COORDINATE LOGIC TRANSFORMS AND THEIR USE IN THE DETECTION OF EDGES WITHIN BINARY AND GRAYSCALE IMAGES

Ethan E. Danahy
Tufts University
edanahy@eecs.tufts.edu

Karen A. Panetta
Tufts University
karen@eecs.tufts.edu

Sos S. Agaian
University of Texas/San Antonio
sos.agaian@utsa.edu

ABSTRACT

This paper introduces coordinate logic (CL) transforms as an alternative method for calculating coordinate logic (CL) filters. Additionally, a new measure and detection technique are introduced, enhancing the capabilities of the basic CL transform for the application of detecting edges within 2D signals (images). Applicable to binary and grayscale images, computer simulations demonstrate the success of this improved procedure on two classes of signals: synthetic (edge maps are know) and natural (edge maps are unknown). Results are evaluated quantitatively (via Pratt's figure of merit) and compared visually to two common edge detection techniques.

Index Terms— coordinate logic filter, coordinate logic transform, edge detection, binary edge map

1. INTRODUCTION

Within scene analysis systems the use of edge detection to determine the boundary between foreground and background objects is very important [1]. From military and robotics to security and data hiding/watermarking, the accurate detection of edges allows improved object recognition and image attribute identification. To do so, the balance between good detection and reducing the amount of errors (false positives) [2] is essential, although often a difficult task as unwanted noise is frequently emphasized by derivative-based techniques [3].

Mertzios and Tsirikolias have presented the idea of using coordinate logic (CL) filters for the purpose of edge extraction [4][5]. Since the CL filters function similar to morphological filters [4], they present several possible methods capable of quickly (fewer computations compared to morphological filters) determining image edge areas.

In this paper the idea of detecting edges in images via coordinate logic methodologies is extended further. First, coordinate logic (CL) transforms are presented, which provide an alternative means for calculating the CL filters' output independent of the basic logic operations. Second, an edge detection algorithm based on these mathematical techniques is modified (and a measure introduced) to rival the results produced by commonly used edge detectors (Laplacian of Gaussian and Canny methods). Computer simulations generate edge maps for a collection of images (both synthetic and natural), with Pratt's figure of merit and visual inspection as the methods employed for evaluation.

This paper is organized as follows. Section 2 reviews the formulation of CL filters, while section 3 provides the CL transform implementation. Section 4 describes the edge detection algorithms and section 5 contains the associated computer simulations. A conclusion, found in section 6, ends the paper.

2. COORDINATE LOGIC (CL) FILTERS

Coordinate logic operations (CLOs) are the basic logic operations (`NOT`, `AND`, `OR`, and `XOR`, and their combinations) applied to corresponding individual binary values or pixels found within two signals or images. `CNOT`, `CAND`, `COR`, and `CXOR` represent the coordinate equivalents for each respectively as applied to multi-bit digital data. Given an image (G) defined by

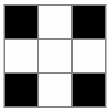
$$G = \{g(i,j);\ i = 1,2,\ldots,M,\ j = 1,2,\ldots,N\} \qquad (1)$$

the evaluation of a CLO (i.e. `CAND`) between two images (here: G_1 and G_2) is performed on a pixel-by-pixel basis and results in output image F:

$$\begin{aligned} F = G_1 \,\texttt{CAND}\, G_2 = \{g_1(i,j) \,\texttt{CAND}\, g_2(i,j)\} \\ i = 1,2,\ldots,M,\ j = 1,2,\ldots,N \end{aligned} \qquad (2)$$

Coordinate logic (CL) filters are the application of the CLOs to a single image as dictated by a binary structuring element B. Since the dimensions of B are often much smaller in size than the original input image G, the resulting output represents local neighborhood characteristics of the image. A possible configuration for B, used in this work as in [4][5], is shown in equation (3).

$$B = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} = \qquad (3)$$



As an example, given structuring element B from (3) centered on input pixel $g(i,j)$ (in image G), the output pixel $f(i,j)$ (in image F) is calculated, for the CLO `COR`, via (4).

$$f(i,j) = g(i-1,j)\,\texttt{COR}\,g(i,j-1)\,\texttt{COR}\,g(i,j) \\ \texttt{COR}\,g(i+1,j)\,\texttt{COR}\,g(i,j+1) \tag{4}$$

