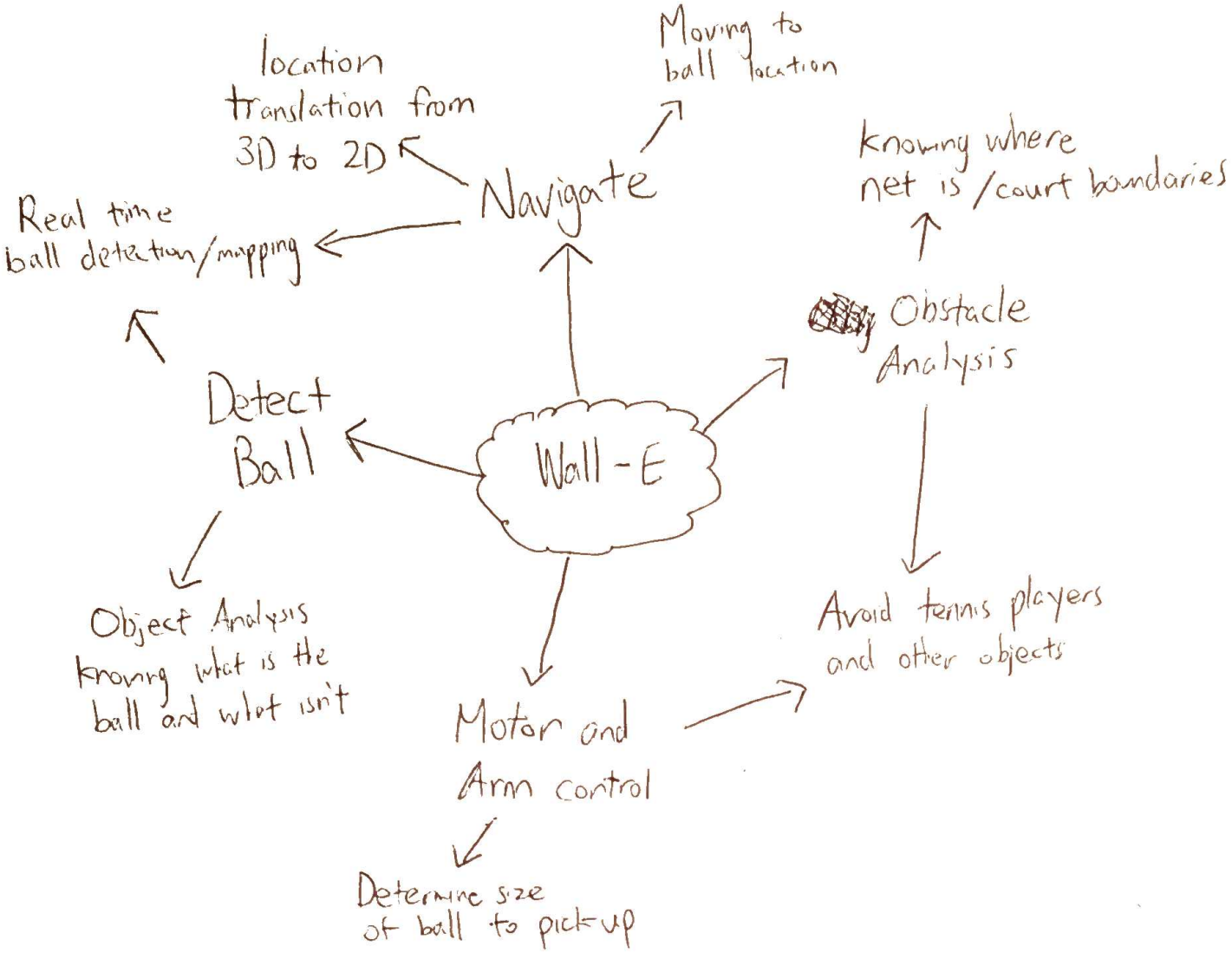
