

```

function [ P_padded ] = pad15( P )
%pad - produces a new matrix like P, except that the number of rows and
%columns are both multiples of 15.
% Note that P is n x m x 3.
% Rows and columns of 0 are added to the "end" of P, if needed.

% save original size of P
[rows,columns,colours] = size(P);

% determine the number of "extra" rows and columns in P
rmLCM = lcm(rows, 15);
cmLCM = lcm(columns, 15);
rm15 = rmLCM-rows;
cm15 = cmLCM-columns;

% Add 15-rm15 rows of zeros to each of the colour matrices
% The number of rows in each of P1, P2, P3 is now a multiple of 15.
if rm15 > 0
    P_pad(:,:,1) = padarray(P(:,:,1),[rm15,0],0,'post');
    P_pad(:,:,2) = padarray(P(:,:,2),[rm15,0],0,'post');
    P_pad(:,:,3) = padarray(P(:,:,3),[rm15,0],0,'post');
else
    P_pad = P;
end;

% Add 15-cm15 columns to the already enlarged matrices
% The number of columns in each of P1, P2, P3 is now a multiple of 15.
[rows,c1,colours] = size(P_pad);
if cm15 > 0
    P_padded(:,:,1) = padarray(P_pad(:,:,1),[0,cm15],0,'post');
    P_padded(:,:,2) = padarray(P_pad(:,:,2),[0,cm15],0,'post');
    P_padded(:,:,3) = padarray(P_pad(:,:,3),[0,cm15],0,'post');
else
    P_padded = P_pad;
end
end

```