# Visual Signature based Identification of Low-resolution Document Images

Ardhendu Behera
Université de Fribourg
Chemin du Musée 3
1700 Fribourg
+41.26.429.66.78
Ardhendu.Behera@unifr.ch

Denis Lalanne
Université de Fribourg
Chemin du Musée 3
1700 Fribourg
+41.26.429.65.96
Denis.Lalanne@unifr.ch

Rolf Ingold
Université de Fribourg
Chemin du Musée 3
1700 Fribourg
+41.26.300.84.66
Rolf.Ingold@unifr.ch

## ABSTRACT

In this paper, we present (a) a method for identifying documents captured from low-resolution devices such as web-cams, digital cameras or mobile phones and (b) a technique for extracting their textual content without performing OCR. The first method associates a hierarchically structured visual signature to the low-resolution document image and further matches it with the visual signatures of the original high-resolution document images, stored in PDF form in a repository. The matching algorithm follows the signature hierarchy, which speeds-up the search by guiding it towards fruitful solution spaces. In a second step, the content of the original PDF document is extracted, structured, and matched with its corresponding high-resolution visual signature. Finally, the matched content is attached to the low-resolution document image's visual signature, which greatly enriches the document's content and indexing. We present in this article both these identification and extraction methods and evaluate them on various documents, resolutions and lighting conditions, using different capture devices.

## Categories and Subject Descriptors

I.7.5 [**Document and Text Processing**]: Document Capture – *document analysis.*

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing – *abstracting methods, indexing methods.*

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *retrieval models, search process*.

## General Terms

Algorithms, Performance, Design, Experimentation, Verification.

## Keywords

Low-resolution document image identification, document visual signature, documents' content extraction, document-based meeting retrieval.


*DocEng'04*, October 28–30, 2004, Milwaukee, Wisconsin, USA.


## 1. INTRODUCTION

Due to advancement in hardware technologies, different kinds of cameras are available on the market. The size of the capture device is often directly proportional to the image quality, and thus mobile devices often propose low-resolution images. Therefore, our aim is to develop an algorithm for identifying documents captured from low-resolution capture devices that can be easily applicable for the high-resolution capture devices without degradation of the performance. Currently, many people daily use devices such as web-cam, digital cameras, mobile phones, etc. Due to the comparatively smaller size of such devices, they can be carried anywhere at any time and can be used to capture documents of interest in the lectures, meetings, conferences, supermarket, etc. It implies lots of possible new applications such as mobile OCR for visually impaired people, for real-time translation, etc. Furthermore, these captured documents can be queried for finding available related information, such as audio or video. For example, during a conference, projected slides of interest could be captured and used afterward for querying the conference repository and retrieve the corresponding original documents (one or more slides presented by a particular speaker), or the related audio/video sequence, or any annotations related to the stored medias [1][12][13]. For this purpose, we make the reasonable assumption that all the documents present in such environments can be stored in advance in a repository.

For most of the existing document analysis systems [7][8][20], the system input is either a scanned document of 300 dpi or higher, or an electronic document (e.g. PDF, etc.). The qualities of these documents are in general quite high and suitable for further low-level processing. If the perceived document image is captured from a low-resolution device and compressed with quality-losing format such as JPEG (50 – 100 dpi), then it becomes difficult to analyze such documents using standard systems. In most of the capture devices (digital cameras, mobile phones, etc.), JPEG compression is used in order to reduce the storage space and to speed up the processing, which unfortunately implies that more noise is brought in and some useful details are lost. Furthermore, the captured document images from such devices are often non-uniform in terms of lighting, because of the use of flash or various lighting conditions. Finally, many other distortions or incomplete information are often present (e.g. varying distance to the object, motion blur, occlusion, etc.). The relatively low resolution and the frequent variations in the captured environments make the noise removal and content extraction very difficult, that causes the drastic decrease in the identification accuracy.

Most of the document image identification systems use classifier to identify the incoming unknown document images. Algorithms used for the classification are either supervised (training requires documents with known class label such as decision tree, neural networks, etc.) or unsupervised (training is based on the features of documents with unknown class label such as K-means, self-organizing maps, etc.) or semi-supervised (combination of both) [21]. The features for classification are mainly based on the layout structure (physical, logical) and on the content of the documents. Unfortunately, in our case it is difficult to extract the complete layout structure and a clean content (using OCR) of the captured documents due to the poor resolution. However, many documents have the same layout structure; all the slide images in a particular presentation very often have the same layout structure with a different content, because they use the same design pattern. This characteristic of slideshows drastically reduces standard methods' identification accuracies.

In this article we propose a novel document identification method that uses a visual signature as a way to symbolize documents and avoid traditional classifiers, which generally introduce errors. The geometric layout analysis we perform attempts to use basic image properties and spatial relations to extract structure without reference to a particular document type [18]. Such analysis is necessary since our input low-resolution image has no structural description. However, the low-resolution of the capture image does not allow extracting a complete layout structure. The visual signature we propose is a shallow and hierarchically structured representation of the document layout structure with a zone's labeling (text, image, etc.). This visual signature is matched with electronic documents available in a repository, in order to identify the corresponding low-resolution image. The matching then follows the hierarchical structures of visual signature and does not visit the entire search tree. The highest-level features stored in the signature are first compared and no-good solutions are removed from the search space. The comparison continues with lower-level features, and so on until the leave's level is reached. We will see that this method is fast, mainly because the visual signature hierarchy guides the search towards fruitful solution spaces. Furthermore, by alternating feature-specific comparison with global distance comparison, it guaranties that no good solutions are avoided. However, we believe that applying standard OCR techniques is not an acceptable solution for extracting textual information from low-resolution images and we propose an alternative method that benefits from the information stored in the original matched electronic documents.

The proposed system is targeting applications such as, browsing on recorded and archived meetings, conferences, lectures, etc. The system can be queried either with a) document images, projected or lying on the table, and captured with low-resolution web-cams, digital cameras, mobile phones, etc. or b) with the keywords, present in the slideshows. Basically, the goal of our low-resolution documents identification method is to link all the multimedia streams captured during the meetings or conferences (audio, video, etc.) with the visible documents. Furthermore, during the query, if the corresponding document is identified from the repository then all the linked multimedia data of interest will be reviewed to the users.

In the next section, we present a brief state-of-the-art of both layout-based and content-based document identification methods. In Section 3, the procedure for building the visual signature, which symbolizes the low-resolution documents, is described. Section 4 explains the hierarchical structuring of the visual signature based on features' priority. In Section 5, both an exhaustive method and a hierarchical search technique, based on the hierarchically structured visual signature are described. Section 6 presents the retrieval performance of various captured slide images using our method. Section 7 explains how the slide content can be extracted without having to perform any OCR technique. Finally, conclusion and future works are described in Section 8.

# 2. STATE OF THE ART

We have found two distinct approaches in the state-of-the-art for identifying documents. The first approach focuses on the document layout, whereas the second uses the document content. In this section, we present both research directions and discuss how they can be applied to the identification of low-resolution images.

## 2.1 Document Layout

The aim of the page segmentation and geometric document layout analysis is to partition documents into homogeneous regions. Various algorithms have been proposed for page segmentation and geometric layout analysis [5][8][9][10][11][19][20][21].

