them is calculated. MeanColor is simplest and cheapest, in the computational sense, descriptor that could be used to measure the color similarity between two images, but it is not enough accurate because it doesn't take into account the color distribution, so the second descriptor - DominantColor- is used. The DominantColor is calculated over the YUV components of the image, it is a compact color descriptor, which stores the color values c_i, their percentages p_i and variance v_i, along with the mean spatial coherency s.

$$DC = \{(c_i; p_i; v_i); s\}; (i = 1; 2; ...; N) \tag{4.19}$$

Since it stores only the color values instead of a color histogram, the storage requirement is very effective, and in case of analysis of image regions which are close to each other it gives enough information about color distribution. For comparison of the similarity between the DominantColor of the segment and CTO the dissimilarity measure is used, which is sensitive not only to the Euclidean distance between colors, but also take into

account the percentage of the presented colors. Given two descriptors $DC1$ and $DC2$, each pair of colors c_{1i} and c_{2j} are considered as similar if their Euclidean distance $d_{1i;2j}$ is below a threshold T_d. A similarity coefficient $a_{1i;2j}$ is then defined as:

$$a_{1i;2j} = \begin{cases} 1 - \frac{d_{i1;2j}}{T_d} & \text{if} \quad d_{1i;2j} \leq T_d \\ 0 & \text{if} \quad d_{1i;2j} > T_d \end{cases} \tag{4.20}$$

Given these coefficients, the dissimilarity measure is:

$$D^2(DC1; DC2) = \sum_{i=1}^{N_1} p_{1i}^2 + \sum_{j=1}^{N_2} p_{2j}^2 - \sum_{i=1}^{N1} \sum_{j=1}^{N_2} 2a_{1i;2j} p_{1i} p_{2j} \tag{4.21}$$

Using the Euclidean distance between the MeanColor values and dissimilarity measure between DominantColor values, the decision, if the segment is a part of the CTO or not, could be done. The segment is a part of the CTO if at least one of the next conditions is fulfilled:

- Euclidean distance between MeanColor values is below 50;
- Euclidean distance between MeanColor values is below 80 and dissimilarity measure between DominantColor values is below 710;
- dissimilarity measure between DominantColor values is below 350;

In case when at least one of the conditions is fulfilled the analyzed segment is included to the CTO masks and CTO parameters are recalculated. In case when conditions are not fulfilled, the segment is marked as checked. In both cases that segment is excluded from the XOR mask, morphological opening with the structuring element 9x9 is done over recalculated XOR mask, and the analysis of the next segment (if there is at least one) starts till all segments in XOR mask are checked.

Then all segments of XOR mask are checked, the candidate node is marked as checked, and also could be marked as node that extends CTOs, if some of segments of XOR mask are included in some CTOs.

Case 3. New CTO is created based on the candidate node. The last case that was implicitly explained before is when and how is created new CTOs.

First of all, due to the fact that subtrees of BPT are analyzed separately, ie candidate nodes are compared only with the CTOs associated with the current subtree, the new

CTO is created when the new subtree is over analysis and there is no CTOs associated with that subtree. The new CTO is created based on the leaf node of the working subtree.

In case when subtree already has several CTOs associated with it, then candidate node is checked to complete and then extend CTOs. If the candidate node neither completes, not extends any of existed CTOs, then the new CTO is created based on the candidate node.

The output of the best node search algorithm is the list of CTOs, that are associated with different subtrees of the BPT.

4.4.1.2 Simplified best node search algorithm

The difference between the general search of the best node algorithm and its simplified version is that the simplified version is applied only to the leaf nodes of BPT. Leaf nodes of the BPT correspond to regions in the search partition of BPT, and, generally, regions in search partition correspond to the single character or to the set of several characters (from 2 to 5) in case the BPT contains relatively big number of regions - 300 and more. When the final mask of the leaf node is created, hole-filling, opening and selection of the biggest connected component are applied (see section 4.3). Opening, with structural element 9x9 pixels, can divide the region masks in several parts, and only the biggest connected component is selected in order to represent the node. But in case the region conatains 1 or several characters, the biggest component could represent only one character or a part of it. According to the best node search algorithm, the mask of the biggest connected component will be used as anchor point to CTO creation, and it could happen that the information about all other characters will be lost. To avoid this problem the simplified best node search algorithm was introduced. It is called "simplified" because:

- only leaf nodes are processed;
- for every leaf node all connected components of the mask after opening are processed separately;

and as a consequence, there are possible only two cases: case there is no CTO in the the current subtree, so the first CTO associated with subtree are created, or case the candidate node completes the existing CTO.

As in the main best node search algorithm, all masks, corresponded to each candidate leaf node as a result of the opening, are checked to fulfill next conditions:

- height: the height of the analyzed mask is limited by the range of values: $[H_{min}, H_{max}]$, where H_{min} is 13 pixels, the and H_{max} is a quarter of the height of the image;
- aspect ratio of the bounding box of the analyzed mask is limited by the range of values: $[AR_{min}, AR_{max}]$, where the minimum value is 1.22 and the maximum 22;
- area of the analyze mask limited by s: $[A_{min}, A_{max}]$, where A_{min} is 300 and A_{max} is one third of the whole image area;
- rectangularity of the analyzed mask is lower-bounded by the value of 0.85.

The scheme of the simplified search of the best node algorithm could be seen at Figure 4.16.

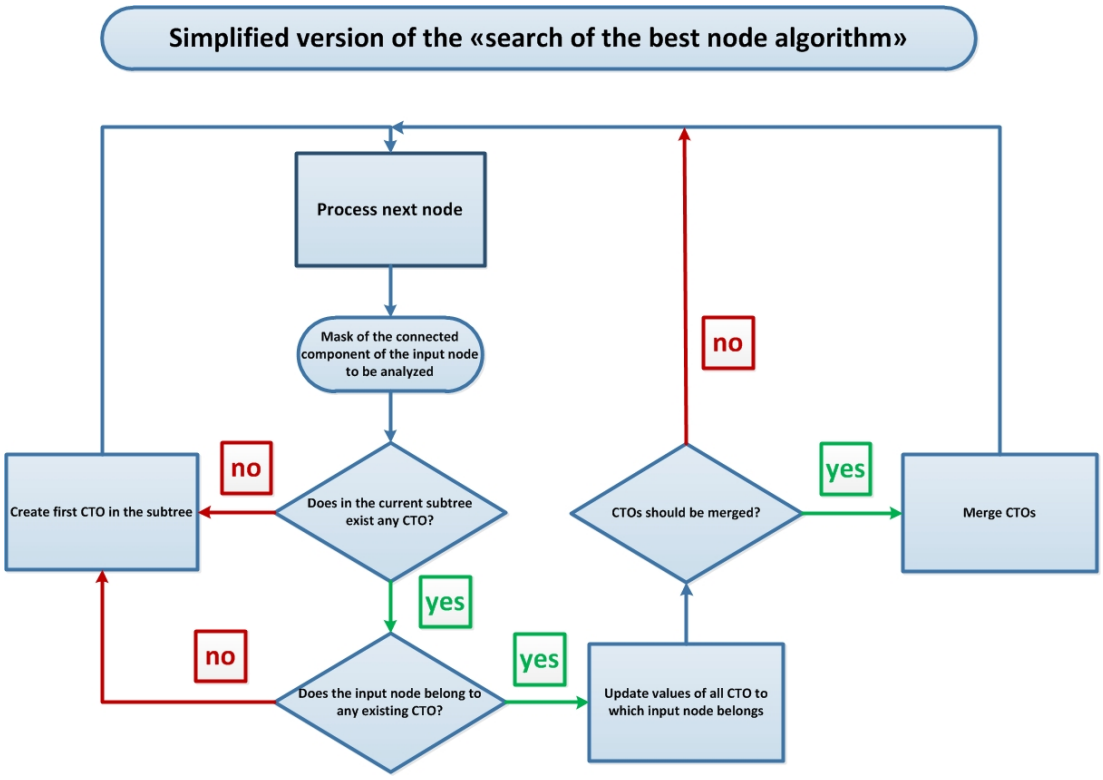


Figure 4.16: Simplified best node search algorithm - .

As it could be seen, the main difference is that each leaf node candidate and all associated with this node masks are checked to complete an already existed CTO (see

section 4.4.1.1), and if these conditions are not fulfilled, new CTO is created based on the current mask.

At the end of the processing of all subtrees of BPT the list of CTO objects is created. Before defining these objects as a final ones, a simple post-process is applied:

- as a final representation of a CTO, the final mask is used, but it is extended according to the width of temporary mask of CTO;
- all CTOs from all subtrees are checked for overlap, and if some CTOs overlap more than at 50% of their areas, they are merged;

After this post-processing the final list of the caption text objects detected in the image are defined. Within these objects there are some false positive, because these objects meet all geometrical, textural and color constrains we defined above and do not contain caption text. The way to identify and exclude these objects is explained in the next step of the algorithm: the binarization.

4.4.2 Binarization of caption text objects

For each caption text object the binarization is performed over the grayscale component of the CTO. The binarization consists in the analysis of the several horizontal lines of pixels (normally 7). For each line mean and variance are calculated.

Lines with low variance are assumed to contain background pixels and can be used for its characterization. For the estimation of the background mean value, median value of each line with low variance are saved and then mean of medians are calculated. If all N analyzed line segments are classified as high variance lines, the median values of short - 10 % of the CTO width - line segments of the left top and bottom corners of the CTO are estimated. The probability of found character in short top or bottom corner segment of CTO is very low, that is why these segments could be used to calculate the background mean value.

Lines with high variance are supposed to contain both text and background and are used for threshold calculation. During the analysis of the high variance lines, the pixel values are saved before and after the big value changes (discontinues in color). Based on the saved values, the threshold are calculated. In case that no line is classified as a

high variance line, the threshold is a mean of the maximum pixel value and minimum pixel value of the CTO.

To improve the binarization, the threshold and background mean value are discriminated according to the next rule:

$$\begin{aligned} &if(BG < T) \\ &\qquad T = (maxvalue + T) * 0.4 \\ &else \\ &\qquad T = (minvalue + T) * 0.6 \end{aligned} \tag{4.22}$$

where BG is the estimated background mean value, T is a threshold value.

Figure 4.17 and figure 4.18 are examples of binarization process.

