AN EFFICIENT IMAGE SEGMENTATION TECHNIQUE TO IDENTIFY TARGET AREAS FROM LARGE-SIZED MONOCHROME IMAGES

Young-Geun Yoon[1], Seok-Lyong Lee[2], Ho-Hyun Park[3], Chin-Wan Chung[4]

[1]School of Industrial and Information Systems Engineering, Hankuk University of Foreign Studies, ygyoon@hufs.ac.kr
[2]School of Industrial and Information Systems Engineering, Hankuk University of Foreign Studies, sllee@hufs.ac.kr
[3]School of Electrical and Electronics Engineering, Chung-Ang University, hohyun@cau.ac.kr
[4]Division of Computer Science, KAIST, chungcw@cs.kaist.ac.kr

ABSTRACT:

In this paper, we propose an efficient image segmentation technique for large-sized monochrome images using a hybrid approach which combines threshold and region-based techniques. First, an image is partitioned into fixed-size blocks and for each block the representative intensity is determined by averaging pixel intensities within the block. Next, the neighborhood blocks that have similar characteristics with respect to a specific threshold are merged in order to form candidate regions. Finally, those candidate regions are refined to get final target object regions by merging regions considering the spatial locality and certain criteria. We have performed experiments on images selected from various domains and showed that our technique was able to extract target object regions appropriately from most images.

KEYWORDS: image processing, image segmentation, multimedia database

1. INTRODUCTION

Image segmentation is a process which partitions an image into nonoverlapping regions such that features in each region are similar while they are different in different regions. More formally, if F is a set of all image pixels and $P(\cdot)$ is a homogeneity predicate defined over groups of connected pixels, then image segmentation is a partitioning of the set F into a set of connected subsets or regions $\{S_1, S_2, \ldots, S_n\}$ such that

$$\bigcup_{i=1}^{n} S_i = F \quad , S_i \cap S_j = \phi, \quad for \ \forall i \neq j.$$

The homogeneity predicate $P(\cdot)$ is such that $P(S_i)=true$ for all regions S_i and $P(S_i \cup S_j)=false$ for any two adjacent regions S_i and S_j (Cheng, 2001).

The segmentation of images is categorized into four main approaches: a threshold technique, a boundary-based method, a region-based method, and a hybrid technique which combine boundary and region criteria (Fan, 2001). A threshold technique is well performed when an image has two distinctive regions. But, if the image has ambiguity of boundary, the technique does not cope well with blurring at boundaries. Further, it neglects all of the spatial information of regions in the image. A boundary-based method depends on the assumption that pixel values change rapidly at the boundary between two regions. It detects edges using Sobel, Robert, or Canny operator, and utilizes those edges to identify boundary lines of regions. But those boundary lines identified by the method are only candidates of the boundary of object, not exact boundaries, thus it must be combined with post-processing techniques, such as edge tracking, gap filling and smoothing. A region-based method is based on the assumption that neighboring pixels in a region have similar characteristics with respect to color, intensity, and texture. A well-known algorithm of the method is a *split and merge* technique (Chang, 1994; Hijjatoleslami, 1998). In this algorithm, a seed pixel is selected, and starting from the pixel similar neighboring pixels are merged. This algorithm shows satisfactory, segmentation results, but the selection of an initial seed pixel influences overall performance, and it is not simple to select the pixel. In (Pavlidis, 1990; Chu, 1993), hybrid techniques which integrate results of above approaches are proposed to provide more accurate segmentation. (Pavlidis, 1990) performs segmentation in such a way that it identifies regions using a *split and merge* algorithm, and then modifies or eliminates boundaries among regions, depending on contrast with boundary smoothness and variation of the image gradient along the boundary. (Chu, 1993) introduces a technique which integrates multiple region segmentation maps and edge maps from different channels.

On the other hand, specific image segmentation techniques (Kim, 2005; Jain, 1988; Choi, 2003) have been studied to extract the text contained in natural images using intensity level. Such methods depend on the assumption that the text is distinctly divided from background, which is general characteristic of the text. But it is not appropriate for an image with a lot of noises and unclear objects.

