

EDGE DETECTION BY SELECTION OF PIECES OF LEVEL LINES

Enric Meinhardt-Llopis

Universitat Pompeu Fabra
Departament de Tecnologia
enric.meinhardt@upf.edu

ABSTRACT

We propose an edge detector based on the selection of well contrasted pieces of level lines, following the proposal of Desolneux-Moisan-Morel (DMM) [1]. The DMM edge detector has the problem of *over-representation*, that is, every edge is detected several times in slightly different positions. In this paper we propose two modifications of the original DMM edge detector in order to solve this problem. The first modification is a post-processing of the output using a general method to select the best representative of a bundle of curves. The second modification is the use of Canny's edge detector instead of the norm of the gradient to build the statistics. The two modifications are independent and can be applied separately. Elementary reasoning and some experiments show that the best results are obtained when both modifications are applied together.

Index Terms— Edge detection, Canny, computational gestalt

1. INTRODUCTION

Edge detection is the task of finding the boundaries between the objects that appear in a digital image. From a mathematical point of view, the detection of edges amounts to find the discontinuities of a function. Because discontinuities are not well defined for discrete digital images, some ingenuity is needed to compute them, and this accounts for the disparity of edge detection methods. People have been using two-dimensional edge detection for years for many tasks. For example, to obtain a visually appealing “primal sketch” of a picture, to reduce the amount of information present in a picture, to get a manageable list of “features” to perform registration of two images, or shape matching; and finally as a first step towards the segmentation of the image into regions.

Most, if not all, edge detectors are based on finding the points of the image where the gray-value changes fast. Equivalently, finding the points where the norm of the gradient is high. Usually, this involves thresholding the response of a discrete differential operator. Canny's edge detector [2, 3] is the most widely used in this class. It needs three parameters: the scale of an initial Gaussian filtering, and two thresholds

of the gradient norm to select a set of connected pieces of boundaries. The optimal values of these parameters depend heavily on the image content.

In the quest for a parameter-free edge detector, Desolneux, Moisan and Morel proposed to use computational gestalt theory for edge detection. Their approach started with the observation that the discontinuities of a function are formed by pieces of level lines where the norm of the gradient is high. The algorithm introduced in [1] consists in traversing the family of all the pieces of all the level lines, and picking those pieces where the gradient is high. This selection is done by comparing the gradient of each piece to the distribution of the gradient over the whole image, in such a way to avoid any manual thresholding. This method is fast and gives good results, but it has a problem of edge replication. The aim of this note is to address this problem.

Let us outline the plan of this work. We start in Section 2 by recalling DMM edge detector. In Section 3 we explain how to filter the output of this edge detector to remove replicated edges. In Section 4 we propose a slight modification of the original detector in order to obtain better localized curves. Finally, in section 5 we compare the results of Canny's and Desolneux's edge detector with ours.

2. THE DMM EDGE DETECTOR

Let us briefly recall the DMM edge detector introduced in [1]. Following this reference, the set of curves which are the output of this detector will be called the *meaningful edges* of the image.

Let Ω be a rectangular grid of pixels. Let $I : \Omega \rightarrow \{0, \dots, 255\}$ be a gray-scale discrete image and $g : \Omega \rightarrow [0, +\infty)$ be the norm of its gradient (henceforth called the *contrast*), computed via finite differences. As in [1], level lines are the boundaries of the level sets $\{x \in \Omega : I(x) \geq \lambda\}$, $\lambda \in \{0, \dots, 255\}$ of I . The distribution of values of g will be treated as a background model of noise. For that, we will need the repartition function of g , defined by $H(\mu) = \frac{\#\{x \in \Omega \mid g(x) \geq \mu\}}{\#\Omega}$ and the total number N_p of connected pieces of level lines (which is computable from the fact that a curve of l discrete points has $l(l-1)/2$ possible connected pieces,

each piece being determined by its two endpoints). The DMM algorithm tries to find the well contrasted (in a statistical sense) pieces of level lines of the image and it depends on a positive parameter ϵ . We can summarize it as:

Algorithm D (Meaningful edges)

1. For each level line C of I :
 - (a) For each connected piece P of C :
 - (i) Let μ the minimum contrast on P and l its length
 - (ii) Compute $NFA(P) = N_p H(\mu)^l$
 - (iii) If $NFA(P) < \epsilon$ keep the curve P , otherwise discard it.
 - (b) Discard the pieces P of C for which there exists another piece P' such that $P \cap P' \neq \emptyset$ and $NFA(P') < NFA(P)$.
2. Output the non-discarded pieces P .

