CS332 Visual Processing in Computer and Biological Vision Systems

Edge Detection in Computer Vision Systems

This handout introduces methods for detecting and describing intensity changes in digital images, focusing on some ideas that are especially relevant to understanding the early stages of processing in the human visual system.

General Points:

Physical changes in the scene, such as the borders of objects, surface markings and texture, shadows and highlights, and changes in surface structure, give rise to changes of intensity in the image. The first step in analyzing the image is to detect and describe these intensity changes – they provide the first hint about the structure of the scene being viewed.

It is difficult to identify where the significant intensity changes occur in a natural image. The raw intensity signal can be very complex, with changes of intensity occurring everywhere in the image and at different scales. We will consider one common approach to the detection of intensity changes that incorporates three operations: smoothing, differentiation, and feature extraction. Two of these operations, smoothing and differentiation, can be combined into a single processing step.

Below is a small portion of an image taken with a digital camera, showing the image intensity at each location. Darker regions of the image correspond to smaller intensity values. The indices of the rows and columns of the array containing this image are indicated along the top row and left column. The full image contains a gray circle against a dark background, and the sample below shows a small portion of the upper left region of the circle. A contour is drawn through the array indicating the border between the gray circle (lower right) and dark background (upper left).

	24	25	26	27	28	29	30	31	32	33	34	35
24	15	20	23	15	22	16	14	17	13	20	14	14
25	21	13	14	15	18	14	17	23	21	17	67	61
26	25	12	16	19	22	22	20	14	25	64	69	63
27	24	19	15	26	27	25	15	55	59	52	61	58
28	13	21	27	20	18	28	54	62	66	56	53	65
29	13	14	24	20	23	68	56	69	61	57	69	62
30	14	28	18	28	63	67	68	57	67	67	69	66
31	24	18	22	59	55	66	59	55	68	54	56	56
32	26	23	23	58	55	57	64	63	57	58	54	56
33	19	22	68	61	56	64	69	56	55	68	53	57
34	22	55	62	57	62	59	59	64	55	68	67	57
35	19	54	60	60	57	55	61	56	57	61	55	60

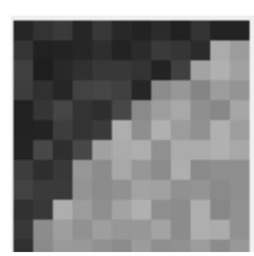


Figure 1: Snippet of a digital image stored in an array of intensity values.

The purpose of the smoothing operation is to remove minor fluctuations of intensity in the image due to noise in the sensors, and to enable the detection of changes of intensity at different spatial scales. In the image of a Handi-Wipe cloth shown in Figure 11, for example, there are changes of intensity at a fine scale that correspond to the individual holes of the cloth. If the image is smoothed by a small amount, variations of intensity due to the holes are preserved (Figure 11(b)). At a coarser scale, there are intensity changes that follow the pattern of colored stripes on the cloth. If the image is smoothed by a greater amount, the intensity fluctuations due to the holes of the cloth may be smoothed out, leaving only the larger variations due to the stripes (Figure 11(c)).

The purpose of the differentiation operation is to transform the image into a representation that makes it easier to identify the locations of significant intensity changes. If a simple step change of intensity is smoothed out, the first derivative of this smoothed intensity function has a "peak" (maximum or minimum) at the location of the original step change, and the second derivative crosses zero at this location. The peaks and zero-crossings of a function are easy to detect and indicate where the significant changes occur in the original intensity function.

The final step of feature extraction refers to the detection and description of the peaks or zero-crossings in the result of smoothing and differentiating the image. There are properties of these features that give a rough indication of the sharpness and contrast of the original intensity changes.

The rest of this document elaborates on the above operations and provides details about their implementation, starting with a general image processing operation known as *convolution*.

Convolution:

The smoothing and differentiation steps can be performed using the operation of *convolution*. This section provides a simplified description of the convolution operation that is adequate for how we will use convolution in practice. This operation involves calculating a weighted sum of the intensities within a neighborhood around each location of the image. The weights are stored in a separate array that is usually much smaller than the original image. We will refer to the pattern of weights as the *convolution operator*. For simplicity, we will assume that the convolution operator has an odd number of rows and columns so that it can easily be centered on each image location.

To calculate the result of the convolution at a particular location in the image, we first center the convolution operator on that location and compute the product between the image intensity and convolution weights at every location where the image and convolution operator overlap. We then sum all of these products over the neighborhood. In Figure 2, a tiny image is shown in (a) and simple 3x3 convolution operator is shown in (b). The array indices are shown along the top row and left column of both. In (c), the convolution weights are shown in red, superimposed on a 3x3 image window centered on location (3,3), which has an intensity of 8 (pink square in the middle). The result of the convolution at image location (3,3) is calculated as follows:

```
C(3,3) = 1*4 + 2*9 + 3*3 + 4*5 + 5*8 + 6*7 + 7*6 + 8*1 + 9*9 = 264
```

In practice, we usually only compute convolution values at locations where the convolution operator can be centered and covers image locations that are all within the bounds of the image array. For the image shown in Figure 2, convolution values would not be calculated along rows 1 and 5, and columns 1 and 5.

