Computer Vision I: Homework 4

Department: CSIE Student ID: d11922016 Name: Jia-Wei, Liao

October 7, 2022

Write programs which do binary morphology on a binary image:

(a) Dilation

(b) Erosion

(c) Opening

(d) Closing

(e) Hit-and-miss transform

Let I be the original image and $K = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$ be the kernel.

1 Dilation

The dilation of I by K is

$$D_K(I) = I \oplus K = \bigcup_{k \in K} I_k,$$

where I_k is the translation of I by $k \in K$.

```
[m, n] = size(img); [k, ~] = size(kernel);
dilation_image = zeros(m, n);
img = padding(img, floor(k / 2));

for i = 1:m
    for j = 1:n
        local_image = get_local_image(img, kernel, i, j);
        dilation_image(i, j) = pixel_dilation(local_image, kernel);
    end
end

dilation_image = uint8(dilation_image);


```

```
function val = pixel_dilation(img, kernel)
[km, kn] = size(kernel);
val = 0;

for i = 1:km
    for j = 1:kn
        if img(i, j) >= 0
            if (kernel(i, j) == 1) && (img(i, j) == 255)
                val = 255;
                return
            end
        end
    end
end

end
```

(a) binary image (b) binary image after dilation

Figure 1: dilation

2 Erosion

The erosion of I by K is

$$E_K(I) = I \ominus K = \bigcup_{k \in K} I_{-k},$$

where I_{-k} is the translation of I by $-k$ with $k \in K$.

```
[m, n] = size(img); [k, ~] = size(kernel);
erosion_image = zeros(m, n);
img = padding(img, floor(k / 2));

for i = 1:m
    for j = 1:n
        local_image = get_local_image(img, kernel, i, j);
```

```
        erosion_image(i, j) = pixel_erosion(local_image, kernel);
    end
end

erosion_image = uint8(erosion_image);


function val = pixel_erosion(img, kernel)
[km, kn] = size(kernel);
val = 0;

for i = 1:km
    for j = 1:kn
        if img(i, j) >= 0
            if (kernel(i, j) == 1) && (img(i, j) ~= 255)
                return
            end
        end
    end
end

val = 255;

end
```

(a) binary image (b) binary image after erosion

Figure 2: erosion

3 Opening

The opening of I by K is

$$O_K(I) = I \circ K = (I \ominus K) \oplus K.$$

```
opening_image = erosion(img, kernel);
opening_image = dilation(opening_image, kernel);
opening_image = uint8(opening_image);
```

(a) binary image (b) binary image after opening

Figure 3: opening

4 Closing

The closing of I by K is

$$C_K(I) = I \bullet K = (I \oplus K) \ominus K.$$

```
closing_image = dilation(img, kernel);
closing_image = erosion(closing_image, kernel);
closing_image = uint8(closing_image);
```

(a) binary image (b) binary image after closing

Figure 4: closing

5 Hit-and-miss transform

Let $J = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ and $K = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ be the kernels.

$$HM(I; J, K) = (I \oplus J) \cap (I^C \oplus K).$$

```
img_c = complement(img);
hit_and_miss_image = intersection(erosion(img, J), erosion(img_c, K));
```

(a) binary image (b) hit-and-miss transform

Figure 5: hit-and-miss

Summary.

In this homework, we use kernels-based algorithms to implement the morphological operations of images. We show the result of the image after dilation, erosion, opening, closing and hit-and-miss transform.