Edge Color Distribution Transform: An Efficient Tool for Object Detection in Images

Jiqiang Song, Min Cai, and Michael R. Lyu
Dept. Computer Science & Engineering, the Chinese University of Hong Kong, Hong Kong, China
{jqsong, mcai, lyu}@cse.cuhk.edu.hk

Abstract

Object detection in images is a fundamental task in many image analysis applications. Existing methods for low-level object detection always perform the color-similarity analyses in the 2D image space. However, the crowded edges of different objects make the detection complex and error-prone. This paper proposes to detect objects in a new edge color distribution space (ECDS) rather than in the image space. In the 3D ECDS, the edges of different objects are segregated and the spatial relation of a same object is kept as well, which make the object detection easier and less error-prone. Since uniform-color objects and textured objects have different distribution characteristics in ECDS, this paper gives a 3D edge-tracking algorithm for the former and a cuboid-growing algorithm for the latter. The detection results are correct and noise-free, so they are suitable for the high-level object detection. The experimental results on a synthetic image and a real-life image are included.

1. Introduction

Object detection in images can be separated into two levels, low-level object detection and high-level object detection, according to the level of target object. The low-level object detection, also known as image segmentation [1,2], is usually based on the analysis of color similarity. The high-level object detection performs the semantic analysis aided by the priori knowledge [3], e.g. templates of the known objects. Plenty of techniques have been proposed on the low-level object detection since it is the basis of the high-level detection. Classification of traditional image segmentation methods can be found in [1] and [2]. In a recent article, Fan *et al* [4] classified popular image segmentation methods into thresholding techniques, boundary-based techniques, region-based techniques, and hybrid techniques. Edge is an important feature used in these methods. However, they all works in the 2D image space where the edges of different objects are crowded, therefore detecting an object is always interfered with its neighboring objects. Since many methods depend on the color similarity, how to handle both uniform-color objects and textured objects is also a difficulty. Some feature-based methods, e.g. using Gabor filters [5], can detect both of them, but they are complex.

When humans detect a low-level object, they are not bothered with its neighboring objects due to the color differences. We then propose a new method – edge color distribution transform – to segregate the objects with different colors efficiently. It first gets the color of each edge point during the edge extraction, and then transforms all edge points to a 3D Edge Color Distribution Space (ECDS), which is constructed by quantizing the image space and the color space. In ECDS, edge points belonging to different objects are segregated spatially rather than overlapping or touching in the 2D image space. Thus, the object detection in such space is much easier and more precise. Since uniform-color objects and textured objects have different distribution characteristics in ECDS, we propose two different algorithms to detect them.

2. Edge color distribution transform

In this paper, we assume the image is of width W, of height H, and of 256-level grayscale color mode.

2.1 Color operator

We choose Sobel edge operator [6] to extract the edges in an image since it generates double edges. Sobel operator has four directional masks to detect horizontal, vertical, left diagonal and right diagonal edges, respectively. Therefore, it can detect edge points and their local direction as well. Since the color of edge point may be different from the color of object due to the color bleeding caused by compression and decompression, we design a color operator (Fig. 1) in order to smooth the difference. This color operator has also four directional masks corresponding to those of Sobel operator. When

1051-4651/02 $17.00 © 2002 IEEE

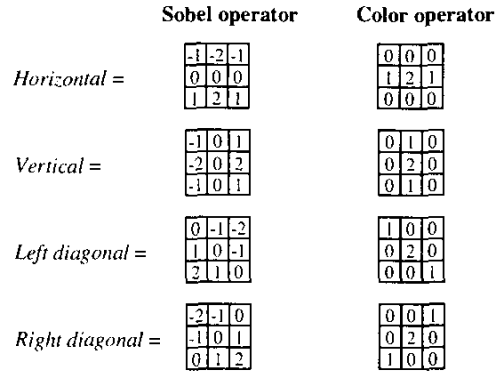


Figure 1. Color operator

doing the edge detection, the Sobel operator determines the direction with the maximal gradient for current point. The color mask corresponding to this direction is selected and centered at current point to calculate its color, which is the quotient of the convolution of the selected mask and the grayscale values of involved pixels divided by 4.

Thus, one detected edge point, which is selected by thresholding its Sobel value, can be described with three parameters (x, y, g), where x and y ($0 \le x \le W$, $0 \le y \le H$) are the image coordinates of the edge point, and g ($0 \le g \le 255$) is its color calculated by the color operator.