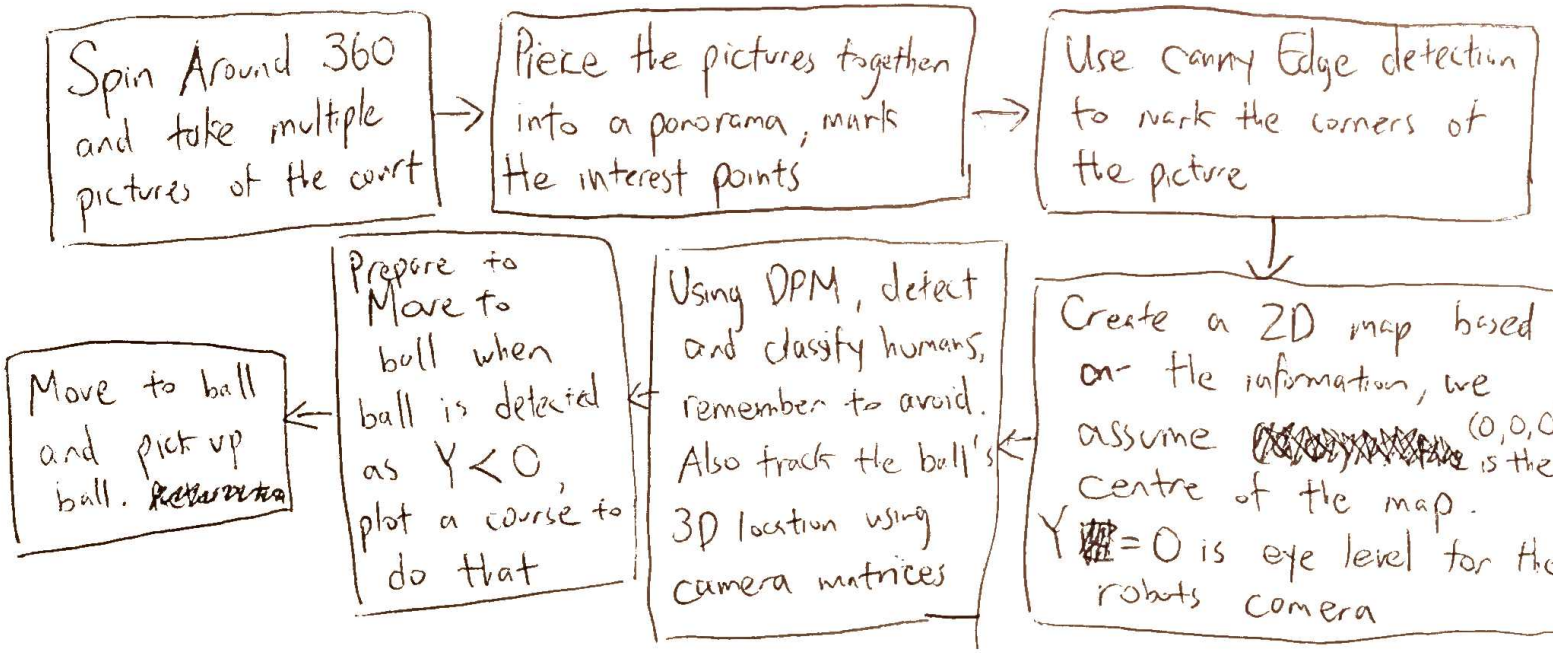
CSC420 Assignment 4

