EAN-13 Barcode Detection and Recognition using Edge, Hough Transform, and Contour-based Decoding

Moldovan Darius-Andrei

Technical University of Cluj-Napoca

November 5, 2025

Outline

What is an EAN-13 Barcode?

- EAN-13 = European Article Number, 13 digits.
- Encodes numeric information in sequences of black and white bars.
- Commonly used on products in European and Romanian retail.

Project Goal

Objective

Develop a robust method to detect and decode an EAN-13 barcode directly from its bar pattern using classical image processing.

- Input: grayscale or color image containing one barcode.
- Output: the recognized 13-digit numeric sequence.
- Must be invariant to lighting, rotation, distance, and perspective.

General Pipeline

1. Image preprocessing.
2. Edge detection.
3. Orientation estimation using Hough Transform.
4. Barcode localization via contours.
5. Perspective correction and flattening.
6. Bar pattern decoding (EAN-13 digit extraction).

1. Image Preprocessing

- Convert to grayscale and apply **CLAHE** to improve local contrast.
- Apply smoothing to reduce random noise.
  - prepares the image for stable edge detection under uneven illumination.

2. Edge Detection

- Use **Sobel** filters to find vertical edges.
- Threshold to keep only strong vertical transitions.
- Perform morphological closing to join fragmented edges.
  - barcode zones show dense, repetitive vertical gradients.

3. Orientation Estimation (Hough Transform)

- Apply **Hough Line Transform** to detect dominant line angles.
- Rotate the image to align the barcode horizontally.
  - simplifies contour extraction and decoding.

4. Barcode Region Extraction

- Find connected regions using **cv::findContours**.
- Filter candidates by aspect ratio (wide/short) and edge density.
- Use **cv::minAreaRect** to compute bounding box.
- Apply **warpPerspective** to correct skew or perspective distortion.

5. Bar Pattern Decoding

- Scan multiple horizontal lines across the barcode region.
- Measure alternating black and white run lengths.
- Normalize widths to the module size (smallest bar width).
- Convert patterns to binary (0 = narrow, 1 = wide).
- Split into left and right halves and decode using EAN-13 tables.
- Validate with checksum:

$$\text{Sum} = 3 \times \text{even digits} + \text{odd digits}$$

$$\text{Check digit} = (10 - (\text{Sum} \bmod 10)) \bmod 10$$

Handling Real-World Variations

- **Lighting:** CLAHE and adaptive thresholding handle shadows/reflections.
- **Rotation:** corrected using Hough-detected dominant angle.
- **Distance:** scale invariance through adaptive module normalization.
- **Perspective:** rectified using **minAreaRect** + **warpPerspective**.

Validation and Error Checking

- Validate decoded digits with the EAN-13 checksum.
- Compare results from multiple scanlines; choose the most consistent.
- Reattempt decoding with adjusted thresholds if checksum fails.

Advantages & Limitations

Advantages

- Fully classical, no OCR or ML models required.
- Robust to lighting, rotation, and perspective.
- Real-time performance on standard hardware.

Limitations

- Sensitive to blur or low contrast.
- Curved surfaces may distort bar widths.

Conclusion

- Edge, Hough, and contour analysis localize the barcode accurately.
- Direct decoding of bar patterns retrieves the 13-digit code.
- Meets robustness requirements with deterministic, interpretable steps.

Future work: improve decoding on curved or blurred barcodes.

Bibliography and References

- OpenCV Documentation - Edge Detection (Sobel, Scharr): https://docs.opencv.org/4.x/d2/d2c/tutorial_sobel_derivatives.html
- OpenCV Hough Transform Reference: https://docs.opencv.org/4.x/d3/de6/tutorial_js_houghlines.html
- OpenCV Contour Detection: https://docs.opencv.org/4.x/d4/d73/tutorial_py_contours_begin.html
- ZXing-C++ (EAN-13 decoding logic reference): https://github.com/zxing-cpp/zxing-cpp
- EAN-13 Barcode Specification (GS1): https://www.gs1.org/standards/barcodes/ean-upc