The evaluation of the associated CL filters for the four basic CLOs on multi-bit ($n \geq 1$, with s_k being the k^{th} bit-plane decomposition representation of g) images is:

$$G_B^{\texttt{CNOT}} = \texttt{CNOT}\,g(i,j) \in B = \sum_{k=0}^{n-1} \left(s_k(i,j)\right)_B^{\texttt{CNOT}} 2^k \tag{5}$$

$$G_B^{\texttt{CAND}} = \texttt{CAND}\,g(i,j) \in B = \sum_{k=0}^{n-1} \left(s_k(i,j)\right)_B^{\texttt{CAND}} 2^k \tag{6}$$

$$G_B^{\texttt{COR}} = \texttt{COR}\,g(i,j) \in B = \sum_{k=0}^{n-1} \left(s_k(i,j)\right)_B^{\texttt{COR}} 2^k \tag{7}$$

$$G_B^{\texttt{CXOR}} = \texttt{CXOR}\,g(i,j) \in B = \sum_{k=0}^{n-1} \left(s_k(i,j)\right)_B^{\texttt{CXOR}} 2^k \tag{8}$$

Note that for the *unary operator* `CNOT`, the size b of the structuring element B must be equal to one (i.e. $|B| = b = 1$) meaning there is only one positive value in B.

3. COORDINATE LOGIC (CL) TRANSFORMS

Since the basic CLOs can be evaluated through simple logical transforms [6], so too can CL filters be calculated via coordinate logic (CL) transforms. The methods for calculating the four basic CLOs via logical transforms applied to the normal forms (binary truth table representations) of a sequence of binary images (here: G_1, G_2, …, G_n) are in equations (9), (10), (11), and (12) and can be shown to be true. Note the following: addition (+) indicates point-wise addition of the elements within two images; (a mod 2) is the modulo-2 based arithmetic on the value a, also applied in a point-wise fashion when operating on an image; and δ_0 is the Dirac delta function, a special case of the Kronecker delta function δ_p where all p-valued elements are set to one, and all others to zero.

$$\texttt{CNOT:} \quad F = \delta_0(G) \tag{9}$$

$$\texttt{CAND:} \quad \begin{aligned} F &= \delta_n(G_1 + G_2 + \ldots + G_n) \\ &= \delta_0(G_1 + G_2 + \ldots + G_n - n) \end{aligned} \tag{10}$$

$$\texttt{COR:} \quad F = \delta_0\left[\delta_0(G_1 + G_2 + \ldots + G_n)\right] \tag{11}$$

$$\texttt{CXOR:} \quad F = \delta_0\left[\delta_0\left[(G_1 + G_2 + \ldots + G_n)\bmod 2\right]\right] \tag{12}$$

Extending these equations to calculate the application of a CL filter (with a given structuring element B) on a single binary image G results in the set of CL transforms. The operation of the sliding window (B) within the CL filter is replaced with a summation of shifted-images during a preprocessing step for the CL transform. The size of the shifts is defined according to the locations of the positive values in the binary structuring element B. For a B of size 3-by-3, the one-valued elements determine a row-shift (Δr) and column-shift (Δc) shift according to figure 1.

$\Delta r = +1$ $\Delta c = -1$	$\Delta r = +1$ $\Delta c = 0$	$\Delta r = +1$ $\Delta c = +1$
$\Delta r = 0$ $\Delta c = -1$	$\Delta r = 0$ $\Delta c = 0$	$\Delta r = 0$ $\Delta c = +1$
$\Delta r = -1$ $\Delta c = -1$	$\Delta r = -1$ $\Delta c = 0$	$\Delta r = -1$ $\Delta c = +1$

Figure 1. Row-shift (Δr) and column-shift (Δc) shift for a 3-by-3 structuring element

Given a structuring element B, a shift to an image G by Δr for the rows and by Δc for the columns is denoted here as $G^{(r+\Delta r, c+\Delta c)}$. The *shifted-sum of G for structuring element B* ($G_B^{\Delta\Sigma}$) is a summation of all the shifted versions of the binary G image according to the positive values in the B structuring element. Mathematically, this is expressed as

$$\begin{aligned} G_B^{\Delta\Sigma} &= B(1)G^{(r+\Delta r_1, c+\Delta c_1)} + \ldots + B(b)G^{(r+\Delta r_b, c+\Delta c_b)} \\ &= \sum_{h=1}^{b} B(h)G^{(r+\Delta r_h, c+\Delta c_h)} \end{aligned} \tag{13}$$

where b is the total size of the structuring element B and $B(h)$ is the value of the h^{th} element.