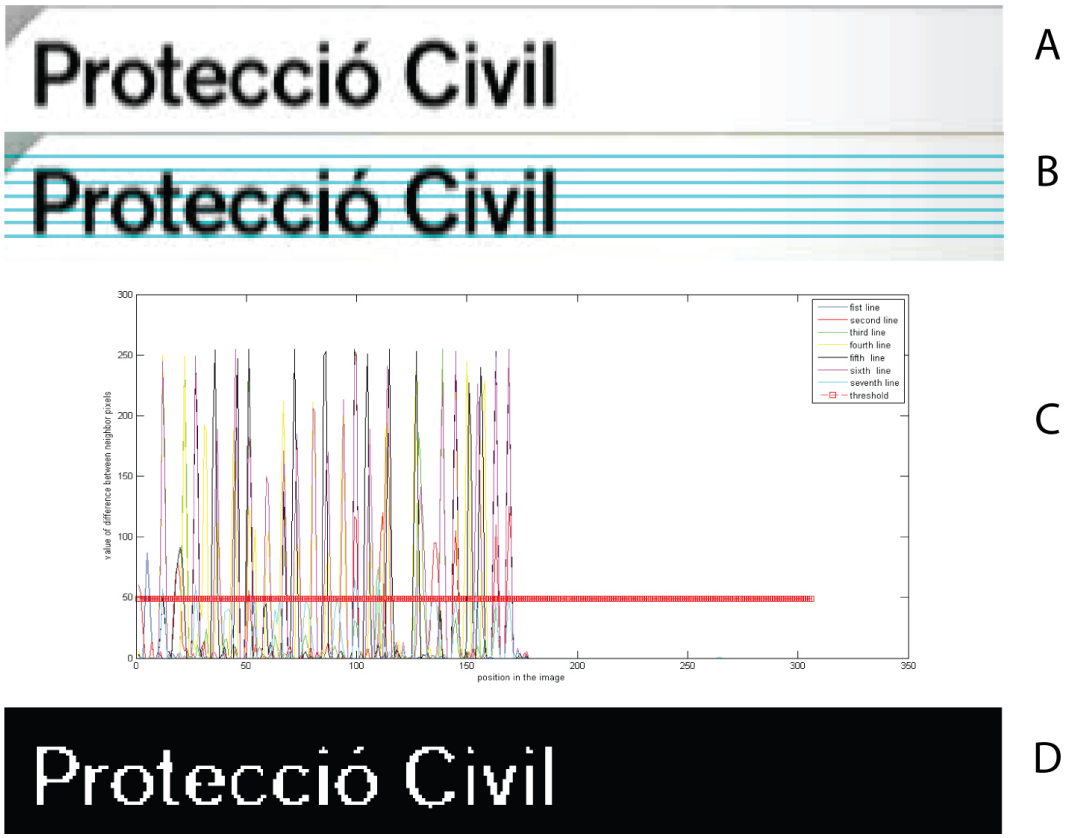


Figure 4.17: The binarization process: (a) original RGB image; (b) grayscale image; (c) threshold selection; (d) the output of the binarization -

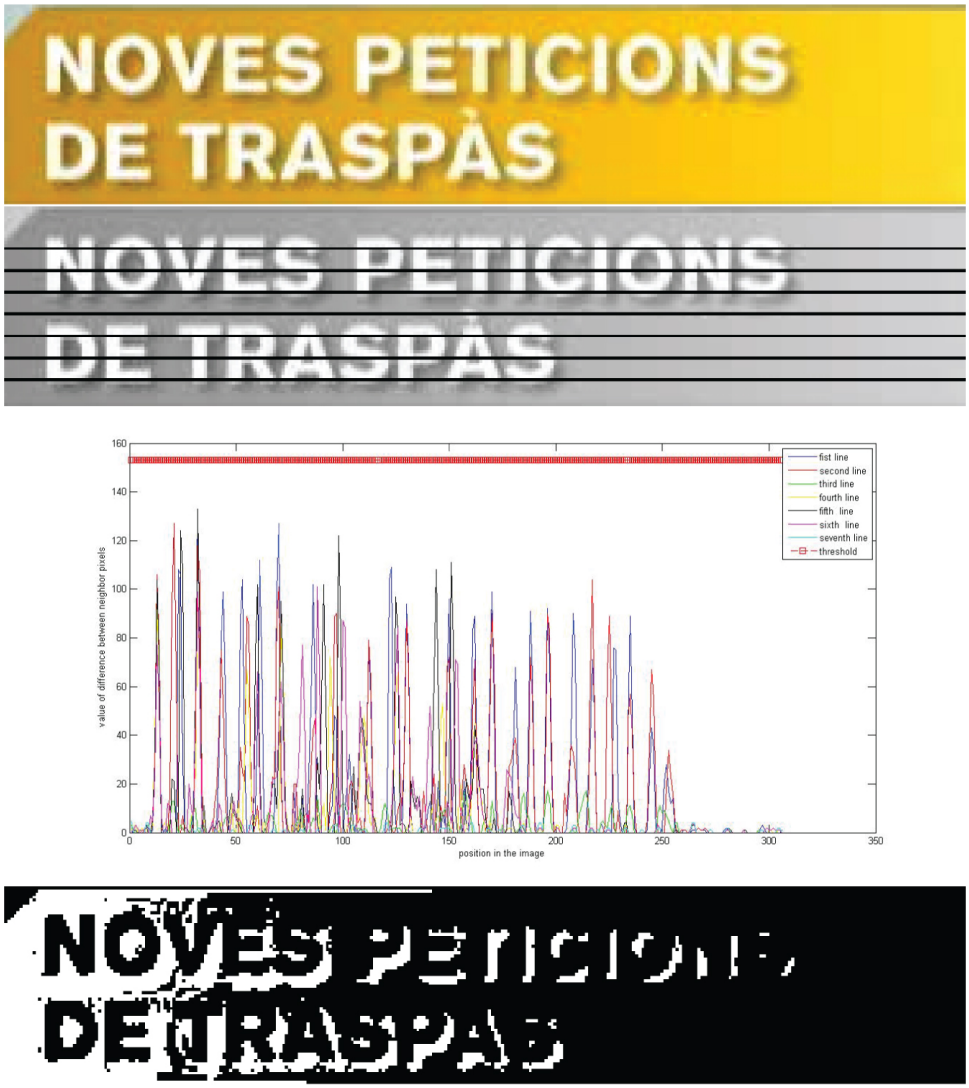


Figure 4.18: The binarization process: (a) original RGB image; (b) grayscale image; (c) threshold selection; (d) the output of the binarization - .

After the binarization of the CTO, some "cleaning" operation is done in order to eliminate the "noise". The cleaning consist of three steps:

- performing the volume opening for deleting all segments with area less then 15 pixels;
- deleting the connected components in the contour of the mask;
- deleting CTO if after the output of the binarization is a white(black) image;

The output of the binarization can be directly used in an OCR system, but it still contains false positive objects. To eliminate these objects a post-binarization analysis of the texture is performed.

4.4.3 Post-binarization analysis

To separate the objects that contain caption text from ones that do not, a SVM classifier is used. As input data for the SVM are used the values of the Homogeneous texture descriptor(HTD), calculated over the luminance component of the possible CTO and calculated over binarized version of CTO. The HTD is used because it is a compact and efficient representation of the image texture. Below there is the small overview of both concepts: homogeneous texture descriptor and support vector machine.

4.4.3.1 Homogeneous texture descriptor

An image can be considered as a mosaic of homogeneous textures so that these texture features associated with the regions can be used to index the image data. For instance, in case of text the space between the characters and strokes of characters is a clear example of a homogeneous textured pattern. The Homogeneous Texture descriptor provides a precise quantitative description of a texture, using 62 numbers (quantified to 8 bits each). The computation of this descriptor is based on filtering using scale and orientation selective kernels, see figure 4.19. The extraction is done as follows; the image is first filtered with a bank of orientation and scale tuned filters (modeled using Gabor functions) using Gabor filters. The first and the second moments of the energy in the frequency domain in the corresponding sub-bands are then used as the components of the texture descriptor. The number of filters used is 5x6 = 30 where 5 is the number of "scales" and 6 is the number of "directions" used in the multi-resolution

decomposition using Gabor functions. An efficient implementation using projections and 1-D filtering operations exists for feature extraction.

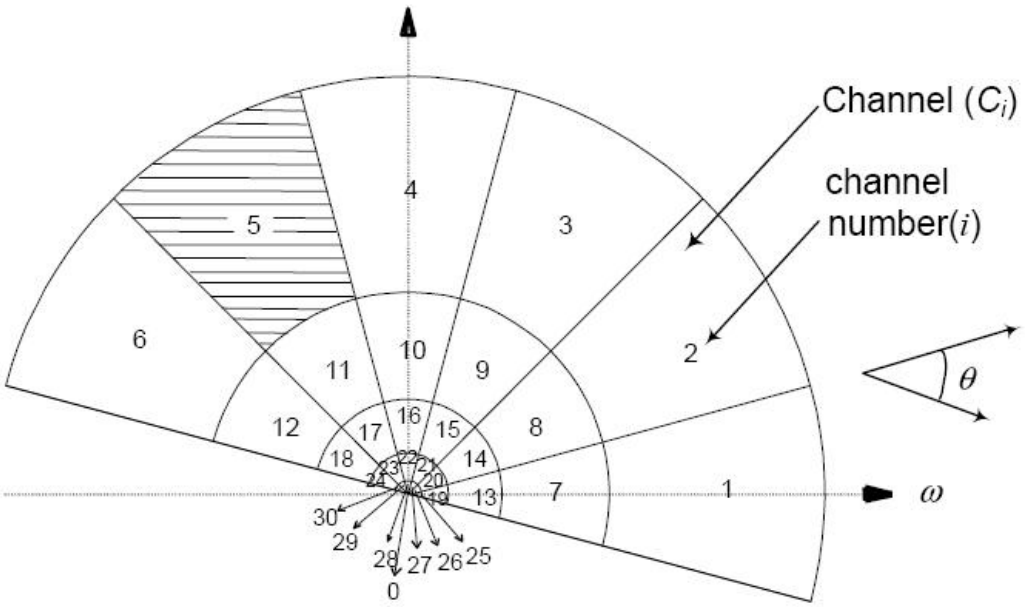


Figure 4.19: Example of Frequency region division with HVS filter extracted from [35] - F

requency region division with HVS filter.

4.4.3.2 Support Vector Machines

Support vector machines are a set of related supervised learning methods that analyze data and recognize patterns, used for classification and regression analysis. The standard SVM takes a set of input data, and predicts, for each given input, which of two possible classes the input is a member of, which makes the SVM a non-probabilistic binary linear classifier. Since an SVM is a classifier, then given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that predicts whether a new example falls into one category or the other. Intuitively, an SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible.

As it said in [36], the standard SVM algorithm aims to find an optimal hyperplane $\mathbf{w} \cdot \mathbf{x} + b = 0$ and use this hyperplane to separate the positive and negative data. The classifier can be written as:

$$f(\mathbf{x}) = \begin{cases} +1 & \text{if} \quad \mathbf{w} \cdot \mathbf{x} + b \geq 0 \\ -1 & \text{if} \quad \mathbf{w} \cdot \mathbf{x} + b < 0 \end{cases} \tag{4.23}$$

The separating hyperplane is determined by two parameters $\mathbf{w}$ and b . The objective of the SVM training algorithm is to find $\mathbf{w}$ and b from the information in the training data. Standard SVM algorithm finds $\mathbf{w}$ and b by solving the following optimization problem.

$$min \frac{1}{2} \parallel \mathbf{w} \parallel^2 + C \sum_i \xi_i \tag{4.24}$$

$$s.t.\ \forall i,\ y_i(\mathbf{w} \cdot \mathbf{x}_i + b) + \xi_i \geq 1 \tag{4.25}$$

$$\xi_i \geq 0 \tag{4.26}$$

The first term $\parallel \mathbf{w} \parallel^2$ controls the margin between the positive and negative data. ξ_i represents the training error of the i^{th} training example. Minimizing the objective function of 4.24 means minimizing the training errors and maximizing the margin simultaneously. C is a parameter that controls the tradeoff between the training errors and the margin. The intuition of standard SVM is shown in Figure 4.20, where $\mathbf{w} \cdot \mathbf{x}_i + b = 1$ and $\mathbf{w} \cdot \mathbf{x}_i + b = -1$ are two bounding planes. The distance between the two bounding planes is the margin.