This algorithm is straightforward, but it requires some explanation, specially step 1.(a)(ii). The quantity $H(\mu)^l$ is the probability that the minimum contrast on a random curve of length l is μ . The acronym *NFA* stands for “Number of False Alarms”, and its use is justified by the following result, explained in [4]: Under a reasonable noise model, the expectation of the number of curve segments output by the algorithm D (when run on noise) is $\leq \epsilon$. Notice that step (ii) of algorithm D computes a positive real number for each piece of level line. This quantity can be interpreted as a “score” of how meaningful is the contrast piece: the closer to zero, the better. Then, step (iii) defines a statistical test in the form of a threshold: only curves whose score is better than ϵ are accepted.

Concerning the efficiency of the algorithm, we have to say that if the pieces of curve on step (a) are traversed in an appropriate order, step (b) can be omitted without affecting the output; but this is an implementation detail.

3. THE EXCLUSION PRINCIPLE

The output of DMM edge detector is usually highly redundant: the edges appear represented as bundles of curves. Here we introduce an exclusion principle to reduce the redundancy of the output, by picking the best representative of each bundle. It is based on a similar principle used in a segment detector [5, 6] in order to reduce output redundancy.

The proposed exclusion principle works by dividing the image domain into small square regions (e.g. of the original pixel size, but not necessarily), and imposing these two requirements on the final set of curves: (1) Each square belongs to at most one curve, and (2) Each curve passes the *NFA* test described above. Here we say that a square Q belongs to a curve C when C crosses through Q . Of course, the first requirement is not usually fulfilled by the original set of curves. The exclusion method works by removing parts of curves until the first requirement is fulfilled. Then, it removes

the remaining pieces of curves that do not pass the test. See Figures 1 to 4 for a graphical explanation.

There are in general non-unique ways to reduce the original set of curves so that the first requirement is true. We propose the following greedy strategy to force uniqueness:

Algorithm E (Exclusion principle)

1. Start with the set of all curves output by algorithm D
2. While there are still curves that pass the test:
 1. Pick the curve C that passes the test with highest score
 2. The curve C owns all the squares that it crosses
 3. Delete the parts of all the other curves that cross through squares owned by C and update their scores accordingly.
 4. Output C and remove it from the set
3. Delete the remaining curves

Remark 1 In the previous algorithm, the “curves” we speak about are not necessarily connected. For example, when we remove a piece in the middle of a curve, the remaining two pieces are still considered “one curve”. This can be seen on the upper curve at Fig. 4(b).

The problem of over-representation was already noted in [1], were the authors proposed a possible solution. The proposal consisted of looking at the tree of level lines (see [7]), and allowing only one detection to occur inside each branch of that tree. However, this criterion may fail in two cases: when there is a single branch of the tree containing separate objects (e.g., the two boundaries of an annulus), and when a branch is cut due to the noise (e.g., the boundary of a blurry disk containing a noisy pixel). In the first case the criterion will miss a boundary and in the second case it will find a duplicated boundary. The proposed exclusion criterion does not have any of these two problems. However, there remains still a problem to solve: when a blurred edge is wider than the scale used for the exclusion principle, multiple representatives are picked for this edge (see Fig. 5(e)) This is addressed on the following section.

4. COUPLING DMM AND CANNY

Our second improvement is aimed to solve the following problem. Let us consider a synthetic image that contains a black object on a white background, and the boundary of the object is blurred, giving rise to all the intermediate gray levels (see Fig. 5(c)). All the level lines surrounding the object are meaningful because they have a high contrast compared with the rest of the image, which is flat. Thus, even if we apply algorithm E , the method will detect the boundary replicated w/s times, where w is the width of the boundary and s the scale of the exclusion principle.

To avoid this replication we propose the following modification to Algorithm D : instead of using the norm of the

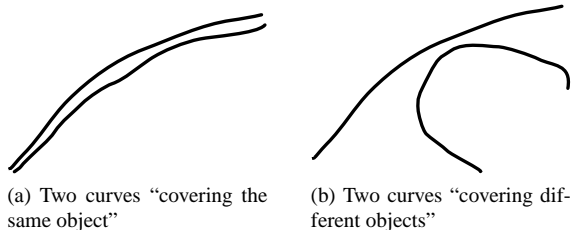


Fig. 1: The two synthetic cases that we are going to consider below

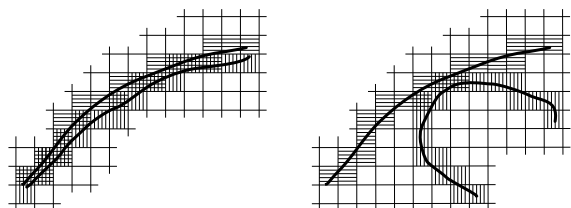


Fig. 2: Marking of the squares according to which curves cross each one.