	1	2	3	4	5
1	5	7	3	2	4
2	8	4	9	3	1
3	7	5	8	7	2
4	9	6	1	9	3
5	2	1	7	8	2

(a)

	1	2	3
1	1	2	3
2	4	5	6
3	7	8	9

(b)

4 1	9 2	3 3
5 4	8 5	7 6
6 7	1 8	9 9

(c)

Figure 2: (a) Tiny 5 x 5 image. (b) 3 x 3 convolution operator. (c) The 3 x 3 convolution operator (values in red) is shown superimposed on a 3 x 3 image window centered on location (3,3), with intensity values shown in black.

The convolution operation is denoted by the * symbol. In one dimension, let I(x) denote the intensity function, O(x) denote the convolution operator, and C(x) denote the convolution result. We can then write:

$$C(x) = O(x) * I(x)$$

In two dimensions, each of these functions depends on x and y:

$$C(x,y) = O(x,y) * I(x,y)$$

The next two sections describe the smoothing and differentiation operations in more detail and show how a convolution operator can be constructed to perform these operations simultaneously.

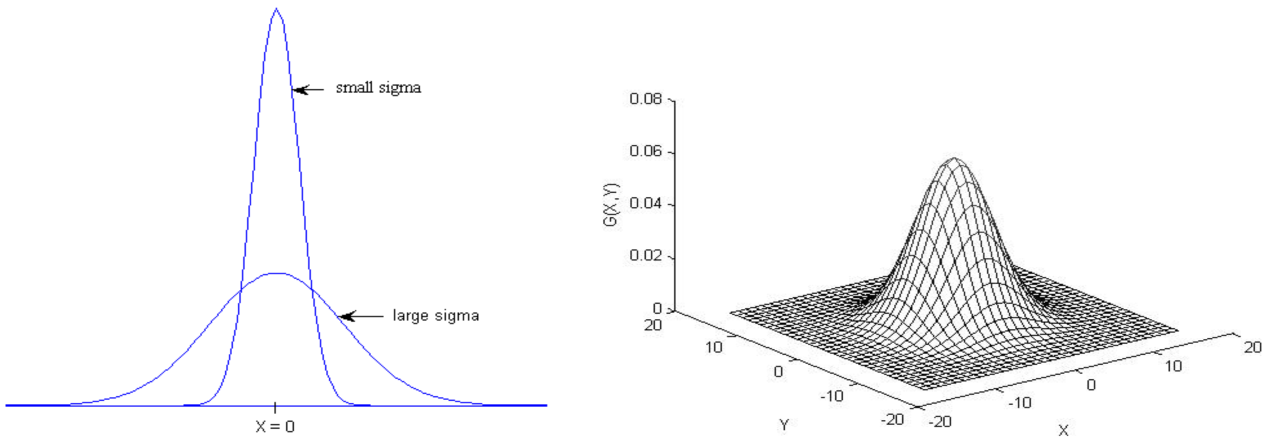
Smoothing:

The smoothing operation blends together the intensities within a neighborhood around each image location. A greater amount of smoothing can be achieved by combining the intensities over a larger region of the image. There are many ways to combine the intensities in a neighborhood to perform smoothing. For example, we could calculate the average of the intensities within a region around each image location. Theoretical studies suggest that a better approach is to weigh the intensities by a Gaussian function that has its peak centered at each image location. This has the effect of weighing nearby intensities more heavily than those further away, when performing smoothing. Biological systems use Gaussian-like smoothing functions in their initial processing of the retinal image. In one dimension, the Gaussian function can be defined as follows:

$$G(x) = \frac{1}{\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

This function has a maximum value at x = 0, as shown in Figure 3(a). In two dimensions, the Gaussian function can be defined as:

$$G(x,y) = \frac{1}{\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$



(a) (b)

Figure 3: (a) One-dimensional Gaussian functions with large and small values of σ.
(b) Two-dimensional Gaussian function, G(x,y).

This function has a maximum value at x = y = 0 and is circularly symmetric, as shown in Figure 3(b). The value of σ controls the spread of the Gaussian – as σ is increased, the Gaussian spreads over a larger area. A convolution operator constructed from the Gaussian function performs more smoothing as σ is increased. For both of the above definitions of the Gaussian function, the height of the central peak decreases as σ increases, but this is not important in practice.

Differentiation:

We noted earlier that if a simple step change of intensity is smoothed out, the first derivative of this smoothed intensity function has a peak at the location of the original step change, and the second derivative crosses zero at this location. This is illustrated in Figure 4, which shows a simple step change of intensity in one dimension (top plot in black), a smoothed version of this step change (red), and the first derivative (green) and second derivative (blue) of the smoothed step change. The dashed line shows that at the location of the original step change, the first derivative has a peak (maximum) and second derivative is zero.