A proposed segmentation method in this paper is using a hybrid approach which combines threshold and region-based techniques, and it is designed and implemented for monochrome images. While existing works select initial seed pixels and merge neighboring pixels of them, our method divides an image into blocks of user-defined size, and determines the representative intensity of a block by averaging the intensity of pixels within the block. Then, an initial block is selected and neighboring blocks with similar characteristics are merged. The order of selecting blocks is from a center block to marginal blocks of image. It is based on experience that in most cases target objects are located in the center of image. Whether neighboring blocks are merged or not is determined by a merging threshold. The proposed algorithm generates regions using blocks instead of pixels and it eliminates pre-processing steps such as filtering and blurring that are adopted by most segmentation methods. It reduces noises since a block itself is represented by the average pixel intensity of the block. This simplification speeds up processing time and reduces the possibility of over-segmentation when a pixel is a merging unit. The proposed algorithm is also designed to be appropriate for images with multiple target regions.

2. REGION-GROWING BASED SEGMENTATION

2.1 Image tiling and threshold determination

Tiling of an image partitions the image of size *MxN* into blocks of size *pxq* ($1 \le p \le M$, $1 \le q \le N$). Then, the image is segmented into *mxn* blocks where $m = \lfloor M/p \rfloor$ and $n = \lfloor N/q \rfloor$. Right and bottom sides of image is not considered by algorithm if *M/p* and *N/q* is not integers. The following equation computes the average intensity $I_{i,j}$ of block(*i,j*) ($0 \le i \le m-1$, $0 \le j \le n-1$).

$$I_{i,j} = \frac{1}{p \times q} \cdot \sum_{h=pi}^{p(1+i)-1} \sum_{v=qj}^{q(1+j)-1} Hist(h,v)$$

where *Hist*(*h,v*) is a value of intensity histogram of location (h,v) of an image. Most natural images have a lot of noises, which requires a pre-processing step such as image filtering or blurring to remove them. But the image tiling has advantage to eliminate overhead of this pre-processing step since the average intensity of pixels in block is used to represent the intensity of the block, which has an effect of removing noises.

On the other hand, a threshold value is determined to convert a monochrome image into a binary image. Using an average intensity value of whole image as a threshold results in poor segmentation since it cause irrelevant neighbor blocks to be merged which is not a part of target object regions. It is because a target object region is not distinct if the intensity average of whole image is quite lower than that of target regions. The problem can be solved by modifying a threshold algorithm slightly to be applied for tiled images.

2.2 Image segmentation algorithm

The proposed segmentation algorithm extracts multiple candidate regions by merging blocks that have similar intensity, and determines target object regions. Segmentation process is divided into (1) tiling image, (2) selecting target blocks with higher intensity than a threshold value, (3) generating multiple candidate regions by merging blocks, and (4) determining target object regions. The tiling step partitions an image into blocks of size *pxq* to process the image by *mxn* blocks instead of *MxN* pixels as merging unit. It is efficient since it reduces amount of information to be processed by 1/(*p*x*q*) compared to the case of using pixels. The block size must be carefully determined considering the accuracy of target regions since it is difficult to represent characteristics of an original image if the size becomes too large.

Each segmented block has a representative intensity which is the average intensity of pixels in the block. All blocks are classified into two classes: *target block* if a block representative intensity is larger than the threshold, and *non-target block* otherwise. The subsequent steps such as block merging and region generation consider target blocks only. In most cases target object regions in monochrome image is found at high-intensity areas. Thus we are able to eliminate non-target blocks from consideration to speed up the process.

The order of evaluating blocks to merge blocks and generate regions is from central to marginal area of an image since target object regions are generally located at the center of the image. When a target block has other target blocks at left, right, top, or bottom side of it, those blocks form adjacent target blocks. For example in Figure 1, if an image has target blocks {1, 3, 4, 6, 8, 12} and non-target blocks {2, 5, 7, 9, 10, 11} then the region generation is performed as follows:

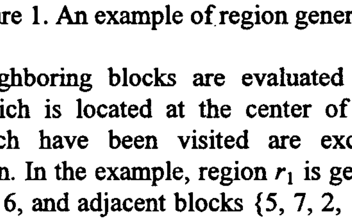


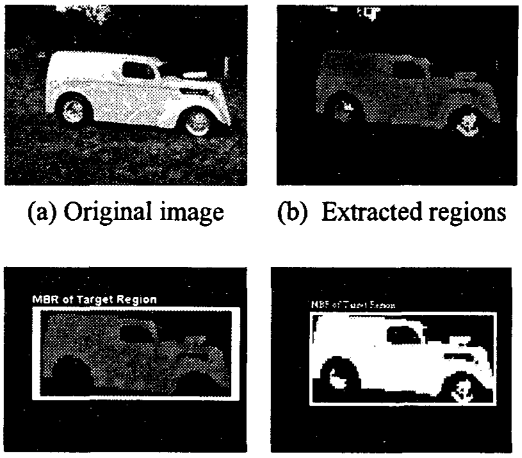
Figure 1. An example of region generation