Traditional approaches for page segmentation and geometric layout analysis are typically referred as top-down methods. Such approaches look for global information on the page (e.g. black and white strips) and partition the page into blocks and then classify them into text lines and finally into words [11][20]. Wong, Casey and Wahl first proposed this kind of approach in 1982 [21]. Wang et al. also investigated a similar algorithm for newspaper layout analysis [19]. A more detailed survey of these approaches can be found in [8]. These approaches perform well for documents assumed to be rectangular in shape with relatively uniform font and size. However, the performance of such approaches degrades significantly when different components are closely adjacent to each other or overlapping.

On the other hand bottom-up methods start with local information (connected components or foreground pixels), determine the words, merge the words into text lines and merge the text lines into paragraphs [5] [18]. The connected components are extracted from the image and then, components of the same type are iteratively grouped together to form progressively higher-level descriptions of the documents (e.g. words, lines, paragraphs, etc.) [5]. The disadvantage of this approach is that the time complexity is higher as compared to top-down approaches due to the identification analysis and grouping of the connected components. Furthermore, bottom-up approaches suffer from the traditional problem of incorrect segmentation due to the early groupings.

Alternatively, texture-based approaches consider the various components of a document image, such as text, images or graphics, as being different textures [9][10]. The problem of such approach is that the time complexity is high and in some cases regions of different types, having a similar texture can be confused or merged. In the case of slides with non-homogeneous background, this kind of method may be inefficient since the foreground objects won't be distinguished from the background texture.

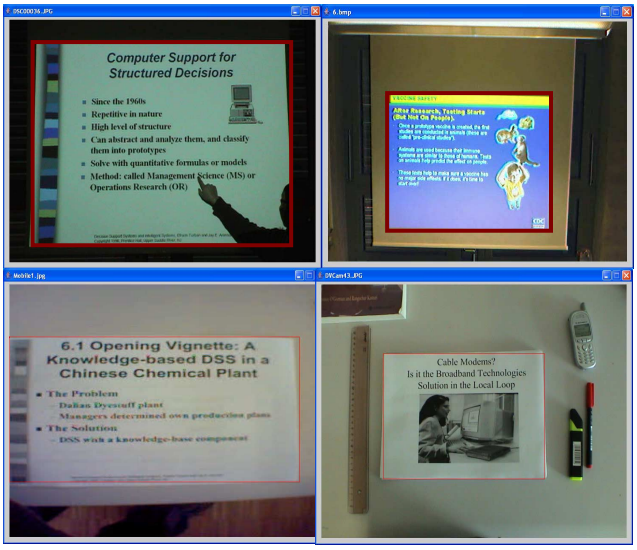


**Figure 1. Documents captured from a digital camera, a mobile phone and a DV camera. The document zone is highlighted**

## 2.2 Document Content

Content-based linking of presentations/meetings/lectures' documents with other medias has been tackled in various research projects [4][6][13][15]. Such methods extract the document content rather than any layout information and the matching is based on either global image comparison or a character string comparison using OCR. Chiu et al. proposed to automatically link multimedia data with a DCT-based image matching of the slide content [4]. The method matches the content of the slideshow's with the captured video. Unfortunately, the method is most suitable for matching high-resolution and high quality slide images. The performance may degrade if the images are of low-resolution and not accurately segmented. Partial occlusion or presence of blur also degrades its performance. Franklin et al. described another technique for linking slideshows with the audio stream of a speaker by matching the speech content with the text content in the presentation slide [6]. Mukhopadhyay et al. proposed a method that matches the content of HTML pages, which contain presentation slides, with the low-resolution video that also includes the presentation slides [13]. The method is based on first binarizing and dilating the segmented slide images and frames to highlight the text regions and then using the Hausdorff distance to compute the similarity between the text lines. The drawback of this method is that the slide region must be accurately segmented and it works well only on slides that contain text. Ozawa et al. proposed a slide identification method for lecture movies by matching characters and images [15]. The method uses OCR to recognize the text and an image matching technique (Dynamic Programming) for matching slides extracted from the video with the original slides. Their method performs well with high-resolution images, captured with a DV camera and slides containing text. For low-resolution images, current OCR techniques fail and thus matching is incorrect. This kind of approach only works with high-resolution slide images and it works only with slides containing text in large fonts (> 24 points).

# 3. VISUAL SIGNATURE EXTRACTION

The visual signature we defined is a hierarchically structured description of a document's shallow physical layout with its respective labeling. In our system we use this signature to describe both a) low-resolution images resulting from the capture of projected slides and b) images converted from the original electronic slide documents. We, then extract various visual features and organize them in the visual signature in a structured manner. The extraction of the visual signature from the electronic documents in the repository is straightforward; the PDF or PowerPoint form of the original electronic documents is converted into a relatively high-resolution image, on which the signature is computed. For the captured images, we first identify the projected part and then up-sampled to the common resolution format. A simple GUI is used to select manually the coordinates of the projected part (see Figure 1) from the captured images. This will be improved in the future with an automatic detection of the projected area within the captured documents.

First of all, the resolution of each document is up-sampled or down-sampled to a common resolution format (720 X 540). Then the image is converted to grayscale and binarized using Otsu binarization for further processing [14]. Furthermore, looking at the mean horizontal run length of both black and white pixels the proper segmentation of foreground objects is checked. For example, for the slide images having dark background and light foreground, the output of the binarization is reversed. In the following sections we detail the extraction of various visual features. However, due to poor resolution of the captured images, the color information is not currently considered for the visual signature.

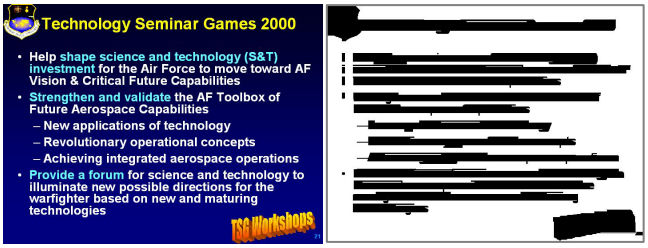


**Figure 2. a) Original image, b) after passing through RLSA**

## 3.1 Bounding Box Extraction and Labeling

The bounding box of each object (text, images, etc.) is extracted using a Run Length Smearing Algorithm (RLSA) [20] both in horizontal and vertical directions. If the distance between two consecutive black pixels is under a specific threshold ($T_h$ = 80 for horizontal, $T_v$ = 100 for vertical) then all the pixels between these two black pixels are turned to black. The bounding box of the blocks are further derived from the background by combining the outputs of horizontal and vertical RLSA with a logical AND operator. Additional horizontal smoothing using the RLSA ($T_s$ = 15) produces the final segmentation result as shown in Figure 2. The values of these thresholds have been evaluated and tuned using about a hundred slide images with resolution of 75 dpi. If the resolution is changed then the corresponding values are scaled accordingly. Then an initial labeling (text, image, etc.) of the bounding box is performed by looking at the various block features.

