

Image Edge Detection Based On Opencv

It is very difficult to distinguish the exact number of the copper core in the tiny wire. This implementation is an efficient image processing application that can be used to check and judge the merits of the industrial product.

Platform - Opencv3 and c++

2 types of implementation: 1. section-wise implementations of paper

Paper Section	Folder Name
---sec II B	-> II.B.ColorToGray
---sec II C	-> II.C.Filter
---sec II D	-> II.D.Threshold
---sec III	-> III.Morphology
---sec IV	-> IV.CounterTrack

2. Final merging (sequential implementation of all section's figure)
Output as a whole -> V.Final_Merge

For all folder : **Source Image** = src.jpg

Source Code = .cpp

For Execution: Open terminal of specific folder and write **./run.sh**

sec II B(ColorToGray): src.jpg is converted to grayscale image. **Output-** out.jpg

sec II C(Filter): In this paper, median filter has been used. I have implemented all the 4 filters.

Homogeneous Blur: **output** - Out_HomogeneousBlur.jpg

Median Blur: **output** - Out_MedianBlur.jpg

Gaussian Blur: **output** - Out_GaussianBlur.jpg

Bilateral Blur: **output** - Out_BilateralBlur.jpg

Of them I think Gaussian is the best.

sec II D (Thresholding): In this paper, binary thresholding has been used. I have implemented all the thresholding approach.

Binary: **output** - Out_BinaryThreshold.jpg

Binary Inverted: **output** - Out_BinaryInvertedThreshold.jpg

Truncate: **output** - Out_TruncateThreshold.jpg

To zero: **output** - Out_TozeroThreshold.jpg

To zero Inverted: **output** - Out_TozeroInvertedThreshold.jpg

sec III (Morphological Processing): Here erosion, dilation and opening have been used.

Erosion/Corrosion: removes pixels on obj. Boundaries. **output** - Out_Erosion.jpg

Dilation/Expansion: add pixels to obj. Boundaries. **output** - Out_Dilation.jpg

Opening: an erosion followed by a dilation. **output** - Out_Opening.jpg

sec IV (Counter Tracking): It is for traversing the border of a region without repetition. It differentiates cores one from the other. **output** - Out_Counters.jpg

Folder V.Final_Merge : It is actually sequential implementation of above all

Out1_ColorToGray.jpg	applied over src.jpg
Out2_Gaussian.jpg	gaussian filter is applied over Out1_ColorToGray.jpg
Out3_Threshold.jpg	Binary Thresholding is applied over Out2_Gaussian.jpg
Out4_Erosion.jpg	
Out4_Dilation.jpg	→ applied one after another over Out3_Threshold.jpg
Out4_Opening.jpg	
Out5_Couters.jpg	shows one distinguished core from another clearly which is the final result of this paper.

Submitted By: 1405100