A Method for Caption Segmentation using Minimum Spanning Tree

Byung Tae Chun, Kyuheon Kim, Jae Yeon Lee

Image Processing Department, Computer & Software Technology Lab.,
ETRI(Electronics and Telecommunications Research Institute)
161 Kajong-Dong, Yusong-Gu, Taejon, 305-350, Korea
E-mail:chunbt@etri.re.kr Fax : +82-42-860-4844

Abstract : Conventional caption extraction methods use the difference between frames or color segmentation methods from the whole image. Because these methods depend heavily on heuristics, we should have a priori knowledge of the captions to be extracted. Also they are difficult to implement.

In this paper, we propose a method that uses little heuristics and simplified algorithm. We use topographical features of characters to extract the character points and use *KMST*(Kruskal minimum spanning tree) to extract the candidate regions for captions.

Character regions are determined by testing several conditions and verifying those candidate regions. Experimental results show that the candidate region extraction rate is 100%, and the character region extraction rate is 98.2%. And then we can see the results that caption area in complex images is well extracted.

1. Introduction

The ability to extract key information from a video program is highly important for a wide variety of applications. Caption in videos can provide important supplemental index informat-ion in video sequences. Examples may include sports, product names, scene locations, speaker names, movie credits, program introductions and special announcements. If caption can be extracted and recognized robustly, we may submit queries such as “Sung Young” and get a list of all movies featuring him, or “stock news” to get relevant financial reports.

Many of the existing approaches on text recognition has focused primarily on optical character recognition in printed and hand-written documents. These systems have attained a high degree of maturity [1].

Conventional researches to extracting caption from images and videos[2,3,4,5,6] suffer from one or more limitations such as locating only the bounding blocks of the caption (therefore requiring human involvement to recognize the characters), sensitivity to font sizes and styles, restrictions on the appearance characteristics of caption can be handled, restrictions on the type of caption that can be extracted(e.g captions only), and inability to handle normal and inverse video modes of captions. Micheal A. Smith and Takeo Kanade briefly describe a method[7] which concentrates on extracting regions from video frames that contain textual information. However, they deal with the preparation of the detected text for standard optical character recognition software. In particular, they do not tray to determine the characters' outline or to segment the individual characters. Another interesting approach to text recognition in scene images is that of Ohya, Shio, and Akamatsu[8]. Text in scene images exists in a 3D space. In view of the many possible degrees of freedom of text characters, Ohya et al restricted them to being almost upright monochrome and not connected in order to facilitate detection

We describe caption extraction method in video using TF(Topographical Features) of character and a graph-theoretic clustering algorithm. The overview of the method is shown in Fig. 1. We use TF of characters to extract the character points and use the graph-theoretic clustering(*KMST*) to extract the candidate regions for captions. Character regions are determined by testing several conditions and verifying those candidate regions.

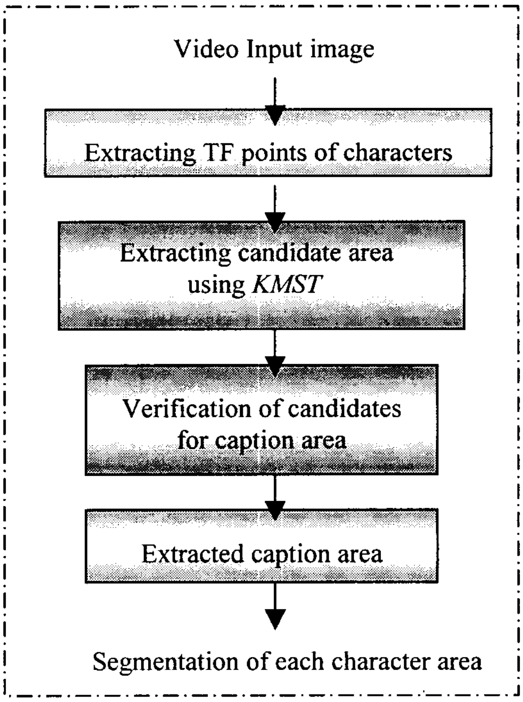
Video Input image
Extracting TF points of characters
Extracting candidate area using *KMST*
Verification of candidates for caption area
Extracted caption area
Segmentation of each character area