In one dimension, we calculate the first or second derivative with respect to x. Let D'(x) and D''(x) denote the results of the first and second derivative computations, respectively. If I(x) is also smoothed with the Gaussian function, then the combination of the smoothing and differentiation steps can be written as follows:

$$D'(x) = d/dx\,[G(x) * I(x)]$$

$$D''(x) = d^2/dx^2\,[G(x) * I(x)]$$

The derivative and convolution operations are associative, so the above expressions can be rewritten as follows:

$$D'(x) = [d/dx\,G(x)] * I(x)$$

$$D''(x) = [d^2/dx^2\,G(x)] * I(x)$$

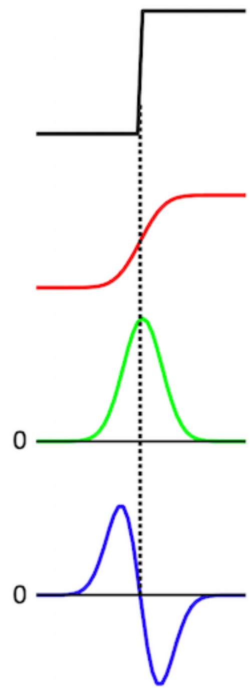


Figure 4: A step change of intensity (black) is smoothed (red) and then the first (green) and second (blue) derivatives are computed from the smoothed step change.

This means that the smoothing and derivative operations can be performed in a single step, by convolving the intensity signal I(x) with either the first or second derivative of a Gaussian function, shown in Figure 5 (the size of the second derivative is scaled in the vertical direction). The value of σ again controls the spread of these functions – as σ is increased, the derivatives of the Gaussian are spread over a larger area.

$$\frac{d}{dx}G(x) = -\frac{x}{\sigma^3}e^{-\frac{x^2}{2\sigma^2}} \qquad \frac{d^2}{dx^2}G(x) = \frac{1}{\sigma^3}\left(\frac{x^2}{\sigma^2}-1\right)e^{-\frac{x^2}{2\sigma^2}}$$

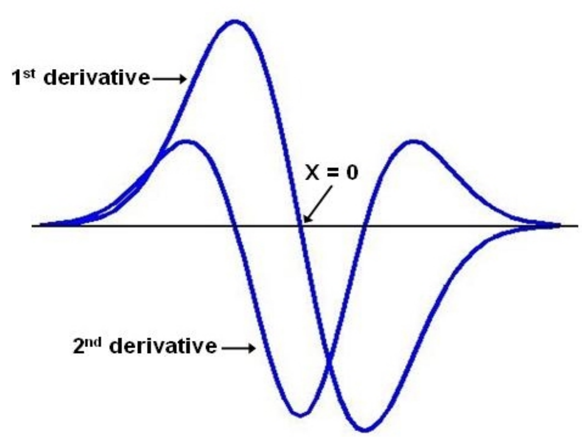


Figure 5: First and second derivative of a Gaussian function, superimposed.

Figure 6 illustrates the smoothing and differentiation operations for a one-dimensional intensity profile. Figure 6(a) shows the original intensity function. Figures 6(b) and (c) show the results of smoothing the image profile with Gaussian functions that have a small and large value for σ. Figure 6(d) shows the first derivative of the smoothed intensity profile in (c), and Figure 6(e) shows the second derivative of (c). The vertical dashed lines show the relationship between intensity changes in the smoothed intensity profile, peaks (maxima or minima) in the first derivative and zero-crossings in the second derivative.

