Pattern Recognition 46 (2013) 2067–2077

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

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

Unsupervised edge detection and noise detection from a single image

Kunal Ray *

Proof and Experimental Establishment, Chandipur 756025, India

ARTICLE INFO

Article history:
Received 1 March 2012
Received in revised form
28 September 2012
Accepted 27 January 2013
Available online 4 February 2013

Keywords:
ADM edge detector
Non-maximal suppression
Ground Truth (GT)
Noise

ABSTRACT

Edge detection is one of the oldest image processing areas that are still active. An important current area of study involves development of unsupervised edge detection algorithms. In this work a paradigm of unsupervised edge detection is proposed that is based on the computational edge detection approach introduced by Canny. It is a simple and computationally cheap technique that achieves non-trivial results. Additionally as a byproduct it generates information about the content and severity of noise in the image. The proposed technique uses a fast edge detector to generate the initial edge mask and subsequently optimizes that by studying the behavior of a proposed details estimator. The study of the same estimator also offers insight about the noise characteristics of the image.

© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

Edge detection in the field of image processing is one of the most important areas of interest. Edges in images provide primary information about the objects present in a scene and the boundaries between them. It is often the crucial first step before sophisticated algorithms for object identification, feature extraction, etc can be applied. Over the time a number of stable paradigms have emerged for edge detection. Some are computationally cheap though may not be very reliable while some are reliable though computationally expensive. Broadly we have the differencing based methods [1,2], anisotropic or non-linear diffusion based methods [3–5] and the active contour based methods [6]. There are also methods which may not be directly categorized among these, like logical operator based or morphology based edge detectors [7,8]. The differentiation based methods are most commonly used and are often very fast compared to others but may not produce closed edge contours. The diffusion or selective smoothing methods use adaptive kernels for image smoothing and can produce good edges in presence of noise but the computational costs are high for them. The active contour based methods also are computationally expensive but can be of use as a preprocessing step for applications involving object or feature extraction.

The basic idea behind differentiation based edge extraction is identification of local maxima or zero crossing of first or second order derivative. However as they are strongly sensitive to noise a preprocessing smoothing step is often essential. Canny [1] in his seminal work identified the three major criteria for good edge detection: good detection, good localization and poor spurious response. It is further shown that for optimal detection first order derivative of Gaussian kernel should be used. The problem arises when we need to identify the optimal size of Gaussian kernel. Small scale ones provide good edges but cannot suppress strong noise while large scale ones suppress noise but also blur genuine edges considerably.

Additional difficulties arise when we consider applying the post-processing step of hysteresis, as proposed by Canny. Hysteresis is based on use of a high threshold to identify a set of pixels as edge points without considering their connectivity information; pixels above the threshold are taken as the edge points. Subsequently a low threshold is used which indicates pixels below it to be non-edges. To classify the pixels between the thresholds, the connectivity information between them and the pixels above the high threshold, is used.

The element of user input in these steps make adaptation of differencing based edge detectors, which follow the paradigm set by Canny, for unsupervised use quite difficult. It has been shown that the performance of the Canny edge detector primarily depends upon the efficiency of the steps of non-maximal suppression (NMS) and hysteresis [9], and use of improper threshold in hysteresis can be quite devastating. Equally, inability of proper smoothing of noise can defeat the objective completely. So any effort to fully automate such edge detectors requires a mechanism to get an idea about the noise present in the image and also a mechanism to automatically guess the hysteresis thresholds.

There have been efforts to automate the Canny edge detector. The primary approach has been directed towards the guessing of

* Tel.: +91 9437409362.
E-mail address: kunalrayind@gmail.com

0031-3203/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2013.01.029

the hysteresis thresholds [10–14]. This may be because there are well established methodologies for noise estimation already [18–20]. However most of the noise estimation and removal techniques work on a sequence of images and are computationally quite expensive. The works approaching the hysteresis problem mostly result in algorithms that are often complex and computationally costly involving trying out sets of parameter values and computing mathematical estimators. They primarily aim toward either making the manual determination of hysteresis thresholds easier, or the unsupervised determination of these thresholds, or in a novel approach searching for the instability zone between the thresholds. In a different approach, edge detector has been proposed with choice of threshold based on statistical variability of gradient vector at each pixel [15].

The problem of automating the edge detection process for differentiation based methods can be summarized into mostly three objectives: (1) solving the noise problem, (2) solving the hysteresis problem and (3) to make sure that the procedure is simple and fast. The first problem mainly requires automated detection and quantification of noise in image, as use of isotropic or anisotropic Gaussian kernel [16,17] for successful noise suppression is well established. The second problem calls for some kind of characterization of the pre-hysteresis edge map output against which the threshold problem can be optimized. This is difficult as two thresholds need to be identified. The third problem is more implementation oriented but is no less important.

