R05922096 資工所 碩 1 李哲安

Computer Vision Hw10 Report

- Discription
    - Zero Crossing Edge Detection
- Algorithm
    - Laplace, Minimum variance Laplacian, Laplace of Gaussian, Difference of Gaussain edge detectors.
    - Kernel : same as website
- Parameters (if any)
    - no
- Principal Code Fragment

  Main (file – /src/hw10/ DemoZeroCrossingEdgeDetection.java)

```
// Laplace
System.out.println("Laplace Edge Detector ...");
BufferedImage laplace = ZeroCrossingEdgeDetector.operate(lena, 30, MaskName.Laplace);
ImgUtil.showImg(laplace, "laplace");

// MinVarLaplace
System.out.println("MinVarLaplace Edge Detector ...");
BufferedImage minVarLaplace = ZeroCrossingEdgeDetector.operate(lena, 15, MaskName.MinVarLaplace);
ImgUtil.showImg(minVarLaplace, "minVarLaplace");

// Laplace of Gaussian
System.out.println("Laplace of Gaussian Edge Detector ...");
BufferedImage LoG = ZeroCrossingEdgeDetector.operate(lena, 5000, MaskName.LoG);
ImgUtil.showImg(LoG, "LoG");

// Difference of Gaussian
System.out.println("Difference of Gaussians Edge Detector ...");
BufferedImage DoG = ZeroCrossingEdgeDetector.operate(lena, 1, MaskName.DoG);
ImgUtil.showImg(DoG, "DoG");
```

GeneralEdgeDetector

(file – /src/cv1.util.cv.edge /ZeroCrossingEdgeDetector.java)

```java
public class ZeroCrossingEdgeDetector {
    public enum MaskName{Laplace, MinVarLaplace, LoG, DoG}

    public static BufferedImage operate(BufferedImage bi, double threshold, MaskName maskName){
        ArrayList<Mask> maskList = ZeroCrossingEdgeDetectorMask.getMasksList(maskName);
        return generalOperation(bi, threshold, maskList);
    }

    private static BufferedImage generalOperation(BufferedImage bi, double threshold, ArrayList<Mask> maskList) {
        BufferedImage result = new BufferedImage(bi.getWidth(), bi.getHeight(), bi.getType());

        for (int y = 0; y < bi.getHeight(); y++) {
            for (int x = 0; x < bi.getWidth() ; x++) {
                double gradientMagnit = 0;
                for(Mask mask : maskList){
                    double maskWeightValue = 0;
                    for (MaskLogic logic : mask.logics) {
                        try {
                            int gray = bi.getRGB(x + logic.x, y + logic.y) & 0xff;
                            maskWeightValue += gray * logic.w;
                        } catch (Exception e) {
                            // Ignore out of bound
                        }
                    }
                    gradientMagnit += maskWeightValue;
                }
                int newGray = gradientMagnit >= threshold ? 0 : 255;
                result.setRGB(x, y, 0xff000000 + (newGray<<16) + (newGray<<8) + (newGray));
            }
        }

        return result;
    }
}
```

- Result Image

Laplace (threshold = 30)

MinVarLaplace (threshold = 15)

LoG (threshold = 5000)

DoG (threshold = 1)