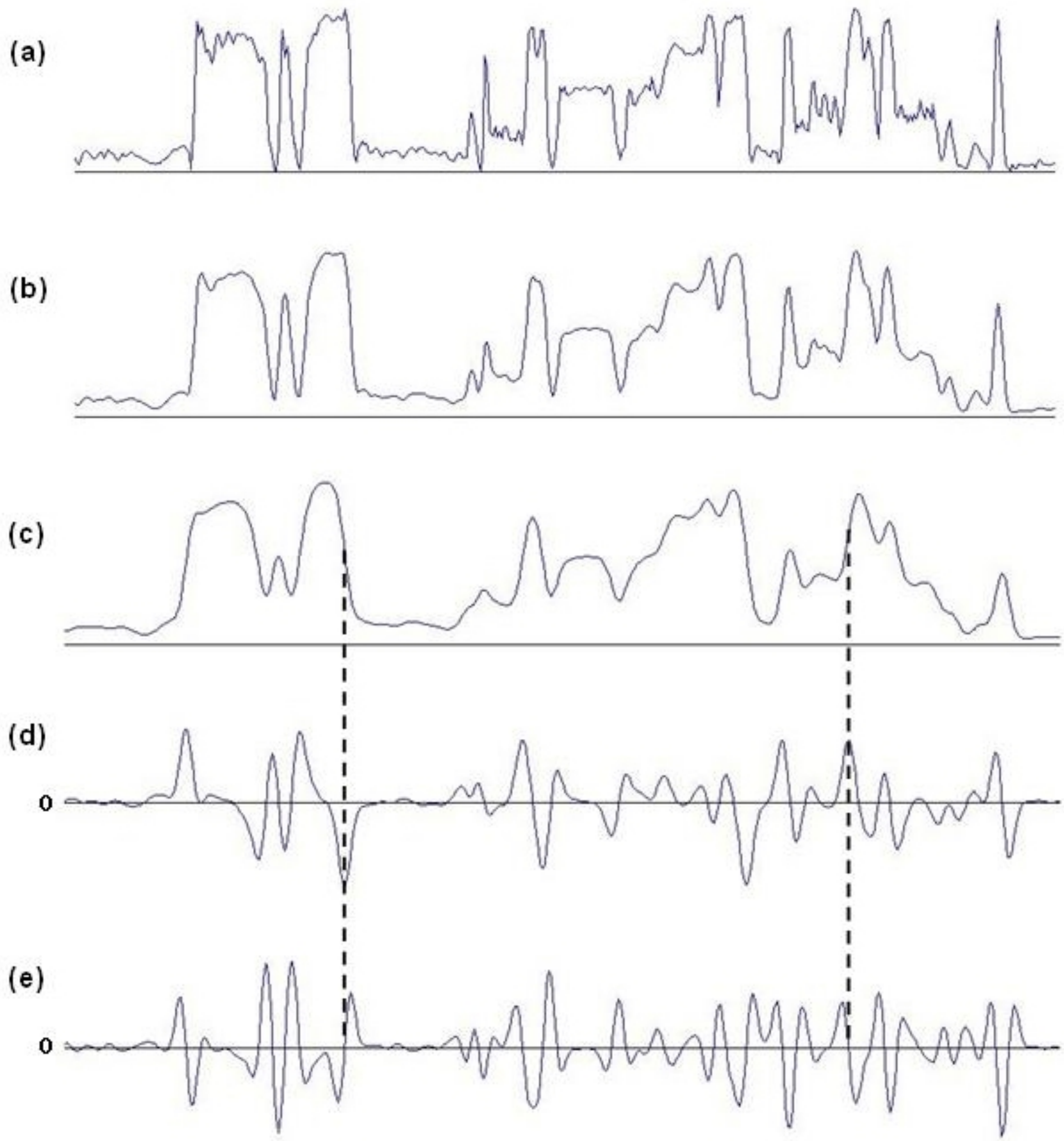


Figure 6: (a) original intensity profile, (b) smoothed intensity, small σ, (c) smoothed intensity, large σ, (d) first derivative of smoothed intensity, (e) second derivative of smoothed intensity. The dashed lines connect the location of an intensity change in the smoothed intensity profile with a peak in the first derivative and zero-crossing in the second derivative.

In two dimensions, intensity changes can occur along any orientation in the image and there are more choices regarding how to compute the derivative. One option is to compute directional derivatives; that is, to compute the derivatives in particular 2-D directions. To detect intensity changes at all orientations in the image, it is necessary to compute derivatives in at least two directions, such as the horizontal and vertical directions. Either a first or second derivative can be calculated in each direction. It is possible to first smooth the image with one convolution step, and then perform the derivative computation by a second convolution with operators that implement a derivative. Alternatively, these two steps can be combined into a single convolution step.

We are going to consider in detail, an alternative approach that involves computing a single, non-directional second derivative referred to as the *Laplacian*. This approach is motivated by knowledge of the processing that takes place in the retina in biological vision systems. The Laplacian operator is the sum of the second partial derivatives in the x and y directions and is denoted as follows:

$$\nabla^2 I = \partial^2 I/\partial x^2 + \partial^2 I/\partial y^2$$

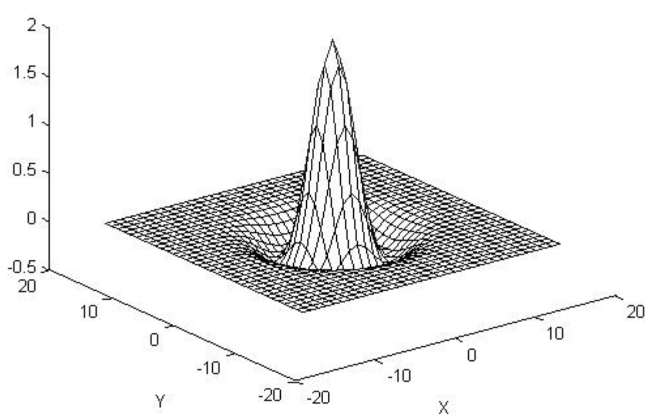
Although the Laplacian is defined in terms of partial derivatives in two directions, it is a non-directional derivative in the following sense. Suppose the Laplacian is calculated at a particular location (x,y) in the image. The 2-D image can then be rotated by any angle around the location (x,y) and the value of the Laplacian does not change (this is not true for the directional derivatives mentioned earlier).