2.2 Distance-weighted accumulation

Each edge point can be easily mapped to a point in the 3D X-Y-G space according to its parameters. However, this one-to-one mapping is too fine to tolerate the minor color difference and to get good clustering effect. Moreover, the memory requirement for such space is too large. Thus, the coordinates and the color are quantized to form the ECDS point, which is defined as $\{(mx, my, gl) \mid 0 \le mx \le \left[\frac{W}{\Delta x}\right], 0 \le my \le \left[\frac{H}{\Delta y}\right], 0 \le gl \le \left[\frac{255}{\Delta g}\right]\}$, where '[ ]' is the round operator.

First, the image is divided into a mesh by Δx and Δy, and the grayscale color space is quantized by Δg. Then one point in the X-Y-G space is transformed to the point in ECDS by the following calculation.

$$mx = \left[\frac{x}{\Delta x}\right];\ my = \left[\frac{y}{\Delta y}\right];\ gl = \left[\frac{g}{\Delta g}\right]$$

Since this is a many-to-one mapping, this transform is actually an accumulation process. Thus, each point in ECDS has an accumulator to record its density.

In fact, one point in X-Y-G space contributes to the density of its neighborhood in a spheral area. However, the transform makes it only contributes to the target point in ECDS. To minimize this quantization error, we use a distance-weighted accumulation to make it contribute to the 3×3×3 neighborhood of the target point. The neighbors are classified into three classes. Those offsetting from the target point in only one dimension are *Close* neighbors; those offsetting in only two dimensions are *Medium* neighbors; others are *Far* neighbors. The weight value of each class is deduced from its Euclidean distance to the target point. Table 1 shows the weight values for all points in the neighborhood.

Relation with target point	Euclidean distance	Weight value
Target point	0	15
Close neighbor	1	5
Medium neighbor	$\sqrt{2}$	4
Far neighbor	$\sqrt{3}$	3

Table 1. Weight values of 3×3×3 neighborhood

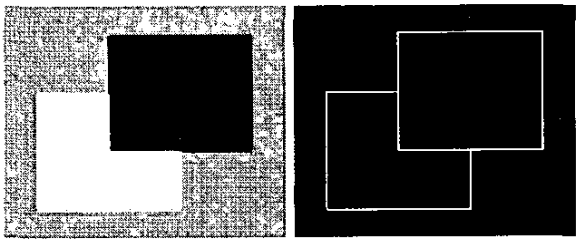
Each point transformed to the ECDS will increase the accumulators of both the target point and its neighbors by the weight values defined in Table 1. Thus, the ECDS reflects the density of both edge and color correctly.

2.3 ECDS

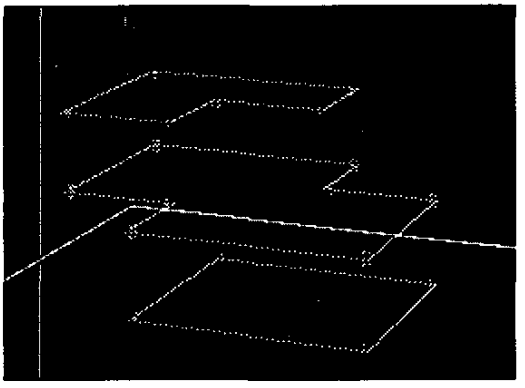
Since the mx-my plane is transformed from the image space by linear quantization, the original spatial relation among the edges of an object is kept in ECDS. Due to the color difference, the edges of different objects, which were overlapped or very close in the image space, are segregated. Thus, a clustered part in ECDS always indicates an object, which is not true in the image space.

Figure 2 shows the ECDS of a synthetic image containing a white rectangle and a black rectangle on the gray background. Two rectangles are partially overlapped (Fig. 2a). Fig. 2b shows the result of edge detection using Sobel operator. The edges of two rectangles touch each other. However, they are totally separated in ECDS, as shown in Fig. 2c, where the horizontal plane is the mx-my plane and the vertical coordinate axis is the gl axis. The top-layer polygon is the edge of the white rectangle, and the bottom-layer rectangle is that of the black rectangle. The middle-layer polygon is the edge of background. Obviously, detecting the rectangles in ECDS is very easy.

