**1. function chain\_code = getChainCodeFour(BW) 's description**

1) find boundary of object and get bound point array.

[B,L,N,A] = bwboundaries(BW, 4);

2) Find first pixel on the boundary

prev\_x = C(r-1, 1);

prev\_y = C(r-1, 2);

3) Traverse the four neighborhood of current pixel in the counter wise order.

for k=1:r-1

x = C(k, 1);

y = C(k, 2);

code = '0';

if( y == prev\_y ) % horizontal direction

if( x > prev\_x )

code = '0';

else

code = '2';

end

end

if( x == prev\_x ) % vertical direction

if( y > prev\_y )

code = '3';

else

code = '1';

end

end

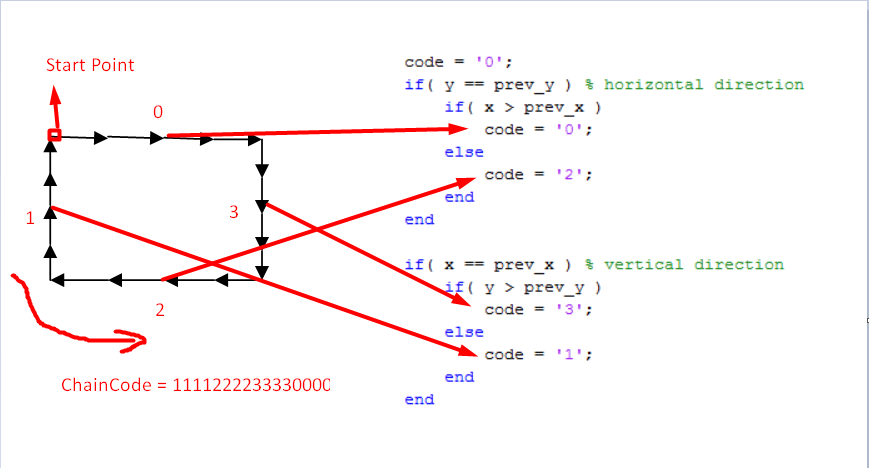
code\_serial = strcat(code\_serial,code); % append chain code

prev\_x = x;

prev\_y = y;

end

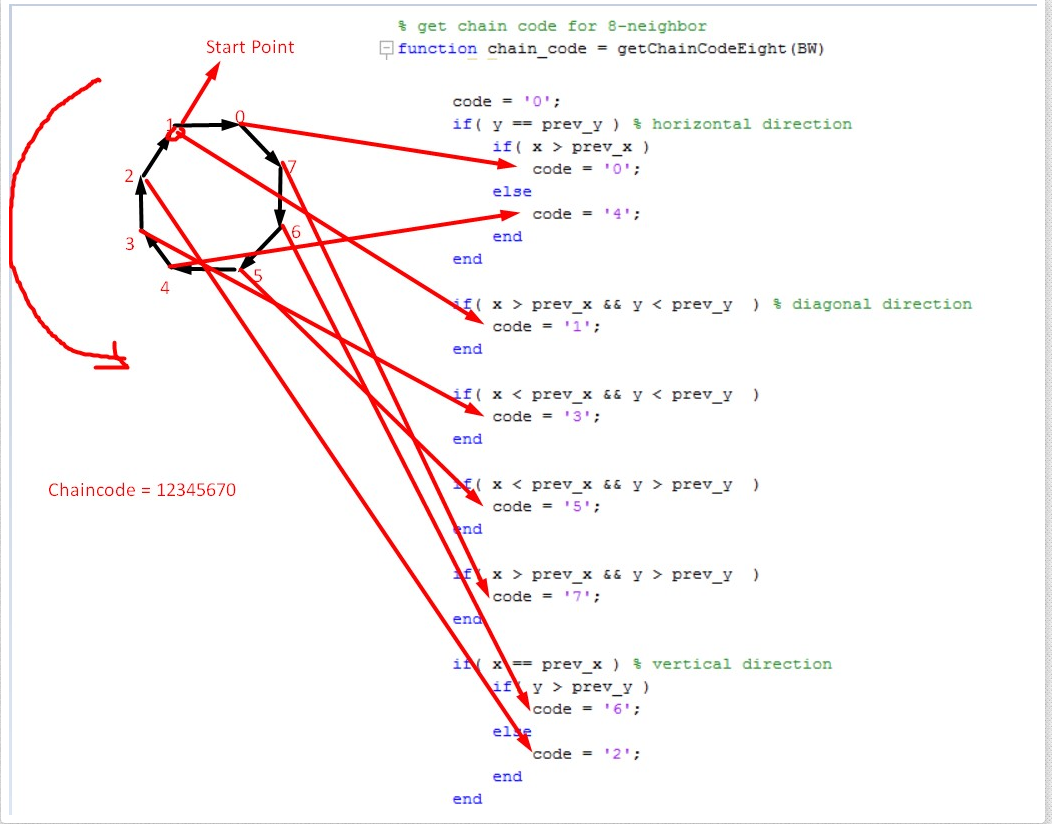
.when above algorithm apply to rectangle object, chain code is as follow