4.4.3.3 CTO probability estimation

The classification task involves separating data into training and testing sets, a training database of CTOs with and without text was created and all samples were labeled. The input data are, on one hand, the HTD over the luminance component of the original CTO and, on the other hand, the HTD over the binarized CTO, which are calculated for each sample. So, two different classifiers are trained, obtaining two separate classification for each CTO. As a result two different decision values are obtained as well as its confidence values. The confidence margins helps to measure how far away the sample is, respect to the limits. The separating value for the classification is zero, a confidence value is calculated as a function of the distance, that is why the probability of text in CTO could be estimated in terms of confidence values. The calculation of the probability is done according to the next rules:

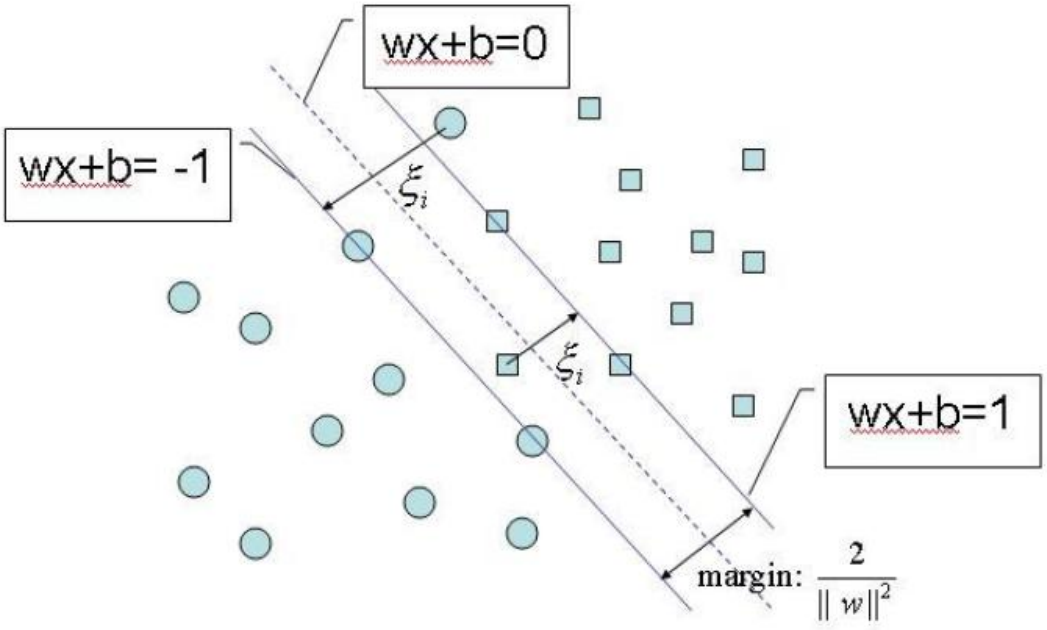


Figure 4.20: Example of SVM separating hyperplane extracted from [36] -

- if the confidence value calculated for the original CTO (CV_o) and confidence value calculated for the binarized CTO (CV_b) both have a negative value, then there is no probability that CTO contains text.

  if $CV_o \in (-\infty,\ 0)$ **and** $CV_b \in (-\infty,\ )$ **then**
  $prob_{CTO} = 0$
  end if

- if sign of confidence values of two classifiers have different sign, then there is some uncertainty if the CTO contains text or not, considering possible error in the classification of the binarized CTO or possible error in the binarization of CTO, the probability of CTO is calculated as a maximum between the re-scaled confidence value of original CTO and bianrized CTO.

  if $CV_o \in (0,\ \infty)$ **and** $CV_b \in (-\infty,\ 0)$ **then**
  $prob_{CTO} = -\frac{1}{4}CV_b * (\frac{1}{4}CV_o + \frac{1}{2})$
  else
  if $CV_b \in (0,\ \infty)$ **and** $CV_o \in (-\infty,\ 0)$ **then**

	Binariztion	
Number of detected CTOs before binarization	512	
	numeric	%
Correctly binarized CTOs	421	82,20%
Partially correctly binarized CTOs	64	12,50%
Incorrectly binarized CTOs	27	5,30%
Number of detected CTOs after binarization	489	95,50%
Number of discarted incorrectly binarized CTOs	23	4,49%

Table 4.1: Results of binarization using CTO detection algorithm

$prob_{CTO} = -\frac{1}{4}CV_o * (\frac{1}{4}CV_b + \frac{1}{2})$

end if

end if

- if both confidence values are positive, then the probability of CTO is calculated as a maximum between the re-scaled confidence values of original CTO and bianrized CTO.

if $CV_o \in [2,\ +\infty)$ **and** $CV_b \in [2,\ +\infty)$ **then**

$prob_{CTO} = max(\frac{1}{4}CV_o + \frac{1}{2}, \frac{1}{4}CV_b + \frac{1}{2})$

end if

Based on the above rules of the probability calculation, all CTOs with non-zero probability are passed to OCR, the others are discarded.

4.5 Results of the algorithm

Number of detected CTOs are given before and after the binarization, all CTOs detected before binarization are classified into three categories: correctly binarized, partially binarized and incorrectly binarized. The CTO is considered to be partially binarized if :

- in case when the background color of CTO is degrading along one of the CTOs dimension, the threshold for binarization is calculated inaccurate and not the whole text is binarized correctly;

- due to the fact that the CTO mask contains false positive results, for example, the CTO mask is bigger then the real caption text, the threshold for the binarization often is calculated incorrectly and not the whole text is properly binarized;
- due to the deleting the connected components in the contour of the CTO mask after binarization some characters, which one that touch the contour mask, were deleted, so the words do not contain some letters and as a consequence could not be recognized b OCR;

As it was observed, all partially correctly binarized CTOs are passed to OCR, while some of incorrectly binarized CTOs are discarded because they have zero probability of contain text according to the CTO probability estimation process.

4.6 Database

Test database consists of images divided into four categories, each category contains images of the same type: images form news, images from talkshows, images from sport videos and one category contains highly textured images with no captions.

The total amount of images is 203, 40 correspond to talkshows, 59 are from sport videos, 54 are images from news and 50 do not contain captions. That amount of images helps to cover diversity of the captions, starting from captions that occupy all the image width to scoreboards that occupy relatively small part od the image, typically in corners. Captions with different backgrounds - from one-color solid to transparent or highly textured - could be found in the database. There are different forms of captions: mainly it is rectangular, but especially in case of talkshows the form could vary from rectangular to elliptical. The example of images from database could be find in figure 4.21.

All images are annotated in therm of bounding boxes. Two types of annotation are corresponded to each caption: first the whole caption (including the whole background) is annotated in term of bounding box, second only text of caption is annotated in terms of bounding box of each word. This double annotation is used for measure the efficiency of the caption text extraction algorithm. The annotation of the whole caption is used in measure the accuracy of the caption object extraction. The word annotation is used for measure the binarization efficiency.

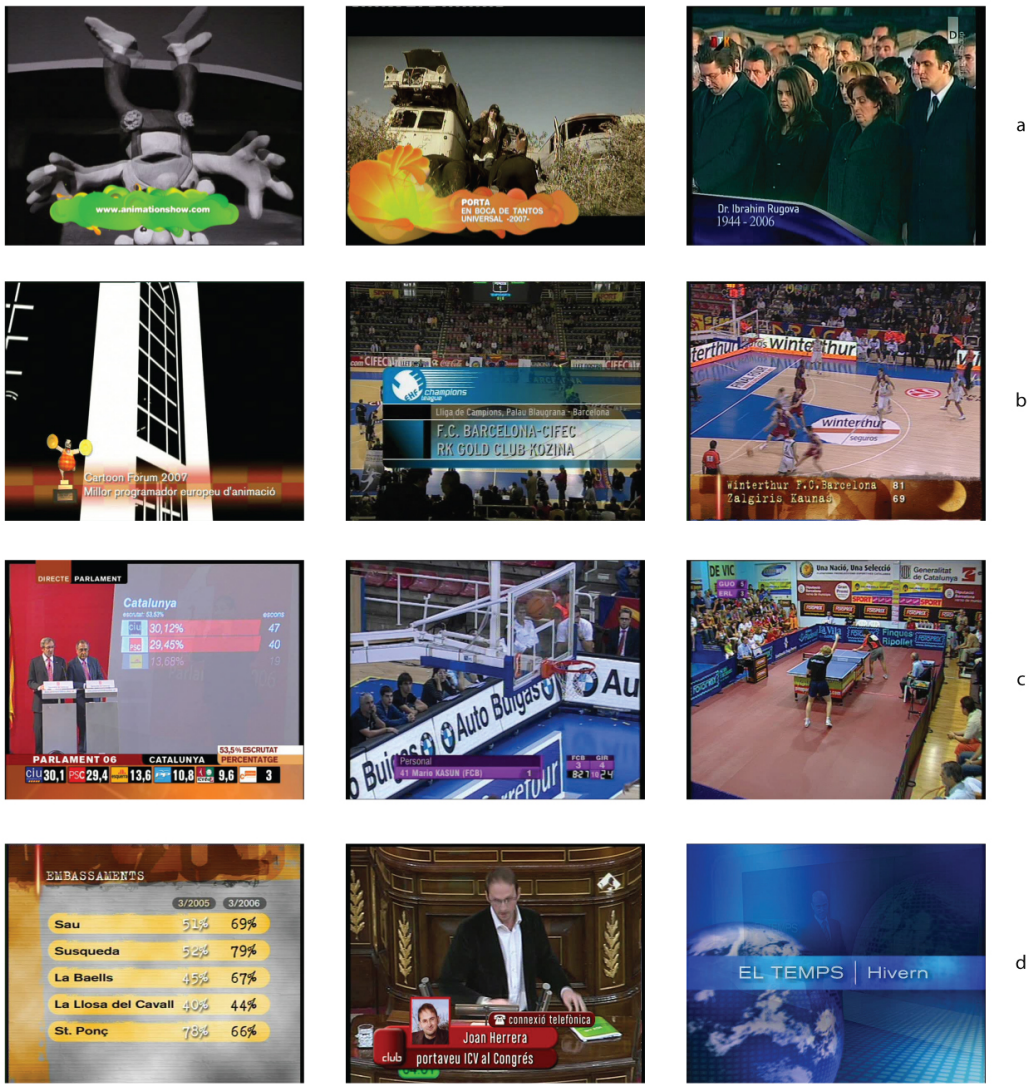


Figure 4.21: Images with captions from databases - (a)Captions do not have a perfect rectangle shape; (b)Captions have a transparent background; (c) Images itself are very high textured for successful caption discrimination ; (d)Captions have almost perfect rectangle shape, but the charters colors changes in the same caption, several captions are overlapped, the background color of caption is similar to the image color

Performance of the caption text extraction algorithm	
recall	precision
0.84	0.72

Table 4.2: Performance of the caption text extraction algorithm

The results of the algorithm can be measured in terms recall and precision metrics. Two metrics of the performance of the algorithm are calculated as:

$$recall = \frac{true\_positive}{true\_positive + false\_negative} \tag{4.27}$$

$$precision = \frac{true\_positive}{true\_positive + false\_positive} \tag{4.28}$$

The performance of the previously explained algorithm over all three databases is shown in table 4.2

4.7 Detected problems

Although the described above algorithm shows good performance, see previous section 4.6, there have been detected several shortcomings:

- One of the main problems is the number of false positives, which are obtained during the CTO extraction. Due to the conservative policy with respect to non-losing candidate text region, after the text candidate spotting procedure some nontext regions are chosen, and some of these regions are propagated through the whole algorithm because of their feature similarities with real text regions;
- The search of the best node is a very complex and time consuming procedure, procedure optimization should be done;
- In the part of the binarization several problems have been detected: the inverse binarization; the lost of single characters because of the discard of the connected components in the contour of the mask; not correct binarization in case of transparent background of caption. Binarization should be improved in term of threshold accuracy
- The algorithm works only for single images, the extension of the algorithm for a sequence of images would be an advantage.