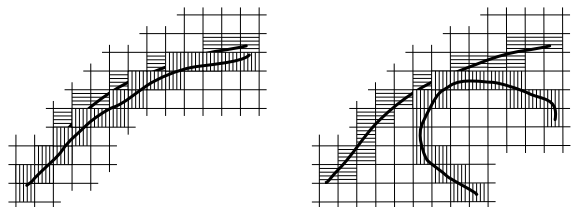


Fig. 3: Assignment of at most one curve to each square, thus fulfilling the first requirement.

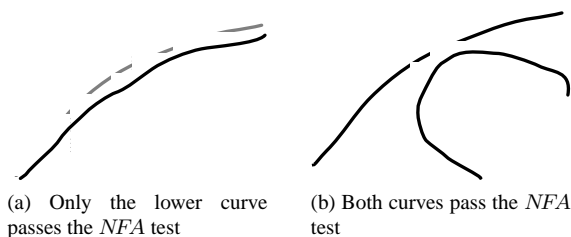


Fig. 4: Run of the *NFA* test for the remaining pieces of curve, thus fulfilling the second requirement.

gradient, use the output Canny's edge detector. More precisely, we compute Canny's filter with the parameters set at their extreme values: omit the initial Gaussian filtering and perform no gradient thresholding at all. This extreme setting of parameters gives an image with a lot of *false positives*: the filter responds too much to noise. However, this is not a problem, because the response inside textured regions is formed by mostly disconnected pixels, which are not covered by long level lines. On the other hand, when there is a long and well contrasted edge, it is completely covered by the Canny's detected pixels, and any level line crossing through them is detected. Furthermore, when there is a wide blurred edge, only its central skeleton of width one pixel is detected by Canny (compare (c) and (d) on Fig. 5); so that the output produced by an edge will be a bundle of curves spanning a width of one pixel at most, independently of the width of the blurred edge.

Algorithm X (Meaningful edges with Canny)

The same as algorithm *D* but using $g =$ output of Canny's filter instead of the gradient norm.

5. EXPERIMENTAL RESULTS

The two modifications we have described here are independent. Each of them improves the output of the original DMM detector, and the combined use of both gives the best results. Let us finally show how do these modifications act upon two concrete examples. The first example, analyzed on Fig. 5, is a synthetic image of a black disk on white background, convolved with a large Gaussian kernel. The original detector finds the boundary replicated over 200 times. Each of the proposed modifications reduces the over-representation of the result. The combined use of both techniques produces the desired result of a single boundary.

The second example, analyzed on Fig. 6, is the lena image. Here we can readily appreciate a great simplification of the output thanks to the exclusion criterion. The effect of using Canny's filter instead of the gradient is more subtle. For instance, the top of the hat closes completely and the shadow on the shoulder is interpreted as a very wide edge.

The only parameter of the whole method is ϵ . The effects of setting this parameter are the same as with the original *DMM* detector, which are discussed at length at [4]. Briefly, except in synthetic or extremely textured images it is safe to set it always to $\epsilon = 1$. This was the setting we used in all our experiments.

The two advantages of the proposed method over Canny's are that it only needs one parameter and that the output is structured as a small set of curves in sub-pixel precision (instead of a set of pixels). The advantage over DMM is that the output is smaller.

