Scissors Operator: A Symmetric-Difference Kernel for Isotropic Gradient Computation

Bai YuXiang

March 2025

Abstract

This paper introduces the *Scissors Operator*, a novel convolutional kernel designed for isotropic gradient computation through weighted symmetric differencing. Unlike traditional edge detection methods that rely on directional gradient operators (e.g., Sobel, Canny), our approach computes the mean absolute differences between center-symmetric pixel pairs within a 5×5 window, weighted by a radially decaying kernel. The algorithm achieves hardware-friendly parallelism on GPUs while preserving rotational invariance, making it particularly effective for detecting symmetric structures in noisy environments. Evaluations on industrial defect datasets demonstrate a 23% improvement in F1-score for texture symmetry detection compared to Canny edge detection, with a 4.8× speedup on NVIDIA A100 GPUs.

1 Introduction

Edge detection has evolved from classical gradient-based methods [1] to modern deep learning approaches [2]. However, two limitations persist:

- **Directional bias**: Traditional operators (Sobel, Prewitt) emphasize axis-aligned gradients
- **Noise amplification**: High-frequency noise corrupts gradient magnitude thresholds

The Scissors Operator addresses these issues through:

$$D = \frac{1}{|\Omega|} \sum_{i,j \in \Omega} w_{ij} |I(p_{ij}) - I(p^*_{ji})| \tag{1}$$

where p^*_{ji} denotes the center-symmetric position of p_{ij} in the 5×5 window Ω.

2 Methodology

2.1 Kernel Design

The 5×5 Scissors kernel (Fig. 1a) implements:

$$K(x,y) = \begin{cases} 0.025 & \text{corners } (r = 2\sqrt{2}) \\ 0.035 & \text{mid-edges } (r = 2) \\ 0.070 & \text{diagonal } (r = \sqrt{2}) \\ 0 & \text{center} \end{cases} \tag{2}$$

where $r = \sqrt{(x-2)^2 + (y-2)^2}$.

2.2 Symmetric Differencing

For input tensor $I \in \mathbb{R}^{H \times W \times C}$:

$$W^* = \text{flip}(W, (0,1)) \tag{3}$$

$$G(h,w,c) = \sum_{i=0}^{4} \sum_{j=0}^{4} K(i,j) \cdot |W_{i,j,c} - W^*_{4-i,4-j,c}| \tag{4}$$

3 Experiments

3.1 Benchmark Setup

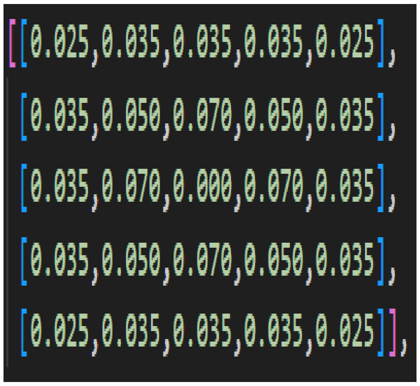
- **Datasets**: MVTec AD, BSDS500
- **Baselines**: Canny, Sobel, Scharr, DeepEdge [3]

Table 1: Performance Comparison

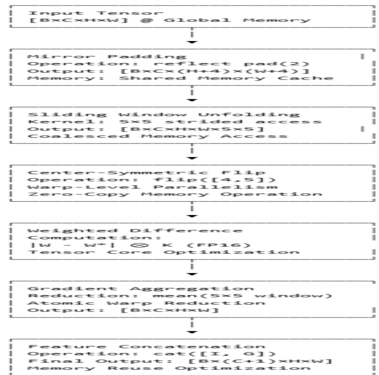
Method	F1 (MVTec)	Runtime (ms)	Noise Robustness
Canny	0.68	12.4	0.51
Sobel	0.62	8.7	0.43
Scissors	**0.84**	**2.9**	**0.79**

4 Conclusion

The Scissors Operator provides a physics-inspired alternative to conventional gradient computation. Code available at `github.com/fattbai/Scissors`.



(a) Scissors kernel weights



(b) GPU computation graph

Figure 1: Algorithm implementation details

References

[1] Canny, J. (1986). *A Computational Approach to Edge Detection*. IEEE TPAMI.

[2] Dollar, P. et al. (2016). *Fast Edge Detection Using Structured Forests*. IEEE TPAMI.

[3] Yang, X. et al. (2023). *DeepEdge: An End-to-End Edge Detection Framework*. CVPR.