First, neighboring blocks are evaluated from target block 6 which is located at the center of image. The blocks which have been visited are excluded from consideration. In the example, region r_1 is generated with initial block 6, and adjacent blocks {5, 7, 2, 10} of block 6 are evaluated. But all adjacent blocks are non-target blocks, causing the merging process to be finished for region r_1. Next turn is block 7 to generate another region. Block 7 is however visited before, thus next block 11 is evaluated which is non-target block. Similarly, blocks 10, 9, 5 are evaluated which are all non-target blocks. Next block 1 is found to be a target block, generating new region r_2. There is no adjacent target block for block 1, thus the merging process for the region r_2 is finished and moves to block 3. New region r_3 is generated with block 3 and target blocks 4, 8, and 12 are added to r_3 since they are adjacent target blocks of block 3, 4, and 8, respectively. After all blocks are visited, the process of merging blocks and generating regions is over. The segmentation algorithm ImageSegmentation is described in Figure 2, where function MergeBlock is used for merging a block into a region.

```
Algorithm ImageSegmentation
Input:   original image I, blocksize parameter p,q
Output: set of target regions R
Step1:  if (I != monochrome image) then
             transform I to a monochrome image MI
         end if
Step2:  // image tiling
         produce a blocked image BI by partitioning
         MI into blocks
Step3:  // determine threshold T
         T = DetermineThreshold(BI, t0)
Step4:  // produce regions by merging blocks
         R = ϕ
         queue Q ← blocks to visit
         VISIT_FLAG of all blocks = OFF
         blk = deQueue(Q)
         while (Q is not empty){
            if (VISIT_FLAG(blk) is OFF &
                        intensity(blk) >= T) then
               initialize a region rgn using blk
               rgn = MergeBlock(blk, rgn)
               R = R ∪ rgn
            end if
            blk = deQueue(Q)
         end while
Step7:  return R
```

Figure 2. Algorithm ImageSegmentation

After generating multiple candidate regions, target object regions are determined among generated regions. There are various approaches to determine target object regions depending on features of an image. In this paper, we merge regions if the distance between centers of two regions is within a predefined value, and determine target object regions among the merged regions when the sizes of them are larger than a predefined size. Figure 3 shows this process. Figure 3(a) is an original image, and Figure 3(b) is an image which includes car, wheel, and other noise regions. Object regions without the region merging process are depicted in Figure 3(c), and Figure 3(d) shows a target object region with the region merging process. As shown in the figure, the region merging process segments an image more exactly when multiple regions are generated.



(a) Original image (b) Extracted regions

(c) Object regions without the region merging process (d) Target object regions with the region merging process

Figure 3. Determining target object regions

3. EXPERIMENTAL EVALUATION

We developed a prototype system called RGBSS (Region-Growing Based Segmentation System) which was implemented with Java program language to consider platform independence. Figure 4 is a screen capture to open an image file and extract multiple candidate regions. Each region generated by RGBSS is displayed using different colors to discriminate regions. Users can choose a block size and a threshold value to evaluate various parameters for segmentation. RGBSS uses the area of regions to determine target object regions, and represents a target object region by an MBR(minimum bounding rectangle).

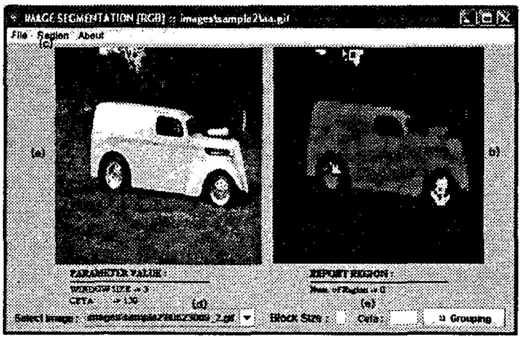
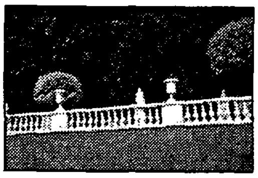


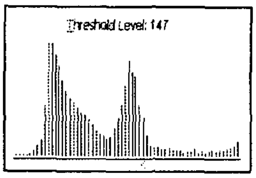
Figure 4. RGBSS system

We conducted experiment on images in which target objects have high intensity compared to the background. In this experiment, we use a simple method to identify target regions using the area of the region, and extract features such as the location of MBR, and the centroid and area of target object regions. We first find multiple candidate regions for images with noises, and then determine target object regions.