The synthetic image is simple since it only contains uniform-color objects and background, so that the edges are consistent and continuous. Real-life images usually contain many textured objects and more complex background. Figure 3 shows an example of real-life image captured from a news video (Fig. 3a), which contains human face, suit, tie, map, icon, logo, and captions. In Fig. 3b, the edges of textured objects, such as captions, tie and logo, are very dense. The edge of suit is broken by the overlapped captions. These factors make the object detection in the image space more complex and error-prone. The ECDS is very helpful in these cases. In Fig. 3c, where the brightness of point is proportional to its density, the edges of textured objects are of high density so that



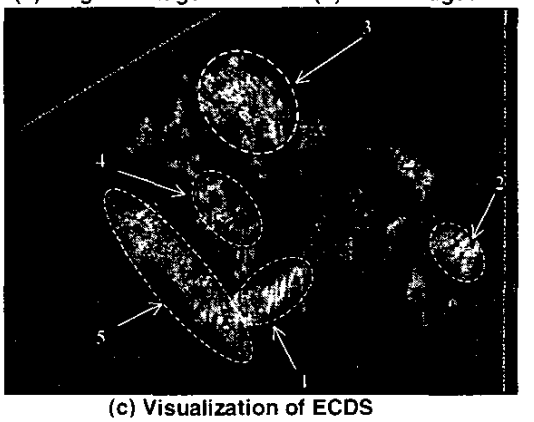
(a) Original image (b) Sobel edges



(c) Visualization of ECDS

Figure 2. ECDS of a synthetic image

(a) Original image (b) Sobel edges



(c) Visualization of ECDS

Figure 3. ECDS of a video image

they are easy to be detected. For example, part 1 is the tie; part 2 is the logo; part 3 is the icon; part 4 and 5 are captions. The edges of suit locate in the different color levels from that of the overlapped objects, so that the detection will be less bothered because there are a certain spatial distance between the interesting edge and the overlapped edges.

3. Object detection

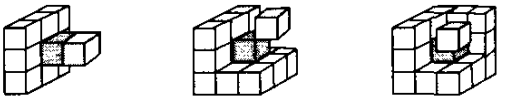
Since the uniform-color objects and textured objects have very different distribution characteristics in ECDS, they should be detected in different ways.

3.1 Uniform-color object detection

The edges of uniform-color objects are isolated and continuous in ECDS, so we propose a 3D edge-tracking algorithm to detect them.

First, the detection scans the ECDS sequentially using a 4×4×3 window stepping 2 in each dimension, and counts the number N of non-zero-density points in the window. The window size in *mx-my* plane is relatively large since we want to get reliable linear feature. The window size in *gl* axis is smaller because the edge color of uniform-color object does not vary much. If N is not less than 4, there is possible a line crossing this window. Then, the straight-line Hough transform is applied to this window to find the collinear points. If a sequence of points is found collinear and both the first one and the last one of them reach the boundary of the window, the sequence of points will be accepted as a seed for tracking the edge.

The edge tracking begins from the extremities of the seed and detects the neighboring points to grow the seed. The tracking processes toward two opposite directions of the seed are performed sequentially. If the tracking from one extremity reaches the other extremity, i.e., the edge is closed, the tracking of the other direction will be cancelled. Since the tracking goes along the edge direction, it is unnecessary to check all connected neighbors, but only necessary to check the *Directional Neighbors* that are determined by the local tracking direction. Let the current extremity point of the seed be P and its last point be P'. The direction of $\overrightarrow{P'P}$ is the current tracking direction, which determines the directional neighbors (Fig. 4). Directional neighbors include all those neighbors that are not neighbors of P' since the neighbors of P' have already been processed in the last tracking step. For example, Fig. 4a-4c show the examples of directional neighbors when P' is the close neighbor, medium neighbor, and far neighbor of P, respectively. The successive tracking point is selected from directional neighbors first by the color



(a) (b) (c)

Saturated gray – P; Light gray – P'

Figure 4. Directional neighbors

similarity with P and second by the distance from P. The tracking stops when no successive point can be found or the edge is closed. The points having been tracked are removed from ECDS to avoid the repetitive detection.

After all linear edges being detected, the merging process starts to merge those edges belonging to the same object but being broken by overlapped objects. It checks each gap between near extremities of neighboring edge pair in similar color levels. If the gap can be bridged by edges of other objects in the *mx-my* plane, the gap will be filled to merge two edges.