### 3.1.1 Feature vector for each bounding box

The blocks drawn in Figure 2b must next be labeled according to their content, so that correct subsequent analysis is further performed. The labeling of each block is done using the following feature vector [20][16]:

- Total number of black pixels in the segmented block; its minimum x-y coordinates and its maximum width and height $(Y_{\min}, X_{\min}, W_{\max}, H_{\max})$; eccentricity $E = W_{\max}/H_{\max}$ of the rectangle surrounding the block; the mean horizontal length of black runs $R_m$ and the bounding box pixel ratio $P$ of the original data for the block, i.e. before running the RLSA.
- The average correlation $C_1$ between adjacent scan lines ($C(1, y)$); the percentage of lines $C_2$ with $C(1, y) > 0.8$; the average correlation $C_3$ between scan lines separated by ten intervening scan lines ($C(10, y)$). The normalized correlations between scan lines at $y$ and $y + r$ is defined as:

$$C(r,y) = \frac{1}{L}\sum_{k=0}^{L-1}[1 - 2p(y,k) \oplus p(y+r,k)]$$

$L$ being the number of pixels in a scan line and $p(y, k)$ is the value of the $k^{th}$ pixel in the scan line y using the original data of the block. The selection of 10 scan lines (10 × 75 / 72) is based on the input resolution (75 dpi) and on the minimum font size (10 points) used in slideshows. If the resolution is changed then the corresponding number of scan lines is to be computed as described above.

### 3.1.2 Bounding box labeling

A unique label is further assigned to each extracted bounding box by considering the previous feature vector. The following rules are applied in order to label the blocks:

1. Text: $C_{1,2} \succ (1-\omega)$ and $C_3 \prec (1-\omega)$
2. Horizontal solid lines: $R_m \succ (1-\omega) \times W_{\max}$ and $E \succ 1/\omega$
3. Graphics and images: $E \succ \omega$ and $C_{1,2,3} \prec (1-\omega)$
4. Vertical solid lines: $E \prec \omega$ and $C_{1,2,3} \prec \omega$
5. Horizontal bar with text: $R_m \prec (1-\omega) \times W_{\max}$ and $E \succ 1/\omega$
6. Vertical bar with text: $E \prec \omega$ and $C_{1,2,3} \succ \omega$

The above-mentioned method has been tested on approximately one hundred documents and the parameter $\omega$ has been set to 0.2 with satisfactory performance. The minimum and maximum heights of the text are computed considering the minimum and maximum size of the fonts. For example, the minimum and maximum font sizes for a typical PowerPoint slide image are 8 and 96 points. Then the corresponding *MIN_TEXT_HEIGHT* and *MAX_TEXT_HEIGHT* are computed based on the specified minimum and maximum font sizes and the resolution (dpi) of the input image. For example, if the input image is 75 dpi, then *MIN_TEXT_HEIGHT* is 8 pixels (8 x 75 / 72) and *MAX_TEXT_HEIGHT* is 100 pixels (96 x 75 / 72). Then the corresponding *MIN_TEXT_WIDTH* and *MAX_TEXT_WIDTH* are computed from *MIN_TEXT_HEIGHT* and *MAX_TEXT_HEIGHT* using default width-height-ratio (7 / 12 for typical Courier font style)[17]. Hence, for the configurations above the *MIN_TEXT_WIDTH* of 5 pixels (8 x 7 / 12) and *MAX_TEXT_WIDTH* of 58 pixels (100 x 7 / 12) are computed.

Each block is further processed to check whether it contains several blocks. Indeed, logos sometimes appear with text (most of the time in the title), captions are close to images and finally several images are often grouped in a same image block. Most often, they could be separated by considering the average block's height, width and also by passing them through horizontal and vertical projection profiles [3]. Finally, when the joined blocks are separated into individual blocks, each block is re-processed in order to extract its feature vector (sub subsection 3.1.1) and labeled accordingly (sub subsection 3.1.2).

## 3.2 Text Line Extraction

In the previous section, the textual blocks are labeled as text but there is no further information about the text alignment (e.g. horizontal and vertical text lines). In this section, we discuss about the extraction of the feature vector of each text line (horizontal or vertical) for the visual signature. The feature vector for each text line is $\{Y_{min}, X_{min}, H_{max}, W_{max}, N_{word}, R_i(Y_i, X_i), P\}$, where $N_{word}$ is the number of words in the text line and $R_i(Y_i, X_i)$ is the relative position of word $i$ ($i = 1, 2, .., N_{word}$) with respect to the bounding box's $Y_{min}$ and $X_{min}$.

### 3.2.1 Horizontal text lines

Often a textual block contains more than one line. Such blocks are passed through horizontal projection profile and separated into individual blocks with one text line per block. Then the number of words in each block is computed using the vertical projection profile (Figure 3). The threshold for word gap detection is selected by looking at a) the average column gap between connected components, b) the mean black run length $R_m$ and c) the average height of each block [3].

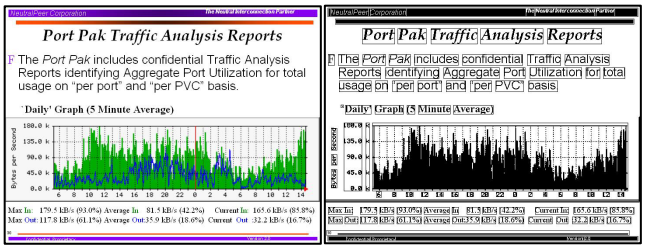


**Figure 3. a) A typical slide with text (horizontal and vertical), image, solid line and bar with text and b) extracted features**

### 3.2.2 Vertical text lines

The vertical text lines are segmented as multiple blocks containing one or more connected components and each block is labeled as either a horizontal text or an image. The width of such blocks is inferior to *MAX_TEXT_WIDTH*. Let $B = \{b_1, b_2, .., b_N\}$ be the set of blocks with label of either text or image such that $\forall b_i \in B : W_{\max}(b_i) \prec MAX\_TEXT\_WIDTH$. Any two blocks in $B$ are merged, if and only if they are aligned vertically and the vertical gap between them is inferior to the *MAX_COL_GAP*. Each block is compared to rest of the blocks in $B$ and if one or more blocks satisfy the condition above then they are merged i.e. $MERGE(b_i, b_j) \Leftrightarrow (A_{i,j} \wedge B_{i,j} \wedge C_{i,j})$ for $\forall b_i, b_j \in B : i \neq j$

$$\text{Where} \begin{cases} A_{i,j} = |X_{\min}(b_i) - X_{\min}(b_j)| \prec T \\ B_{i,j} = |X_{\min}(b_i) + W_{\max}(b_i) - X_{\min}(b_j) - W_{\max}(b_j)| \prec T \\ C_{i,j} = |Y_{\min}(b_j) - Y_{\min}(b_i) - H_{\max}(b_i)| \prec MAX\_COL\_GAP \end{cases}$$

Finally, $B$ consists of zero or more vertical text lines. Each element in $B$ is checked and if labeled as a vertical text line then it is passed through the horizontal projection profile to compute the number of words in the text line and their relative positions [3]. The threshold for the vertical word gap detection is selected by looking at a) the average row gap; b) the width and c) the vertical mean black run length $R_v$ of each vertical text line. Finally, the feature vector for each vertical text line is updated. In our system the threshold $T$ is set to 10 pixels and it works for most of the slide images.