Let L(x,y) denote the result of calculating the Laplacian of an image that has been smoothed with a 2-D Gaussian function. Then L(x,y) can be written as follows:

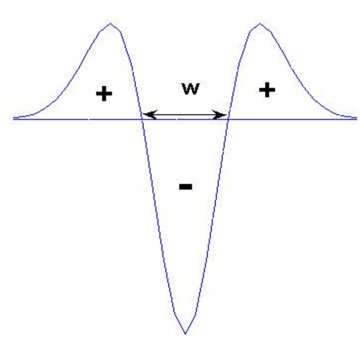
$$L(x,y) = \nabla^2[G(x,y) * I(x,y)] = [\nabla^2 G(x,y)] * I(x,y)$$

The Laplacian and smoothing operations can be performed in a single step, by convolving the image with a function whose shape is the Laplacian of a Gaussian. This function is defined as follows:

$$\nabla^2 G = \frac{1}{\sigma^2}\left(\frac{r^2}{\sigma^2} - 2\right)e^{-\frac{r^2}{2\sigma^2}} \qquad r^2 = x^2 + y^2$$



(a)



(b)

Figure 7: (a) The Laplacian of a Gaussian, shown with its sign reversed. (b) 1D cross-section.

This function is circularly symmetric and shaped like an upside-down Mexican hat. Figure 7(a) shows this function with its sign reversed. It has a minimum value at the location x = y = 0 (maximum in the Figure 7(a) rendering). Because the Laplacian is a second derivative, the features in the result of the convolution of the image with this function, which correspond to the locations of intensity changes in the original image, are the contours along which this result crosses zero.

The space constant σ again controls the spread of this function. As σ is increased, the Laplacian of a Gaussian function spreads over a larger area. A convolution operator constructed from this function performs more smoothing of the image as σ is increased, and the resulting zero-crossing contours capture intensity changes that take place at a coarser scale. The diameter of the central negative portion of the Laplacian of a Gaussian function, which we denote by w, is related to σ as shown below. We will often refer to the size of the Laplacian of a Gaussian function by the value of w, depicted in the one-dimensional cross-section of the function shown in Figure 7(b).

$$w = 2\sqrt{2}\sigma$$

In practice, we can scale all the values of the Laplacian of a Gaussian function by a constant factor and flip its sign, without changing the locations of the resulting zero-crossing contours. To construct a convolution operator for this function, it can be sampled at evenly spaced values of x and y and the resulting samples can be placed in a 2-D array. In Figure 8, we illustrate a convolution operator that was constructed in this way. This operator was derived by computing samples of the following function, which differs from the Laplacian of a Gaussian function shown earlier by a constant scale factor and a change in sign:

$$\nabla^2 G = 25\left(2 - \frac{r^2}{\sigma^2}\right)e^{-\frac{r^2}{2\sigma^2}}$$

In the example shown, $\sigma = \sqrt{2}$, so the diameter w of the central region of the function, which is now positive, is 4 picture elements (pixels). The array has 11x11 elements and the center element (location (6,6)) represents the origin, where x = y = 0. The maximum positive value of the operator occurs at this location. Note that the convolution operator is circularly symmetric.

	1	2	3	4	5	6	7	8	9	10	11
1	0	0	0	0	0	-1	0	0	0	0	0
2	0	0	-1	-1	-2	-3	-2	-1	-1	0	0
3	0	-1	-2	-4	-6	-7	-6	-4	-2	-1	0
4	0	-1	-4	-7	-4	0	-4	-7	-4	-1	0
5	0	-2	-6	-4	15	29	15	-4	-6	-2	0
6	-1	-3	-7	0	29	50	29	0	-7	-3	-1
7	0	-2	-6	-4	15	29	15	-4	-6	-2	0
8	0	-1	-4	-7	-4	0	-4	-7	-4	-1	0
9	0	-1	-2	-4	-6	-7	-6	-4	-2	-1	0
10	0	0	-1	-1	-2	-3	-2	-1	-1	0	0
11	0	0	0	0	0	-1	0	0	0	0	0

Figure 8: Convolution operator created from samples of the Laplacian-of-Gaussian function.

Figure 9 shows a portion of the result of convolving the image of the noisy circle (see Figure 1) with this Laplacian-of-Gaussian operator (the values were scaled by 0.1). There are positive and negative values in the result, along the border of the original circle. A contour of zero-crossings along the edge of the original circle is drawn on the figure. There are a few spurious zero-crossing contours not highlighted here, indicating that this convolution operator is still somewhat sensitive to the fluctuations of intensity due to the added noise. The convolution of the image with a larger Laplacian-of-Gaussian function would yield fewer of these spurious zero-crossings.

	24	25	26	27	28	29	30	31	32	33	34	35
24	13	31	13	-21	-42	-66	-111	-196	-273	-253	-188	-119
25	-2	-61	-85	-59	-67	-151	-240	-287	-242	-48	161	230
26	32	-74	-110	-85	-111	-222	-278	-212	-50	193	360	334
27	60	-33	-87	-121	-175	-208	-128	51	184	255	267	182
28	-13	-81	-138	-195	-175	-39	147	296	307	204	137	74
29	-126	-197	-227	-183	-6	231	358	360	278	174	134	83
30	-153	-259	-241	-55	211	383	362	256	180	137	111	45
31	-124	-255	-177	73	268	311	228	122	71	39	-15	-71
32	-133	-197	-35	171	221	179	128	53	5	-30	-103	-142
33	-185	-69	179	273	179	112	92	29	3	9	-53	-99
34	-216	55	299	284	124	52	40	-1	6	48	27	1

Figure 9: Result of convolving image from Figure 1 with convolution operator shown in Figure 8.