5

Improvements on the caption text detection algorithm.

5.1 Overview

This chapter explains different experiments that have been made in order to improve accuracy and speed of the caption text detection algorithm. In the first section several proposals to enhance the text candidate spotting step are explained. Second and third sections focus on experiments in consistency analysis for output: in the second section are explained changes in the search of the best node to represent CTO algorithm, and the third section concentrates on problems and improvements in binarization.

5.2 Text candidate spotting

Two different methods have been applied to improve the accuracy of the text candidate spotting.

First approach consists in using the feature extraction based on the structural model of text. The structural features used here are ridges detected at several scales in the image. A line of text is considered as a structured object, modeled by a ridge at a coarse scale representing its center line and by numerous short ridges at a smaller scale representing the skeletons of characters.

Second approach consists in using MPEG-7 Edge Histogram Descriptor(EHD) [37] for text candidate spotting. EHD represents the spatial distribution of five types of edges, namely four directional edges and one non-directional edge in the image. Due to

the fact that text is characterized by a significant amount of the edges of different types, but mostly vertical and horizontal, values of EHD corresponding to the mentioned types of edges are used for classifying text and non-text regions.

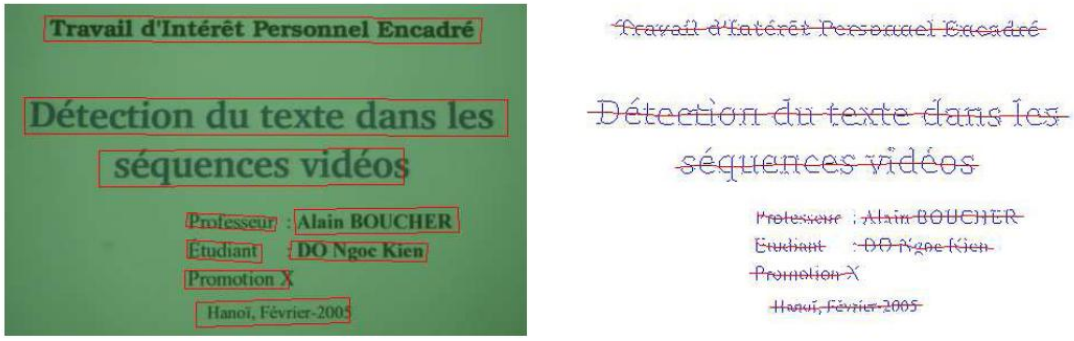
5.2.1 Structural model of text approach

To find text in images, structural features as ridges are used. Analyzing ridges in scale space allows to capture information about details as well as global shape, see [38]. A line of text is considered a structured object. At small scales can be clearly distinguished the strokes, and, at lower resolution, the characters are blurred and the text string forms an elongated cloud. This situation can be characterized by ridges at small scales representing skeletons of characters and at coarser scale representing the center line of the text string. In order to find text regions in images based on ridges, it is necessary to perform 2 steps:(1) ridges extraction in scale space and (2) classifying regions corresponding to ridges into two classes: text or non-text. The BPT regions which are marked as a text-contain regions will be preselected as candidate text regions.

5.2.1.1 Ridge extraction

This section briefly explains ridge detection. For more technical details, see [39]. Given an image $I(x;y)$ and its laplacian $L(x;y)$ a point $(x_r;y_r)$ is a ridge point if the value of its laplacian $L(x_r;y_r)$ is a local maximum in the direction of the highest curvature; it is a valley point if the value of its laplacian $L(x_r;y_r)$ is a local minimum. The term "ridge" is used to indicate these two types of points. Ridge points are invariant to image rotation and translation.

To detect ridge points, the main curvatures and associated directions at each pixel using the eigenvalues and eigenvectors of the Hessian matrix are computed. Then ridge points are linked to form ridge lines by connected components analysis. Scale space adds a third dimension σ to the image such that $I_\sigma(x;y)$ is the original image I smoothed by a Gaussian kernel with standard deviation σ. For capturing structures of different sizes the smoothed images were computed for values $\sigma_i = \sqrt{2}^i$ where the values of $i = 1, 2, 3, 4, 5, 6$ were estimated based on the dimensions of characters and text strings which could appear in caption text, see section 4.2. Figure 5.1 shows an image with detected text and its ridges detected at two scales.



(a) (b)

Figure 5.1: Example of ridges detection extracted from [38] - (a) Image of a slide; detected text regions are bounded by red rectangles. (b)Ridges detected at two levels $\sigma_1 = 2\sqrt{2}$ (blue) and $\sigma_2 = 16$ (red): red lines represent the center lines of text strings, blue lines represent skeletons of characters

5.2.1.2 Classification of candidate text blocks

The output of the previous step is k images containing ridge lines detected at k scale levels. Now, for each ridge at level i,$i = 0, ..., k-1$, the region corresponding to the ridge as text region or non-text region is classified. The region corresponding to a ridge detected at scale σ is defined as the set of points such that the distance from each point to the ridge is smaller than σ. The ridge at the coarse scale representing a text line is called the *central ridge*, the region corresponding to this ridge is called the *ridge region* and all ridges at smaller scale in this ridge region which best fit character skeletons are called the *skeleton ridges*. The scale of the skeleton ridges is half the width of their strokes. All detected ridges may be considered as central ridge starting from the largest scale σ_{k-1}. If the following criteria are fulfilled then the region, corresponding to a central ridge, is classified as text region:

- Ridge Length Constraint: Generally, the length of skeleton ridges representing the skeleton of the characters is approximately equal to the height of characters, which is 2 times the scale σ of the central ridge. For round characters like O, U, the length can reach up to 4 times σ. So the skeleton ridge length must be inside the interval $[\sigma;4\sigma]$. For the central ridge, it is supposed that $Ncharacter_{min}$ is the minimal number of characters in each text string, $Wcharacter_{min}$ is the minimal

width of a character. Thus the length of the central ridge has to be longer than $Ncharacter_{min}Wcharacter_{min}$, where $Ncharacter_{min}$ is a 1 and $Wcharacter_{min}$ is 5.

- Spatial Constraint: In printed latin characters, skeleton ridges often are perpendicular to the central ridge at their center points. However, this is not true for some fonts (e.g. italic), and for other character sets (e.g. chinese or japanese). To construct a generic text detection system, we weaken the perpendicularity constraint by applying a non-parallel constraint. Thus, a text ridge region must contain an important number of skeleton ridges which are not parallel to the central ridge. Above, we suppose that there is at least $Ncharacter_{min}$ in the text string, as each character contributes to at least one skeleton ridge, so the number of skeleton ridges inside the central ridge region has to be bigger than $max\{Ncharacter_{min}; length\_of\_central\_ridge/Ncharacter_{min}\}$.

The example of central ridges, skeleton ridges and also text regions could be seen in figure 5.2.

5.2.2 Edge-histogram approach

In the current text candidate spotting approach, the HL- and LH- subbands are used because of the high presence of vertical and horizontal lines in the text, see section 4.2. There is a MPEG-7 texture descriptor which indicates presence and amount of the edges of different types in the region: the Edge Histogram Descriptor(EHD). The EHD represents the spatial distribution of five types of edges, namely four directional edges and one non-directional edge. An image is divided into 4 by 4 subimages, and local edge histograms for each of these sub-images are computed. To generate the histogram, edges are categorized according to next types: vertical, horizontal, 45° diagonal, 135° diagonal, and non-directional edges. Thus, each local histogram has five bins corresponding to the above five categories. Since text is characterized by the significant amount of vertical and horizontal edges, it was expected that regions could be selected based on high values corresponding to vertical and horizontal edges and significant values corresponding to the diagonal (45° and 135°) edges. Values were calculated relatively to the area of the region, and taking into account that text regions

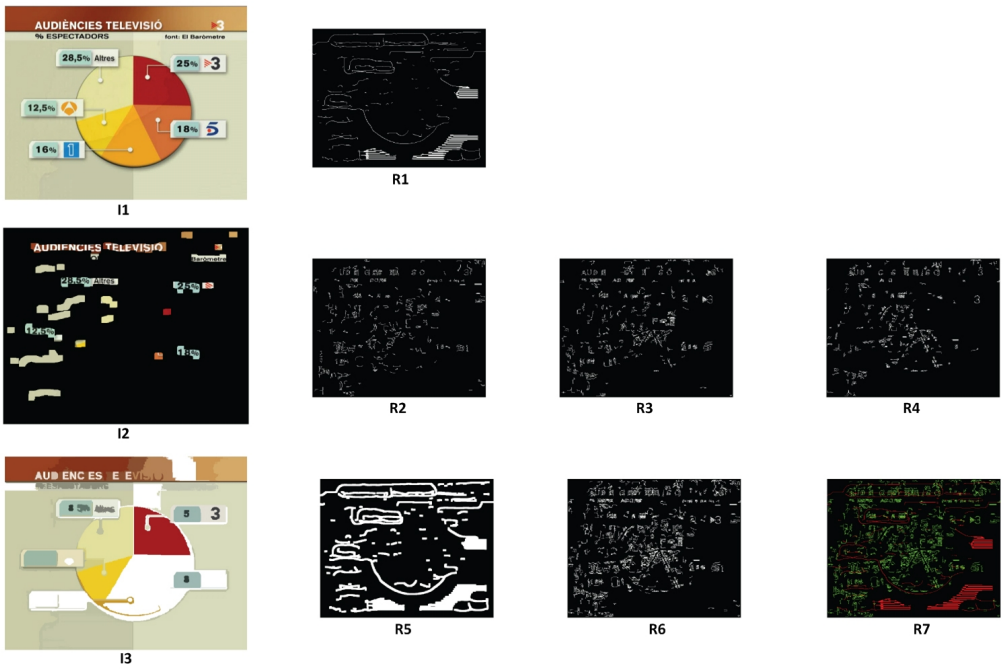


Figure 5.2: Example of text candidate spotting based on ridges - (I1) Original image; (I2) ridges corresponding to text; (I3) BPT-regions corresponding to text candidate spotting based on ridges; (R1)central ridges corresponded to character size, detected at level $\sigma = 8$; (R5) central ridge region detected at level $\sigma = 8$; (R2,R3,R4) sleleton ridges detected detected at two levels $\sigma = 1, \sqrt{2}, 2$; (R6) all previously detected skeleton rigdes together; (R7) Ridges detected at four levels $\sigma_1 = 8$ (red) and $\sigma_2 = 1$, $\sigma_3 = \sqrt{2}$, $\sigma_4 = 2$ (green): red lines represent the center lines of text strings, green lines represent skeletons of characters

are comparatively small, it is expected that the threshold based on the presence of vertical/horizontal/diagonal edges gives good text candidate spotting.

5.2.3 Structural model approach and edge histogram approach. Results and conclusion

Several experiments have been done to evaluate the performance of text candidate spotting based on the ridge analysis. Two drawbacks have been detected: first, text candidate spotting based on ridge analysis fails to distinguishing between text and non-text images when both images have textured zones; second, for images with captions, ridge analysis works acceptable, but text candidate spotting based on Haar power analysis has better performance, see figure 5.3 for comparison. One of the main criterium to accept the new approach for text candidate spotting would be that the number of false detected regions (false positives) decreases while the number of false rejected regions (false negative) does not increase. As it can be seen in the comparison figure 5.3, both goals are not achieved. Consequently, it was decided that text candidate spotting based on ridge extraction does not perform well in the scope of the caption text detection problem.

