J. Vis. Commun. Image R. 36 (2016) 159–171

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

PEL: A Predictive Edge Linking algorithm ☆

Cuneyt Akinlar *, Edward Chome

Anadolu University, Computer Engineering Department, Eskisehir, Turkey

A R T I C L E I N F O

Article history:
Received 15 December 2015
Accepted 28 January 2016
Available online 6 February 2016

Keywords:
Edge detection
Edge linking
Edge segment detection
Canny
CannySR
Edge Drawing (ED)
Contour detection
gPb

A B S T R A C T

We propose an edge linking algorithm that takes as input a binary edge map generated by a traditional edge detection algorithm and converts it to a set of edge segments; filling in one pixel gaps in the edge map, cleaning up noisy edge pixel formations and thinning multi-pixel wide edge segments in the process. The proposed edge linking algorithm walks over the edge map based on the predictions generated from its past movements; thus the name Predictive Edge Linking (PEL). We evaluate the performance of PEL both qualitatively using visual experiments and quantitatively within the precision-recall framework of the Berkeley Segmentation Dataset and Benchmark (BSDS 300). Both visual experiments and quantitative evaluation results show that PEL greatly improves the modal quality of binary edge maps produced by traditional edge detectors, and takes a very small amount of time to execute making it suitable for real-time image processing and computer vision applications.

1. Introduction

Edge detection is a very important and fundamental first step in many computer vision and image processing applications. A traditional edge detection algorithm [1–4] takes a grayscale image as input and produces a binary edge map (BEM) as output, where an edge pixel (edgel) is marked (e.g., its value in the edge map is 255), and a non-edge pixel is unmarked (e.g., its value in the edge map is 0).

The binary edge maps produced by traditional edge detectors are usually of low quality, consisting of gaps between the edgels, unattended edgels and noisy notch-like structures, ragged and multi-pixel wide edgel formations, etc. An example of such an edge map with low quality artifacts is shown in Fig. 1 for the famous Lena image. This edge map was obtained by the OpenCV implementation of the widely-used Canny [2] edge detector (cvCanny) [5], which is the fastest known Canny implementation. To obtain this edge map, the input image was first smoothed by a Gaussian kernel with $\sigma = 1.5$ (using cvSmooth from OpenCV), and cvCanny was called with low and high threshold values set to 20 and 40 respectively, and the Sobel kernel aperture size set to 3. Fig. 1 also shows the close-up views of two separate sections of the edge map to illustrate the low quality artifacts, which can be grouped in three categories as follows: (1) There are discontinuities and gaps between edgel groups as can clearly be seen in the close-up views of the two enlarged sections of the edge map. Some of these gaps need to be filled up. (2) There are noisy, unattended edgel formations and notch-like structures. This is more evident in the close-up view of the upper-left corner of the edge map. These noisy artifacts needs to be removed. (3) There are multi-pixel wide edgel formations in a staircase pattern especially around the diagonal edgel formations (both 45 degree and 135 degree diagonals). Such formations can be seen in many places in the edge map, and they need to be thinned down to 1-pixel wide chains.

To improve the modal quality of the binary edge maps produced by traditional edge detectors, edge linking methods have been proposed in the literature [7–24]. The goals of these methods are commonly to remove noisy edgel formations and clean up the edge map, and to fill in gaps between edgels to form longer edgel groups.

Snyder et al. [7] is one of the first researchers to present a method to deal with errors in edge detector results. The authors propose a method to close gaps in edge maps while preserving edge structure and connectedness based on the concept of a chamfer map. Xie [6] presents a method to link edge pixels for the purpose of line segment detection. The method makes use of the concepts of horizontal edge element and causal neighborhood window to realize edge linking and consequently line segment detection. Given a binary edge map, the author performs linking of pixels aligned on the same image line into what is called a horizontal edge element, which is then converted to a line segment. The

☆ This paper has been recommended for acceptance by K. Chung.

* Corresponding author.
E-mail addresses: cakinlar@anadolu.edu.tr (C. Akinlar), edwardchome@anadolu.edu.tr (E. Chome).