Fig. 1. Caption segmentation method using Kruskal minimum spanning tree

2. Caption segmentation using *KMST*

2.1 Extracting topographical feature points of character

If we examine the character regions in a video image, we can see that those regions have some fixed colors and sizes, and are densely located in the horizontal direction, as shown in Fig. 2. However, colors and shapes are not regular in the background area.

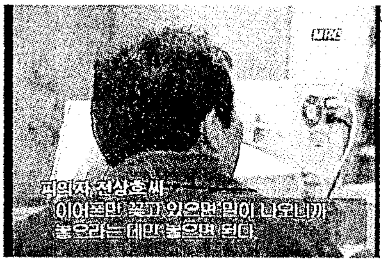

Fig. 2. A caption image from video sequence

Except for the binary (black-and-white) images input by a scanner, the following two cases exist in an ordinary input image. The first case is when the intensity increases from the background region to the character region and decreases from the peak of the character region. That is, the character region is brighter than the background region. The second case is contrary to the first one; the gray value decreases from the background region to the character region and increases from the valley of the character region. That is, the character region is darker than the background region. The extraction method of TF points based on this idea can be presented algorithmically as in Fig. 3. The method is presented in detail in[9]. The n value reflects the width of character stroke and α values reflect the difference for averages of neighborhood pixels. The TF points extracted from the image of Fig.2 are shown in Fig. 4.

```
==================================
 $A_1$ = center pixel value of character
 $A_2$ = average of 3*3 neighborhood
              ⋮
 $A_n$ = average of n*n neighborhood
 if( (($A_1 > A_2+\alpha_1$ ) and ( $A_2 > A_3+\alpha_2$ )
     .. ($A_{n-1} > A_n+\alpha_n$) )          OR
    (($A_1 < A_2+\alpha_1$ ) and ( $A_2 < A_3+\alpha_2$ )
     .. ($A_{n-1} < A_n+\alpha_n$) ))
          character
  else
          not character
==================================
```

where : n = size of mask

$\alpha_1, \alpha_2, \ldots \alpha_n \geq 0$

Fig.3. Algorithm for character points extraction

2.2 Extracting candidate caption area using *KMST*

An graph is usually represented as $G=(V,E)$ and consists of sets of vertices and edges. Let $G=(V,E)$ be an undirected connected graph. A sub-graph $T=(V,E')$ of G is a spanning tree of G iff T is tree. In this paper, we use *KMST*(Kruskal minimum spanning tree) to extract caption area in videos.

We can define the vertices as the extracted TF points in Section 2.1. And then we can define all the edges that connect these vertices. In this paper we analyze the characteristic of the edges based on three edge features : color distance(*Th-cd)*, angle(*Th-a)* and distance*(Th-d*).