## 3.3 Image, Horizontal Line and Vertical Line

For these kinds of blocks, no processing is necessary. We keep only the feature vector ($\{Y_{min}, X_{min}, H_{max}, W_{max}, P\}$) for the visual signature for each of this block (see Figure 3).

## 3.4 Horizontal and Vertical Bar with Text

Often in presentations, a rectangular bar behind the title is used (see Figure 3). This bar generally has a different background than the slides in order to highlight some textual information (below title, above footnotes, etc.). During binarization, often the foreground and background are reversed for such blocks. If we do not consider this case, the block will be either considered as an image or as a horizontal (resp. vertical) line. However, it is useful to analyze these kinds of blocks for adding further features to the visual signature. Thus, horizontal bars with text block (resp. vertical bar with text) are first converted to horizontal (resp. vertical) text blocks, i.e. white background with black foreground. Then the new textual block is treated like other text blocks (horizontal, vertical text extractions, subsection 3.2). The feature vector of a horizontal (resp. vertical) bar with text is thus the same as a horizontal (resp. vertical) text line but with a different labeling (see Figure 3).

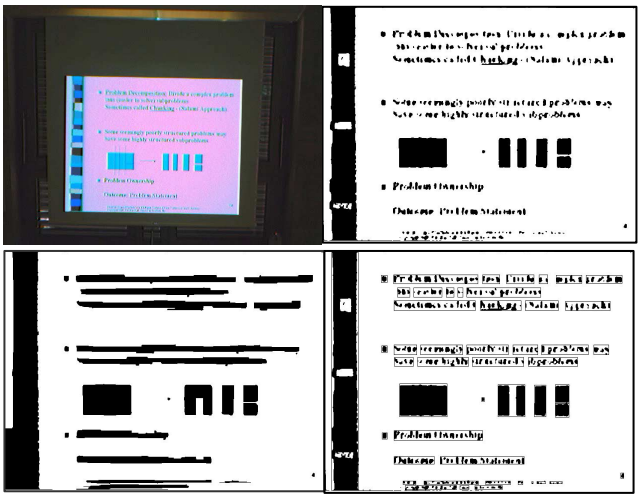

**Figure 4. a) Captured image, b) projected part is up-sampled and binarized c) output of RLSA d) features' bounding boxes**

## 3.5 Bullet Extraction

Bullets are often used in slideshows and appear normally at the beginning of a text line. It is thus useful information to extract and store in the visual signature. Looking at Figure 2, it is clear that in some cases bullets are attached to horizontal text lines. In other cases, bullets are segmented as separate blocks and labeled either as images or as horizontal texts having a width inferior to $MAX\_TEXT\_WIDTH$. Let $B = \{b_1, b_2, .., b_{|B|}\}$ be the set of blocks labeled either as text or as image, having the property $\forall b_i \in B : W_{\max}(b_i) \prec MAX\_TEXT\_WIDTH$ and $L = \{l_1, l_2, .., l_{|L|}\}$ be the set of the rest of horizontal text lines. The method looks for the presence of a block $b_i$ ahead of a text line $l_j$, by comparing the relative position and bounding box properties (height, width) of $b_i$ with line $l_j$. If the block $b_i$ is satisfying the condition given below then the block $b_i$ is considered as a bullet and associated with line $l_j$. There can only be one bullet per text line. Moreover, the line $l_j$ is removed from $L$ and the method continues to operate for the rest of the elements in $L$. Let $M = \varnothing$ be the set of bullets.

- If the condition $(p_{i,j} \wedge q_{i,j} \wedge r_{i,j})$ is satisfied then $ADD(M, b_i)$, $REMOVE(L, l_j)$ for $1 \le \forall i \le |B|$ and $1 \le \forall j \le |L|$

$$\text{Where} \begin{cases} p_{i,j} = Y_{\min}(b_i) \ge Y_{\min}(l_j) \\ q_{i,j} = H_{\max}(b_i) \le H_{\max}(l_j) \\ s_{i,j} = X_{\min}(l_j) - X_{\min}(b_i) - W_{\max}(b_i) \prec MAX\_COL\_GAP \end{cases}$$

- If $L \neq \varnothing$, then possibly a bullet is within $l_j$ and should be present in the first word. The average width and height of the connected components and the average column gap between connected components in $l_j$ are then computed. The following steps are further performed for the bullet extraction in the first word of $\forall l_j \in L$:

1. If there is only one connected component in the word, and if its width is inferior to the $MAX\_TEXT\_WIDTH$ and if either its height is inferior to the average height or its width is inferior to the average width, then this connected component is a bullet (e.g. solid rectangle, circle, horizontal bar, picture, etc.).
2. For two connected components in the word: If either the width and height of the first one is two times greater than the corresponding width and height of the next one or the height of the first one is inferior to the next one, then the entire word is a bullet (e.g. I., 1., 2., 3., numbering, a), 1), etc.).
3. For more than two and less than five connected components in the word: if the height of all connected components except the last one are less or equal to the average height of the text line, and the height and width of the last connected component is inferior to the half of the height and width of all previous ones (e.g. II., III., IV., etc) then the entire word is a bullet.

- If a bullet is found in $l_j$, then $l_j$'s feature vector is updated by removing its first word and the word is moved to the bullet set $M$ ( $ADD\{M, FIRST\_WORD(l_j)\}$ ).

Finally, if $M \neq \varnothing$, then the feature vector for each element in the $M$ is built for visual signature. The feature vector for bullets is the same as that of image $\{Y_{min}, X_{min}, H_{max}, W_{max}, P\}$.

In this section, we explained how to extract various features and further stored them in a visual signature. Figure 4 represents one of the typical captured images from a web-cam, corresponding RLSA output of the projected part and bounding

box of various extracted features. Looking at Figure 4, it is clear that the resolution of the captured image is very poor in order to apply any standard OCR techniques, whereas it can be applicable to the original images (see Figure 3).

In the following section, the visual signature structuring is explained and presented as an alternative to classification techniques.

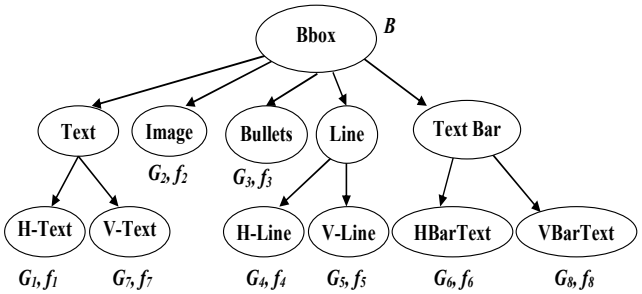


**Figure 5. Tree representation of features in visual signature**

## 4. STRUCTURING VISUAL SIGNATURES

We have not used any classifier for organizing the electronic documents and their corresponding images, in the repository. The main idea is to structure the visual signature rather than structuring the repository. We organized our visual signature based on the feature's priority; higher-level features appear first in the XML hierarchy and lower level features stand at the leaves (Figure 6). It is a breadth first matching approach, which considers higher-level features first. If one traverses from the root feature nodes to the leaf feature nodes, the priority slowly decreases (Figure 5). There are few reasons for choosing such hierarchy:

1. The hierarchy of the visual signature corresponds to the visual features' extraction process. Features requiring less processing are first extracted. They are more reliable than the lower-level features, which need more processing, and thus may introduce errors. For this reason, we chose to have the most reliable features at the top of the visual signature tree.
2. The textual layouts vary more than other features in most of the slide images. For example, a person often select an existing design patterns (e.g. PowerPoint application) and thus only the textual and image content varies. Thus, the textual feature is of highest priority. Observing real world slideshow presentations, other features (bullets, horizontal and vertical lines, bar with text, etc) have been prioritized accordingly.
3. It speeds-up during the matching of visual signatures by giving more importance to the high-level features, which narrows the search path.