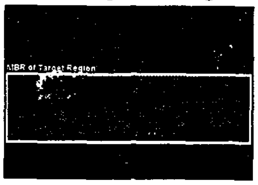
Figure 5 shows an experiment to identify target object regions where the bridge rail is a target object with high intensity. Figure 5(a) is an original image and 5(b) is a histogram of that. In Figure 5(c), a target object region is properly extracted using RGBSS and in Figure 5(d) an image is segmented with a threshold 147 which is most suitable using histogram 5(b). But we can observe that the method based on a threshold technique couldn't identify a target object exactly and thus a secondary process is needed since the segmented image have noises.



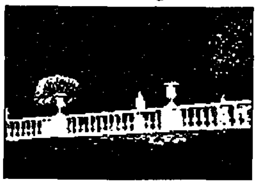
(a) Original image



(b) Histogram



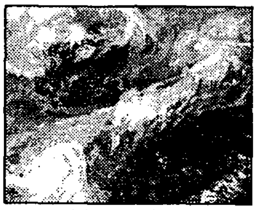
(c) Using RGBSS



(d) Using a threshold

Figure 5. Identifying target object regions

Figure 6 shows an image with multiple candidate regions. RGBSS extracts and displays multiple regions as candidates for user selection. Here a typhoon with high-intensity clouds could be a target object region. RGBSS extracts multiple regions and determines the typhoon located at 7 o'clock direction as a target object region.



(a) Original

(b) Identifying multiple regions

Figure 6. Determining target object regions from multiple candidate regions

4. CONCLUSION

Image segmentation techniques have been recognized importantly since multimedia data are widely used in various domains such as industry, education, culture, and Web. In this paper, we proposed an efficient image segmentation technique for large-sized monochrome images using a hybrid approach which combines threshold and region-based techniques, and implemented it as a prototype system called RGBSS. The proposed algorithm is efficient since it eliminates the preprocessing step such as filtering and blurring that are adopted by most segmentation methods in order to reduce noises. We performed experiments on images selected from various categories and were able to extract proper target object regions from most images. As future improvements, we plan to extend the proposed algorithm to process color images, and use diverse features such as color, edge, and texture, as well as intensity.

REFERENCES

Chang, Y. L. and Li, X., 1994. Adaptive Image Region Growing. IEEE Transactions on Image Processing, Vol. 3, pp.868-872.

Cheng, H. D. et al., 2001. Color Image Segmentation: advances and prospects. Pattern Recognition, 34, pp.2259-2281.

Choi, Y., 2003. Scene Text Extraction in Natural Images Using Hierarchical Feature Combining and Verification. The 2nd KAIST-Tsinghwa Joint Workshop on Pattern Recognition, pp.76-102.

Chu, C. and Aggarwal, J. K., 1993. The Integration of Image Segmentation Maps Using Region and Edge Information. IEEE Transactions on Pattern Analysis and Machine Intelligence, 15, pp.1241-1252.

Fan, J. and Yau, D. K., 2001. Automatic Image segmentation by Integrating Color-Edge Extraction and Seeded Region Growing. IEEE Transactions on Image Processing, 10, pp.1454-1466.

Hijjatoleslami, S. A. and Kittler, J., 1998. Region Growing: A New Approach. IEEE Transactions on Image Processing, 7, pp.1079-1084.

Jain, A. K. and Yu, B., 1988. Automatic Text Location in Images and Video Frames. Pattern Recognition, Vol. 31, No. 12, pp.2055-2076.

Kim, J. S., Kim, S. H., and Choi, Y. W., 2005. Text Region Extraction of Natural Scene Images using Gray-level Information and Split/Merge Method. Journal of KISS: Software and Applications, 32(6), pp.0502 - 0511.

Pavlidis, T. and Liow, Y. T., 1990. Integrating Region Growing and Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 12, pp.225-233.

ACKNOWLEDGEMENT

This research was supported by the Agency for Defense Development, Korea, through the Image Information Research Center at Korea Advanced Institute of Science & Technology.