Alex Chang
1000064681

1)



b) Flowchart



1c)

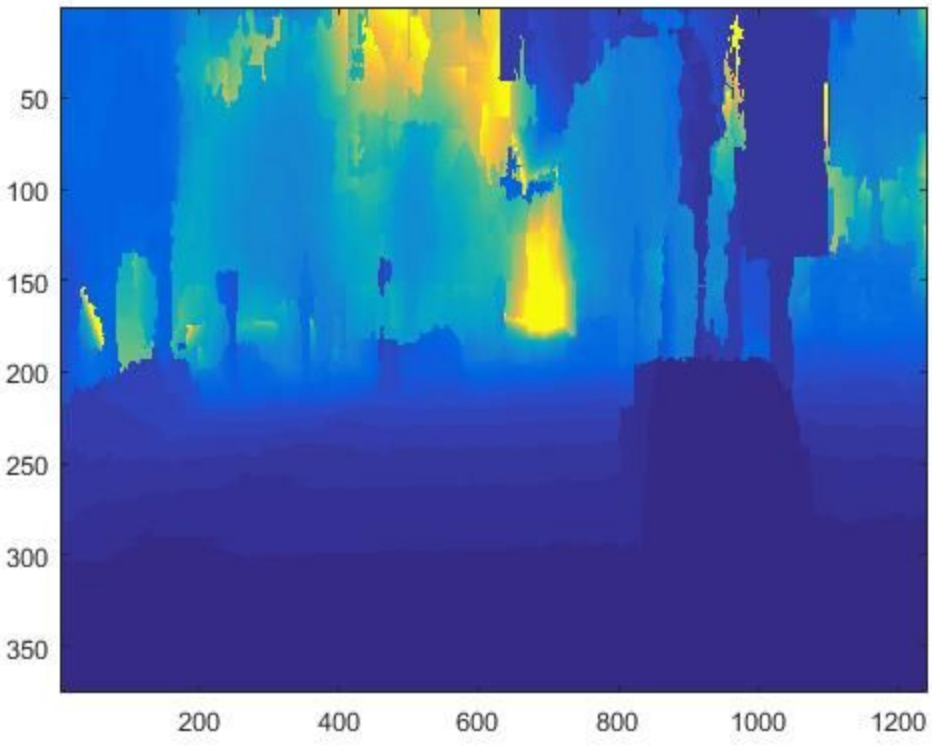
Psuedo code

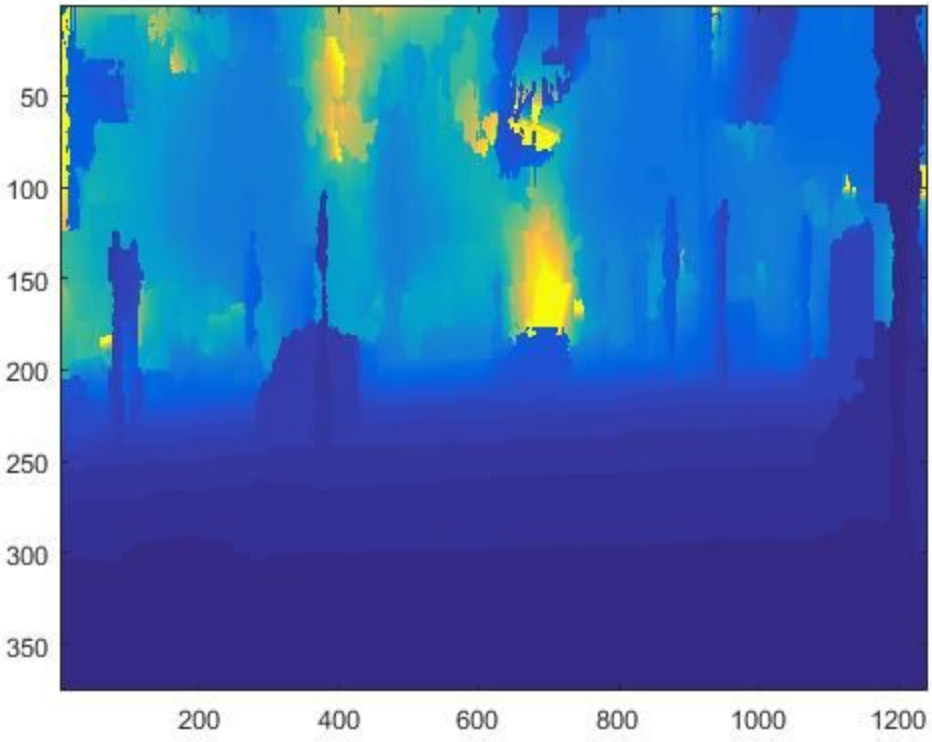
```
Capture 360-degree images of court
Find Features (imgs)
mark location of features # net, bench, etc
Combine Panorama( imgs, features )
Canny Edge detect (imgs)
2D map = (Panorama, features, canny edges)
& Set (0,0,0) as centre of map
While (true)
    Capture 360-degree images
    Find Ball (imgs) &
    (X,Y,Z) = calculate ball location (imgs, detected_img)
     if (Y<0) # ball may be not in play
          Capture 360 degree images
          (X2,Y2,Z2) = calculate ball distance (imgs, detected_img)
          if (X,Y,Z)==(X2,Y2,Z2) # ball is stopped
          Human[X,Y,Z] = calculate human distance (imgs)
          - plan route (Human[], X, Y, Z, imgs, panorama, features)
          go to ball (BX, B.Y, B Z)
```

