Proceedings of the Federated Conference on Computer Science and Information Systems pp. 261–270
DOI: 10.15439/2016F541
ACSIS, Vol. 8. ISSN 2300-5963

# Caption-guided patent image segmentation

Jerzy Sas, Urszula Markowska-Kaczmar
Wroclaw University of Technology
Faculty of Computer Science and Management
Wyb. Wyspianskiego 27, 50-370 Wroclaw, Poland
Email: jerzy.sas, urszula.markowska-kaczmar@pwr.edu.pl

Anastasia Moumtzidou
Information Technologies Institute,
Centre for Research and Technologies - Hellas,
6th km Charilaou - Thermi, 57001, Thessaloniki, Greece,
Email: moumtzid@iti.gr

***Abstract*—The paper presents a method of splitting patent drawings into subimages. For the image based patent retrieval and automatic document understanding it is required to use the individual subimages that are referenced in the text of a patent document. Our method utilizes the fact that subimages have their individual captions inscribed into the compound image. To find the approximate positions of subimages, first the specific captions are localized. Then subimages are found using the empirical rules concerning the relative positions of connected components to the subimage captions. These rules are based on the common sense observation that distances between connected components belonging to the same subimage are smaller than distances between connected components belonging to various subimages and that captions are located close to the corresponding subimages. Alternatively, the image segmentation can be defined as a specific optimization problem, that is aimed on maximizing the gaps between hypothetical subimages while preserving their relations to corresponding captions. The proposed segmentation method can be treated as the approximate solution of this problem.**

## I. Introduction

BEFORE inventors prepare a patent application they should spend some hours doing a good search for patents that are related to their idea. Usually, searching for patents, they prepare phrases that in the best way describe the core concept. However, the list of results often contains hundreds or even thousands of patents depending on the popularity of the term or phrase selected. It also happens that one thing is described with different names and labels. Therefore, the results obtained are not always relevant.

Many specific tools exist that support patent databases searching ([4]), but in most cases they are mainly based on textual analysis. It is worth noticing however, that patent drawing is almost always required to illustrate the invention. Frequently, patents include numerous drawings showing a variety of views. Therefore, it seems that patent search results would be much more relevant, and it would be much easier to access the patent if a query considered illustrations included in patent documents. This observation leads the concept of content-based patent image search ([12], [13]). When applying content-based search paradigm, patent searchers are browsing thousands of patents looking only on images contained in drawings section. This task can be accelerated by using patent image search engines, which can retrieve images, based on their visual content. The importance of images in patent search can be further emphasized by the fact that images are both language independent and independent of the scientific terminology that may evolve over the years. They are also important in attempts to understand a patent.

Usually, a patent includes several figures or drawings showing how an invention looks. Drawings in patents can contain reference numerals that are used in the detailed description to identify parts of the drawings and to draw reader's attention, but they introduce difficulties in the case of image segmentation.

It happens that the invention is compound, and the patent document shows drawings of one or more parts. The drawings in patents are specific and differ much from other illustration in other types of documents.

A variety of styles can be encountered in patent images, e.g., surface shading, plots, pattern area fills, broken lines and varying line thickness. However, in most cases, patents include some views prepared as black-and-white line art. The drawings can be made using different techniques. Sometimes, CAD tools are used, but there are also numerous old patents with drawings prepared manually with ink. In many cases, component subimages are not separated by distinct wide areas of background, so naive approach based merely on the segmentation by wide background bands fails. The example of a compound patent image consisting of many subimages is presented in Fig.1.

Thus, to develop an image content based patent search engine, it is necessary to employ techniques to identify the number and the position of the figures on the page to isolate them. To this end, an efficient segmentation of the page in its figures is required. Its accuracy will determine to a large degree the performance of image patent search process. Our research is focused on drawings segmentation from patent documents.

All these mentioned above features cause that images and subimages segmentation in patents is a challenging task and methods developed in other areas, e.g. for segmentation of color or gray shade images included in journals are not appropriate in the application are being considered here.

The paper is organized as follows. Section II contains a short review of other works related to compound image segmentation into parts. In the next two sections, the segmentation

This work was supported by the statutory funds of the Department of Computational Intelligence, Faculty of Computer Science and Managament, Wroclaw University of Science and Technology and partially by EC under FP7, Coordination and Support Action, Grant Agreement Number 316097, ENGINE European Research Centre of Network Intelligence for Innovation Enhancement ( http://engine.pwr.edu.pl/). All computer experiments were carried out using computer equipment sponsored by ENGINE project.

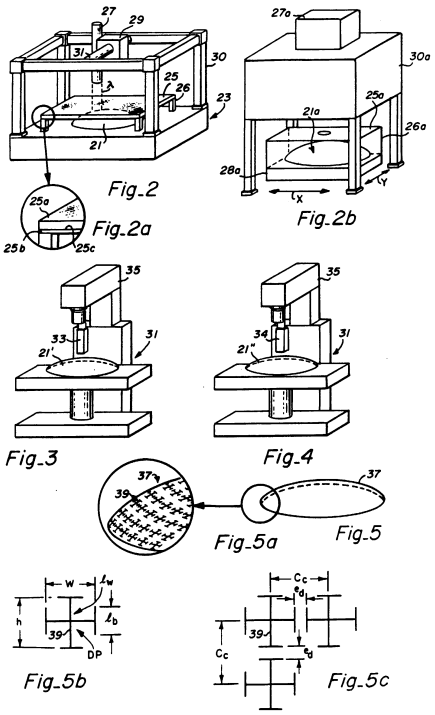


Fig. 1. Example of patent image consisting of many captioned subimages

problem is defined, first intuitively and then more formally. Sections V-A and V-B describe two possible algorithms of segmentation that we tested. In Section VI the specific method that can be applied to simple grid-layout images segmentation is presented. The method of caption detection used here, based on text extraction with OCR support is shortly described in Section VII. The method of automatic evaluation of human-defined and automatic segmentation consistency is explained in Section VIII. Experimental results obtained with proposed segmentation methods on a patent images benchmark database are presented in Section IX. Finally, some conclusions and suggestions for further works are formulated in Section X.