The following equations (14) to (17) are the definitions for the basic set of coordinate logic (CL) transforms.

$$G_B^{\texttt{CNOT}} = \delta_0\left(G_B^{\Delta\Sigma}\right) \tag{14}$$

$$G_B^{\texttt{CAND}} = \delta_n\left(G_B^{\Delta\Sigma}\right) = \delta_0\left(G_B^{\Delta\Sigma} - n\right) \tag{15}$$

$$G_B^{\texttt{COR}} = \delta_0\left[\delta_0\left(G_B^{\Delta\Sigma}\right)\right] \tag{16}$$

$$G_B^{\texttt{CXOR}} = \delta_0\left[\delta_0\left[\left(G_B^{\Delta\Sigma}\right)\bmod 2\right]\right] \tag{17}$$

These CL transform formulations are as applied to binary image G, although an extension to grayscale data is possible through the individual evaluation of each bit plane.

4. EDGE DETECTION ALGORITHMS

In [4] are several edge detection methods involving CL filters, including

$$F = \left[\left(G_B^{\texttt{CAND}}\ \texttt{CXOR}\ G\right) - \left(G_B^{\texttt{COR}}\ \texttt{CXOR}\ G\right)\right] \tag{18}$$

which gives very similar results to the morphological $G_B^{\texttt{CAND}} - G_B^{\texttt{COR}}$ edge detector [5]. This method applied to binary images results in edge maps that are competitive with traditional methods. However, with multi-bit (grayscale) data the technique is applied to each bit-plane (as defined by the CL filter equations (5) to (8)) and outputted edge maps often include errors in locations of low contrast or gradients. Figure 2 demonstrates this, where the resulting binary edge map from equation (18) (middle) contains misidentification of edges (false positive) in the gradient section of the original shapes image (left).

To overcome this, a new measure and associated procedure are introduced. Here, grayscale images are instead treated as a sequence of binary images (achieved through progressive thresholding) rather than a set of bit planes. Each is processed independently and the combined result provides a measure indicating likely edge locations. Equation (19) is the formulation of this summation measure (F_{sum}) where G_t is grayscale image G thresholded at level t.

$$F_{sum} = \sum_{t} \left[\left(G_{tB}^{\text{CAND}} \text{ CXOR } G_t \right) - \left(G_{tB}^{\text{COR}} \text{ CXOR } G_t \right) \right] \qquad (19)$$

The final binary edge map is derived from the summation measure F_{sum} through a second step where localized thresholding at level T_{edge} is performed on blocks of size b. This final threshold value, shown in equation (20), is determined automatically as being proportional to the average within F_{sum}. The scaling factor α can be varied, although in this work it is held constant at 2, a value that produced decent results for all inputted images. Figure 2 (right) shows the output from the procedure described by equations (19),(20), and more accurately detects the edges.

$$T_{edge} = \frac{\sum_{i=j=0}^{M,N} \alpha F_{sum}(i,j)}{MN} \qquad (20)$$

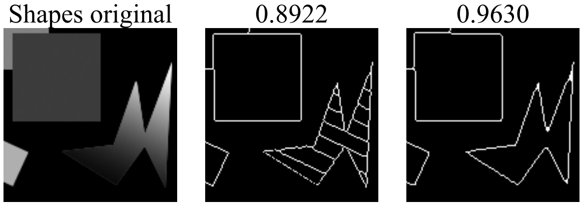


Figure 2. Portion of grayscale shapes (left) and edge maps found by methods from equations (18) (middle) and (19),(20) (right). Values are Pratt's figure of merit.