After evaluating EDH approach, it was concluded that for candidate text spotting it performs worse than the Haar power analysis. Even though the amount of false positives detected with EHD approach, is lower than those detected with Haar power analysis, the increasing of false negative is not acceptable in text candidate spotting.

5.2.4 Text candidate spotting. Restriction on the area of the region.

Based on the performed experiments and obtained results, it is easy to conclude that text candidate spotting based on the power of HL and LH subbands of Haar wavelet transforms is optimal. Once the result mask is obtained, this information must be translate in terms of regions of BPT. At this point is where an improvement has been introduced: the leaf region of BPT that is under the Haar power mask is considered as a text region only if at least 40% of the area of the region is superimposed with the Haar power mask. This modification helps to limit the amount of regions to be processed and exclude that regions which were marked as text regions only because they are neighbor regions of real text regions. As table 5.1 shows, the amount of false positive results using the restrictions on area is the same as the amount of the false positives

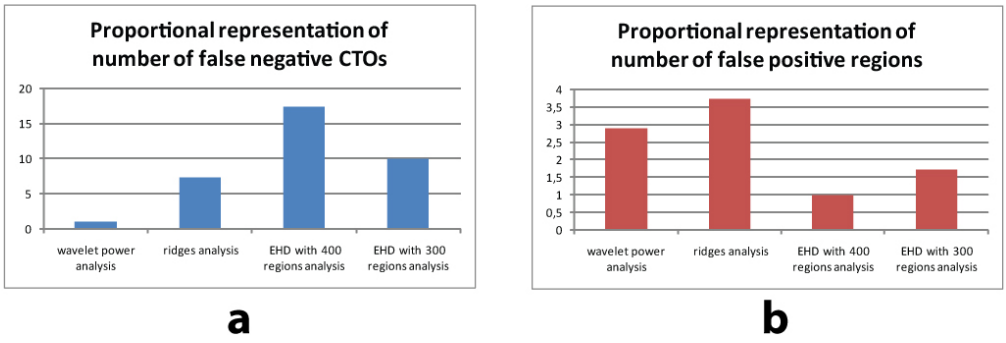


Figure 5.3: Comparison of the performance of structural model approach, EHD approach and Haar Wavelet analysis - (a) Analysis of the false negative results: all values are scaled with respect to the method with minimum number of false negatives. (b) Analysis of the false positive results: all values are scaled with respect to the method with minimum number of false positives

if the restriction is not applied, but the number of regions which are processed are significantly smaller, and, as a consequence, the mean processing time for each image is much lower. Also the result table shows that in the case with restriction only one previously correctly detected CTO is lost, but it is compensated by the increase of the amount of the partially detected CTOs.

The more significant improvement in the precision in text candidate spotting based on the Haar power analysis is done by applying temporal redundance, when the sequence of several key frames with the same captions is processed, see chapter 6.2

5.3 Consistency analysis of the output

5.3.1 Extraction of caption text objects. Bottom-up approach

In this section will be explained the experiments that have been made in order to improve tree analysis of verified text candidate nodes. The main goal is to increase the accuracy of caption text extraction algorithm without increasing the complexity.

Caption text extraction problem is a part of a more general problem: object extraction. In the context of current work the object extraction is done in terms of image segmentation through binary partition tree representation. There mainly exist two

	With restriction	Without restriction
Total number of CTOs	552	552
Correctly detected CTOs	410	411
Partially detected CTOs	73	69
Number of FN CTOs	69	72
Number of FP CTOs	60	60
Total number of processed regions	4608	8067
Mean time of processing per image	9,36 sec	11,62 sec

Table 5.1: Comparison of the performance of CTO detection algorithm with and without restriction of the area of region below Haar power mask. The results are obtained processing images from the test database. The BPT of each image contains 200 regions

conceptually opposed approaches for image segmentation: Top-down methods, which are knowledge-based and search for pre-defined models into the image given some prior information. Bottom-up methods, which are generic methods aiming at linking visual features to perceptual meaningful primitives. To find caption both methods can be used, because, on one hand, it is possible to model a caption, thus regions could be analyzed for belonging to the caption model (this is the top-down approach), on the other hand by checking regions characteristics, the decision about object presence in the region could be done (bottom-up approach).

As it was explained in section 4.4.1.1, the method that is used in the search of best node is a hybrid top-down bottom-up method. Candidate regions are analyzed starting from the top of BPT to find subtrees, that is where the top-down strategy is applied; then, each subtree is analyzed bottom-up starting from the subtree leaf region. CTOs of each subtree are compared with CTOs of different subtree only at the end of processing all subtrees, see chapter 4.4.1.2.

The idea behind the bottom-up approach is to avoid the separation of CTOs by subtrees and compare captions between themselves during the analysis of the whole BPT. As the top-down strategy is applied only for separating the whole BPT in subtrees, and the idea is to perform the analysis of the whole BPT, the bottom-up search through the BPT was performed in the search of the best node algorithm. To compare these two different BPT scanning strategies - hybrid and bottom-up - , the comparison in terms of recall and precision metrics was done, see 4.6.

The objective of the improvements is to decrease the number of false positives while the number of false negatives does not increase, in other words, any previously detected CTO should not be discarded. The experiment were done over 3 different databases (see chapter 4.6): news, sport and talkshow. As a result of comparison, see figure 5.4, precision was degraded about 8 % for all 3 databases, while non significant improvements of recall value were observed for talkshow database - around 5% - and for sport database - around 11%, while for news database the recall value has degraded about 9%. As a conclusion of this experiment, it was decided that the original hybrid top-down bottom-up strategy of the BPT scanning is optimal for caption text objects extraction.

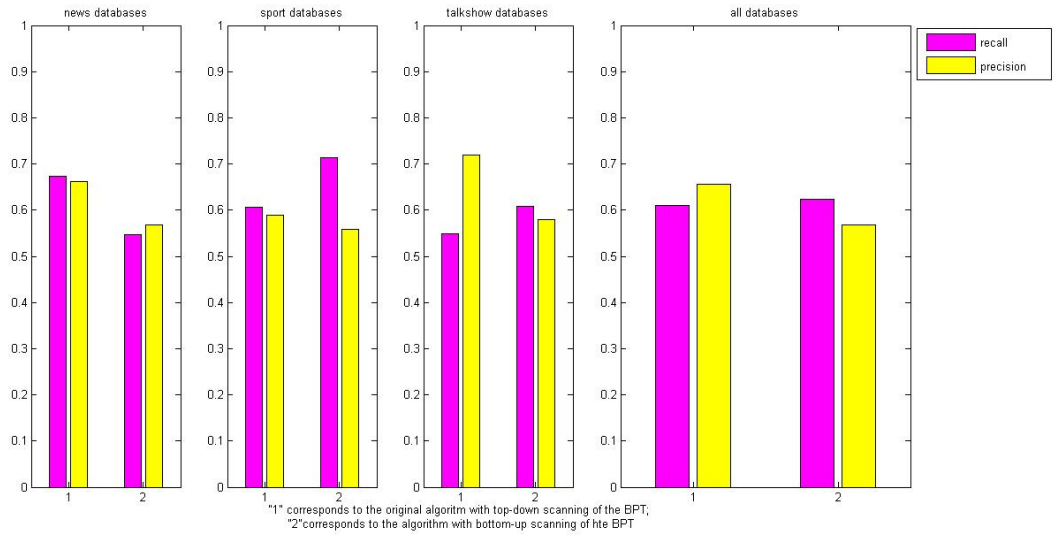


Figure 5.4: Comparison of performance of hybrid and bottom-up strategies in analysis of text candidate nodes in BPT - Three databases were analyzed (from left to right): news database, sport database, talkshow database. The last plot represents an average recall and precision values for all databaes

5.3.2 Binarization

Binarization results from chapter 4.4.2 show that there are some problems in that step. Main difficulties in binarization are related to the estimation of the appropriate threshold to separate text characters from the background. There is no difficulties in binarization if the caption is correctly extracted, but in case when the search of the

best node algorithm fails to extract the whole CTO or when background is transparent or when there is degradation of characters color or background color or both colors in the caption it could be difficult to estimate the optimum threshold for the whole CTO. To solve that problems the binarization by words of CTO was introduced. In order to split a CTO in different words, next steps are performed:

- to convert CTO image to the grayscale image, see figure 5.5(b);
- to extract all edges, using Canny edge detector ; the output of this step is a binary image, where are edges represented by pixel with TRUE value, see figure 5.5(c);
- to connect characters of the word into one component: dilation with a structuring element (9 by 2 pixels) is applied to the previous binary image; the size of the structuring element was experimentally defined, based on the mean distance between the characters of the same word; the output of this step is the image where connected components correspond to the words, see figure 5.5(d);
- connected components are classified in order to discard those that are noise; classification is done based on the geometrical properties of the components; all component for which width of bounding box is lower than 13 pixels and the height of bounding box is lower than 5 pixels, are discard;
- all connected components classified as a text-contained are the inputs to the binarization process, see figure 5.5(e,f);

Figure 5.5 illustrates all above explained process.

Table 5.2 proofs the increase of accuracy in binarization using binarization by words. An advantage of the method is that the number of false positives remains or decreases during the binarization by words. Accuracy increases mostly because of correct binarization of previously inverse binarized objects. During the initial binarization those object were discarded because they had zero probability after the CTO probability estimation process, now these objects are classified as a correct results. In the initial binarization process, CTOs that represent captions with degraded color background or with several colors background were binarized partially correct, partially not. With binarization by words the whole CTO is binarized correctly, assuming that some words of caption are binarized as white letters on black background, and some words of the

Figure 5.5: Steps of the binarization by words process - Up-down: initial caption text object, grayscale component of initial CTO, detected edges of CTO, result of dilation of detected edges, selected components and its binarized versions

	Initial binariztion		Binarization by words	
Number of detected CTOs before binarization	512			
	numeric	%	numeric	%
Correctly binarized CTOs	421	82,20%	456	89,06%
Partially correctly binarized CTOs	64	12,50%	38	7,40%
Incorrectly binarized CTOs	27	5,30%	18	3,50%
Number of detected CTOs after binarization	489	95,50%	496	96,80%
Number of discarted incorrectly binarized CTOs	23	4,49%	16	3,12%

Table 5.2: Table with CTO detection results using initial binarization method and binarization by words

same caption are binarized as a black characters at white background. For examples see figure 5.6. A drawback of the method is the increase of computational cost. Taking into account that for correctly detected CTO with solid background the output of the binarization for both methods is the same, it was decided to include binarization by words as an option of the binarization process in the software, and therefore keep both binarization methods. If the user knows about image or video content the best choice of binarization could be taken. Other way, the option with more accurate results is the binarization by words.

