end 8=0;%background only

%Pad label matrix with zeros 0 .This lets us write the

boundary\_following loop

%without worrying about going off the edge of the image.

1p.padarray(L,[1 1],0,1poth.),

%Compute the linear indexing offsets to take us from a pixel to

its

%neighbors. M.size(lp,1), if conn..8

%Order is N NE E SE S SW W NW.

offsets.(-1,M-1,14,M+1,1,-M+1,-M,-M-11;

else

%Order is N E S W.

end offsets=[-1"1,41];

%next\_search\_direction\_lut is a lookup tabl.Given the direction

from pixel

%k to pixel k+1,what is the direction to start with when

examining the

%neighborhood of pixel k+1?

if conn..8

next\_search\_dire lu

ction\_t[8 8 2 2 4 4 6 6];

else

next\_search\_direction\_lut[4 1 2 3];

end

%next\_direction\_lut is a lookup table.Given that we just looked at neighbor

%in a given direction,which neighbor do we look at %next?

if corm..

next\_direction\_lut.42 3 4 5 6 7 8 1];

else

next\_direction\_lutq2 3 4 *1];*

*end*

%Values used for marking the starting and boundary pixels. START.-1;

BOUNDARY.-2;

%Initialize scratch space in which to record the boundary pixels as well as

%follow the boundary.