Feature Extraction:

The final step in the detection of intensity changes involves locating the peaks (in the case of the first derivative) or zero-crossings (in the case of the second derivative) in the convolution result. At the location of a peak (maximum or minimum), the value of the convolution result has a larger magnitude than surrounding values. At the location of a zero-crossing, the convolution result changes between positive and negative values. As noted previously, for the case of the Laplacian derivative operator, the important features in the result of the convolution are the zero-crossing contours. Typically, a separate array is filled with values that indicate the locations of these features. This array could contain a 1 at the locations of peaks or zero-crossings, and 0 elsewhere. Figure 10 illustrates the results of convolving the image in Figure 10(a) with the Laplacian of a Gaussian function. The convolution result is displayed in Figure 10(b), with large positive values shown in white and large negative values shown in black. In Figure 10(c), all of the positive values in the convolution result are shown as white and negative values are black. The zero-crossings in this result are located along the borders between positive and negative values, shown in Figure 10(d). These zero-crossing contours indicate where the intensity changes occur in the original image.

If the image is convolved with multiple operators of different size, each convolution result can be searched for peaks or zero-crossings that correspond to intensity changes occurring at different scales. Figure 11 shows edge contours derived from processing an image of a Handi-Wipe cloth with convolution operators of different size. Laplacian-of-Gaussian operators with w = 4 and w = 12 pixels were used to generate the two results. The result of the smaller operator preserves the changes of intensity due to the holes in the cloth, while the result of the larger operator captures the intensity changes due to the pattern of colored stripes. Some computer vision systems attempt to combine these multiple representations of intensity changes into a single representation of all of the intensity changes in the image. Other systems keep these representations separate, for use by later stages of processing that might, for example, compute stereo disparity or image motion.