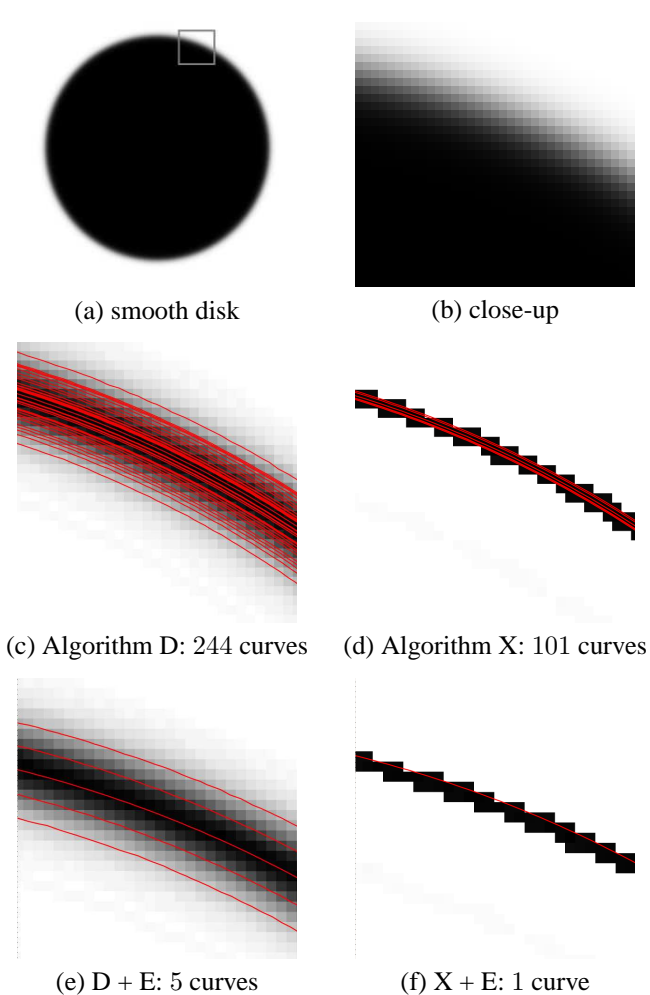


Fig. 5: Comparison of the discussed methods on a synthetic image of a blurred disk.

6. ACKNOWLEDGEMENT

Special thanks to Vicent Caselles for his collaboration in this work. The author is indebted to his colleagues for many remarks and corrections, and more particularly to Pablo Canela and Rafael Grompone. The research was partially supported by the PNPGC project MTM2006-14836, the FPI grant BES-2004-3723, and ALFA project CVFA II-0366-FA.

7. REFERENCES

- [1] A. Desolneux, L. Moisan, and J.M. Morel, “Edge Detection by Helmholtz Principle,” *Journal of Mathematical Imaging and Vision*, vol. 14, no. 3, pp. 271–284, 2001.
- [2] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.

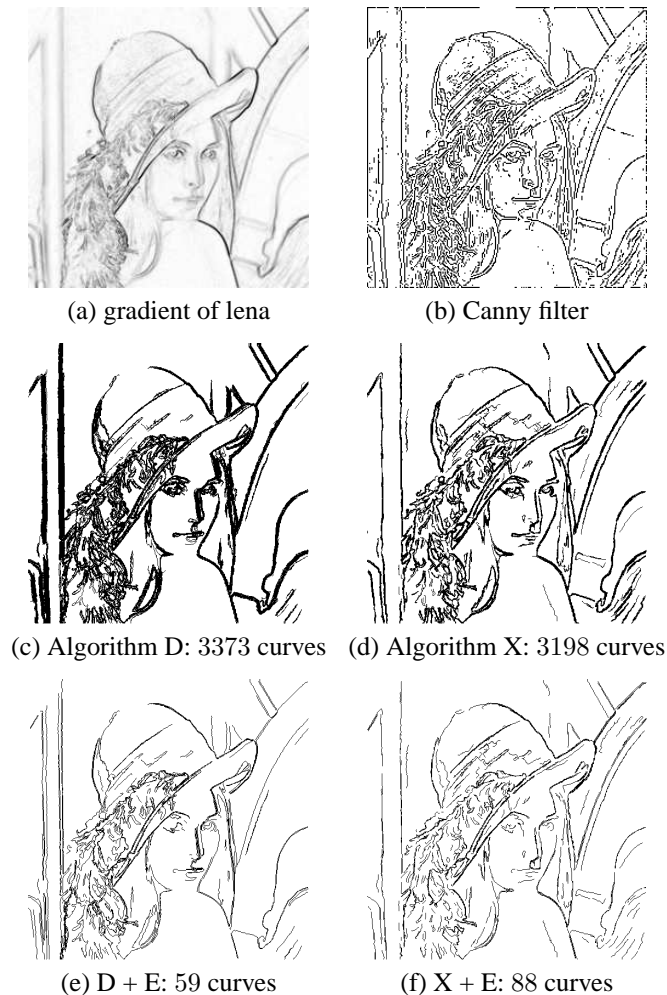


Fig. 6: Comparison of the discussed methods on a real image.

- [3] R. Deriche, “Using Canny’s criteria to derive a recursively implemented optimal edge detector,” *International Journal of Computer Vision*, vol. 1, no. 2, pp. 167–187, 1987.
- [4] F. Cao, P. Musé, and F. Sur, “Extracting Meaningful Curves from Images,” *Journal of Mathematical Imaging and Vision*, vol. 22, no. 2, pp. 159–181, 2005.
- [5] A. Desolneux, L. Moisan, and J.M. Morel, “Meaningful Alignments,” *International Journal of Computer Vision*, vol. 40, no. 1, pp. 7–23, 2000.
- [6] R. Grompone von Gioi, J. Jakubowicz, and G. Randall, “Multisegment detection,” in *IEEE International Conference on Image Processing, ICIP 2007. San Antonio, Texas, 2007*, pp. II: 253–256.
- [7] C. Ballester, V. Caselles, and P. Monasse, “The tree of shapes of an image,” *ESAIM: Control, Opt. and Calc. of Variations*, vol. 9, pp. 1–18, 2003.