**2. function chain\_code = getChainCodeEight(BW)'s description**

This algorithm is similar to above 4 neighborhood chain code algorithm

.when 8 neighborhood chain code algorithm apply to circular like object, chain code is as follow



**3. function normal\_code = normalizeChainCode(code\_array)'s description**

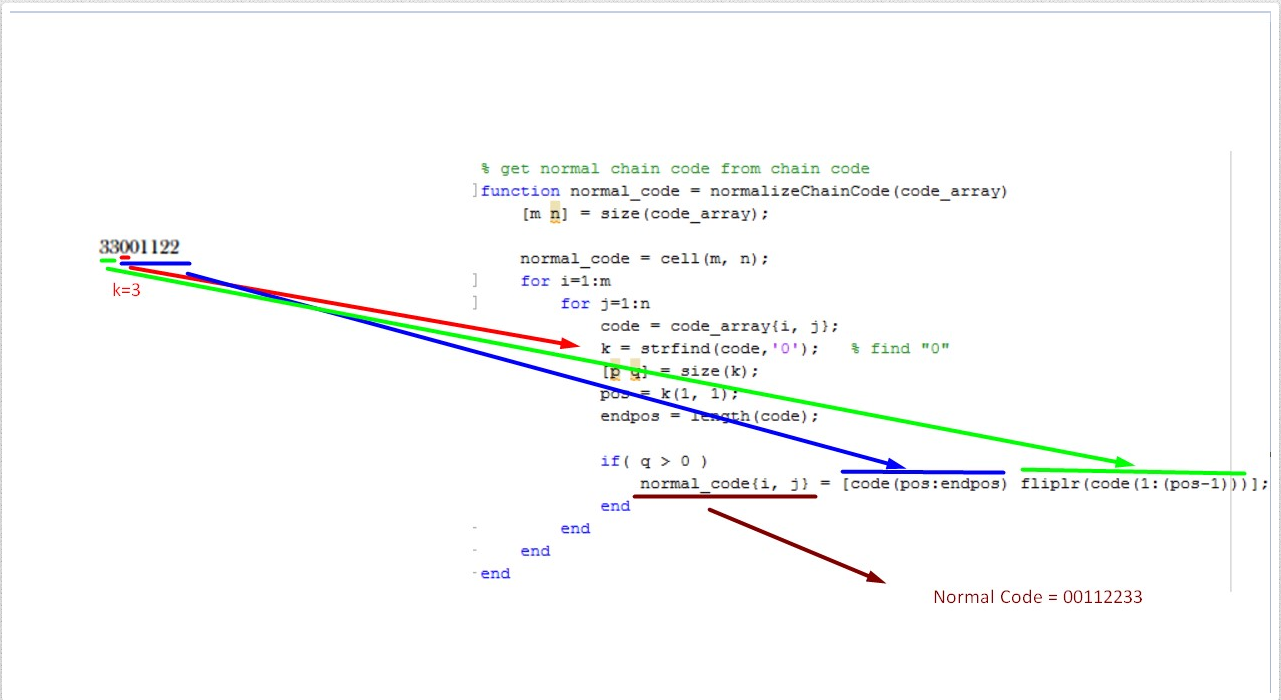
1) find first 0 code position

k = strfind(code,'0'); % find "0"