5. COMPUTER SIMULATIONS

This section presents simulation results demonstrating the edge detection method on several images. Synthetic inputs, with known edge locations, are quantitatively evaluated using Pratt's figure of merit (*FOM*, defined in [7]), which provides a numerical comparison between the detected and actual edge maps as a way to assess algorithms.

$$FOM = \frac{1}{\max\{N_I, N_A\}} \sum_{k=1}^{N_A} \frac{1}{1 + \alpha d^2(k)} \qquad (21)$$

Binary images: From the set of images tested (20 images total), three synthetic images [3] and two natural images were randomly selected and are presented here (figure 3). The binary images were created as thresholded versions of the grayscale originals from figure 6. The resulting output from the CL transform (abbreviated CLT in the figures) implementation of the algorithm described by equation (18) is compared against two commonly used edge detection algorithms: the Laplacian of Gaussian (LoG) and Canny methods. Table 1 provides Pratt's figure of merit comparison for the synthetic binary images (the edge maps generated by the CLT are in figure 4; the other methods' output, being visually similar at this resolution, is not included). The values (higher figure of merit) indicate the CL transform more accurately detecting edges. The edges in the natural images (figure 5) detected by the CL transform are visually comparable to the other two methods.

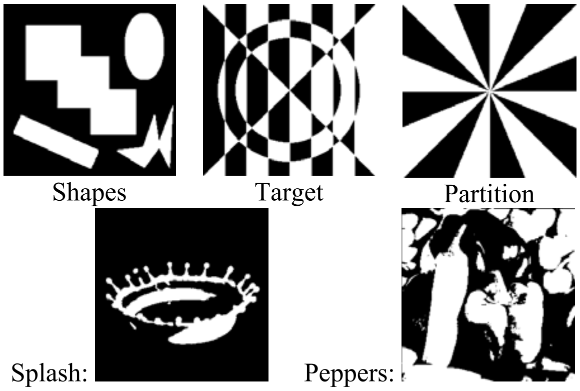


Figure 3. Synthetic and natural input binary images

Table 1: Pratt's figure of merit for binary images

Image:	Shapes	Target	Partition
CLT:	0.9791	0.9972	0.9834
LoG:	0.9131	0.9842	0.9795
Canny:	0.8617	0.9511	0.9391

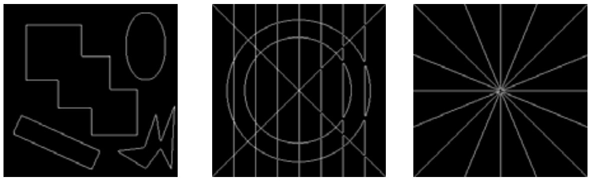


Figure 4. CLT edge maps for synthetic binary images: shapes (left), target (center) and partition (right)

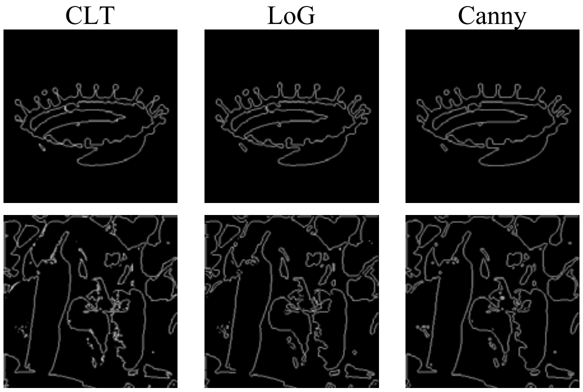


Figure 5. Edge maps for natural binary images: splash (top) & peppers (bottom)

Grayscale images: To test the detection measure/threshold ((19) & (20)) technique introduced in section 4, the grayscale counterparts of the binary images are processed. The original grayscale images are in figure 6.

As in the binary examples, the CL transforms for edge detection applied to grayscale synthetic images (figure 7) generally outperforms the other two methods quantitatively.

For the natural images (figure 8), all three algorithms have difficulty when processing portions of the images low in contrast or containing gradients. Even still, the CL transform method is able to achieve results again visually comparable to the other methods for these natural images.

6. CONCLUSION

This paper introduced a set of CL transforms as methods for calculating CL filters, as well as presented a new measure and thresholding technique for the detection of edges within grayscale images. This extension implementation of the edge detection algorithm shows improved capabilities, as demonstrated on a collection of synthetic and natural images. The technique quantitatively outperforms commonly used edge detection methods, indicating the potential use for detection via CL transforms in many image processing applications. Future explorations will investigate varying the noise level within the input images.

7. REFERENCES

[1] Koschan and M. Abidi, "Detection and Classification of Edges in Color Images," *IEEE Signal Processing Magazine, Special Issue on Color Image Processing*, Vol. 22, No. 1, p. 64-73, 2005.