## II. Related Work

The idea of searching documents on the basis of figures included in documents is not new. For instance, Lu et al. in [7] proposes to utilize the content of figures in searching scientific literature in digital libraries. The work described in this paper is only focused on the categorization of figures included in scientific documents. Authors developed a machine learning based method for document image categorization. A figure is categorized as a photograph or a non-photograph. A non-photograph is then further classified as plot, diagram, or another graph. The aim of a method presented in [6] is similar - image classification using machine learning methods. The authors also propose numerical data extraction from pictures.

Although the idea of figure content based document retrieval exists since many years, there are still many challenging tasks to implement. The problems attract the attention of researchers. It is important to detect figures in documents and separate them from a text. The exemplary approaches can be found in [9], [3], [1], [10]. Some of the figures are composed of subfigures. Their separation is also a challenge [2]. The next problem is a classification of figures to the defined groups. Research devoted to image content understanding is a rapidly developing area.

Considering the area of our research in this short survey, we have particularly focused on image and subimage extraction (segmentation). The paper [2] presents a technique of compound image separation. It is based on systematic detection and analysis of uniform space gaps. The method assumes the separation of subimages by thin horizontal or vertical uniform space that separate compound figures from a major part of compound figure images.

The paper [5] describes a solution to the similar problem but located in biomedical literature. Articles in this area are often composed of multiple subfigures and may illustrate diverse methodologies or results. This solution is similar to our approach in that it also is based on capture recognition. Their method first analyzes figure captions to identify the label style used to mark panels, then determines the panel layout, and finally, each figure partition into panels is performed. To identify the number of panels, authors developed three stage procedure. First, a simple lexical analysis is made to determine the potential panel labels in the captions. Then, the list of potential panel labels is analyzed in order to identify and remove false positive ones. Finally, the segmentation of captions according to the set of identified panels is carried out. In the next step, the image processing is executed. First, the optimum threshold value for segmenting the figure is computed. The text embedded in images is also detected in order to find panel labels. Next, the number of panels in the figure is determined. Then panels are partitioned it into a set of panel-subcaption pairs, on the basis of the set of subcaptions, panel labels, and connected components. To partition the figure, they first create a node for each recovered panel label and then create an edge connecting each node to its closest horizontal and vertical neighbors. It means that the method assumes the specific horizontal or vertical relative position of subimages.

Another method of figure classification and subimage extraction is that described in [14]. It also refers to the biomedical area. The method of subimage extraction retrieves subimages using reconstruction from Hough peaks.

Comparing to the presented methods, in the case of patent documents, subdrawings rarely have easy to detect vertical or horizontal separating spaces. Therefore it is a much challenging task.

## III. Problem formulation

Let us consider a binary black and white image. Black pixels are assumed to represent drawn lines, captions and inscribed text (foreground), while white pixels constitute background. The image consists of subimages, where each subimage is associated with its individual text caption. We assume that

captions can be reliably detected in earlier stages of the image segmentation procedure. The applied method of caption detection is described shortly in section VII. So, at the current stage, we know the number of subimages, which is equal to the number of detected captions. If no caption is detected, then it can be assumed that the whole image constitutes the single component, unless there are sufficiently wide background areas separating clouds of foreground pixels. In the later case, one of segmentation methods based merely on separation by background (described in the previous section) can be applied. We will not deal with such cases in this paper.

We are considering here only such images where the set of detected captions is not empty. Our aim is to subdivide the set of all foreground pixels into disjoint subsets constituting subimages associated with captions in such a way, that it corresponds to the image author intent and to the intuition of a human observing and interpreting the image.

Unfortunately, in practice, there are no strict rules followed when drawing compound images, so depending on the degree of the image complexity, shape of components and the logical (semantic) relation between them, the subimages may be arranged on the image plane quite freely. For this reason, it is hardly possible to define the formal segmentation principle, so that it always corresponds to the intent of the image creator. Nevertheless, usually, some basic principles are applied when arranging the layout of the compound image. The intuitive rules typically used are as follows:

- subimages are separated by relatively wide areas of background,
- in most cases, subimages are not connected by foreground elements; if it is not the case, only few simple lines connect subimages (the example can be a single line connection between subimages captioned *"Fig.2"* and *"Fig.2a"* or *"Fig.5"* and *"Fig.5a"* in the drawing in Fig.1,
- each subimage contains at least one "dominant" element consisting of foreground pixel which size is comparable or greater than the size of the caption,
- captions are close to elements of the subimage associated with them,
- in certain cases, the subimages are regularly arranged in a grid-like manner, where subimages are clearly separated by horizontal and vertical bands of foreground pixels, which often include subimage captions (an example of such layout is presented in Fig.2.

Images complying the latter principle can be segmented very easily and also such a regular layout is easily distinguishable. The method proposed here first tries to detect whether or not the image being analyzed a case of a regular layout. Regular images are segmented with a specific (easy) method and excluded from further considerations. For other images, we apply the segmentation procedures based on remaining intuitive rules.

In the approach described in this paper, we follow these informal rules to build the clustering method, which groups foreground elements of the image into subsets associated with individual captions. The method of image segmentation

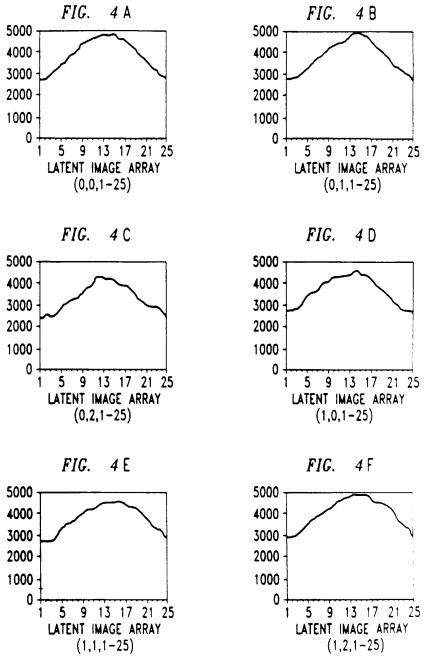


Fig. 2. Example of the regular grid-like layout of subimages

described here; clusters connected components of foreground pixels into sets representing captioned subimages. It means that, in most cases, each subimage consists of entire connected components. The only exception from this rule is where there exists a substantial evidence that some subimages share commonly connected components. In such situation, some connected components are split. We will start with defining the notion of connected components more formally, and then some proposals of connected components clustering will be described.