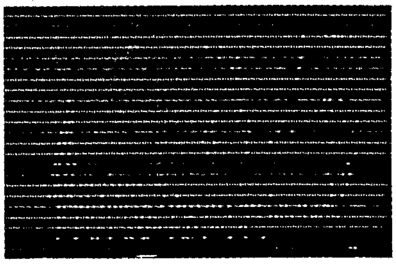

Fig. 4. extracted TF points

A *KMST* graph is constructed by connecting all pairs of vertices for which the edge connecting the two vertices satisfies certain conditions(*1*) on the three edge features. This procedure can be described algorithmically as in Fig. 5. When we assume that we don't have any a priori knowledge about the characters to be extracted, we use default values for edge analysis. The default value for *Th-d* can be calculated by averaging distances between TF points in Fig.4. The default value for *Th-a* is set to 0' ~ 45' when we want to extract a horizontal caption. The decision for *Th-cd* value is very difficult, because this value relates to extracted caption color. We know in experiment result that it is acceptable to set the default value for *Th-cd* to 50.

```
------------------------------------------------------------
  Th-cd   = default (unknown) / user define (known)
  Th-a    = default (unknown) / user define (known)
  Th-d    = default (unknown) / user define (known)

  $E_{CD}$   = Calculate Color Distance for E(v1 ,v2)
  $E_A$    = Calculate Angle for E(v1 ,v2)
  $E_D$    = Calculate Distance for E(v1 ,v2)

  If($E_{CD}$ < Th-cd && $E_A$ < Th-a && $E_D$ < Th-d)
      Satisfied
  Else
      Not satisfied                         ………… (1)
------------------------------------------------------------
```

where :: Th-cd : Threshold value for color distance

Th-a : Threshold value for angle

Th-d : Threshold value fol distance

```
============================================
i = 0
While (E != empty)
{
   $T_i$ = {}
   while(($T_i$ <= n-1) and(E != empty))
   {
        Choose an edge($v_1$, $v_2$) from E when they are
          satisfied by condition(1)
        Delete ($v_1$, $v_2$) from E
        if(($v_1$, $v_2$) dose not create a cycle in $T_i$)
              add($v_1$, $v_2$) to $T_i$
        else
              discard($v_1$, $v_2$)
    }
    if($T_i$ != {})
    {
        i = i + 1;   // Creat New $T_{i+1}$
    }
}
=======================================
```

Fig. 5. Algorithm for extracting candidate caption using *KMST*

Fig. 6 shows the graph made from the feature points of Fig. 4 using the algorithm. And we discard those graphs that has too small a region or has too large a color variance to be a character region. Extracted candidate caption areas are shown in Fig.7 .

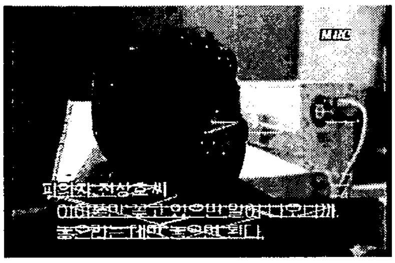

Fig. 6. Constructed graphs using *KMST*

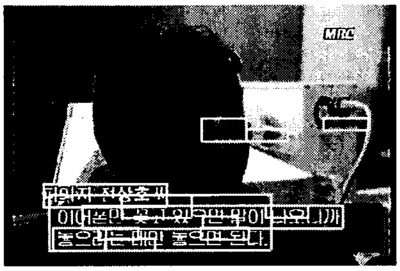

Fig. 7. Extracted character candidate regions

2.3 Verification of candidates of caption area

The post-processing method is offered to decide whether a candidate has characters or not. We perform thresholding for candidate caption area before verification processing. It is shown in Fig.8.

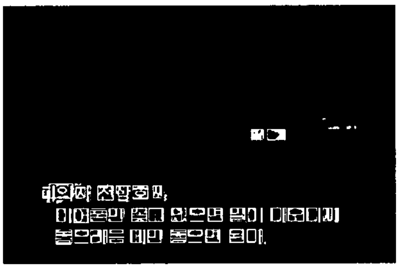

Fig. 8. Shows the candidate caption area after threshoding of image.

The candidate text regions are verified by two stages. In the first stage, the size and other characteristics of the text lines are analyzed to remove non-text regions. If the area, width and height of candidate text region is too small or big, the corresponding regions are discard. If the fill factor is too low, the corresponding regions are discarded. Next, the width-to-height ratio of the blocks is calculated. If it exceeds limits, i.e. dose not lie between *min_ratio* and *max_ratio*, the corresponding regions are also discarded.

In the second stage, the text lines are segmented and the character sizes and the number of characters are examined. If certain conditions are satisfied, the text region is extracted Fig.9 shows verification operation. Fig. 10 shows the extracted text area after verification.