An example of the current features' hierarchy is displayed in Figure 6. In the following section, matching techniques based on our structured visual signature, are described.

## 5. MATCHING VISUAL SIGNATURES

The proposed signature-based matching technique has numerous advantages. Firstly, it is better than the global image matching, for example pixel-by-pixel comparison [13][15], mainly because a) the resolution of the extracted images is very poor for an

```
<VisualSign>
  <BoundingBox NoOfBb="16">
    <Text NoOfLine="14">
      <HasHorizontalText NoOfSentence="14">
        <Sentence y="1" x="8" width="570" height="32"
          NoOfWords="4" PixelRatio="0.36" />
        ...
      </HasHorizontalText>
      <HasVerticalText NoOfSentence="0" />
    </Text>
    <HasImage NoOfImage="1">
      <Image y="0" x="0" width="79" height="75"
        PixelRatio="0.43" />
    </HasImage>
    <HasBullet NoOfBullets="6">
      <Bullet y="206" x="34" width="6" height="6"
        PixelRatio="0.92" />
      ...
    </HasBullet>
    <Line NoOfLine="0">
      <HasHLine NoOfLine="0" />
      <HasVLine NoOfLine="0" />
    </Line>
    <BarWithText NoOfBar="0">
      <HBarWithText NoOfBar="0" />
      <VBarWithText NoOfBar="0" />
    </BarWithText>
  </BoundingBox>
  <GlobalFeatures>
    <ImagePelRatio FullImageRatio="0.18"
    PixelRatioWin1="0.2" PixelRatioWin2="0.14"
    PixelRatioWin3="0.17" PixelRatioWin4="0.2" />
  </GlobalFeatures>
</VisualSign>
```

**Figure 6. An example of visual signature in XML**

effective matching, b) rotation and translation affects the pixels locations, which further affects the distance computation ($d(A,B) \leq d(A,X) + d(X,B)$, where $A$, $B$ and $X$ are the respective source, target and intermediate images). Secondly, this is novel document identification and matching technique that avoid traditional classifiers and uses a visual signature as a way to represent documents.

In this section, we present both the exhaustive and hierarchical search techniques for the identification of captured images. At each feature node level (Figure 6), the matching score is calculated by considering the number of total matches divided by the total number of elements located in the corresponding feature node. Both comparison directions are considered, mainly because the number of features and the number of elements in a feature is rarely equal for the same high-resolution and low-resolution images. This fact is a direct implication of the resolution difference and of further errors in the RLSA, projection profiles and bounding boxes' labeling. For a particular feature, the total number of matches is computed by looking at the difference in the feature vector of each element in both the extracted and the original visual signatures. The final matching score is the average of scores in both directions. The threshold values for computing the feature vector are shown in Table 1. All the given thresholds (pixels) have been computed for a resolution of 75 dpi and they should be scaled accordingly if the input resolution changes. For example, if a threshold value is 5 for the resolution of 75 dpi then the corresponding value for 100 dpi is 7 (5 × 100 / 75). The same rule is also applicable for selecting the thresholds for the RLSA.

## 5.1 Exhaustive Search

This is the brute force search method for matching visual signatures. First the matching score for each available feature ($f_i$) (see Figure 5) is calculated and then the global score is computed as the ratio of the sum of all features' score upon the number of features having non-zero score. In this method, two types of mechanisms have been used for computing the global score: with ($\sum \omega_i f_i$ for $\forall i$) and without weighted value of features ($\sum f_i$ for $\forall i$) (for weight values, see Table 2). The signature having the highest global score is returned after comparison with all the signatures in the repository.

**Table 1. Threshold used for feature vector comparison**

| Elements | Threshold |
|---|---|
| $Y_{min}$, $X_{min}$ | 5 (pixels) |
| $H_{max}$, $W_{max}$ | 10 (pixels) |
| $N_{word}$ | 2 (word count diff.) |
| $X_i$, $Y_i$ for each word | 5 (pixels) |

**Table 2. Weight for various features**

| Features | Weight |
|---|---|
| Horizontal text ($f_1$) | 0.8 |
| Image ($f_2$) | 0.6 |
| Bullet ($f_3$) | 0.5 |
| Others ($f_3, f_4, f_5, f_6, f_7, f_8$) | 0.3 |

## 5.2 Hierarchical Search

This matching technique is based on simple heuristics taking into consideration the hierarchy of the visual signature. Higher-level features are first compared, and then the lower-level features are matched, and so on until the matching reaches the leaves of the hierarchy. The hierarchy of the visual signature normally guides the search path. In this case the score for any feature is considered, if it is greater than a certain minimal value (0.2). Let $E$ be the signature of the captured image and $D = \{d_1, d_2, ..., d_{|D|}\}$ the set having all the signatures in the repository and $T$ the matching threshold.

1. $B = \{b_1, b_2, .., b_{|B|}\}$ is derived from $D$ ($B \subseteq D$), considering all the signatures with the difference in bounding box number is inferior to $T_b$ i.e. $\forall b_i \in B \Rightarrow \exists d_i \in D : Diff\_Bbox(E, d_i) \le T_b$.

2. First, the horizontal text feature's matching-score $f_1$ is computed. The subset $G_1$ (see Figure 5) is created from $B$ ($G_1 \subseteq B$) by considering elements, whose score $f_1$ is superior to $T$. If no element in $G_1$ satisfies the above condition ($f_1 > T$), then the same subset $B$ ($G_1 = B$) is kept for the next feature comparison i.e. $\forall g_i \in G_1 \Rightarrow (\exists b_i \in B : f_1(b_i, E) \succ T), (G_1 = B) \Leftrightarrow (G_1 = \varnothing)$.

3. The same procedure as in step 2 is used for deriving the next subset from the previous subset for all other features following the features hierarchy (from higher-level to lower-level features, see Figure 5 and Figure 6 for features hierarchy) i.e. $\forall g_i \in G_j \Rightarrow (\exists g_k \in G_{j-1} : f_j(g_k, E) \succ T), G_j \subseteq G_{j-1}, \; 2 \le j \le 8$.

   At any feature level, elements fulfilling the criteria in all the previous feature levels are kept. Furthermore, at any feature level, if there is only one solution and its matching score is greater than 80%, then all other existing features for this solution are looked for. If the global matching score exceeds 90%, the search is then terminated and the current solution is returned by the system (winner one).