## IV. Finding Subimages by Connected Components Clustering

Let us define the connected component (*CC*) of the image as the set of foreground pixels that are linked by other foreground pixels belonging to the same CC. A pixel $p$ will be here represented by the pair of its coordinates:

$$p_i = (x_i, y_i), x_i \in \{0, ..., x_{res}-1\}, y_i \in \{0, ..., y_{res}-1\}, \quad (1)$$

where $x_{res}, y_{res}$ denote the image horizontal and vertical resolution. Let $\mathcal{F}$ denotes here the set of all foreground pixels in the image being segmented. We divide the set $\mathcal{F}$ into the set of disjoint CCs:

$$C = \{c_1, ..., c_M\}, c_i \subseteq \mathcal{F}, c_i \cap c_j = \emptyset, \bigcup_{i=1}^{M} c_i = \mathcal{F}. \quad (2)$$

The connected component $c$ is the set of pixels such that if two pixels $p_a, p_b \in c$ then there exists the sequence of pixels $(p_a = p_1, p_2, ..., p_k = p_b)$, all of them belonging to $c$ that

for each pair of pixels $p_i, p_{i+1}, i = 1, ..., k-1$, pixels $p_i$ and $p_{i+1}$ are direct neighbors. We consider two pixels to be direct neighbors if they are *8-connected*, i.e. they have common edge or corner. A CC is maximal if no more foreground pixels can be attached to it. Further on, we will be considering *maximal CCs* only. We define the distance $d(c_i, c_j)$between two CCs $c_i$ and $c_j$ as:

$$d(c_i, c_j) = \underset{p_l \in c_i, p_k \in c_j}{min}( \quad e(p_l, p_k) \quad ), \tag{3}$$

where $e(p_l, p_k)$ denotes Euclidean distance between pixels in 2D. Similarly, we can define the distance between two sets (clusters) $s_i, s_j$ of CCs:

$$d(s_i, s_j) = \underset{c_l \in s_i, c_k \in s_j}{min}( \quad d(c_l, c_k) \quad ). \tag{4}$$

Our aim is to split the whole binary image into a set of subimages $S_A$, where each subimage $s_i \in S_A$ is the set of CCs (we will call it *cluster*), $s_i \subseteq C$. We also assume that clusters are disjoint and they all sum up to the whole image, i.e. each foreground pixel from $\mathcal{F}$ belongs to exactly one cluster (or in other words - subimage). The partitioning of the image into clusters should be as close as possible to the intent of the image creator. Initially, we make the assumption that each CC belongs entirely to a single subimage, which may be not true in some cases (as shown in Fig.1). We will deal with such cases later and will propose a method of connected component splitting. By taking into account the set of intuitive rules given in the previous subsection we assume that: a) subimages consist of graphical elements (corresponding to CCs) that are located close each to other within the single subimage and b) disjoint subimages are separated with relatively large bands of background (white) pixels. Additionally, we assume that each subimage is associated with its individual caption. We assume that captions are reliably detected in earlier stages of the image segmentation procedure. So, at the current stage we know the number of subimages, which is equal to the number of detected captions. It seems also reasonable to make assumption, that caption areas are located close to the corresponding subimage. The problem is therefore how to split connected components set into disjoint subsets such that the obtained partitioning corresponds to subimages, as a human perceives it.

Formally, the presented intuitive assumptions correspond to finding such clustering of connected components, where the number of clusters is fixed (and is equal to the number of detected captions $k$) and the minimal distance between clusters of CCs is maximized. The graphical elements constituting captions (characters and their graphical elements) are being considered here as ordinary foreground image elements. We assume that each caption (as a graphical element) is entirely included in a single cluster and each cluster includes exactly one caption. The elements of a caption are being considered as a single indivisible CC (even though actually a caption consists of many "true" CCs). Let $\Gamma$ denotes the set of all possible partitionings of the set $C$ into $k$ clusters $\{s_1, s_2, ..., s_k\}, s_i \subseteq C$, which satisfy the above restriction. By $s$ we denote here the cluster of CCs. We need to find such "optimal" clustering $\gamma^* \in \Gamma$ that:

$$\gamma^* = \arg\max_{\gamma \in \Gamma}( \underset{s_i, s_j \in \gamma}{min} d(s_i, s_j)), \tag{5}$$

where $d(s_i, s_j)$ is the distance between clusters defined in equation 4. Because the number of possible partitioning in $\Gamma$ is very big if the number of CCs is high (equal to the Stirling number of the second kind that determines the number of possible partitioning of $n$-element set into $k$ nonempty subsets) the problem is computationally hard and a suboptimal solution must be applied.

### A. Connected component splitting

In certain subimages, the single large CC may span many subimages as shown in Fig. 1, where subimages 2 and 2a or 5 and 5a share a common CC. Any method based on complete CCs clustering cannot retrieve the correct segmentation in cases like this. The selected CCs need to be split into smaller ones, so that the principle of composing subimages from complete CCs can be still used. The method applied there consist of: first, carying out the segmentation procedure using the original CCs set, detecting CCs that possibly need to be split, splitting them into smaller CCs and executing the clustering procedure again. In the second clustering, the new set of CCs is used, where original large ones are replaced by their parts.

*1) Detection of CCs that need to be split:* The CCs that are "suspect" of spanning between adjacent subimages are detected by examining the position of sufficiently large CCs, with relation to the nearest captions. The detection is performed after initial clustering with original CCs. In this way we can consider only these captions that are close enough to obtained clusters. If two captions are close to a large CC then, instead of having only one cluster containing this large CC, we should split the large CC and allocate its parts to two smaller clusters labeled by captions close to "suspect" large CC. CC splitting seems especially adequate if two candidate captions are closer to this CC than to other clusters.