[2] P. Bao, L. Zhang, and X. Wu, "Canny Edge Detection Enhancement by Scale Multiplication," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 9, p. 1485-1490, Sept 2005.

[3] E. Danahy, S. Agaian, and K. Panetta, "Directional edge detection using the logical transform for binary and grayscale images," *SPIE Defense and Security Symposium: Mobile Multimedia/Image Processing for Military and Security Applications*, Vol. 6250, April 2006.

[4] Mertzios and K. Tsirikolias, "Coordinate Logic Filters: Theory and Applications in Image Analysis," in *Nonlinear Image Processing*, S. Mitra and G. Sicuranza, Eds. Academic, New York, 2000.

[5] Mertzios and K. Tsirikolias, "Applications of Coordinate Logic Filters in Image Analysis and Pattern Recognition," *IEEE International Symposium on Image and Signal Processing and Analysis*, p. 125-130, 2001.

[6] S. Agaian, J. Astola, and K. Egiazarian, *Binary Polynomial Transforms and Nonlinear Digital Filters*, Marcel Dekker, Inc., New York, 1995.

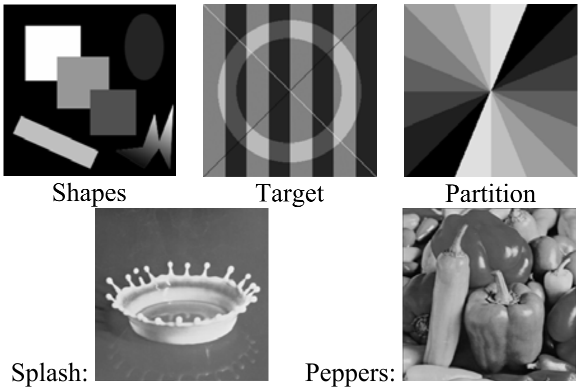


Figure 6. Input grayscale images tested: synthetic (top) and natural (bottom)

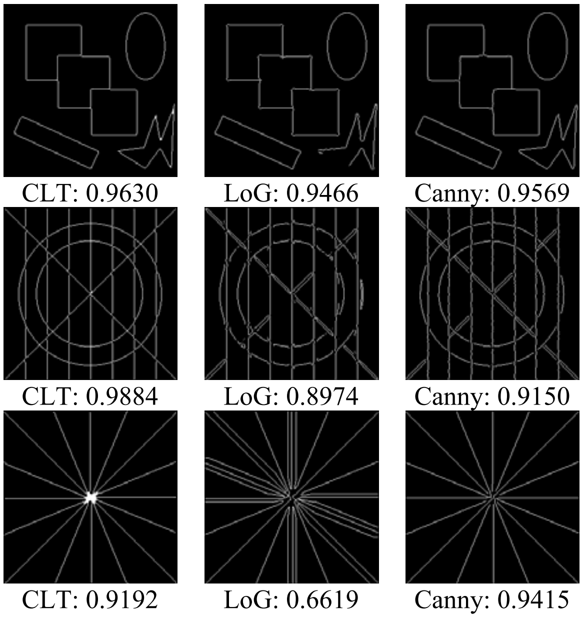


Figure 7. Edge maps comparing methods for synthetic grayscale images: shapes (top), target (middle) and partition (bottom) result. Values are Pratt's FOM.

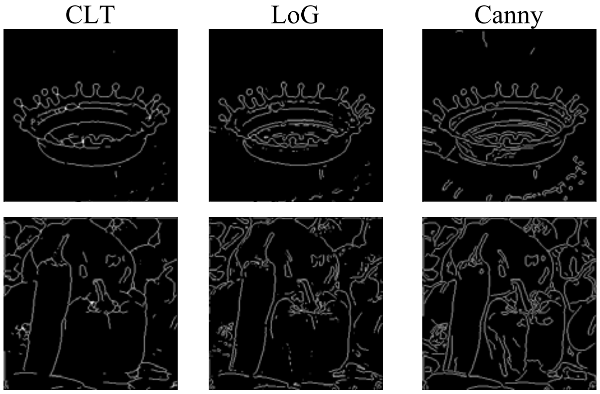


Figure 8. Edge maps for natural grayscale images: splash (top) & peppers (bottom)

[7] W. Pratt, *Digital Image Processing*, John Wiley & Sons, New York, 2nd edition, 1991.