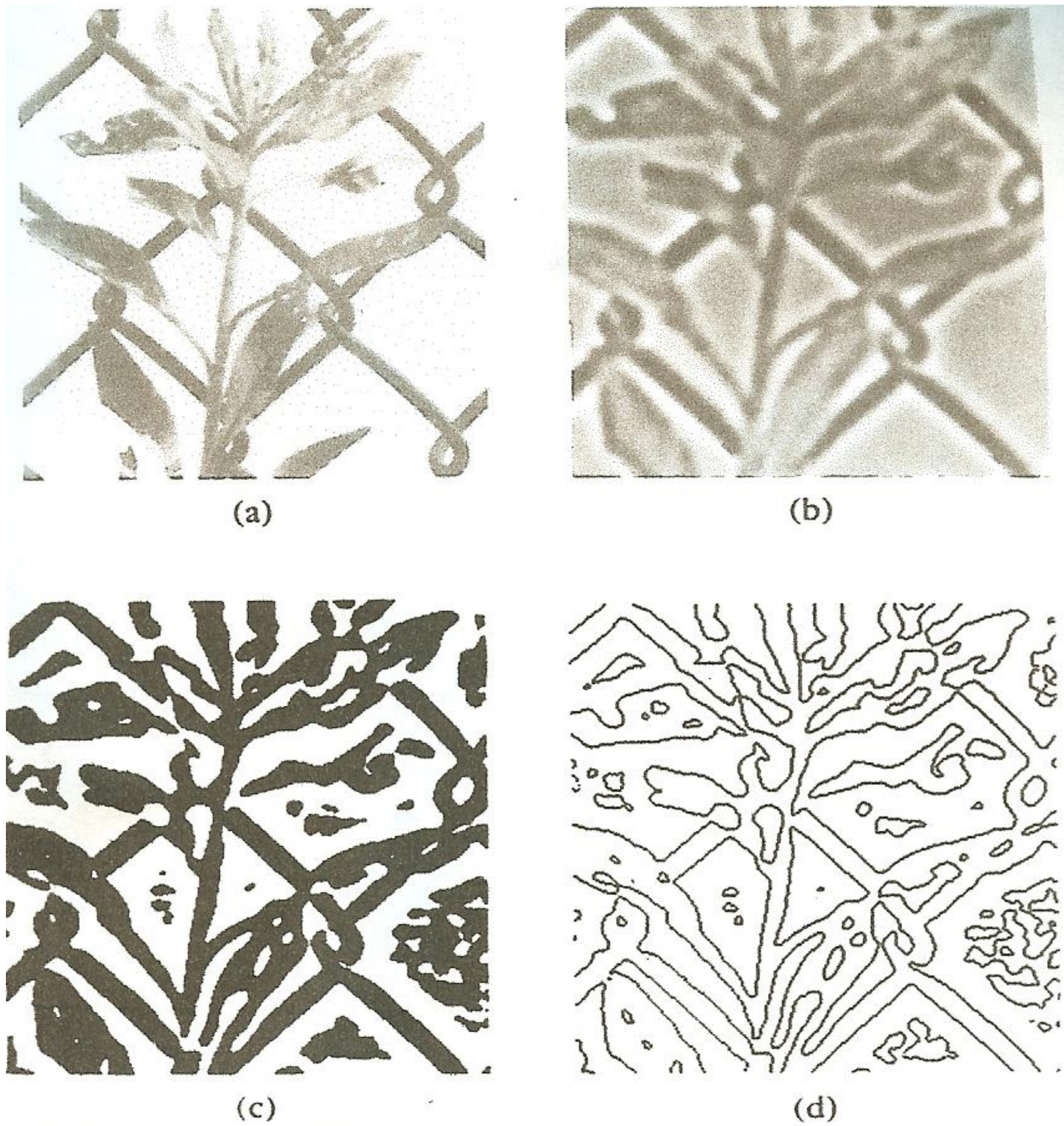
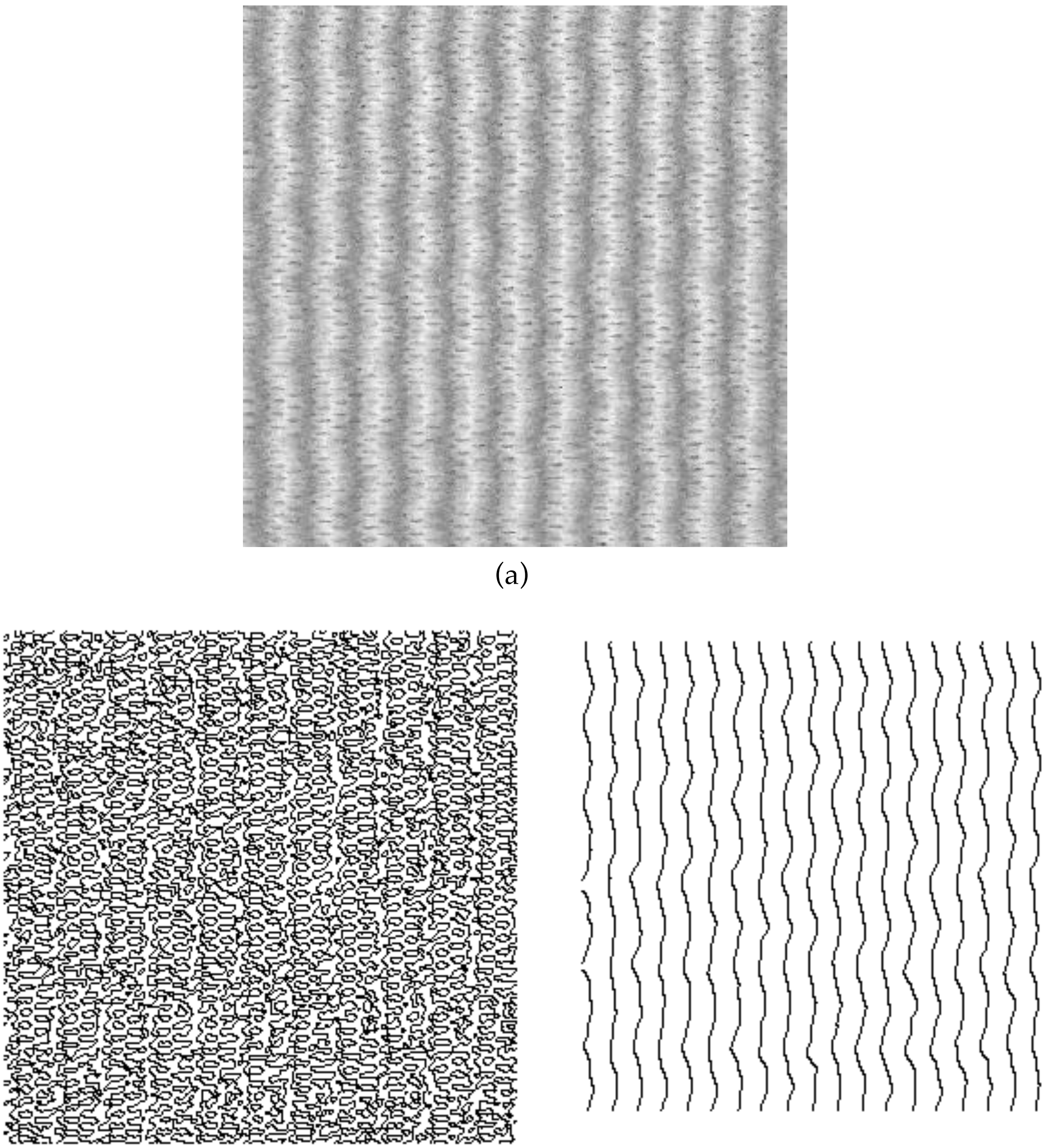


Figure 10: (a) Original image. (b) Result of convolving the image with a Laplacian-of-Gaussian operator. (c) Positive values in the result shown in (b) are displayed in white and negative values are shown in black. (d) Zero-crossing contours derived from the result in (b).