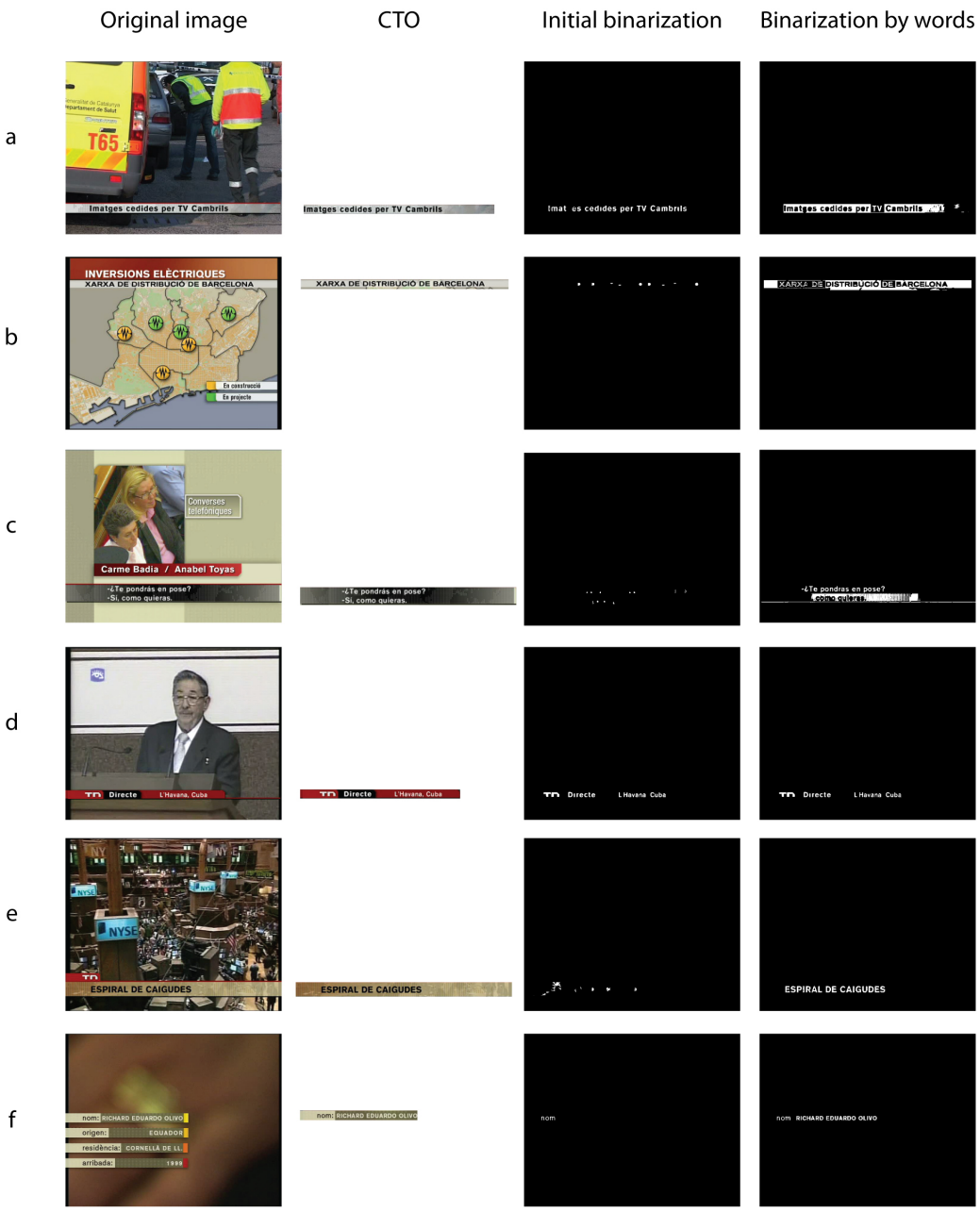


Figure 5.6: Examples of binarization by words - (a) Result of initial binarization is more correct even through the output of the binarization by words is considered as correct; (b),(c) Initial binarization is incorrect and could not be read by OCR meanwhile the binarization by words creates more readable output; (d) There is no difference in outputs between after initial binarization and binarization by words; (e) The initial binarization is incorrect while the output of binarization by words gives the correct result; (f) Due to the difference in background and characters colors of CTO with initial binarization only part of CTO is binarized correctly while with the binarization by words the whole CTO is processed correctly

5. IMPROVEMENTS ON THE CAPTION TEXT DETECTION ALGORITHM.

6

Temporal approach

6.1 Overview

This chapter explains the extension of the caption text extension algorithm over several key frames containing the same caption text. Usually caption text is added during several seconds in the video, because the viewer needs some time to read text information on the image. To improve caption text extraction accuracy, it would be possible to analyze all sequential frames in the video, but due to the computational cost, computational complexity and temporal redundancy between neighboring frames, only some key frames of the video segment are extracted and analyzed. Key frame is a type of video frames that in terms of video temporal compression does not require another video data to be decoded, as it does not refer to previous or subsequent going frames. For example, within MPEG-2 video, key frames are named Intra-frames (I-frames). Since between inter frames there is a temporal redundancy that increase the risk of incorrect detection of the CTO, the information only from different key frames will be fused in several parts of the algorithm. In the first section, how the information from different frames is used to increase the accuracy of text candidate spotting is explained. In the second section, improvements in the consistency analysis of the output is highlighted, and in the third section, enhancements in the binarization are explained. The overview of the temporal approach for the caption text extraction algorithm is in figure 6.1

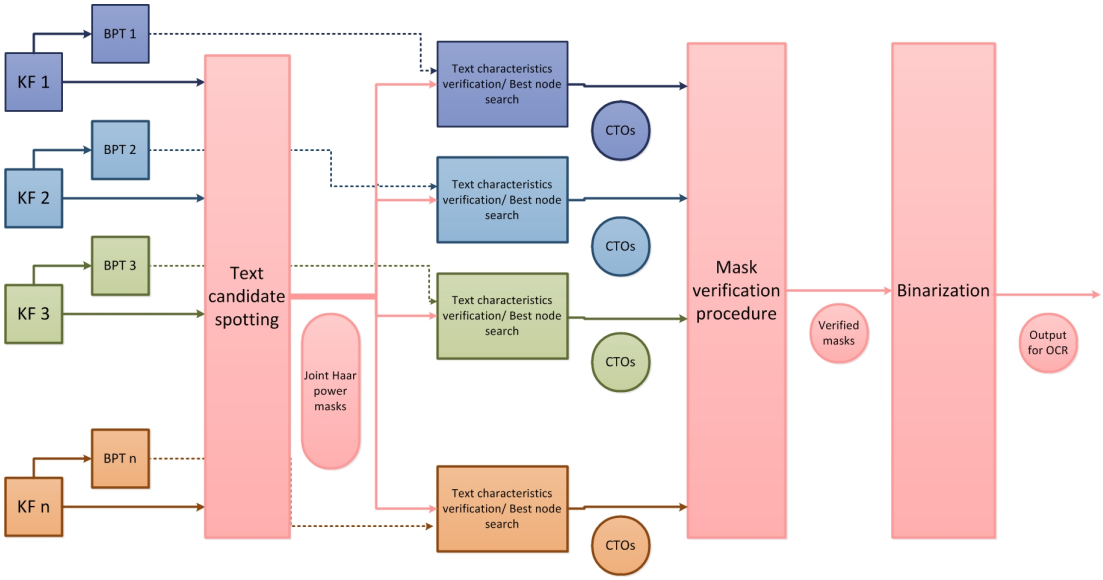


Figure 6.1: The overview of the temporal approach for the caption text algorithm -

6.2 Temporal approach for text candidate spotting

Due to the fact that there are several key frames with the same caption text, placed in the same position, we can merge the information of the text placement from different images. In chapter 5.2, the analysis of the power of HL and LH subbands of the Haar wavelet gives reasonably accurate text candidate spotting. To increase the accuracy of text candidate spotting, Haar power mask is calculated for each key frame, and then the joint Haar power mask is created as a result of the union of the individual masks. For each pixel (x, y) of the joint mask, a mark is assigned: the value of the mark is equal to the number of frames, where at position (x, y) of each individual Haar power mask is $TRUE$. If the value of the mark is more than $ceil(0.7 * \max_{i=1:N} marks)$, where N is the total number of key frames, the pixel (x, y) of joint mask gets the $TRUE$ value. Roughly speaking, the pixel (x, y) will get the $TRUE$ value in the joint Haar power mask if values of this pixel are marked as $TRUE$ in the majority of the individual Haar power masks. This approximation helps to make the joint Haar power mask more precise. In case when the analysis is done only for 2 key frames with the same caption text, the joint Haar power mask will be the union of the individual haar power masks of two key frames. Increasing the accuracy of the joint Haar power mask implies a

reduction of the number of the regions that will be checked as possible text candidates, and as a consequence the processing time for the extraction of the CTO will decrease. An example of the individual and joint Haar power masks could be seen in the 6.2.

After creating the joint mask, this mask is used as individual Haar power mask for each key frame. Then each key frame is processed separately because the BPT associated with each key frame differs.

6.3 Temporal approach for consistency analysis of the output

To include information from all analyzed frames with the same captions, two modifications of the consistency analysis of the output have been done: first, the output masks of the best node search step for each key frame are jointly analyzed, and, second, the information from different key frames are merged for better text color mean estimation in the binarization.

6.3.1 Verification of the CTO mask

The output of the best node search algorithm is the list of CTOs with their associated masks and geometrical parameters: width, height, center of masses and rectangularity. Due to the fact that in all analyzed images there is the same number of captions and theirs positions are the same, output masks should be equal, but they usually present little differences. Key frames contain the same text, but their background differs, which leads to different BPTs, with different regions. As a result, in some images are detected false positive CTOs, and in some images there are false negatives. But using all detected CTOs from all images, lost and false positive CTOs could be detected.

CTO mask verification can be divided into two parts: first, masks of CTOs detected in all key frames are adjusted to fit better real captions, second, the correction of false positive/negative results is done. This is possible, because the comparison between the same CTO mask in different images could be done. Based on that, the decision could be taken.

Masks of all CTOs can also be divided into two groups: first when the same CTO mask is presented in all images, second when the CTO mask is presented not in all analyzed images. The order of the masks verification plays an important role. All