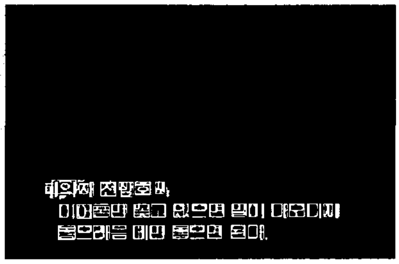

Fig. 9. Verification of candidate of text area

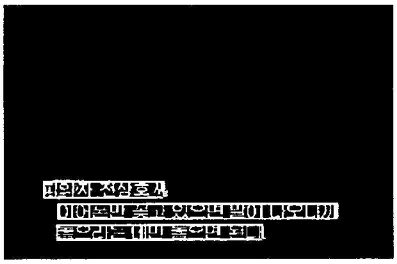

Fig.10. Extracted text area after applying verification

3. Experiments and Results

The program is implemented in Visual C++ Ver.6.0. The experiments have been performed on a Pentium PC with 333 MHz CPU. And used OS is Windows-NT. The MPEG1 data have been used for the experiments, and the data have been acquired with an RT5 image board. The video image is a news images. The captions in images usually stay for 2 to 5 seconds. The image frames were extracted at 3 frames per second.

The experiment shows that the result of extracting candidate areas was 100% when this approach was applied to 3211 frames. Finally, 98.2% of extracting candidate areas was acquired when the verification was performed with the combination of various kinds of background and character colors. Experimental result show that the character extraction method using *KMST* clustering extract character regions accurately.

4. Conclusions

This paper introduced a new method for extracting caption areas of video image using *KMST* clustering. Through a few experiments, the performance is approved. But the following issues should be researched more in the future. The first problem is a method for automatically obtaining the threshold value (*Th-cd* value). The second one is a method of forming the character candidate regions by another clustering method. In addition, the verification methods through learning by examples would be better than that by isolated and multiple area.

References

[1]Shunji Mori, Ching Y. Suen, Kazuhiko Yamamoto,"Historical review of OCR research and Development", Proceeding of the IEEE, Vol.80, No.7, pp.1029-1058, July 1992.

[2] K.Matsuo, K.Ueda, and M.Umeda, "Extaction of character string region on signboard from scene image using adaptive threshold method," IEICE, Vol.J80-D, No.6, pp.1617-1626, 1997.

[3] Rainer Lienhart and Frank Stuber, "Automatic text recogntion in videos," SPIE Storage and Retrieval for Image and Video DB V, Vol. 3022, pp.368-378, Feb. 1997

[4] Shoji Kurakake, Hidetaka Kuwano, Kazumi Odaka," Recognition and visual matching of text region in video for conceptual indexing," SPIE Image and Video Processing IV, Vol. 2666, pp.180-188, Feb. 1996

[5] J. Zhou, D.Lopresti, and T.Tasdizen, "Finding Text in color," Proc. of SPIE on Document Recognition V, pp.130-140, 1998.

[6] B.L. Yeo and B. Liu,"Visual content highlight -ing via automatic extraction of embedded captions on MPEG compressed video", In Proc. SPIE Digital Video Compression : Algorithm and Technologies, Vol.2668, Feb. 1996.

[7]Michael A. Smith and Takeo Kanade, "Video Skimming for Qucik Browsing Based on Audio and Image Characterization," Carnegie Mellon University,Technical Reprot CMU-CS-95-186, July 1995

[8] Jun Ohya, Akio Shio and Shigeru Akamatsu, "Recogniton characters in scene images," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 16, No.2, pp.214-220, 1884

[9] Byung Tae Chun, Younglae Bae and Tai-Yun Kim,"Text extraction in videos using Topographical features of character," The 8th IEEE Inter. Conf. on Fuzzy System (FUZZ-IEEE'99), Vol.2, pp.1126-1130, Aug., 1999.