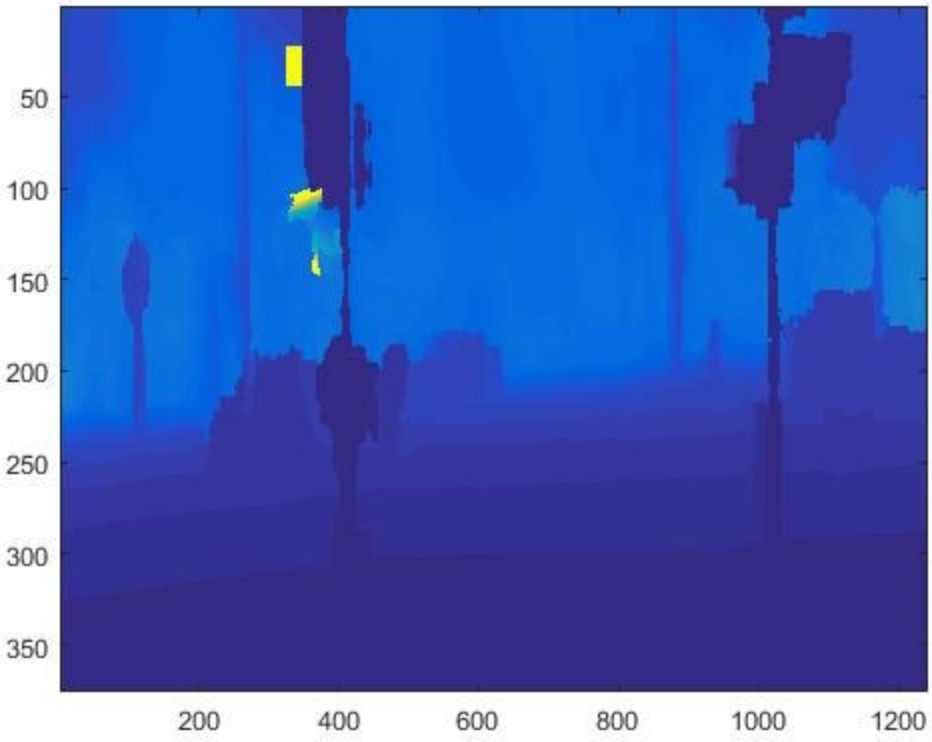
Q2a: Matlab Code + Output

q2a.m | q2bc.m | q2d.m | calc3D.m | q2e.m | se

```
data = getData([],'test','list');
ids = data.ids(1:3);

for i= 1:3
    calib = getData(ids{i}, 'test', 'calib');
    disp = getData(ids{i}, 'test', 'disp');
    disparity = disp.disparity;
    figure;imagesc(disparity);
    fT = calib.f*calib.baseline;
    depth = fT./disparity;
    dfosho = depth;
    dfosho(dfosho>255)=255+eps;
    figure;imagesc(dfosho);
end
```