Let $l(C)$ denotes the caption $l$ assigned to the cluster $C$. By $L_C$ we denote the set of captions being candidates for captions assigned to subimages obtained by possible split of the cluster $C$ into smaller ones. The set $L_C$ consists of: a) the caption $l(C)$ assigned to $C$ by the primary clustering carried out using the original CCs found in the image and b) other captions, which distance to $C$ is comparable to the distance between $C \backslash \{l(C)\}$ and $l(C)$. Only these captions are included in $L_C$ which are not "strongly bound" to other clusters. We assume here that $l(C')$ is strongly bound to $C'$ if its distance to $C'$ is much lower than the distance of any other caption to $C'$. The threshold of distances used for classifying the caption as "strongly bound" can be determined experimentally using the set of validation images.

The captions collected in $L_C$ are then used to detect sufficiently large CCs $c \in C$ that possibly span many actual subimages. For simplicity, let us consider the pair of captions

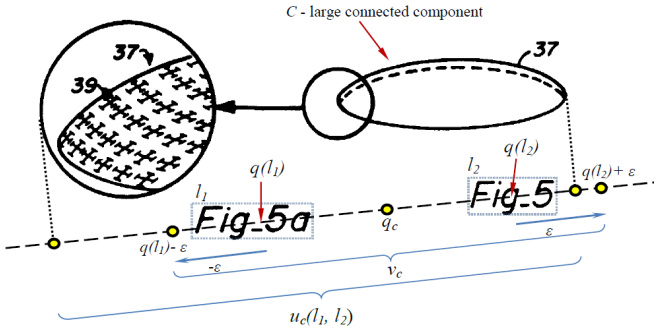


Fig. 3. Evaluation of a CC as a candidate for split with respect to two captions

$l_1, l_2 \in L(C)$. They define the line containing the centroids of captions $q(l_1), q(l_2)$ and the line segment with end points at $q(l_1)$ and $q(l_2)$. Each CC $c \in C$ can be projected onto the line $(q(l_1), q(l_2))$. Because, by definition, each CC is compact then its projection onto the line also constitutes the compact line segment $u_c(l_1, l_2)$. We consider the CC $c \in C$ to be a candidate for split if $u_c(l_1, l_2)$ covers the center of the line segment $q_c = (q(l_1), q(l_2))$ and it is sufficiently long. In our experiment we assumed the minimal length of $u_c(l_1, l_2)$ to be at least $0.75 * \|q(l_1) - q(l_2)\|$. In this way, for a pair of captions in $L_C$ we can selected some CCs in $C$ that are candidates for splitting. If there are more than two elements in $L_C$ then they can be ordered into the sequence $(l_1, l_2, ..., l_m)$ so that the path $(q(l_1), q(l_2), ..., q(l_m))$ is the shortest. Hence we obtain pairs of captions that are close each to another and possibly are assigned to adjacent subimages. Then for each pair of captions $(l_i, l_{i+1}), i = 1, ..., m-1$ adjacent in the ordered sequence, the described above procedure is carried out resulting is some CCs as candidates for splitting. The idea of finding CCs for splitting in presented in Fig. 3.

*2) Finding CC split point:* The next element of the proposed procedure is finding the split point for a CC that has been selected as a candidate for split with respect to two captions $l_1$ and $l_2$. We want to split it into two parts which are close to captions $l_1, l_2$ and so that the minimal number of connections between resultant components exists. The later assumption follows from the observation that in cases where the large CC needs to be split, it consists of fragments connected by a few lines, most often - by just a single one. We applied the method where the boundary between components of a CC is the straight line which satisfies the following conditions:

- it is perpendicular to the line defined by $(q(l_1), q(l_2))$,
- it crosses the extended line segment $v_c = (q(l_1) - \epsilon, q(l_2) - \epsilon)$, where $\epsilon = (q(l_2) - q(l_1))/4$,
- it minimizes the number of split lines of the skeletonized image of the CC being considered; if there is more than one position of the split line where this minimum is reached then the position closest to the center of the line segment $(q(l_1), q(l_2))$ is selected.

If the described above CC splitting procedure creates at least one subdivided CC then clustering is being carried out again using the set of modified CCs.

## V. Image Segmentation by Connected Components Clustering

We propose two methods to find suboptimal solution of the problem defined in eq. 5. The first one is based on human intuition that binds subimage elements to the close captions and initially focuses attention on big graphical elements. The second method is typical k-means algorithm application to the set of CCs. The only introduced constraint is that the components corresponding to captions cannot be assigned to the same cluster.

### A. CCs clustering by human perception imitation

In this method we follow the human way of reasoning when dividing the image into subimages. Later on, we will call it *intuitive segmentation*. Although precise way of human reasoning is unknown, it seems that when splitting an image into subimages, humans proceed as follows. Initially, we seek for big graphical components that are located close to captions. They become cores of further activities. If a smaller element is completely surrounded by closed shapes of big components already associated with a caption, a human tends to assign it to the same caption. All remaining smaller elements are assigned to these already constituted subimages to which the distance is smallest. The detailed procedure that can be applied is defined as Algorithm 1.

### B. CCs clustering by k-means

k-means algorithm is a widely known and successfully applied method of clustering, so we will not describe it here in details. Its primary description can be found in [8]. In order to apply it to our problem, we need to specify how the initial clustering is obtained and how centers of clusters are determined. Initial clustering is obtained creating $k$ clusters that initially contain CCs representing captions. Remaining CCs are assigned to initial clusters so that the including cluster corresponds to the caption closest to a CC. We mean here the distance computed as in equation 3 where in place of $c_j$ the single pixel is used that is the center of the caption bounding box. The core of k-means algorithm is the computation of cluster means and computing the distance of elements being clustered to cluster means. Here we are clustering CCs (not individual foreground pixels). Therefore two methods of cluster center positions computing can be proposed:

- a cluster center is the centroid of all equally weighted foreground pixels belonging to CCs in the cluster,
- a cluster center is the weighted mean of bounding box centers that enclose CCs belonging to the cluster, the component weights can be based either on the size (maximal length along $x$ and $y$, or the bounding box area) or on the number of pixels belonging to a CC.

Both variants were evaluated experimentally and the better one is finally recommended. Details are presented in Sec.IX.