http://dx.doi.org/10.1016/j.jvcir.2016.01.017
1047-3203/

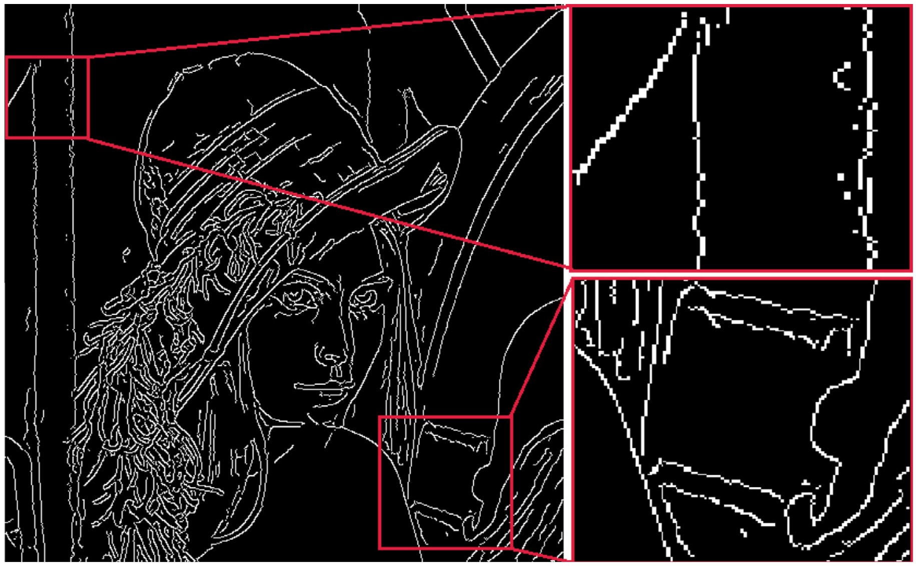

Fig. 1. Left: Lenas edge map result by OpenCV Canny with low and high threshold values of 20 and 40 respectively. The image was first smoothed by a Gaussian kernel with $\sigma = 1.5$. Right: Close-up view of two sections of the edge map showing low quality edge group formations: Discontinuities and missing edgels, noisy, notch-like edge group formations, and multi-pixel wide staircase edgel structures.

algorithm performs poorly especially in highly textured regions. Basak et al. [8] also consider the problem of line detection via edge linking and propose two neural network architectures.

Farag and Delp [9] define edge linking as a graph search problem. Beginning at a start node, they use the A^* search algorithm to reach a goal node. The search makes use of the gradient magnitude and angles, and the swath of edge information estimated by the zero crossings of Laplacian of Gaussian operator. They propose a linear path metric function to guide the A^* search. To limit the search space, the algorithm only looks at three neighbors for transitioning, which reduces the running time but affects the accuracy of the algorithm. The biggest problem with this algorithm is the large number of parameters that needs to be adjusted. Also some statistical parameters and a priori information about the edge map needs to be known for the algorithm to work correctly. Zhu et al. [10] present an algorithm to link discontinuous edge segments. They model an edge map as a global potential field with energy dispositions at detected edges. A directional potential function measures the energy charges and guides the linking process to fill the gaps between the broken edge fragments. Their method can fill small gaps between broken edge fragments successfully, but produces unacceptable results when the gaps are large or the image is noisy.

Saber et al. [11] propose a method for segmentation of the color images. They model the segmentation map as a Gibbs random field and use the gradient magnitude of the 3-channel color image to find the regions. The boundaries of the closed regions constitute the final linked edge map. Maeda et al. [13] also concentrate on image segmentation and propose incorporating a linking algorithm based on a directional potential function into an edge preserving smoothing filter. They show that the unnecessary details of the image are smoothed out before region growing is performed. Hajjar and Chen [12] propose an edge linking algorithm and its VLSI implementation for real-time edge linking. Their method is based on the break points' direction and weak level points and try to fill the gaps between edge groups. Ghita and Whelan [14] also propose an algorithm to close the gaps in an edge image by local analysis of the edge break points (terminators). They mark the edge termination points, and determine the connection path between different edge termination points by an analysis of the local edge structure. Their algorithm requires a single pass through the edge map and can be applied without a priori information. Shih and Cheng [15] apply mathematical morphology to edge linking to fill in the gaps between edge segments. Broken edge segments are extended along their slope directions by using the adaptive dilation operation with suitable elliptical structuring elements. They also apply thinning and pruning as a post-processing step. Although their algorithm is shown to fill gaps in the edge map of an ellipse, it is not clear how it can be applied without a priori information about the edge map.

Sappa and Vintimilla [18] present an algorithm that takes in an edge map and the original intensity image, and generates closed contours using graph theoretical approaches. In the process, they close small gaps and remove spurious edge pixels, but their algorithm is more tailored towards image segmentation than edge linking. Lu and Chen [16] apply ant colony optimization to compensate broken edges. They propose four moving policies for ants and apply a finite number of iterations to fill up the broken edge segments. Jevtic et al. [19] propose a similar ant based edge linking algorithm to combine broken edges. Wang and Zhang [17] propose the use of application-specific local neighborhood to compute edge direction and geodesic distance for measuring proximity between candidate edge points to be linked. Lin et al. [20] make use of an edge linking algorithm with directional edge-gap closing to produce complete edge-links that are then used for lane detection.

Flores et al. [21] present a method for the segmentation of intensity images that combines an optical contouring technique with edge linking. Ji et al. [22] present a method for segmentation of satellite imagery. Their idea is to make use of the gradient magnitude and directions and perform a heuristic A^* search to link the original edge points. Their algorithm is also tailored toward image segmentation. Guan et al. [23] recently proposed a Partial Differential Equation (PDE)-based method to link the edges to obtain closed contours for image segmentation. They then show their algorithm's performance on the segmentation of cell images.

We can classify most of the edge linking algorithms found in the literature in two categories: (1) Those that try to fill the gaps between broken edge groups in the edge map, which is only part