50
100
150
200
250
300
350
200
400
600
800
1000
1200

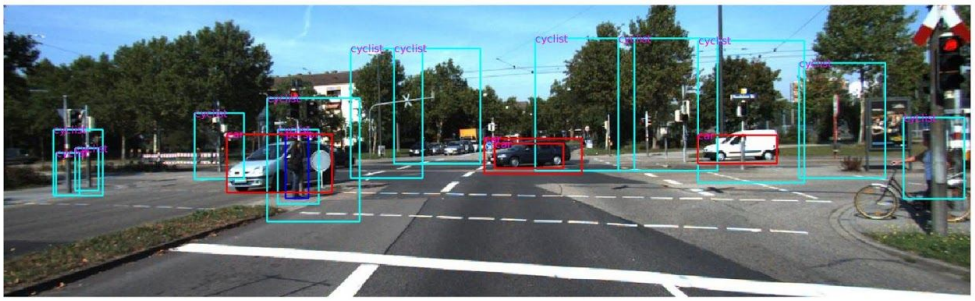

50
100
150
200
250
300
350
200
400
600
800
1000
1200

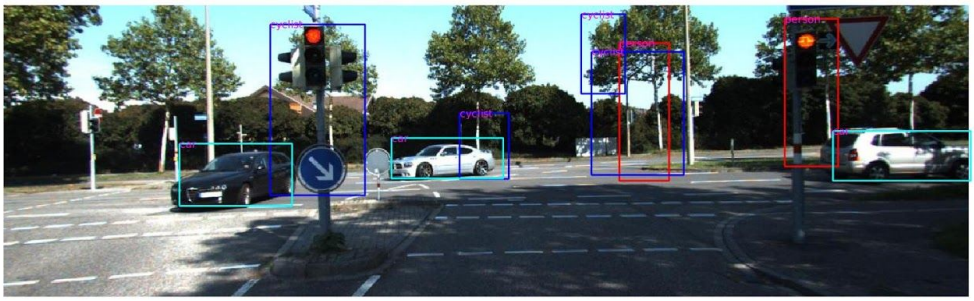
Q2b/Q2c Code and Output

q2a.m | q2bc.m | q2d.m | calc3D.m | q2e.m | segim2.m | getData.m | q2f.m

```
data = getData([], 'test','list');
ids = data.ids(1:3);
dtype = {'person', 'bicycle', 'car'};
person_threshold = -0.5;
cyclist_threshold = -0.355;
car_threshold = -0.5487;
for idx= 1:3

    if idx == 1
        dtype = 'detector-person';
        thresh = person_threshold;
        rtype = 'person';
    elseif idx == 2
        dtype = 'detector-cyclist';
        thresh = cyclist_threshold;
        rtype = 'cyclist';
    else
        dtype = 'detector-car';
        thresh = car_threshold;
        rtype = 'car';
    end
    for k= 1:3
        data = getData([], [], dtype);
        model = data.model;
        imdata = getData(ids{k}, 'test', 'left');
        im = imdata.im;
        f = 1.5;
        imr = imresize(im,f); % if we resize, it works better for small objects

        % detect objects
        fprintf('running the detector, may take a few seconds...\n');
        tic;
        [ds, bs] = imgdetect(imr, model, thresh); % you may need to reduce the threshol
        e = toc;
        fprintf('finished! (took: %0.4f seconds)\n', e);
        nms_thresh = 0.5;
        top = nms(ds, nms_thresh);
        if model.type == model_types.Grammar
          bs = [ds(:,1:4) bs];
        end
        if ~isempty(ds)
            % resize back
            ds(:, 1:end-2) = ds(:, 1:end-2)/f;
            bs(:, 1:end-2) = bs(:, 1:end-2)/f;
        end;
```

```
47 -            fprintf('detections:\n');
48 -            ds = ds(top, :);
49 -            result=sprintf('../data/test/results/%s-%s',ids{k}, rtype);
50 -            save(result, 'ds');
51 -        end
52 -    end
53
54 -    color = {'blue', 'cyan', 'red', 'magenta'};
55 -    for k = 1:3
56 -        data = getData(ids{k}, 'test', 'detection-results');
57 -        imdata = getData(ids{k}, 'test', 'left');
58 -        im = imdata.im;
59 -        figure; axis ij; hold on; drawnow;
60 -        imagesc(im);
61 -        for c = 1:3
62 -            showboxesMy(im, data.score{c}.ds(:,1:4), color{c});
63 -            text(data.score{c}.ds(:,1), data.score{c}.ds(:,2), data.class{c},'Color',color{
64 -        end
65 -    end
66
```


cyclist
cyclist
cyclist
cyclist
cyclist


cyclist
cyclist
cyclist
person

Q2d/Q2e Code and helper functions calc3D, segim2 and outputs (Also changes made to getData.m; added case detection-results to obtain results from calculations)