**Algorithm 1** "Intuitive" CC clustering

**Require:** $C$ - set of all connected components; $L$ - set of captions; ($L \subseteq C$)

sort CCs by size in decreased order;
create the family $G$ of initially empty sets of CCs assigned to captions;
**while** empty sets in $G$ exist **and** there exist unassigned CCs **do**
  get the first biggest CC that is not assigned to any caption;
  assign it to the set in $G$ corresponding to the closest caption;
**end while**
**if** there exist empty sets in $G$ **then**
  **for** all captions with no CCs assigned **do**
    find CC $c$ closest to unassigned caption which is not the only element of the family in $G$ that contains it;
    remove $c$ from the set in $G$ that currently contains it;
    assign $c$ to the current caption;
  **end for**
**end if**

/* Now we have big CCs assigned to captions */

make initial clusters of CCs assigned to labels so far;
**while** not all CCs are assigned to clusters **do**
  get the next unassigned CC;
  find the cluster closest to it;
  assign CC to the closest cluster;
**end while**

**return** ($G$) - the family of CC sets assigned to individual labels

**Algorithm 2** "CC clustering with modified k-means"

**Require:** $C$ - set of all connected components; $L$ - set of captions; ($L \subseteq C$)

$G \leftarrow$ the family $k$ initial clusters containing individual captions from $L$;
compute the centroid position for each cluster in $G$;
assign all CCs in $(C \setminus L)$ to clusters in $G$ using minimal distance to the centroid as a criterion;
$G' \leftarrow G$;
**repeat**
  $G \leftarrow G'$;
  compute the centroid position for complete clusters in $G$;
  $G' \leftarrow \{g_i : i = 1, ..., k; g_i = \emptyset\}$;
  **for all** $c \in C \setminus L$ **do**
    $g^* \leftarrow \underset{g \in G}{argmin}(X(g), c)$;
    $I(g^*|G, G') \leftarrow I(g^*|G, G') \cup \{c\}$;
  **end for**

/* Now we have all non-caption components initially clustered */

sort clusters in $G'$ in descending order of their sizes;
$L' = L$;
**for all** $g \in G$ in decreasing order of size **do**
  $l^* \leftarrow \underset{l \in L'}{argmin}(d(g, l))$;
  $g \leftarrow g \cup l^*$;
  $L' \leftarrow L' \setminus \{l^*\}$;
**end for**
**until** $G' \leftarrow G$ - no change in clustering
**return** ($G'$) - the family of CC sets assigned to individual labels

In order to preserve the restriction that each cluster contains exactly one CC representing a caption, the typical k-means algorithm must be modified. In the initial partitioning, this restriction is obviously satisfied, provided that it is established according to the recipe described above. In the k-means algorithm phase, where cluster centroids are computed, all CCs assigned to a cluster in the previous iteration are taken into account. However, when the new clusters are being build in the next iteration only CCs not being elements of captions are a subject to be moved to the cluster with nearest centroid. In this way, new clusters are constituted, which do not contain CCs representing clusters. The caption CCs are then uniquely assigned to these clusters. The assignment is achieved by considering created clusters in decreasing order of its size (biggest first) and assigning such already unassigned cluster to the current cluster for which the distance to a cluster is minimal. Finally, centroids of new clusters that include captions are computed and the next k-means iteration starts. The modified algorithm is presented in Algorithm 2. In the algorithm the following symbols are used: $d(\alpha, \beta)$ is the distance between elements $\alpha$ and $\beta$, $L$ is the set of CCs that are graphical elements constituting captions (each caption is treated as a single CC), $X(g)$ is the centroid of the cluster $g$. Let $G$ and $G'$ denote two families of CC sets, such that in both $G$ and $G$, each caption $l \in L$ belongs exactly to a single set in $G$ and $G'$. $I(g|G, G'), g \in G$ denotes the element from $G'$ that contains the same caption $l \in L$ as $g$.

## VI. Segmentation of grid-like composed images

In certain cases subimages are located in a compound image, so that they create grid-like layout as presented in Fig. 2. Correct segmentation is quite easy in such cases and it can be carried out using a simplified method. Sometimes, general methods described in preceding sections fail in grid-like layouts, so it seems reasonable to apply specific method to this class of images which very often leads to the segmentation consistent with an image creator intent.

By grid-layout of subimages we mean here such a placement of subimages where

- captions can be enclosed by narrow horizontal bands spanning from left to right edge of the image, which do not contain significant elements of subimages and all CCs located in caption strips can be separated by the path consisting only of background pixels that connects left and right edge of the image;
- the subimages are consistently located either under or above their captions - it means that there are no significant CCs either below the lowest caption band or above the highest caption band. In result, horizontal areas between caption bands can be uniquely assigned to their caption bands;
- if there are more than one captions in a caption band then all CCs in the corresponding image band can be separated into as many clusters as the number of captions in the caption band. The separation areas are vertical areas consisting only of background pixels.

The above conditions can be easily tested programmatically. The testing procedure immediately provides the segmentation of the image, which appears to be very reliable. In our approach, each image being segmented is subject to grid layout test before other segmentation methods are tried. If the test passes then final segmentation is a byproduct of the grid layout testing procedure. If the grid layout test fails then the image is passed to general segmentation procedure based on concepts presented in preceding sections.

## VII. Caption localization

Text extraction methods are used in order to find subimage captions. In this work we used the method described in [11]. The procedure consists of two phases. In the first phase, some rectangular areas are detected which are likely to contain (any) text inscribed to the image. In the second phase, areas recognized as containing text are tested for occurrence of specific text patterns being the actual captions.

*1) Detection of text areas in the image:* The first phase consists in turn of three stages. In the first stage, the procedure finds candidate areas that possibly contain texts, by grouping small CCs into rectangular areas of shapes and sizes typical for areas including individual words of text inscribed into the image. In the next stage, candidate areas are passed through the classification procedure which classifies them as "text" or "non-text" using the set of features related to distribution of foreground pixels and line segments within the candidate area. Areas recognized as "text" are passed to OCR module running in no-dictionary mode. Finally, only such candidate areas are considered to be text areas, for which the results of OCR recognition satisfy an empirical criterion related to the ratio of untypical characters occurrence in the recognized textual string.