For some applications, it is important to obtain a rough idea of the contrast and sharpness of intensity changes in the image. The contrast refers to the amount of change of intensity and the sharpness refers to the number of pixels over which intensity is changing. These properties are illustrated in Figure 12. The shape of the convolution result in the vicinity of zero-crossings can indicate the sharpness and contrast of the original intensity change. We can compute properties such as the rate of change of the convolution output as it passes through zero (sometimes referred to as the *slope* of the zero-crossing), the height of the positive and negative peaks on either side of the zero-crossing, or the distance between these two peaks. In general, a high-contrast, sharp intensity change gives rise to a zero-crossing with a steep slope and high peaks on either side of the zero-crossing that are closely spaced. A lower contrast intensity change gives rise to shallower zero-crossings and lower peaks, and a shallower intensity change gives rise to a zero-crossing with shallower slope and with peaks on either side that are more spread apart.



(a)

(b) (c)

Figure 11: (a) Original image. (b) Zero-crossing contours from result of convolving the image with a Laplacian-of-Gaussian operator with small σ. (c) Zero-crossing contours from result of convolution with an operator with large σ.

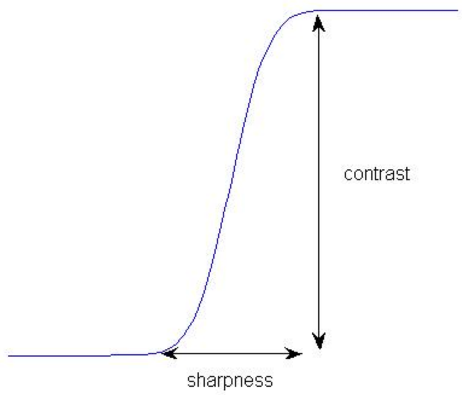


Figure 12: Sharpness and contrast of an intensity change.

In the result of convolving a 2-D image with the Laplacian of a Gaussian, the rate of change of the convolution result as it crosses zero, which we will call the *slope* of the zero-crossing, can be used as a rough measure of the contrast and sharpness of the intensity changes. Intensity changes can occur at any orientation in the image. At the location of a zero-crossing contour, the slope of the convolution result is always steepest in the direction perpendicular to the contour. The *gradient* of a 2-D function is a vector defined at each location of the function that points in the direction of steepest increase of the function. This vector can be constructed by measuring the derivative of the function in the x and y directions. The horizontal and vertical components of the gradient vector, ∇, are the derivatives in the x and y direction: $\nabla = (dx,dy)$. The magnitude of this gradient vector indicates the rate of change of the function at each location. If we again let L(x,y) denote the result of convolving the image with the Laplacian of a Gaussian function, then at the location of a zero-crossing, the slope of the zero-crossing can be defined as follows:

$$slope = \sqrt{\left(\frac{\partial L}{\partial x}\right)^2 + \left(\frac{\partial L}{\partial y}\right)^2}$$

The derivatives of L(x,y) in the x and y directions can be calculated by subtracting the convolution values in adjacent locations of the convolution array. For example, in the convolution result shown in Figure 9, suppose we want to calculate the slope of the zero-crossing that occurs between location (29, 28) (i.e. row 29 and column 28) and location (29, 29). The location (29, 29) contains the value 231. The derivative in the x direction can be calculated by subtracting the value at location (29, 28), yielding 237, and the derivative in the y direction can be calculated by subtracting the value at location (28, 29), yielding 270. The result of the slope calculation is then 359. In practice, only the relative values of the slopes at different zero-crossings are important, so the slopes can be scaled to smaller values. The image in Figure 13(a) was first convolved with the Laplacian of a Gaussian. The zero-crossing contours were detected and the slopes of the zero-crossings were calculated as described above. In Figure 13(b), the zero-crossings are displayed with the darkness of the contour proportional to this measure of slope. Higher contrast edges in the original image, such as the outlines of the coins, give rise to darker (more steeply sloped) zero-crossing contours.

(a)

(b)

Figure 13: (a) Original image. (b) Zero-crossing contours of the convolution of the image with a Laplacian-of-Gaussian operator, with the slope of each zero-crossing conveyed by the darkness of the contour.

Summary:

This handout introduced some basic concepts related to the detection and description of intensity changes in an image. This process is referred to as *edge detection* in the literature, as intensity changes often correspond to edges in the scene, such as borders of objects, outlines of surface textures and markings, shadow boundaries, and so on. We will see that intensity changes also play an important role in the perception of images and early stages in the neural processing of the retinal image. We only touched the surface of computer vision work on edge detection, focusing on the Laplacian-of-Gaussian operator that resembles spatial filtering operations in the primate retina. More information about edge detection methods used in computer vision can be found in numerous sources – I just mention one, the textbook, *Computer Vision: Algorithms and Applications*, by Richard Szeliski, available online at http://szeliski.org/Book/.