+4 | q2a.m | q2bc.m | q2d.m | calc3D.m | q2e.m | segim2.m | getData.m

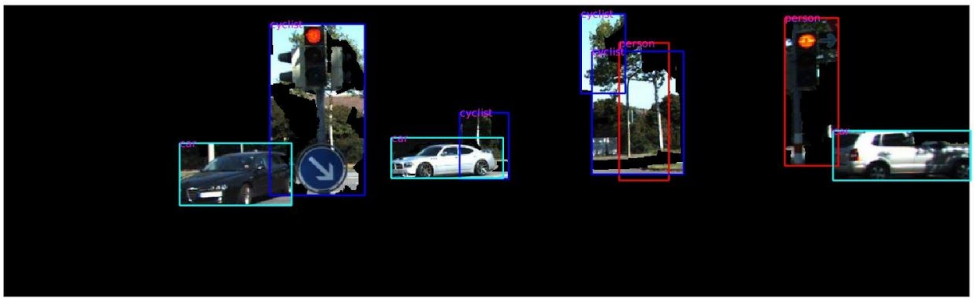
```
1 -   data = getData([], 'test','list');
2 -   ids = data.ids(1:3);
3 -   for i = 1:3
4 -       data = getData(ids{i}, 'test', 'detection-results');
5 -       calib = getData(ids{i}, 'test', 'calib');
6 -       disp = getData(ids{i}, 'test', 'disp');
7 -       disparity = disp.disparity;
8 -       depth = (calib.f*calib.baseline)./disparity;
9 
10 -      for j = 1:3
11 -          data.score{j} = calc3D(data.score{j}, depth, calib);
12 -          ds = data.score{j}.ds;
13 -          fname=sprintf('../data/test/results/%s-%s',ids{i}, data.class{j});
14 -          save(fname, 'ds');
15 -      end
16 -  end
17 
18 
```

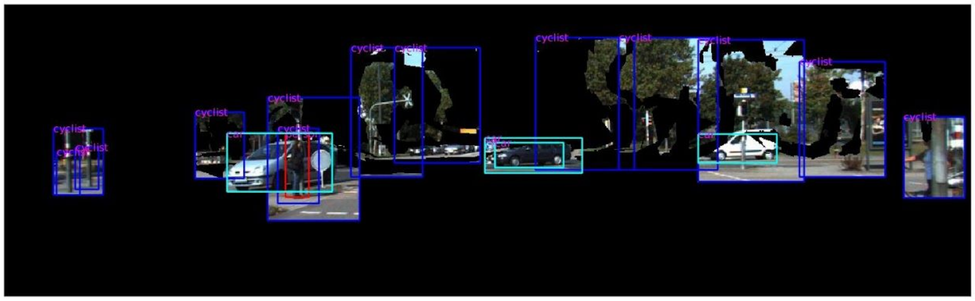
+4 | q2a.m | q2bc.m | q2d.m | calc3D.m | q2e.m | getData.m | q2f.