Initially, connected components (CCs) that possibly enclose individual characters or their sequences are found. The series of criteria were experimentally elaborated that must be satisfied in order to reject CCs that are not likely to contain text characters. All used criteria are discussed in [11]. One of the most important criteria appeared to be the ratio $f_i^{(hv)} = h_i/v_i$ of the height of the shape contained in CC $h_i$ to the average line width $v_i$ in the $i$-th CC. If the ratio value is out of the interval covering the range typical for text areas in images, the test fails and the area is not further considered as a candidate area. The acceptance interval is determined by considering text areas appearing in the validation set of patent images, for which "true" text areas were manually annotated. By analyzing the set of manually confirmed text areas in the validation set, the histogram of $f_i^{(hv)}$ has been created. As the acceptable range of $f_i^{(hv)}$ we assumed the interval covering 98% of values encountered in the validation set, leaving aside 1% of very small and 1%of very high text areas as outliers. Other parameters of criteria used in candidate text areas selection were established in the similar way.

The candidate areas that passed two first stages of text extraction are finally subject to OCR that runs in no-dictionary mode. The result of OCR recognition is first used as the data for the last stage of text area detection. It has been observed that in the case of "false" candidates that contain no text, the string created by OCR includes many punctuation characters (like . , ; : -). The final criterion of a candidate area acceptance as the text area (no necessarily the area of caption) is the ratio of punctuation characters occurrence in the whole recognized string. Details are described in [11].

*2) Selection of text areas containing caption patterns:* If the OCR module were absolutely accurate then caption detection would be a trivial problem. It would be possible to simply compare the OCR results to one of predefined text strings that can be a caption. However, in patent images, inscribed textual elements are hard to recognize. In most cases, the main reason is that images are hand-drawn, so is the included text. While OCR works well with machine printed text, its performance seriously deteriorates when it is presented with handwriting or hand-printed text. Therefore, it seems reasonable to consider as captions not only areas where OCR produced exact caption patterns, but also to accept such ones, where the recognized sequence is in some sense similar to one of acceptable caption patterns.

The idea of recognizing the text area as a caption is based on finding in the sequence recognized by the OCR module, the subsequence that is similar to one of *caption patterns*: "FIG" "fig" "Fig". The visual similarity of characters in the recognized subsequence to the corresponding characters in these patters is taken into account. It may happen that the recognized sequence is a correct word containing the pattern as a subsequence (e.g. "conFIGuration", "FIGhter"). In order to exclude such texts from considerations, first the result of recognition is compared with the list of correct words containing the pattern as a subsequence. Only words longer that 4 characters are used. If the length of the sequence recognized by OCR module is longer than 4 characters, then the minimal edit distance to words in the dictionary is found with Levenshtein algorithm. If the edit distance is less than one fourth of the number of characters in the closest word, then the label is rejected as rather being the part of ordinary

word appearing in the image.

If the OCR recognition result was not rejected by the dictionary test then the similarity to caption patterns is computed. The procedure finds a three-character substring in the recognized sequence that is most similar to patterns. When computing the similarity, we multiply similarity factors for the individual characters in a pattern. The similarity measures should be specific to properties of used OCR module and used fonts. The similarity factors can be computed as a relative frequency of errors consisting in replacing the actual character $c_a$ by another erroneous character $c_r$: $n(c_r|c_a)/n(c_a)$. $n(c_r|c_a)$ is the count of errors consisting in replacing $c_a$ by $c_r$ and $n(c_a)$ is the number of $c_a$ occurrences. In our experiments we however used similarity factors based on visual intuitive similarity of various character shapes. We assumed than nonzero similarities are given only to the following sets of characters recognized by OCR:

- for F : F E f P
- for I : I l i 1 J L T
- for G : G C 6 c
- for f : f t
- for i : i j 1 l
- for g : g 9

For remaining characters we assume that the similarity is equal 0.0.

Let us consider the character sequence $(c_1, c_2, ..., c_m)$ returned by OCR module running in no-dictionary mode. For each subsequence consisting of three consecutive characters $s^{(3)}(j) = (c_j, c_{j+1}, c_{j+2})$, the similarity to the three-characters pattern sequence $f$ is defined as:

$$\Delta(s, f) = \prod_{i=1}^{3} \delta(s_i, f_i), \tag{6}$$

where $s_i$, $f_i$ is the $i$-th character in the sequence, $\delta(a, b)$ is the similarity between characters $a$ and $b$. Here we assume that $0.0 \leq \delta(a, b) \leq 1.0$ and $\delta(a, a) > \delta(a, b)$ for all $a, b; a \neq b$.

The final likelihood that the sequence $s$ represents a caption is computed as:

$$Q(s) = \max_{j=1,...,m-2} \max_{f \in \Phi} \left(\Delta(s^{(3)}(j), f)\right), \tag{7}$$

where $\Phi = \{"fig", "Fig", "FIG"\}$ is the set of caption patterns. In order to make the final decision whether or not the text area passed to OCR is the caption, the computed similarity is compared to a threshold. The lower is the threshold value, the more actual captions are recognized but also the likelihood to accept "false" captions increases. On the other hand - the higher it is, the higher is the likelihood that an actual caption will be omitted. The threshold value is determined as a metaparameter. The threshold should be set depending on he losses following form two types of caption recognition errors (i.e. rejection of true caption and false acceptance of non-caption text as a caption). The threshold value was fixed using the validation set in images containing captions and non-caption texts. Details are described in Section IX.

## VIII. Automatic segmentation evaluation

In order to evaluate the accuracy of the automatic segmentation, the results obtained automatically need to be compared with "ground truth". By ground truth we mean here the results of manual segmentation of images in the test set, done by humans We assume that each manually defined segment has at most one automatically created counterpart that most closely matches it. Each automatic segment can be then used at most once as a counterpart of a certain manual segment. Some automatic segments may remain unassigned to any manual segment, as well as some manual segments may have no corresponding automatic segments. More formally, let us denote the set of manual and automatic segments by $S_M$ and $S_A$. At the first stage, we define the function $f(s) : S_M \rightarrow S_A + \{\phi\}$, where $\phi$ denotes here "no automatic segment matches the segment $s$". The function $f(s)$ can be constructed in such way that maximizes the matching measure between elements of $S_M$ and $S_A$. In the case of small number of elements in these sets, the exhaustive search that tries all possible mappings can be applied. In other cases, suboptimal procedures of $f(s)$ construction must be applied.

