

bwlabel

Label connected components in 2-D binary image

Syntax

```
L = bwlabel(BW)
L = bwlabel(BW,conn)
[L,n] = bwlabel( __ )
[gpuarrayL,n] = bwlabel(gpuarrayBW,conn)
```

Description

L = bwlabel(**BW**) returns the label matrix **L** that contains labels for the 8-connected objects found in **BW**. [example](#)

L = bwlabel(**BW**,**conn**) returns a label matrix, where **conn** specifies the connectivity.

[**L**,**n**] = bwlabel(__) also returns **n**, the number of connected objects found in **BW**.

[**gpuarrayL**,**n**] = bwlabel(**gpuarrayBW**,**conn**) performs the labeling operation on a GPU. The input image and output label matrix are gpuArrays. The variable **conn** can be a numeric array or a gpuArray. This syntax requires the Parallel Computing Toolbox™. [example](#)

Examples

[collapse all](#)

▼ Label Components Using 4-connected Objects

Create a small binary image.

Try it in MATLAB

```
BW = logical ([1 1 1 0 0 0 0 0
               1 1 1 0 1 1 0 0
               1 1 1 0 1 1 0 0
               1 1 1 0 0 0 1 0
               1 1 1 0 0 0 1 0
               1 1 1 0 0 0 1 0
               1 1 1 0 0 1 1 0
               1 1 1 0 0 0 0 0]);
```

Create the label matrix using 4-connected objects.

```
L = bwlabel(BW,4)
```

L = 8×8

```
1 1 1 0 0 0 0 0
1 1 1 0 2 2 0 0
1 1 1 0 2 2 0 0
1 1 1 0 0 0 3 0
1 1 1 0 0 0 3 0
1 1 1 0 0 0 3 0
1 1 1 0 0 3 3 0
1 1 1 0 0 0 0 0
```

Use the find command to get the row and column coordinates of the object labeled "2".

```
[r, c] = find(L==2);
rc = [r c]
```

```
rc = 4x2
```

```
2     5
3     5
2     6
3     6
```

▼ Label Components Using 4-connected Objects on a GPU

Create a small binary image and create a gpuArray object to contain it.

```
BW = gpuArray(logical([1 1 1 0 0 0 0 0
                      1 1 1 0 1 1 0 0
                      1 1 1 0 1 1 0 0
                      1 1 1 0 0 0 1 0
                      1 1 1 0 0 0 1 0
                      1 1 1 0 0 0 1 0
                      1 1 1 0 0 1 1 0
                      1 1 1 0 0 0 0 0]));
```

Create the label matrix using 4-connected objects.

```
L = bwlabel(BW,4)
```

Use the find command to get the row and column coordinates of the object labeled "2".

```
[r,c] = find(L == 2)
```

Input Arguments

[collapse all](#)

▼ BW — Binary image

2-D real, nonsparse, numeric or logical matrix

Binary image, specified as a 2-D, real, nonsparse, numeric or logical matrix. For numeric input, any nonzero pixels are considered to be on.

Example: BW = imread('text.png'); L = bwlabel(BW);

Data Types: single | double | int8 | int16 | int32 | int64 | uint8 | uint16 | uint32 | uint64 | logical

▼ conn — Connectivity

8 (default) | 4

Connectivity, specified as the values 4, for 4-connected objects, or 8, for 8-connected objects.

Example: BW = imread('text.png'); L = bwlabel(BW,4);

Data Types: double

▼

gpuarrayBW — Binary image when run on a GPU
gpuArray

Binary image when run on a GPU, specified as a gpuArray.

Example: BW = gpuArray(imread('text.png')); L = bwlabel(BW);

Output Arguments

collapse all

▼

L — Label matrix
matrix of nonnegative integers

Label matrix of contiguous regions, returned as matrix of nonnegative integers with the same size as [BW](#). The pixels labeled 0 are the background. The pixels labeled 1 make up one object; the pixels labeled 2 make up a second object; and so on.

Data Types: double

▼

n — Number of connected objects
nonnegative integer

Number of connected objects in [BW](#), returned as a nonnegative integer.

Data Types: double

▼

gpuarrayL — Label matrix when run on a GPU
gpuArray

Label matrix when run on a GPU, returned as a gpuArray.

Tips

- The functions `bwlabel`, `bwlabeln`, and `bwconncomp` all compute connected components for binary images. `bwconncomp` replaces the use of `bwlabel` and `bwlabeln`. It uses significantly less memory and is sometimes faster than the other functions.

	Input Dimension	Output Form	Memory Use	Connectivity
<code>bwlabel</code>	2-D	Double-precision label matrix	High	4 or 8
<code>bwlabeln</code>	N-D	Double-precision label matrix	High	Any
<code>bwconncomp</code>	N-D	CC struct	Low	Any

- You can use the MATLAB® `find` function in conjunction with `bwlabel` to return vectors of indices for the pixels that make up a specific object. For example, to return the coordinates for the pixels in object 2, enter the following:.

```
[r, c] = find(bwlabel(BW)==2)
```

You can display the output matrix as a pseudocolor indexed image. Each object appears in a different color, so the objects are easier to distinguish than in the original image. For more information, see [label2rgb](#).

- To compute a label matrix having a more memory-efficient data type (e.g., uint8 versus double), use the [labelmatrix](#) function on the output of [bwconncomp](#).
- To extract features from a binary image using [regionprops](#) with default connectivity, just pass BW directly into [regionprops](#), i.e., [regionprops](#)(BW).
- The [bwlabel](#) function can take advantage of hardware optimization for data types logical, uint8, and single to run faster. Hardware optimization requires marker and mask to be 2-D images and conn to be either 4 or 8.

Algorithms

[bwlabel](#) uses the general procedure outlined in reference [1], pp. 40-48:

1. Run-length encode the input image.
2. Scan the runs, assigning preliminary labels and recording label equivalences in a local equivalence table.
3. Resolve the equivalence classes.
4. Relabel the runs based on the resolved equivalence classes.

References

[1] Haralick, Robert M., and Linda G. Shapiro, Computer and Robot Vision, Volume I, Addison-Wesley, 1992, pp. 28-48.

Extended Capabilities

> C/C++ Code Generation

Generate C and C++ code using MATLAB® Coder™.

See Also

[bwconncomp](#) | [bwlabeln](#) | [bwselect](#) | [label2rgb](#) | [labelmatrix](#) | [regionprops](#)

Introduced before R2006a