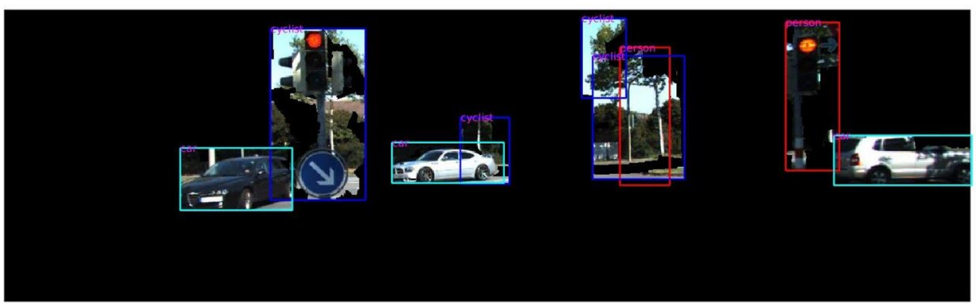
```
1   function output = calc3D(score, depth, calib)
2
3 -     x_thresh = 350;
4 -     y_thresh = 1199;
5 -     size_i = size(score.ds,1);
6 -     for i = 1:size_i
7
8 -         x = score.ds(i,2):score.ds(i,4);
9 -         y = score.ds(i,1):score.ds(i,3);
10 -        x = round(x);
11 -        y = round(y);
12 -        x = x(x<=x_thresh);
13 -        y = y(y<=y_thresh);
14 -        z_depth = depth(x,y);
15 -        Z = mode(round(z_depth(:)));
16 -        cX = score.ds(i,1)+(score.ds(i,3)-score.ds(i,1))/2;
17 -        cY = score.ds(i,2)+(score.ds(i,4)-score.ds(i,2))/2;
18 -        X = (Z.*(cX - calib.K(1,3)))/calib.f;
19 -        Y = (Z.*(cY - calib.K(2,3)))/calib.f;
20 -        data.score.ds(i,7) = X;
21 -        data.score.ds(i,8) = Y;
22 -        data.score.ds(i,9) = Z;
23
24 -    end
25 -    output = score;
26 - end
```

```
    case {'detector-car', 'detector-person', 'detector-pedestrian', 'detector-cyclist'}
        cls = strrep(whatdata, 'detector-', '');
        files = dir(fullfile(DETECTOR_DIR, sprintf('%s_final*.mat', cls)));
        if isempty(files)
            fprintf('file doesn''t exist!\n');
        else
            data = load(fullfile(DETECTOR_DIR, files(1).name));
        end;
    case {'superpixels'}
        dispdir = fullfile(DATA_DIR, imset, 'results');
        spfile = fullfile(dispdir, sprintf('%s_segment.png', imname));
        spim = [];
        if ~exist(spfile, 'file')
            fprintf('you haven''t ran spsstereo code yet...\n');
        else
            spim = imread(spfile);
            spim = double(spim);
        end;
        data.spim = spim;
    case {'detection-results'}
        dispdir = fullfile(DATA_DIR, imset, 'results');
        class = {'person', 'cyclist', 'car'};
        data.class = class;
        for i = 1:3
            results = fullfile(dispdir, sprintf('%s-%s.mat', imname, class{i}));
            data.score{i} = load(results);
        end
    otherwise
        disp('unknown data type, try again');

end;
```

```
+4 | q2a.m | q2bc.m | q2d.m | calc3D.m | q2e.m | segim2.m | getData.m | q2f.m | +

1 -   data = getData([], 'test','list');
2 -   ids = data.ids(1:3);
3 -   col = {'red', 'blue', 'cyan', 'magenta'};
4
5
6 -   for i = 1:3
7 -       data = getData(ids{i}, 'test', 'detection-results');
8 -       imagedata = getData(ids{i}, 'test', 'left');
9 -       image = imagedata.im;
10 -      calib = getData(ids{i}, 'test', 'calib');
11 -      disp = getData(ids{i}, 'test', 'disp');
12 -      disparity = disp.disparity;
13 -      depth = (calib.f*calib.baseline)./disparity;
14
15 -      segment_im = segmentation_img(depth, calib);
16 -      for c = 1:3
17
18 -          X = data.score{c}.ds(:,7);
19 -          Y = data.score{c}.ds(:,8);
20 -          Z = data.score{c}.ds(:,9);
21 -          image_section = seg_im2(X,Y,Z);
22
23 -      end
24 -      figure; axis ij; hold on
25 -      imagesc(image_section);
26 -      for c = 1:3
27 -          showboxesMy(image_section, data.score{c}.ds(:,1:4), col{c});
28 -          text(data.score{c}.ds(:,1), data.score{c}.ds(:,2), data.class{c},'Color',col{4},'
29 -      end
30 -      hold off;
31
32
33 -  end
```




cyclist
cyclist
cyclist
cyclist
cyclist
cyclist
cyclist
cyclist
cyclist
cyclist
cyclist
cyclist