4. When the search reaches the right-most leaf node, the number of elements in the final subset $|G_8|$ is checked. If it is more than one, a weight is assigned to each feature's score, according to its position in the hierarchical visual signature. The corresponding weight for each feature is shown in Table 2. The sum of the weighted score of all features for all elements in $G_8$ is then calculated. If the difference between the highest weighted sum and the second highest weighted sum of $G_8$ is superior to 0.4, then the element having the highest weighted sum is considered as the required visual signature. Otherwise, $B$ is assigned to the top 20% elements in $G_8$ having the highest sum.

5. $T$ is increased with a step size of 0.2 and the same matching procedure starts again from step 2 and it continues until only one matching slide is found or no more elements are present in the set. If there are no more elements in any of the above subset, it means that the slide is not present in the repository. The above matching technique is summarized in Figure 7.

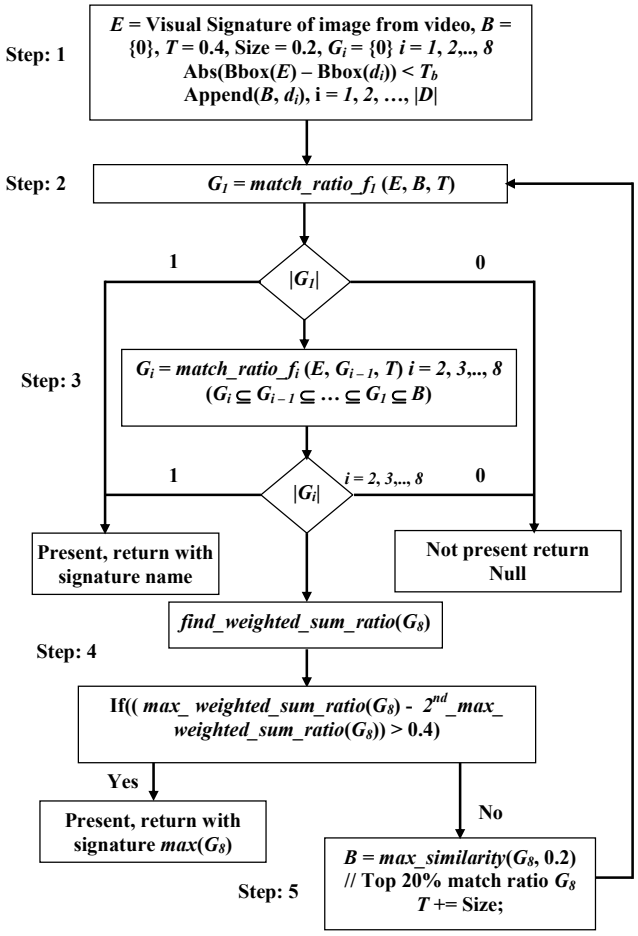


**Figure 7. Flow chart of the hierarchical matching technique**

## 6. EVALUATION AND RESULTS

We have developed an application that automatically evaluates the matching of the visual signatures. The visual signatures corresponding to the original electronic documents are stored in the repository along with the original slide images. The images captured from the web-cams, digital cameras and mobile phones are first processed to build their corresponding visual signatures. The extracted visual signature is then used in order to query the repository and find the best matching visual signature (Figure 8). For this purpose, 16 slideshows have been captured as video streams, using a web-cam. One image per stable period has been extracted, which corresponds to a sequence where only one slide is displayed [2].

We evaluated the proposed matching methods for slideshows having a homogeneous background without complex textures. The evaluation is based on a recall and precision metrics rather than on a simple identification rate, which is the ratio of the number of documents correctly identified upon the number of documents queried. While using a simple identification, one could know the accuracy of the identification from the user point of view, but there is no information about the accuracy from the system part, which is the ratio of the number of documents correctly identified upon the total number of documents returned by the system. Using the metrics above, we get the identification rate from both sides and the relationship between them. Recall ($R$), Precision ($P$) and F-measure ($F$) are defined as:

$$R=\frac{D_c}{D_c+D_n+D_\phi},\ P=\frac{D_c}{D_c+D_f},\ F=\frac{2\times R\times P}{R+P}\ \text{and}\ \frac{R}{P}=\frac{D_c+D_f}{D_c+D_n+D_\phi}$$

Where
- $D_c$ = number of correct documents retrieved
- $D_n$ = number of correct documents not retrieved
- $D_\phi$ = number of documents are not in the repository
- $D_f$ = number of incorrect documents retrieved
- $D_c+D_n$ = total documents queried are in the repository
- $D_c+D_f$ = total documents returned by the system

In this evaluation, we queried only documents that are in the meeting repository, i.e. $D_\Phi = 0$. Thus, the system returns either a document (correct or wrong) or null, when the system could not make a trustful decision. Hence the following rules can be stated: $(D_c+D_f \le D_c+D_n) \Rightarrow (D_f \le D_n) \Rightarrow (R \le P)$ and $(D_n - D_f)$ is the number of times the system returns null. The ratio $R$ / $P$ is the answering rate of the system. $R$ conveys the identification rate from the user part (i.e. ratio of the correct documents identified upon total documents queried) whereas $P$ says the trustfulness (accuracy) of the answer returned by the system (i.e. ratio of the correct documents identified upon total documents returned by the system). Finally, $F$ corresponds to the combined performance of both $R$ and $P$.

All the following evaluations have been performed on a repository containing more than 1000 slide images. For all the captured devices used, the images mostly contain the slide region with a rotation of less than ±5 degree. Note that because only the signature matching is employed and because OCR is avoided, our hierarchical search takes less than 1 second in order to compare 1000 image visual signatures. In the first set of experiments, our matching algorithms have been tested on 626 (16 slideshows) slide images, captured with a web-cam, as queries.

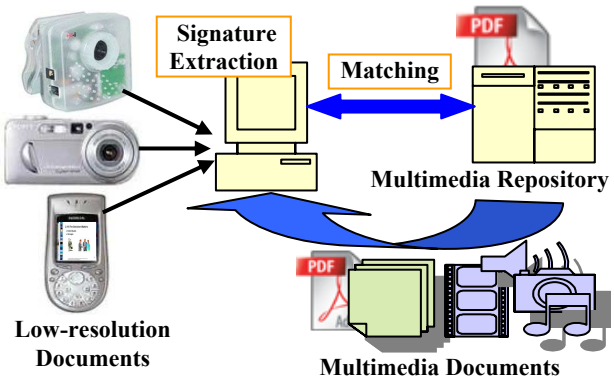


**Figure 8. Retrieval of original documents and related media**

The performances of three matching algorithms have been compared: two exhaustive searches (with and without weighting mechanisms) and one hierarchical search (presented in the above section). The results are displayed in Table 3 and Table 4. For the exhaustive search, the average recall (66%) and precision (71%) with weighted features is better than the respective recall (62%) and precision (67%) without weight, which tells about the benefits of having weighted visual features for matching. This score should be improved with a correct weighting of each visual features, using statistical methods. We did some preliminary experiment using the Hausdorff distance [13] and it resulted in a precision below 50%, with a recall below 40%. We plan in the near future to compare precisely the score of our visual signature with a standard Hausdorff distance.

