MEASURING ANGLE OF INTERSECTION

%% Measuring Angle of Intersection

```
RGB = imread('gantrycrane.png');
imshow(RGB);

text(size(RGB,2),size(RGB,1)+15,'Image courtesy of Jeff Mather',...
    'FontSize',7,'HorizontalAlignment','right');

line([300 328],[85 103],'color',[1 1 0]);
line([268 255],[85 140],'color',[1 1 0]);

text(150,72,'Measure the angle between these beams','Color','y',...
    'FontWeight', 'bold');

% you can obtain the coordinates of the rectangular region using
% pixel information displayed by imtool
start_row = 34;
start_col = 208;

cropRGB = RGB(start_row:163, start_col:400, :);

imshow(cropRGB)

% Store (X,Y) offsets for later use; subtract 1 so that each offset will
% correspond to the last pixel before the region of interest
offsetX = start_col-1;
offsetY = start_row-1;

I = rgb2gray(cropRGB);
BW = imbinarize(I);
BW = ~BW;  % complement the image (objects of interest must be white)
imshow(BW)

dim = size(BW);

% horizontal beam
col1 = 4;
row1 = find(BW(:,col1), 1);

% angled beam
row2 = 12;
col2 = find(BW(row2,:), 1);

boundary1 = bwtraceboundary(BW, [row1, col1], 'N', 8, 70);
```

```
% set the search direction to counterclockwise, in order to trace downward.
boundary2 = bwtraceboundary(BW, [row2, col2], 'E', 8, 90,'counter');

imshow(RGB); hold on;

% apply offsets in order to draw in the original image
plot(offsetX+boundary1(:,2),offsetY+boundary1(:,1),'g','LineWidth',2);
plot(offsetX+boundary2(:,2),offsetY+boundary2(:,1),'g','LineWidth',2);

ab1 = polyfit(boundary1(:,2), boundary1(:,1), 1);
ab2 = polyfit(boundary2(:,2), boundary2(:,1), 1);

vect1 = [1 ab1(1)]; % create a vector based on the line equation
vect2 = [1 ab2(1)];
dp = dot(vect1, vect2);

% compute vector lengths
length1 = sqrt(sum(vect1.^2));
length2 = sqrt(sum(vect2.^2));

% obtain the larger angle of intersection in degrees
angle = 180-acos(dp/(length1*length2))*180/pi

intersection = [1 ,-ab1(1); 1, -ab2(1)] \ [ab1(2); ab2(2)];
% apply offsets in order to compute the location in the original,
% i.e. not cropped, image.
intersection = intersection + [offsetY; offsetX]

inter_x = intersection(2);
inter_y = intersection(1);

% draw an "X" at the point of intersection
plot(inter_x,inter_y,'yx','LineWidth',2);

text(inter_x-60, inter_y-30, [sprintf('%1.3f',angle),'{\circ}'],...
    'Color','y','FontSize',14,'FontWeight','bold');

interString = sprintf('(%2.1f,%2.1f)', inter_x, inter_y);

text(inter_x-10, inter_y+20, interString,...
    'Color','y','FontSize',14,'FontWeight','bold');
```