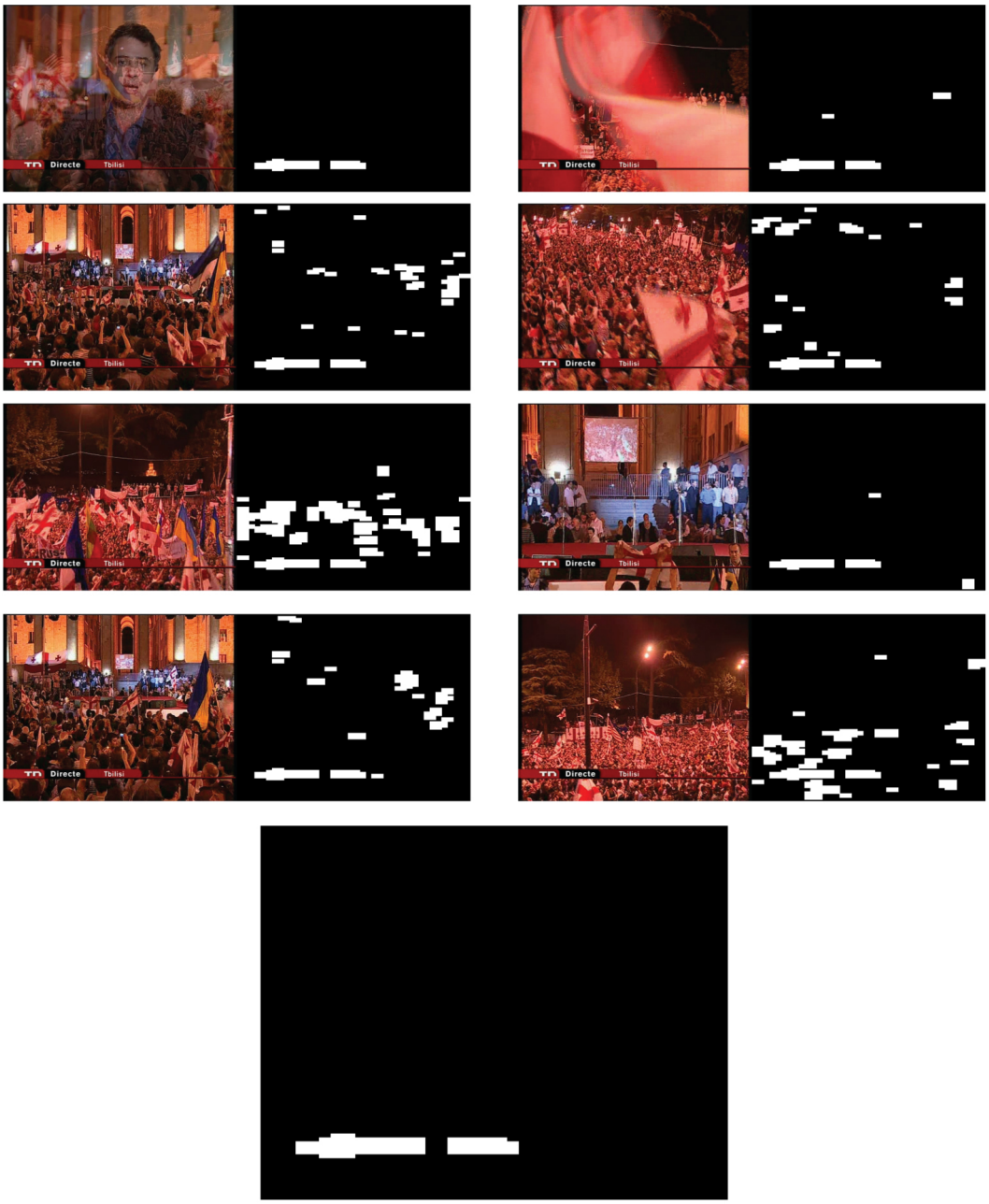


Figure 6.2: Examples of binarization by words - Eigth key frames with the same captions, each one with corresponding individual haar power mask, bottom image - joint Haar power mask

masks are checked according to the probability of text containing. This probability is defined as a function of the number of frames where this mask appears. Masks, detected in bigger number of key frames, are verified first. Specially, it is selected a set of masks that are detected in all images, because these masks with the highest probability contain text.

The algorithm of mask verification consists of several steps:

First step analyzes thoses CTO's masks, that are detected in all key frames.

1. A list of all CTOs final masks detected in all key frames is created;

2. For each CTO mask, detected in all key frames, an intersection masks is calculated. The resulting mask is called working mask. This way for each detected CTO in all key frames, a working mask is created.

3. Finally, it is checked if the working mask represents the same caption in all key frames. The following color descriptor measurements should be fulfilled:

   (a) The Euclidean distance between MeanColor descriptor values could efficiently characterize the color similarity between images. But this descriptor does not include information about spacial distribution of color in an image, that is why the ColorLayout descriptor is used;

   (b) The Color Layout(CL) captures the global spatial layout of the colors in an image by computing the DCT coefficients of the set of mean colors of a 8 by 8 grid. Mathematically, the descriptor is represented by N_i coefficients for each of the N_c channels, as follows:

   $$CL = \{\{c_j^i\}_{j=1}^{N_i}\}_{i=1}^{N_c} \tag{6.1}$$

   Two descriptors CL_1 and CL_2 are compared in MPEG-7 as follows:

   $$D(CL_1; CL_2) = \sum_i^{N_c} \sqrt{\sum_j^{N_i} w_j^i (c_{j1}^i - c_{j2}^i)^2} \tag{6.2}$$

   where w_j^i is the weight associated to the j-th coefficient of the i-th channel, allowing the measure to penalize more the differences in lower frequencies (see [40]).

To check if two masks of CTOs are similar enough to represent the same caption, first the MeanColor descriptors are obtained. If the Euclidean distances between these MeanColor values are below the threshold, which is 3.5, these two masks represent the same caption with certainty. If the MeanColors difference is in the confidence interval $[3.5; 20]$, then it can not be assured that both masks represent the same caption text. Then, the ColorLayout descriptors and their similarity measure are calculated. If the similarity measure is below the empirically fixed threshold 13.5, then it is decided that both masks belong to the same CTO, otherwise masks do not describe the same CTO. Also it is conclude that masks do not from the same caption if the Euclidean distance between ColorMean descriptor values is above the threshold, which is 20. That way masks said to represent the same caption according to the next rule:

- **if** $(CM_d < 3.5)$ **or** $((CM_d \in [3.5,\ 20])$ **and** $(CL_d \leq 13.5))$ **then**
  masks represent the same caption
  else
  masks do not represent the same caption
  end if

where CM_d is the Euclidean distance between values of ColorMean descriptors and CL_d is the similarity measure between ColorLayout descriptor.

There are two possible situations when working masks are checked based on the above described color-based measurements:

(a) If measurements are fulfilled, the current working mask is considered as a "text mask"(TM). Then using this TM as a reference, it is searched in the list of all CTOs masks, those which contain this TM. From all these masks, the biggest mask is chosen, and it is checked, if the same color-based measurements are fulfilled. If so, then this biggest mask acts as TM; if the measurements are not fulfilled, the process runs recursively for the next biggest mask. The process stops when the first mask, satisfying the conditions is found, or when all masks have been checked. After that, CTO parameters are updated and TM is considered as the CTO mask. To exclude this mask from further mask verification process, this TM is excluded from the list of all CTOs masks detected in all key frames.

(b) If measurements indicate that the working mask is not a TM, the working mask represents a false positive. This working mask is excluded from further verification process by excluding it from the list of all CTOs masks detected in all key frames.

After checking all CTO masks that are detected in all key frames, the list of text masks and associated CTOs are obtained. This list represents CTOs that will be binarized. Also the list of all CTOs (LoCTOs) masks detected in all key frames is updated to not include already checked masks. The next step is to check masks that are detected in some key frames, or, in other words, check masks from the LoCTOs. After the excluding checked mask from all masks, there are some masks that are splitted or bisected, these masks could contain elongated thin parts. As these parts cannot contain text, we are interesting in discarding them. for this purpose, the opening with structural element 9 by 9 is performed over each mask from LoCTOs.

Second step is an analysis of masks of CTOs, that are detected just in several or even in one key frame. After first step, the list of all CTOs masks detected in all key frames contains only non-checked masks. For each non-checked mask the same verification based on the same color measurements is performed. The order of masks check depends on the number of key frames where this mask was detected. Masks, detected in bigger number of frames are checked first.

6.3.1.1 Results

Results of the mask verification process is a list of CTOs and their verified masks. The database of 176 images with 332 captions was used to calculate the precision of the extension of CTO detection algorithm over several key-frames with the same captions. These challenging 176 images with text of different size and color, and complex background textures are divided in 51 sequences of key frames. The length of sequence differs from 2 to 18 images each one. The comparison of the results obtained by running the CTO detection algorithm over each frame separately and over several frames appears in the figure 6.3 and in table 6.1. The comparison of the algorithms is done using the same method of binarization to exclude possible influence of the binarization procedure on results.

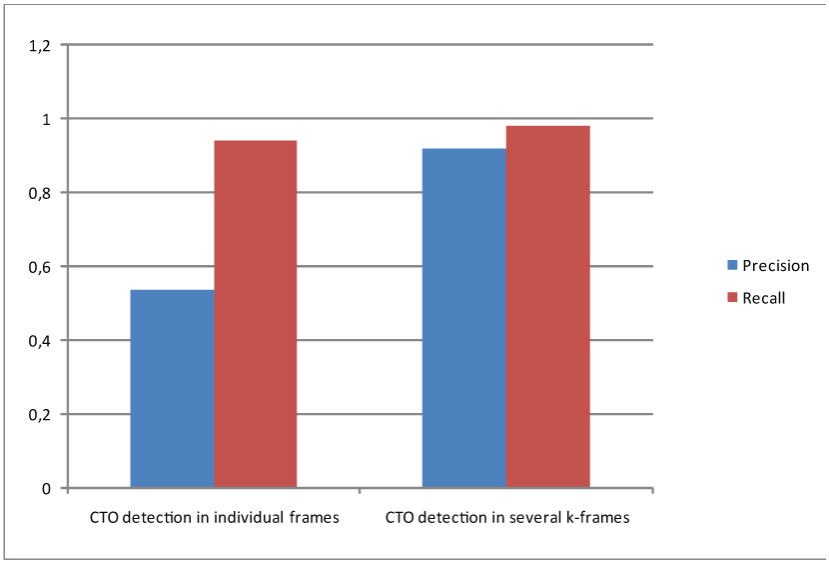


Figure 6.3: Comparison of the performance of CTO detection algorithm in individual frames and in the set of several key frames -

	CTO detection			
Images	176			
CTOs ground truth	332			
	Individual		Several K-frames	
	numeric values	% over all CTOs	numeric values	% over all CTOs
Correctly detected	251	75,60%	318	95,78%
Partially detected	61	18,37%	8	2,41%
False negative	20	6,02%	6	1,81%
False positive	269		29	
Recall	0.94		0.98	
Precision	0.54		0.92	

Table 6.1: Detection results using CTO detection algorithm in individual frames and in several key frames, where PD is the number of partially detected objects, FP is the number of false positives and FN is the number of false negatives

Results are classified as correctly detected, partially detected, false positives and false negatives. The object is considered to be partially detected if at least 50% of the object is correctly detected. That is why objects, that are classified as partially detected objects, in terms of recall and precision values are considered as detected objects. Advantages of the temporal approach are:

- The decrease of false positives, that is around 90% while the amount of false negative results decreases a 70%. Therefore the goal of minimizing number of false positives without increasing false negatives is achieved.
- The significant amount of CTOs, that in the individual caption text extraction algorithm are partially detected, in case of temporal analysis are fully detected.
- And not so obvious, the time of jointly processing several key frames is lower than time process separately the same key frames.

It is easy to conclude that temporal extension of the algorithm gives better detection results in less time.

6.3.2 Binarization

As usual, if the same caption with the same text is presented in several consecutive key frames, the color of characters do not change from frame to frame. Color of background could change significantly, especially in the case of the transparent background of captions. Binarization process consists of separating background information from foreground(text) by thresholding. To improve quality of binarization, a better estimation of the threshold should be calculated. As it has been explained in chapter 4.4.2, to determine the threshold, the mean color value of characters and mean color value of background are used. Several horizontal lines are passed over the CTO, and based on pixel values of these lines, the color of background and color of text are calculated. In case when all analyzed lines are classified as a high-variance lines, that means that these lines contain text, the background color value is estimated using small patches from the corners of CTO, where probability to find text is very low. But due to the degradation in text/background color or some visual effect that are applied to call the attention to the caption, not always threshold that separates text from background

could be estimated correctly using only several values before and after the significant color changes.