**Table 3. Results (Exhaustive) for images from web-cam**

| | Without weight | | | | With weight | | | |
|---|---|---|---|---|---|---|---|---|
| **Slideshow** | **R** | **P** | **F** | **R/P** | **R** | **P** | **F** | **R/P** |
| 1 (47) | 0.91 | 0.91 | 0.91 | 1.00 | 0.96 | 0.96 | 0.96 | 1.00 |
| 2 (75) | 0.91 | 0.91 | 0.91 | 1.00 | 0.95 | 0.95 | 0.95 | 1.00 |
| 3 (73) | 0.82 | 0.87 | 0.84 | 0.94 | 0.84 | 0.88 | 0.86 | 0.94 |
| 4 (72) | 0.72 | 0.72 | 0.72 | 1.00 | 0.82 | 0.82 | 0.82 | 1.00 |
| 5 (87) | 0.41 | 0.43 | 0.42 | 0.95 | 0.55 | 0.58 | 0.56 | 0.95 |
| 6 (27) | 0.19 | 0.19 | 0.19 | 1.00 | 0.30 | 0.30 | 0.30 | 1.00 |
| 7 (24) | 0.87 | 0.95 | 0.91 | 0.92 | 0.87 | 0.95 | 0.91 | 0.92 |
| 8 (21) | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 9 (14) | 0.57 | 0.57 | 0.57 | 1.00 | 0.71 | 0.71 | 0.71 | 1.00 |
| 10 (30) | 0.60 | 0.60 | 0.60 | 1.00 | 0.63 | 0.63 | 0.63 | 1.00 |
| 11 (13) | 0.38 | 0.38 | 0.38 | 1.00 | 0.31 | 0.31 | 0.31 | 1.00 |
| 12 (44) | 0.54 | 0.62 | 0.58 | 0.89 | 0.54 | 0.62 | 0.58 | 0.89 |
| 13 (34) | 0.32 | 0.48 | 0.39 | 0.68 | 0.32 | 0.48 | 0.39 | 0.68 |
| 14 (16) | 0.44 | 0.50 | 0.47 | 0.88 | 0.37 | 0.43 | 0.40 | 0.88 |
| 15(25) | 0.76 | 0.76 | 0.76 | 1.00 | 0.80 | 0.80 | 0.80 | 1.00 |
| 16(24) | 0.54 | 0.87 | 0.67 | 0.63 | 0.58 | 0.93 | 0.72 | 0.63 |
| Total (626) | 0.62 | 0.67 | 0.65 | 0.93 | 0.66 | 0.71 | 0.68 | 0.93 |

**Table 4. Result (Hierarchical) for images from web-cam**

| Slideshow | R | P | F | R/P |
|---|---|---|---|---|
| 1 (47) | 0.93 | 0.98 | 0.96 | 0.96 |
| 2 (75) | 0.92 | 1.00 | 0.96 | 0.92 |
| 3 (73) | 0.77 | 0.98 | 0.86 | 0.78 |
| 4 (72) | 0.67 | 1.00 | 0.80 | 0.67 |
| 5 (87) | 0.42 | 0.97 | 0.60 | 0.44 |
| 6 (27) | 0.52 | 0.93 | 0.67 | 0.56 |
| 7 (24) | 0.79 | 1.00 | 0.88 | 0.79 |
| 8 (21) | 0.86 | 1.00 | 0.92 | 0.86 |
| 9 (14) | 0.57 | 0.88 | 0.70 | 0.64 |
| 10 (30) | 0.57 | 0.74 | 0.64 | 0.77 |
| 11 (13) | 0.46 | 1.00 | 0.63 | 0.46 |
| 12 (44) | 0.25 | 1.00 | 0.40 | 0.25 |
| 13 (34) | 0.24 | 0.53 | 0.33 | 0.44 |
| 14 (16) | 0.18 | 0.60 | 0.28 | 0.31 |
| 15(25) | 0.24 | 1.00 | 0.39 | 0.24 |
| 16(24) | 0.29 | 0.88 | 0.44 | 0.33 |
| Total (626) | 0.54 | 0.91 | 0.65 | 0.59 |

Further, as we can see, our hierarchical search drastically increases the average precision, i.e. the truthfulness of the system (Table 4). This means that when the system gives back a document, the answer can be trusted in 91% of the cases. Although only 54% of the documents queried are identified (recall), documents incorrectly identified are known since the system returns null in case of uncertainty, and thus those documents can be re-queried on another system that retrieves the slideshow to which the documents belongs. This low recall value is due to either the removal of solutions in the initial matching step, which is the bounding box comparison or due to the fact that not even a single feature of the visual signature qualifies the minimum matching threshold (Section 5.2). However, this should be fixed in the future by properly setting up the various thresholds used by the hierarchical algorithm and by enhancing the bounding box extraction procedure as well as the matching technique, so that no good solution is removed in the initial step. Finally, from a preliminary study, it seems that the hierarchical search is only 2-3 times faster than the exhaustive search. However, with the increasing of the number of images in the repository, this ratio is proportionally growing and our hierarchical search will become greatly necessary for real-time applications, in order to avoid uninteresting search spaces.

For the exhaustive searches, two slideshows (6 and 11 of Table 3) gave back recall and precision values inferior to 40%, which drastically decreased the overall average performance. But the performance is better in the hierarchical search for the same slideshows. Indeed, in the exhaustive search, the solution having the highest non-zero global score is returned (Section 5.1) and it may not be the correct one, whereas in the hierarchical search, the score is feature-specific and is compared to a threshold for acceptance in each feature level. In the near future, we will improve the recall and precision of the exhaustive search by considering a pool of relevant solutions, let's say the top $N$ solutions and then the final solution will be chosen among them, using an adaptive weighing method.

In the second set of experiments, we evaluated the retrieval accuracy of the slides captured using digital cameras (2-4 Mpixel) as queries. The results are displayed in Table 5 and Table 6. Images were taken while the presentation was going on. In this case the performance is comparatively lower than the web-cam. The main reason is due to the distance varies and the camera rotation. A more subsequent evaluation will be performed in the near future.

**Table 5. Results (Exhaustive) for images from digital cameras**

| Slide images | Without weight | | | | With weight | | | |
|---|---|---|---|---|---|---|---|---|
| | R | P | F | R/P | R | P | F | R/P |
| 20 | 0.46 | 0.73 | 0.56 | 0.63 | 0.64 | 0.78 | 0.70 | 0.82 |

**Table 6. Results (Hierarchical) for digital cameras**

| Slide images | R | P | F | R/P |
|---|---|---|---|---|
| 20 | 0.43 | 0.71 | 0.54 | 0.61 |

In this evaluation process, both the recall and precision should be increased in order to improve the identification accuracy from both user and system sides. The lower values of recall & precision are due to a) the existence of tables, small font size (< 10 points), and complex figures, which obstructs the extraction of an effective visual signature and b) in some cases, the extracted slide images were so bad (distorted or too small) that the matching gave back no result. Since the number of correct retrieved documents ($D_c$) can be increased by enhancing the bounding box extraction procedure along with the matching technique as explained earlier, and because both recall & precision are directly proportional to $D_c$ ($R$ & $P$ α $D_c$), then not only can be increased $R$ and $P$ but also $D_n$ (correct documents not retrieved) and $D_f$ (incorrect documents retrieved) can be decreased. Moreover, $P$ is inversely proportional to $D_f$ ($P$ α $1/D_f$) and since $D_f$ can be decreased by tuning the threshold so that null is returned instead of incorrect documents, then $P$ can be increased without affecting recall and $D_c$.

