Homework 07, UST
Dilating and Eroding in OpenCV

Mr. Saeed Ullah,

Korean Institute of Science & Technology Information (KISTI),
University of Science & Technology (UST)
Saeedonline12@gmail.com

Summary—This report presents Dilation and Erosion functions in OpenCV. Tools used in this project are OpenCV 3.2 Library which is a library used for Computer Vision, and Visual Studio 2015 (64 bit).

I.   INTRODUCTION

Dilation and Erosion are the two types of morphological operations. Based on the given input image a "structural element" is developed. This might be done in any of the two procedures. These are aimed at removing noise and settling down the imperfections, to make the image clear.

◆   **Dilation**

This procedure follows convolution with some kernel of a specific shape such as a square or a circle. This kernel has an anchor point, which denotes its center.

This kernel is overlapped over the picture to compute maximum pixel value. After calculating, the picture is replaced with anchor at the center. With this procedure, the areas of bright regions grow in size and hence the image size increases.

◆   **Erosion**

Erosion is quite a similar process as dilation. But the pixel value computed here is minimum rather than maximum in dilation. The image is replaced under the anchor point with that minimum pixel value.

With this procedure, the areas of dark regions grow in size and bright regions reduce.

II.   OPENCV SOURCE CODE

Both of the methods dilate() and erode() accepts the following parameters

src − A Mat object representing the source (input image).

dst − A Mat object representing the destination (output image).

kernel − A Mat object representing the kernel.

```cpp
//Load Image
src = imread("..\\data\\lena1.jpg",
CV_LOAD_IMAGE_COLOR);

// Create a structuring element
int erosion_size = 6;
Mat element =
getStructuringElement(cv::MORPH_CROSS,
Size(2 * erosion_size + 1, 2 * erosion_size + 1),
Point(erosion_size, erosion_size));

// Apply erosion on the image
erode(src, dst, element);

// Apply Dilation on the image
dilate(src,dst,element);
```

III.   RESULTS

Fig. 1. Original Image

Fig. 2. Dilation Applied Image   Fig. 3. Erosion Applied Image

CONCLUSION: - dilate() and erode() built-in functions are powerful functions for applying Dilation and Erosion on images which can easily be implemented.