Having $f$ function defined, we can evaluate the matching between manual and automatic segmentation by evaluating the matching defined by $f(s)$ for each segment $s \in S_M$. Popular $F1$ measure is used to evaluate the matching. Let for some $s \in S_M$ we have $a \in S_A \cup \{\phi\}$. We treat here $s$ and $a$ as sets of foreground pixels. If $a = \phi$ then $a$ is assumed to be the empty set. The pixels in $s$ are "relevant" elements, while the pixels in $a$ are "retrieved" elements. The precision and recall can be then computed as

$$prec = \frac{\mid s \cap a \mid}{\mid a \mid} \tag{8}$$

$$rec = \frac{\mid s \cap a \mid}{\mid s \mid} \tag{9}$$

$$F1 = 2\frac{prec * rec}{prec + rec} \tag{10}$$

In the case when $\mid a \mid = 0$ (no automatic subimage is assigned to the manual one) we assume that $F1 = 0$. The $F1(s)$ score is computed for each $s \in S_M$. The overall automatic segmentation accuracy for the whole image is calculated as:

$$F1_T = \frac{\sum_{s \in S_M} \mid s \mid * F1(s)}{\sum_{s \in S_M} \mid s \mid}. \tag{11}$$

Thus, bigger subimages are assigned higher weights than smaller ones in the overall evaluation. In order to compute the assessment for the collection of images, the measures $F1_T$ computed for individual images are averaged.

## IX. Experimental results

For the sake of drawings segmentation testing we used the subset of images collected in *PATExpert* project conducted by ITI CERTH institute and available at http://mklab.iti.gr/project/patentbase web page. The image set is described in [13]. Images in the database are scanned images of manually

created drawings. Only a few of them seem to be created with drawing software. Also in these cases they were printed and the image was finally acquired by scanning. Near-duplicates (i.e. identical or almost identical images) included in the original database were excluded.

For experiment purposes, 1461 images containing at least one caption were selected from *PATExpert* database. Each selected image was manually segmented into subimages and caption areas were marked. In such a way, we obtained "ground truth" information about the actual segmentation and localization of captions. The set of images was divided into two parts: a) the subset of images containing exactly one caption (1135 images) and b) the set of images containing multiple captions (296 images). 600 images from the part a) were assigned to the validation set. It was used for caption detection metaparatemers tuning. Remaining images from the set a) and images from the set b) were used as the testing set.

The experiment carried out consists of two phases. In the first phase we evaluated the accuracy of caption area recognition. The second phase was aimed on the assessment of the automatic segmentation consistency with the manual segmentation.

### A. Evaluation of caption recognition accuracy

The aim of this experiment was to set the metaparameters used in the caption detection algorithm and then to estimate the performance of the proposed method. Tesseract OCR module with its default character set for English was used for extracted text recognition.

The validation set consisting of 600 images from the part a) was used for tuning the text area detection algorithm as described in [11]. It was also used to set the likelihood threshold $Q(s)$ defined by equation 7. The threshold value was set as the maximal value that accepts 98% of all true captions appearing in text areas in the validation set.

The test set for caption detection consisted of 535 remaining images from the part a) and all 296 images from the part b). They included total 1322 subimages with captions.

The caption recognition errors occurred in 168 of 1322 caption occurrences, so the caption detection error rate was 12.71%. The error caused by insertion of false captions occurred only in 5 images. In two cases, captions were erroneously recognized where actually they were parts of longer words inscribed in the image. Only three false captions were recognized in text candidate field that in fact contained graphic elements. 163 errors out of the whole number 168 were caption omissions. The reasons of all observed caption errors are summarized in the Table I.

It is evident that the most frequent reason of caption omissions is related to handwriting and unusual fonts. In future, some of this errors can be avoided by providing OCR module with samples of character shapes used in patent images. Validation set can be used for this purpose. Another possibility is to use OCR recognizer aimed more on handwriting. Tesseract program used in our experiments is trained on machine fonts, so it performs poorly in case of handprinted texts.

### B. Automatic segmentation assessment

For the sake of automatic segmentation, only these elements of the test set were used, for which all captions were recognized correctly. 267 images from the part b) passed caption detection test successfully and they were used as the test set in the next experiment.

First, the images of grid-like structure were selected using the procedure described in Section VI. The procedure is purely technical and very accurately selects grid-like layout. The method detected 152 grid-like subimages. The segmentation defined by caption bands corresponded exactly to the actual segmentation in 150 images from this group. Only in two images, small and probably well-tolerated inaccuracies occurred. They were caused by existence of subcaptions appearing on the opposite side of the caption than the side including the related subimage. The proposed procedure for grid-like layouts detection and segmentation based on grids can be therefore assessed as very accurate and useful in many cases.

Remaining 115 images from the test set were segmented by intuitive segmentation described and by the modified k-means algorithm described in Sections V-A and V-B. For methods evaluation we used F1-score as explained in Section VIII. We observed that the F1-measure computed in this way is related to the degree of inaccuracy of "true" and automatic subimage matching. The intervals of F1 values roughly correspond to the following inaccuracy levels:

- $F1 \in (0.99, 1.0 >$ - subimages match almost perfectly, the differences are not meaningful and concern only individual pixels and dot-like graphical elements;
- $F1 \in (0.97, 0.99 >$ - differences in matching images concern usually misplaced very small graphical elements, in most cases of minor importance for image understanding;
- $F1 \in (0.95, 0.97 >$ - small elements like individual characters or digits appearing in descriptions, pointing arrows, distant and not graphically connected elements are displaced; usually it does not impair the concept presented in a image;
- $F1 \in (0.85, 0.95 >$ - relatively big elements of subimages are misplaced, but the presented concept is usually still readable;
- $F1 \leq 0.85$ - major elements of subimages are misplaced or missing (i.e. - they remain unassigned to subimage), the presented concept is not readable.

