

## Re: chain codes

---

*Source:* <http://sci.tech-archive.net/Archive/sci.image.processing/2007-02/msg00077.html>

---

- *From:* "ImageAnalyst" <[imageanalyst@xxxxxxxxxxxxxx](mailto:imageanalyst@xxxxxxxxxxxxxx)>
  - *Date:* 8 Feb 2007 05:15:40 -0800
- 

I presume you've already figured this out since the answer is so simple, but. . .

Your code doesn't compute chain codes at all. It merely follows borders in the image and builds up an edge detection image, kind of like you ran a Sobel filter over it. So you have an image with edges in it, but you're not keeping track of the individual edges. Keep at it and you'll get it.

By the way, your code only processes one "image" which may have several "blobs" in it. If you have another image with another circle in it, you'll have to open each image file individually and process each file in turn.

Regards,  
ImageAnalyst

On Feb 7, 12:58 pm, "cindy" <[olivemckenzie2...@xxxxxxxxxx](mailto:olivemckenzie2...@xxxxxxxxxx)> wrote:

I would like to know if any one can describe to me how to read the chain code of two circles side by side in a image and how do I would know when one image has finished and another image has begun. using C++ code. I have tried the typical neighbourhood code and it does not seem to work.

```
for(int h=1; h<ROWS-1;h++)
{
for(int w=1; w<COLS-1;w++)
{
int value=Array[h][w];

if (val<255)
cout<<h<<" " <<w<<endl;

getchar();

if(Array[h][w]<255)
{
Shape_Array[h][w] = value;
}
}
```

## Re: chain codes

```
else
if (Array[h][w+1] < 255 && Shape_Array[h][w+1] == 255)
{
Shape_Array[h][w+1] = value;
w++;
}
else if (Array[h][w-1] < 255 && Shape_Array[h][w-1] == 255)
{
Shape_Array[h][w-1] = value;
w--;
}

else
if (Array[h+1][w-1] < 255 && Shape_Array[h+1][w-1] == 255)
{
Shape_Array[h+1][w-1] = value;
h++;
w--;
}
else if (Array[h+1][w] < 255 && Shape_Array[h+1][w] == 255)
{
Shape_Array[h+1][w] = value;
h++;
}
}
```