Using several CTOs that represents the same caption, text mean color could be estimated. All key frames is processed using previously explained procedure of line analysis, which is applied to every CTO. In case of temporal binarization, firstly, only lines with high color variance are analyzed. All values of the line are divided into two classes by 2-means clustering algorithm. To define what class characterizes the text, and what class the background, the color mean values of lines with low color variance are calculated. The median value of the previously mentioned means is assumed to represent color mean of the background. That way, the color mean of text and color mean of background are calculated. The threshold value is calculated as $0.5 * (Color_{background} + Color_{text})$. More accurate background color estimation allows to represent text in white color after binarization, and the deleted white connected components in the contour of the binarized CTO (see chapter 4.4.2) does not lead to loss of the text.

	one frame		several Kframes	
Number of detected CTOs before binarization	318			
	numeric	%	numeric	%
Correctly binarized CTOs	259	81,14%	286	89,90%
Partially correctly binarized CTOs	42	13,20%	20	6,28%
Incorrectly binarized CTOs	17	5,30%	12	3,77%

Table 6.2: Comparison of the binarization results using CTO detection algorithm in individual frames and in the set of several key frames

The increase of the performance in the binarization process could be seen in table 6.2. The results are divided into three groups: correctly binarized, partially binarized and incorrectly binarized. The binarization is defined as correct if character appears in white color on black background or in black color in white background. Otherwise the binarization is classified as incorrect. The binarization is considered as partially correct if not all characters of the word are correctly binarized. Performance evaluation in binarization for both methods was done on the same captions previously extracted from images. The set of captions containing only captions with words does not contain false positive captions. The amount of incorrectly binarized CTOs decreases from 17

to 5 (on 70%), while the amount of partially correctly binarized CTOs decreases from 42 to 20, that is about 52%. As a consequence the amount of correctly binarized CTOs increases. But athe temporal analysis of binarization requires extra computational cost and there is always the trade-off between the performance and computational cost, this binarization method was also included in the software as optional, binarization can be done individually for each key frame or jointly for all key frames.

7

Conclusions

This document presents the system developed for caption text extraction. The system is based on a hierarchical region-based image representation. In this work, the image is represented as a Binary-Partition-Tree (BPT), where image regions are represented as nodes of the BPT, see chapter 3.

The desired image text or caption is modelled as a set of descriptors. The descriptors used in this work are the following:

- Shape descriptors: height, aspect ratio, compactness and rectangularity;
- Texture descriptors: Haar wavelet decomposition;
- Color descriptors: Color Mean, Color Layout and Dominant Color descriptors.

A similarity measure between caption model and image regions, or BPT nodes, has been implemented, see chapter 4. The image representation is analyzed to select those image regions or BPT nodes that better fit the given caption model according to the similarity measure described in see chapter 4. The selected image regions are slightly modified to better fit the model. The resulting caption images are binarized and processed by an OCR.

The following section presents the most important contributions of this work.

7.1 Contributions

Several experiments were carried out to improve the performance of the system or to confirm that the current system configuration works in an optimal way:

- In the caption modelling step (text candidate spotting), new approaches were tested, such as structural representation of text through ridge detection and text modelling through Edge Histogram Descriptor; it could be conclude that the text modelling based on the Haar wavelet decomposition shows better results in terms of accuracy and computational costs; small changes in text candidate spotting were introduced without changing the main concept of Haar wavelet analysis; with these modification the better performance, in respect to the original text candidate spotting, is obtained.

- In the image fitting to model step (best node search), bottom-up BPT scanning approach was tested; as a result of test, it could be conclude that the current hybrid top-down bottom-up scanning method is optimal for caption text extraction;

- In the binarization step a more efficient binarization method by words has been proposed;

- The temporal approach for several key frames with the same caption is implemented in the steps of text candidate spotting and consistency analysis of output. Simultaneous analysis of several key frames increases significantly the recall and precision of the algorithm;

7.2 Future work

Experiments performed in this work confirm that for single image the initial algorithm [2] performs in optimal way in the steps of text candidate spotting, text candidate verification and best node search. The majority of false positives are detected when the classification of the final binarization results is done. Therefore, the current results could be improved at the binarization stage. However, it has to be taken into account that the BPT creation procedure can not be modified for backwards compatibility.

The temporal analysis of several key frames could benefit a lot from creating temporal binary partition tree for several images, and adaptation the search of best node algorithm to the temporal BPT.

Bibliography

[1] M. Leon; A. Gasull;. **Text Detection in Images and Video Sequences**. *International Conference on Multimedia, Image Processing and Computer Vision*, 2005. 1, 3

[2] M. Leon. **Caption text extraction for indexing purposes using a hierarchical region-based image model**. *2009 IEEE International Conference on Image Processing*, 2009. v, 1, 13, 14, 80

[3] R. Lienhart; W. Effelsberg;. **Automatic Text Segmentation and Text recognition for Video Indexing**. *Technical Report TR-98-oo9, Praktische Informatik IV, University of Mannheim*, 1998. 3

[4] H. Li; D.Doermann; O. Kia;. **Automatic Text Detection and Tracking in Digital Video**. *IEEE Trans. on Image Processing*, 2000. 3, 6, 8

[5] G.K. Myers; R.C. Bolles; Q.-T. Luong; J.A. Herson;. **Recognition of Text in 3-D Scenes**. *Fourth Symposium on Document Image Understanding Technology*, 2001. 3

[6] K. Jung; K. Kim; A.K. Jain;. **Text Information Extraction in Images and Video: a Survey**. *Pattern recognition, Vol. 37*, 2004. 3

[7] L. Agnihotri; N. Dimitrova; M.Soletic. **Multi-layered Videotext Extraction Method**. *IEEE International Conference on Multimedia and Expo (ICME) 2002*, 2002. 4, 5, 6

[8] D. Chen; J.-M. Odobez; H. Boulard;. **Text Detection and Recognition in Images and Video Frames**. *Pattern Recognition the journal of the pattern recognition society*, 2003. 5, 6, 7

[9] Y. Zhong; H. Zhang; A.K. Jain;. **Automatic Caption Localization in Compressed Video**. *IEEE Trans. PAMI*, 2000. 5, 6

[10] B. Shen; I.K. Sethi;. **Direct feature extraction from compressed images**. *SPIE Storage and Retrieval for Image and Video Databases IV*, 1996. 5

[11] U. Gargi; S. Antani; R. Kasturi;. **Indexing Text Events in Digital Video Databases**. *ICPR*, 1998. 5

[12] H. Li; D. Doermann; O. Kia;. **Automatic Text Detection and Tracking in Digital Video**. *Univ. of Maryland, College Park, Tech.Reps. LAMP-TR-028, CAR-TR-900*, 1998. 6

[13] N. Chaddha; A. Gupta;. **Text Segmentation Using Linear Transforms**. *Proc. of Asilomar Conf. Circuits and Computers*, 1996. 6

[14] D. Zhang; R.K. Rajendran; S.-F. Chang;. **General and Domain-Specific Techniques for Detecting and Recognizing Superimposed Text in Video**. *IEEE International Conference in Image Processing (ICIP 2002)*, 2002. 6

[15] L. Agnihotri; N. Dimitrova. **Text Detection for Video Analysis**. *IEEE Workshop on CBAIVL 1999*, 1999. 6

[16] X.-S. Hua; X.-R.Chen et al. **Automatic Location of Text in Video Frames**. *ICPR, Intl Workshop on Multimedia Information Retrieval (MIR2001, In conjunction with ACM Multimedia*, 2001. 6

[17] M.A. Smith; T. Kanade;. **Video Skimming and Characterization thought the Combination and Language Understanding Techniques**. *IEEE Computer Vision and Pattern Recognition*, 1997. 6

[18] T. Sato; T. Kanade; E. K. Hughes; M.A. Smith; S. Satoh;. **Video OCR: Indexing Digital News Libraries by Recognition of Superimposed Captions**. *ACM Multimedia Systems: Special Issue on Video Libraries*, 1999. 6

[19] S. Tekinalp; A.A. Alatan;. **Utilization of Texture, Contrast and Color Homogeneity for Detecting and Recognizing Text from Video Frames**. *IEEE International Conference in Image Processing (ICIP 2003)*, 2003. v, 6, 7

BIBLIOGRAPHY

[20] M. Leon. **Extraction of Semantic Entities from Images and Video Sequences for Text Detection**. *PhD. Thesis Proposal, Department of Signal Theory and Communications, UPC, Barcelona, Spain*, 2006. v, 7, 9

[21] R. Lienhart; F. Stuber;. **Automatic Text Recognition in Digital Videos**. *Proceedings of SPIE Image and Video Processing IV 2666*, 1996. 8

[22] R. Lienhart;. **Text Segmentation and Text Recognition in Digital Videos**. *http://www.informatik.uni-mannheim.de/informatik/pi4/projects/MoCA/ProjecttextSegmentationAndRecognition*, 1998. 8

[23] A.K. Jain; B. Yu;. **Automatic Text Location in Images and Video Frames**. *Pattern recognition, Vol. 31, No. 12*, 1998. 8

[24] A.K. Jain; S. Bhatarcharjee;. **Text Segmentation using Gabor filters for automatic document processing**. *Machine Vision and Application*, 1992. 8

[25] H-K Kim;. **Efficient Automatic Text Location Method and Content-based Indexing and Structuring of Video Database**. *Journal of Visual Communication and Image Representation, Vol. 7*, 1996. 8

[26] V. Wu; R. Manmatha;. **TEXTFINDER: An Automatic System to Detect and Recognize Text in Frames**. *IEEE Transactions on Pattern analysis and machine intelligence, Vol. 21, No.11*, 1999. 8

[27] V. Wu; R. Manmatha; E.M. Riseman;. **Automatic text Detection and recognition**. *Proceedings of Image Understanding Workshop*, 1997. 8

[28] E.K. Wong; M. Chen;. **A new Robust Algorithm for Video Text Extraction**. *Pattern Recognition 36*, 2003. 9

[29] E.K. Wong; M. Chen;. **A Robust Algorithm for Text Extraction in Color Video**. *Proceedings of IEEE International Conference on Multimedia and Expo*, 2000. 9

[30] X. Tang et al.;. **Video Text Extraction using Temporal Feature Vectors**. *2002 IEEE International Conference on Multimedia and Expo. ICME'02*, 2002. 10

[31] P. Salembier; L. Garrido;. **Binary Partition Tree as an Efficient Representation for Image Processing, Segmentation, and Information Retrieval**. *IEEE Transactions on image processing, vol. 9, no. 4*, 2000. 12

[32] L.G. Ostermann. **Hierarchical Region Based Processing of Images and Video Sequences: Application to Filtering, Segmentation and Information Retrieval**. *vol. 1, Department of Signal Theory and Communications, UPC, Barcelona, Spain*, April 2002. v, 12, 13

[33] V. Vilaplana; F. Marques; M. Leon; A. Gasull;. **Object detection and segmentation on a hierarchical region-based image representation**. *Proceedings of 2010 IEEE 17th International Conference on Image Processing*, 2010. 17

[34] P.L. Rosin. **Measuring rectangularity**. *Journal of Machine Vision and Applications*, 1999. 23

[35] Yong Man Ro; Munchurl Kim; Ho Kyung Kang. **MPEG-7 Homogeneous Texture Descriptor**. *Journal of ETRI*, 2001. vi, 45

[36] D. Zhuang. **Efficient Text Classification by Weighted Proximal SVM**. *Fifth IEEE International Conference on Data Mining (ICDM05)*, 2005. vi, 46, 47

[37] B.S. Manjunath; P. Salembier; T. Sikora. **Introduction to MPEG-7: Multimedia Content Description Interface**. *Wiley, 2002. ISBN: 0-471-48678-7*, 2002. 53

[38] H. Tran; A. Lux; A. Boucher;. **A Novel Approach for Text Detection in Images Using Structural Features**. *Proc. of ICAPR (1)'2005*, 2005. vi, 54, 55

[39] H. Tran; A. Lux;. **A method for ridge extraction**. *Proc. of the 6th Asean conference on Computer Vision, ACCV'04*, 2004. 54

[40] Xavier Giro i Nieto; Carles Ventura; Jordi Pont-Tuset; Silvia Cortes; Ferran Marqus;. **System architecture of a web service for Content-Based Image Retrieval**. *Proc. of the conference on Image and Video Retrieval, CIVR 2010*, 2010. 71