cyclist
car
car
cyclist
cyclist
person
person

Q2f Code and output

+4 | q2a.m | q2bc.m | q2d.m | calc3D.m | q2e.m | segim2.m | getData.m | q2f.m | +

```
data = getData([], 'test','list');
ids = data.ids(1:3);
for i = 1:3

    data = getData(ids{i}, 'test', 'detection-results');
    fprintf('image %d\n', i);
    closest_d = Inf;
    closest = {};
    for j = 1:3

        X = data.score{j}.ds(:,7);
        Y = data.score{j}.ds(:,8);
        Z = data.score{j}.ds(:,9);

        for k = 1:size(X)
            distance = norm([X(k), Y(k), Z(k)]);
            label = data.class{j};

            if X(k) >= 0
                txt = 'to your right';
            else
                txt = 'to your left';
            end
            fprintf('There is a %s %0.1f meters %s \n', label, abs(X(k)), txt);
            fprintf('It is %0.1f meters away from you \n', distance);

            if closest_d > distance
                closest_d = distance;
                closest{1} = label;
                closest{2} = distance;
                closest{3} = X(k);
            end
        end
    end

    fprintf('\nThe %s is closest to you at %0.1f meters\n', closest{1}, closest{2});
    if closest{3} >= 0
        ctxt = 'to your right';
    else
        ctxt = 'to your left';
    end
    fprintf('It is %0.1f meters away %s \n', abs(closest{3}), ctxt);
    fprintf('\n\n');
end
```

Command Window

```
>> q2f
image 1
There is a person -11.1 meters to your left
It is 78.9 meters away from you
There is a cyclist 19.0 meters to your right
It is 67.0 meters away from you
There is a cyclist 7.8 meters to your right
It is 18.7 meters away from you
There is a cyclist -3.8 meters to your left
It is 26.3 meters away from you
There is a cyclist 12.2 meters to your right
It is 69.2 meters away from you
There is a cyclist -26.9 meters to your left
It is 78.0 meters away from you
There is a cyclist 3.5 meters to your right
It is 129.1 meters away from you
There is a cyclist -24.9 meters to your left
It is 47.1 meters away from you
There is a cyclist -26.7 meters to your left
It is 105.5 meters away from you
There is a car 10.2 meters to your right
It is 48.1 meters away from you
There is a car 3.3 meters to your right
It is 7.8 meters away from you
There is a car -3.4 meters to your left
It is 35.2 meters away from you
There is a car 10.0 meters to your right
It is 48.1 meters away from you

The car is closest to you at 7.8 meters
It is 3.3 meters away to your right
```

```
image 2
There is a person -5.5 meters to your left
It is 17.9 meters away from you
There is a cyclist -5.4 meters to your left
It is 17.9 meters away from you
There is a cyclist 12.2 meters to your right
It is 65.3 meters away from you
There is a cyclist 18.3 meters to your right
It is 57.1 meters away from you
There is a cyclist -11.2 meters to your left
It is 70.0 meters away from you
There is a cyclist -8.3 meters to your left
It is 19.8 meters away from you
There is a cyclist -15.3 meters to your left
It is 26.0 meters away from you
There is a cyclist -5.3 meters to your left
It is 18.8 meters away from you
There is a cyclist -18.8 meters to your left
It is 32.9 meters away from you
There is a cyclist -14.9 meters to your left
It is 25.8 meters away from you
There is a cyclist 28.7 meters to your right
It is 65.7 meters away from you
There is a cyclist 14.9 meters to your right
It is 27.4 meters away from you
There is a cyclist -6.9 meters to your left
It is 96.4 meters away from you
There is a cyclist 7.3 meters to your right
It is 11.6 meters away from you
There is a car 2.9 meters to your right
It is 32.1 meters away from you
There is a car 3.1 meters to your right
It is 32.2 meters away from you
There is a car 16.2 meters to your right
It is 38.6 meters away from you
There is a car -6.4 meters to your left
It is 19.1 meters away from you

The cyclist is closest to you at 11.6 meters
It is 7.3 meters away to your right
```

```
image 3
There is a person 13.1 meters to your right
It is 46.0 meters away from you
There is a person 4.8 meters to your right
It is 9.3 meters away from you
There is a cyclist 0.3 meters to your right
It is 25.0 meters away from you
There is a cyclist 9.9 meters to your right
It is 45.6 meters away from you
There is a cyclist 13.8 meters to your right
It is 50.0 meters away from you
There is a cyclist -2.3 meters to your left
It is 8.3 meters away from you
There is a car -1.3 meters to your left
It is 25.0 meters away from you
There is a car -6.4 meters to your left
It is 16.4 meters away from you
There is a car 13.6 meters to your right
It is 22.6 meters away from you

The cyclist is closest to you at 8.3 meters
It is -2.3 meters away to your left
```