TABLE I
CAPTION DETECTION ERROR REASONS

| Error reason | Number of occurrences |
|---|---|
| Strange but repeatable font | 35 |
| Handwritten captions | 66 |
| Oversegmentation of text fields | 43 |
| Caption patterns in ordinary words | 2 |
| Untypically rotated image | 7 |
| Falsely inserted captions | 3 |
| Unexplained | 12 |
| **TOTAL** | **168** |

TABLE II
$F1$ SCORE DISTRIBUTION FOR TWO PROPOSED SEGMENTATION PROCEDURES

| $F1$ **interval** | **Fraction of subimages** | |
|---|---|---|
| | intuitive segmentation | k-means segmentation |
| $F1 \in (0.99, 1.0>$ | 72.65 | 66.85 |
| $F1 \in (0.97, 0.99>$ | 11.87 | 10.22 |
| $F1 \in (0.95, 0.97>$ | 3.59 | 4.14 |
| $F1 \in (0.85, 0.95>$ | 5.52 | 6.91 |
| $F1 \leq 0.85$ | 6.35 | 11.88 |

Taking the above observations into account, we can assume that the segmentation fails if for at least one of subimages in the image, its $F1$ score is less than 0.95.

In order to evaluate the accuracy of automatic segmentation procedures and to compare them, $F1$ scores were evaluated for all automatically created subimages assigned to their "true" counterparts. Table II shows fractions of all tested subimages falling into $F1$ score intervals described above.

Overall performance of k-means segmentation is lower than the performance of intuitive segmentation. In the test, intuitive segmentation provided usable results for 88.13% of images, while k-means segmentation gave good results only in 81.20%. The main weakness of k-means based segmentation lies in its tendency to create unbalanced subimages, where one automatically determined subimage contains numerous big CCs, while others consist only of small components. In many cases it leads to segments that contain CCs actually belonging to various subimages. One of reasons of such situations is frequent appearance of empty clusters created in the course of the algorithm execution. The correction consists of forced assignment of some CCs to the empty cluster. The implemented cure consists in selection of the CC which is closest to the caption not assigned to any nonempty cluster. Sometimes it is a small CC, what can lead to the phenomena of unbalanced clusters.

The results of k-means segmentation is usually close to the human intent in the case of images containing many small, almost equally sized CCs. In such images, k-means often outperforms the intuitive segmentation. However, the k-means segmentation accuracy decreases significantly in images consisting of a few large CCs located close to each other.

## X. CONCLUSIONS AND FURTHER WORKS

In this paper we presented the complete procedure of patent image segmentation supported by caption detections. In both phases of the procedure (detecting captions and segmenting) we obtained the correctness at the level of about 90%. This result seems to be practically acceptable and makes it possible to recommend it for practical application in patent document processing.

In the future research we plan to introduce the third approach to clustering problem which is most closely related to the formal definition of the problem in equation 5. We are going to apply the simulated annealing method which seems well suited to our discrete optimization problem with a huge search space. . Caption detection method should be also improved to avoid some segmentation errors induced by true caption omission. Promising direction seems to apply 2D Fourier spectrum analysis in order to raise text area detection accuracy.

## REFERENCES


[1] K. Suchet Chachra, Z. Xue, S. Antani, D. Demner-Fushman, and G. R. Thoma. Extraction and labeling high-resolution images from pdf documents. In *Proc. SPIE 9021, Document Recognition and Retrieval XXI, 90210Q (24 March 2014);*, 2014. doi:10.1117/12.2042336.

[2] A. Chhatkuli, A. Foncubierta-Rodriguez, D. Markonis, F. Meriaudeaub, and H. Mueller. Separating compound figures in journal articles to allow for subfigure classification. In *Proc. SPIE 8674, Medical Imaging 2013: Advanced PACS-based Imaging Informatics and Therapeutic Applications*, 2013. doi:10.1117/12.2007971.

[3] C. Clark and S. Divvala. Looking beyond text: Extracting figures, tables, and captions from computer science paper. In *Scholarly Big Data: AI Perspectives, Challenges, and Ideas: Papers from the 2015 AAAI Workshop*, pages 2–8, 2015.

[4] D. Hunt, L. Nguyen, and M. Rodgers. *Patent Searching: Tools & Techniques*. Wiley, 2007.

[5] L. D. Lopez, J. Yu, C. O. Tudor, C. N. Arighi, H. Huang, K. Vijay-Shanker, and C. H. Wu. Robust segmentation of biomedical figures for image-based document retrieval. In *2012 IEEE International Conference on Bioinformatics and Biomedicine*, 2012. doi: 10.1109/BIBM.2012.6392706.

[6] X. Lu, S. Kataria, W. J. Brouwer, J. Z.Wang, and M. Prasenjit ůand C. Lee Giles. Automated analysis of images in documents for intelligent document search. *IJDAR*, 2009. doi:10.1007/s10032-009-0081-0.

[7] X. Lu, P. Mitra, J. Z. Wang, and C. Lee Giles. Automatic categorization of figures in scientific documents. In *Joint Conference on Digital Library, JCDL 06, USA.*, 2006. doi:10.1145/1141753.1141778.

[8] J. Macqueen. Some methods for classification and analysis of multivariate observations. In *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[9] P. A. Praczyk, J. Nogueras-Iso, and S. Mele. Automatic extraction of figures from scientific publications in high-energy physics. *Information Technology and Libraries*, pages 25–52, December 2013.

[10] M. Prasenjit S. R. Choudhury and G. Clyde Lee. Automatic extraction of figures from scholarly documents. In *DocEng '15 Proceedings of the 2015 ACM Symposium on Document Engineering*, pages 47–50, 2015. doi:10.1145/2682571.2797085.

[11] J. Sas and A. Zolnierek. Three-stage method of text region extraction from diagram raster images. In *Proceedings of the 8th International Conference on Computer Recognition Systems CORES 2013, Milkow, Poland, 27-29 May 2013*, pages 527–538, 2013. doi:10.1007/978-3-319-00969-8-52.

[12] A. W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(12):1349–1380, December 2000. doi:10.1109/34.895972.

[13] S. Vrochidis, S. Papadopoulos, A. Moumtzidou, P. Sidiropoulos, E. Pianta, and I. Kompatsiaris. Towards content-based patent image retrieval; a framework perspective. *World Patent Information Journal*, 32(2):94–106, 2010. doi:10.1016/j.wpi.2009.05.010.

[14] X. Yuan and D. Ang. A novel figure panel classification and extraction. *Int. J. Data Min. Bioinformatics*, 9(1):22–36, November 2014. doi:10.1504/IJDMB.2014.057779.