2) divide chain code to two parts and re-arrange in order to 0 code comes first position.

normal\_code{i, j} = [code(pos:endpos) fliplr(code(1:(pos-1)))];

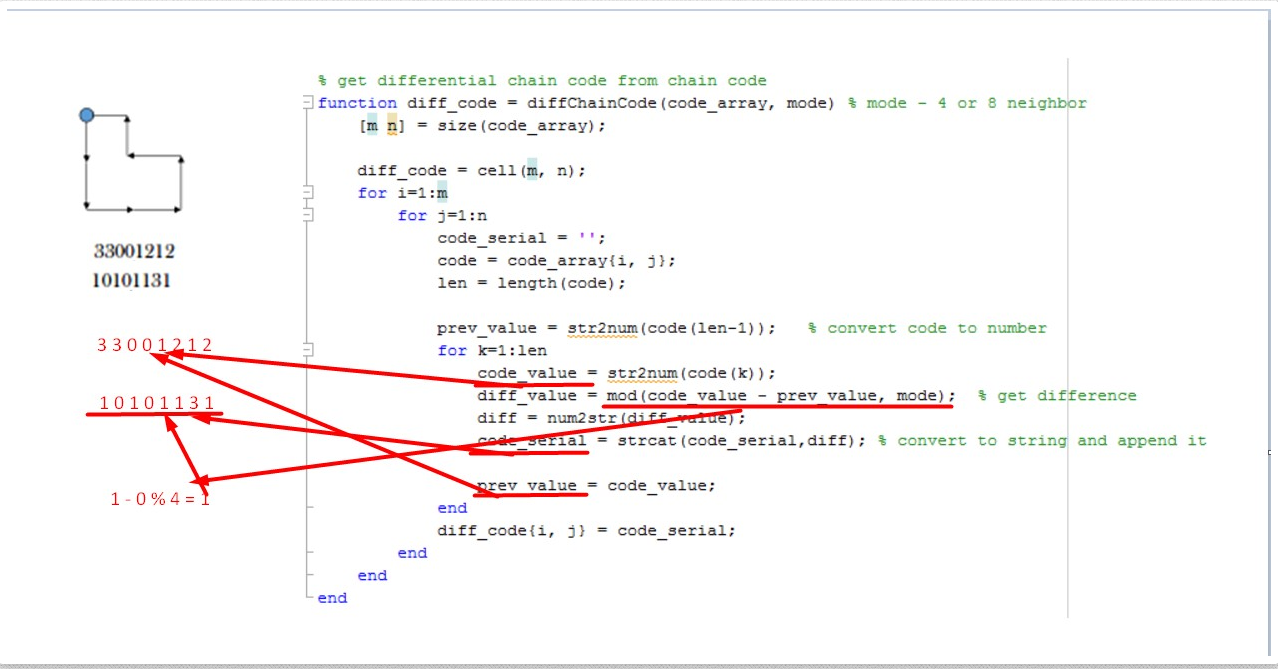
When above normal function is applied to 33001122 chain code, result normalized chain code is as 00112233



**4. function diff\_code = diffChainCode(code\_array, mode)'s description**

1) interate each code value in chain code series. k =1, ... , N, N: chain code length

2) diff\_code(k) = code(k) - code(k-1) % mod



**4. function f\_code = getFourierDescriptor(BW)'s description**

1) Find boundary of object

[B,L,N,A] = bwboundaries(BW, 8)

2) Make complex number for Fourier transform

x = 0:(np - 1);

m = ((-1) .^ x)';

s(:,1) = m .\* s(:, 1);

s(:,2) = m .\* s(:, 2);

s = s(:,1) + i \* s(:, 2);

3) Perform Fourier transform

z = fft(s); % perform fft

**5. function getShapeDistribution(code\_array, mode, num)'s description**

1) cumulate histogram count based code value bin.

for k=1:len

code\_value = str2num(code(k));

hist(code\_value + 1) = hist(code\_value + 1) + 1;

end

2) display histogram bar.

bar(hist);