# 7. CONTENT EXTRACTION FROM ELECTRONIC DOCUMENTS

Both the visual signature, extracted from low-resolution image, and the layout structure of the corresponding electronic document,

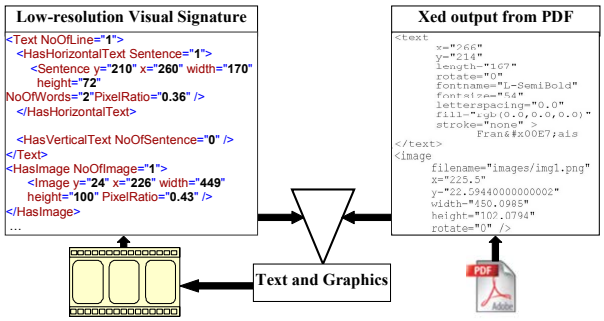


**Figure 9. Linking of visual signature with Xed's output**

extracted with Xed [7], are XML files. They can be further linked, and the visual signature can thus be enriched with the textual content available in the electronic document. The procedure compares the geometrical locations of the various feature blocks in our visual signature with the layout structure of the corresponding electronic document. Sometimes two or more feature blocks need to be merged in our visual signature so that it can be aligned with the electronic document's layout structure. Once the alignment is performed, the textual and graphical parts are extracted along with their characteristics. An example of this extraction process is shown in Figure 9.

## 8. CONCLUSION AND FUTURE WORK

In this paper, we described a new document identification method for low-resolution document image. We have explained how to extract various useful visual features and how to structure them in a visual signature by looking at their importance in a typical real world slideshow presentation. A fast matching technique based on a structured visual signature that does not require any classifier, has been then presented along with evaluation results. Finally, the resolution of the extracted images being too poor to perform OCR, an alternative way to get the textual content directly from the PDF documents [7] has been presented.

In the near future, we plan to improve our identification method to consider slideshows having complex background texture. We also plan to consider the color information by correcting the color values of the low-resolution capture devices and by identifying different background pattern for the matching technique. Finally, we plan to evaluate a) the performance of our visual signature for identifying low-resolution documents, using or not color information, and b) the performance of our matching techniques on slideshow repositories of various sizes.

## 9. ACKNOWLEDGEMENTS

We would like to thank the University of Applied Sciences of Fribourg and Didier Von Rotz for helping us setting up the meeting capture environment.

## 10. REFERENCES

[1] Abowd, G. D., Atkeson, C. G., Feinstein C. H. A., et.al. Teaching and learning as Multimedia Authoring: The Classroom 2000 Project, In *Proc. ACM Multimedia*, Boston MA, (Nov. 1996), 187-198.

[2] Behera, A., Lalanne, D., and Ingold, R. Looking at projected documents: Event detection & document identification, *Intl. Conf. on Multimedia Expo (ICME '04)*, 2004.

[3] Cattoni, R., Coianiz, T., Messelodi, S., and Modena C. M. *Geometric Layout Analysis Techniques for Document Image Understanding a review*: Technical Report, ITC-IRST, Trento, Italy 1998.

[4] Chiu, P., Foote, J., Girgensohn, A., and Boreczky, J. Automatically linking multimedia meeting documents by image matching, *Proc. of Hypertext '00, ACM Press,* 2000, 244-245.

[5] Drivas, D., and Amin, A. Page Segmentation and Classification Utilizing Bottom-Up Approach, *In Proc. of ICDAR,* 1995, 610-614.

[6] Franklin, D., Bradshaw, S., and Hammond, K. J. Jabberwocky: you don't have to be a rocker scientist to change slides for hydrogen combustion lecture, *Intelligent User Interface,* 2000, 98-105.

[7] Hadjar, K., Rigamonti, M., Lalanne, D., and Ingold, R. A new tool for eXtracting hidden structures from Electronic Documents, *DIAL '04*, PA, January 2004.

[8] Haralick, R. Document Image Understanding: geometric and logical layout, *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* 8, 1994, 385-390.

[9] Jain, A., and Zhong,Y. Page segmentation using texture analysis, *Pattern Recognition,* vol. 29 (1996), 743-770.

[10] Jain, A., and Bhattacharjee, S. Text segmentation using gabor filters for automatic document processing, *Machine Vision and Application,* 5, 1992, 169-184.

[11] Krishnamoorthy, M., Nagy, G., Seth, S., and Viswanathan, M. Syntactic segmentation and labeling of digitized pages from technical journal, *IEEE Tran. on Pattern Analysis and Machine Intelligence,* 15, 7 (July 1993), 737-747.

[12] Lalanne, D., Sire, S., Ingold R., Behera, A., Mekhaldi, D., Rotz, D. V. A research agenda for assessing the utility of document annotations in multimedia databases of meeting recordings. *3rd Intl. Workshop on MDDE '03, in conjunction with VLDB-2003*, Berlin, Germany, 2003.

[13] Mukhopadhyay, S., and Smith, B. Passive capture and structuring of lectures. *In Proc. of ACM*, 1999, 477-487.

[14] Otsu, N. A threshold selection method from gray-level histograms. *IEEE Trans. on Systems, Man and Cybernetics*, 9, 1 (1979), 62-66.

[15] Ozawa, N., Takebe, H., Katsuyama, Y., Naoi, S., and Yakota, H. Slide identification for lecture movies by matching characters and images. *In Proc. SPIE-Document Recognition and Retrieval XI,* 5296 (2004), 74-81.

[16] Palvidis, T., Zhou, J. Page segmentation and classification. *Computer Vision, Graphics, and Image Processing*, 54, 1992, 484-496.

[17] Shin, C., Doermann, D., and Rosenfeld, A. Classification of document pages using structure-based features. *Int. J. Document Analysis and Recognition*, 3, 2001, 232-247.

[18] Simon, A., Pret, J., and Johnson, A. A fast algorithm for bottom-up document layout analysis, *IEEE Trans. Pattern Analysis and Machine Intelligence,* 19, 1997, 273-276.

[19] Wang, D., and Srihari, S. N. Classification of newspaper image blocks using texture analysis, *Computer Vision, Graphics, and Image Processing,* 47, 1989, 327-352.

[20] Wong, K.Y., Casey, R.G., Wahl, F.M. Document Analysis system. *IBM J. Res. Dev*., 26, 1982, 647-656.

[21] Yang, J.Y., and Ersoy, O. K. *Combined Supervised and Unsupervised Learning in Genomic Data Mining*, Technical Report TR-ECE 03-10, ECE, Purdue University, West Lafayette, IN 47907-2035, 2003.