We propose in this paper a simple algorithm which aims to solve the second problem and as by-product of which we get the direction for solving the first problem. The edge detector we propose to use is the Absolute Difference Mask (ADM) [26], as it is one of the fastest edge detectors and can be easily implemented in hardware for high speed applications. We subsequently propose in the paper a modified non-maximal suppression in place of the usual one and dropping of the step of hysteresis. We introduce an estimator against which the edge map after the non-maximal suppression is characterized. We further show that by studying the behavior of the estimator against the edge map the presence of noise as well as its severity can be estimated.

This paper is organized as follows. Section 2 discusses the ADM edge detector and its implementation. Section 3 introduces the modified NMS algorithm and the proposed estimator for characterization of the post NMS edge map. It further discusses the use of the estimator. Section 4 discusses the noise problem and the use of the estimator for noise estimation. Section 5 reports evaluation of the proposed algorithm. The paper is concluded in Section 6.

2. The absolute difference mask (ADM) edge detector

The ADM edge detector is one of the fastest edge detectors to find out the edge strength and edge direction and subsequently generating the final edge map. The steps for finding the edge strength and direction can be broken into three steps: preparing inputs to find the absolute differences for a pixel, finding all the absolute differences for the pixel and finally using the absolute differences to get the edge strength and edge direction for the pixel. For finding out the relevant data it uses the values for the eight immediate neighbors and eight further neighbors. The scheme can be understood as follows:

For a pixel at location (i,j) if the intensity is $I(i,j)$ then we have:

$$V_u = I(i,j-1)+I(i,j-2) \tag{1a}$$

$$V_l = I(i,j+1)+I(i,j+2) \tag{1b}$$

$$H_r = I(i+1,j)+I(i+2,j) \tag{2a}$$

$$H_l = I(i-1,j)+I(i-2,j) \tag{2b}$$

$$Pd_u = I(i-1,j-1)+I(i-2,j-2) \tag{3a}$$

$$Pd_l = I(i+1,j+1)+I(i+2,j+2) \tag{3b}$$

$$Nd_u = I(i+1,j-1)+I(i+2,j-2) \tag{4a}$$

$$Nd_l = I(i-1,j+1)+I(i-2,j+2) \tag{4b}$$

After these inputs have been found the absolute differences are:

$$V = |V_u - V_l| \tag{5a}$$

$$H = |H_r - H_l| \tag{5b}$$

$$Pd = |Pd_u - Pd_l| \tag{5c}$$

$$Nd = |Nd_u - Nd_l| \tag{5d}$$

Subsequently, the edge strength and direction at (i,j) is given by

$$\mathrm{Edge}(i,j) = \max(V,H,Pd,Nd)/2 \tag{6a}$$

$$\mathrm{dir}(i,j) = \mathrm{dir}(\min(V,H,Pd,Nd)) \tag{6b}$$

The direction as generated by Eq. (6b) is the direction quantized to four values of vertical, horizontal and the two diagonals.

We implemented this edge detector by setting up arrays for the four parameters V, H, Pd and Nd. The data were generated by accessing the specified neighboring pixel locations for each pixel though a single for loop. Once Eqs. (1)–(4) above are evaluated, Eq. (5) are evaluated and the four values are kept in a single array. Finally Eq. (6a) and 6b are evaluated and the resulting edge map values and the edge direction are stored. This algorithm is extremely fast compared to any mask processing based edge map detector. The speed of this algorithm is primarily due to the fact that only addition and subtraction operations are performed in contrast with the convolution performed by the mask based edge detectors.

However the edge map generated by this algorithm is not usable directly as the procedure is extremely sensitive and generates huge number of edges of variable thickness. To clean up the edge map and generate single pixel edge signature subsequent post-processing step(s) are needed.

3. The modified non-maximal suppression algorithm and the proposed estimator

The post-processing step of non-maximal suppression is designed for retaining only the most powerful edge map signature and suppressing the rest. This results in a single pixel wide edge signature which is often essential for application of subsequent sophisticated algorithms. This step is necessary for retaining only the strongest part of the detected edge signature and has been shown to be a major component behind the success of the Canny edge detector. However the non-maximum suppression also generates quite a large number of edges and to filter the output Canny introduced the two threshold hysteresis. The traditional non-maximal suppression algorithm takes as input the edge strength and edge direction map. Eq. (7) illustrates the algorithm:

$$\forall(i,j)\ ,\ I(i,j) = \text{no\_edge if} \begin{cases} I(i,j) = \text{zero\_edge\_magnitude} \\ I(i,j) < I(i+n_x,\quad j+n_y) \\ I(i,j) < I(i-n_x,\quad j-n_y) \end{cases}$$
$$\forall(i,j), I(i,j) = \text{edge otherwise}$$
$$\text{where } -1 \le n_i \le 1,\quad i = x,y \tag{7}$$

Download English Version:

https://daneshyari.com/en/article/531060

Download Persian Version:

https://daneshyari.com/article/531060

Daneshyari.com