After merging, those edges that are closed or whose two extremities are both on the boundary of ECDS are accepted, and the others are discarded. For closed edges, there is an additional check: the color inside the edge should be same as the color of edge. This check is to eliminate the background edges such as the middle-layer edge in Fig. 2c. Finally, each edge indicates a uniform-color object in the image. Fig. 5a shows the polygonized edges detected from the image in Fig. 3.

3.2 Textured object detection

Since the edges of textured objects cluster in ECDS, we use a cuboid-growing algorithm to detect them.

First, the detection scans the ECDS sequentially using a $w \times w \times w$ cubic window stepping 1 in each dimension to detect the high-density parts. If the average density of all points in the window is not smaller than MIN_DENSITY, which is determined by Δx, Δy, and Δg, this cube will be recorded as a seed to detect the object. Then, all seeds are processed in the descending order of the average density. For a cuboid whose boundary is defined as (*mx_max, mx_min, my_max, my_min, gl_max, gl_min*), the cuboid-growing algorithm is described as follows.

```
BOOL stop = False;
WHILE (NOT stop) {
  Dmx+ = density after mx_max + growing_step;
  Dmx− = density after mx_min − growing_step;
  Dmy+ = density after my_max + growing_step;
  Dmy− = density after my_min − growing_step;
  Dgl+ = density after gl_max + growing_step;
  Dgl− = density after gl_min − growing_step;

  IF (maximum of 6 densities < MIN_DENSITY)
  THEN stop = True;
  ELSE grow the cuboid in the direction with the
       maximum density for one growing_step.
}
```

Window size w is 3 in this algorithm. Using a smaller w, more textured objects will be detected; however, the detection will be more noise sensitive. The *growing_step* is 1 in this algorithm. It can be adjusted to speed up the detection. The points in the detected cuboid are also removed to avoid the repetitive detection. This cuboid-growing algorithm is robust for texture types and noise.

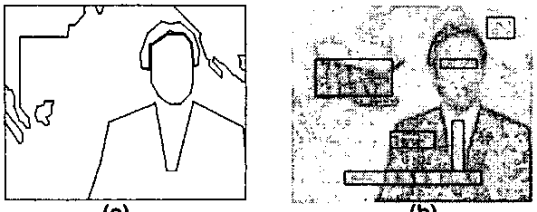


Figure 5. Result of object detection

Fig. 5b shows the textured objects detected from the image in Fig. 3. The limitation of this algorithm is that it cannot yield precise boundaries for arbitrary-shaped objects.

4. Conclusions

This paper proposes to detect objects in a new edge color distribution space rather than in the image space. In the 3D ECDS, the edges of different objects are segregated and the spatial relation of a same object is kept as well, which makes the object detection easier and less error-prone. Since uniform-color objects and textured objects have different distribution characteristics in ECDS, this paper gives a 3D edge-tracking algorithm for the former and a cuboid-growing algorithm for the latter. The detection results are correct and noise-free. The quantization intervals Δx, Δy of the ECDS transform have a direct impact on the detection rate. Smaller Δx, Δy brings all objects up, while larger ones would only bring the larger objects. Future research will focus on improving the textured object detection algorithm.

Acknowledgement

The work described in this paper was fully supported by two grants from the Hong Kong Special Administrative Region: the Hong Kong Research Grants Council under Project No. CUHK4222/01E, and Innovation and Technology Fund, under Project No. ITS/29/00.

References

[1] R.M. Haralick and L.G. Shapiro, "Survey, Image segmentation techniques", *Computer Vision, Graphics, and Image Processing*, 1985, vol.29, pp. 100-132

[2] N.P. Pal and S.K. Pal, "A review on image segmentation techniques," *Pattern Recognition*, 1993, vol.26, pp. 1277-1294

[3] T. Gevers and A.W.M. Smeulders, "Color-based object recognition," *Pattern Recognition*, 1999, vol.32, pp. 453-464

[4] J. Fan, D.K.Y. Yau, A.K. Elmagarmid, and W.G. Aref. "Automatic image segmentation by integrating color-edge extraction and seeded region growing." *IEEE Trans. Image Processing*, 2001, 10(10): 1454-1466.

[5] A.K. Jain, N.K. Ratha, and S. Lakshmanan, "Object detection using Gabor filters", *Pattern Recognition*, 1997, 30(2): 295-309

[6] W.K. Pratt, *Digital Image Processing*, New York, NY: Wiley, 1978.