CMSC 828D: Fundamentals of Computer Vision Homework 6[1]

Instructors: Larry Davis, Ramani Duraiswami,
Daniel DeMenthon, and Yiannis Aloimonos

Solution based on homework submitted by Haiying Liu

Camera calibration using a linear method

We want to calibrate the camera of a robot vehicle. We place a large cubic frame of size 4 meters on the road several meters in front of the vehicle. The positions of the eight corners of the cubic frame are defined with respect to a world coordinate system with its axes parallel to the cube edges and with its origin at the center of the cube. The world coordinates of the cube vertices are

2	2	2
-2	2	2
-2	2	-2
2	2	-2
2	-2	2
-2	-2	2
-2	-2	-2
2	-2	-2

There are bright light sources at the corners of the cube that are all visible and are easy to detect in the camera image.

We detect the cube corners at the following pixel positions in the camera image:

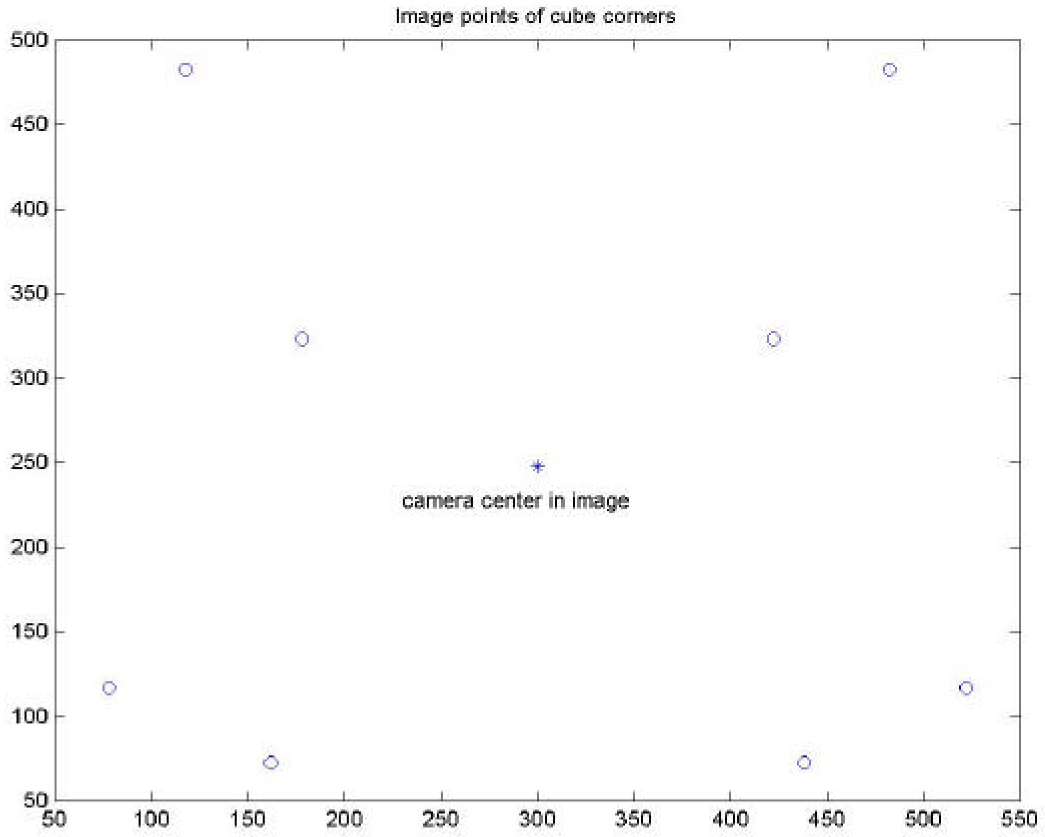
422	323
178	323
118	483
482	483
438	73
162	73
78	117
522	117

For each image point, the left coordinate defines its horizontal position (pixel column) and the right coordinate defines its vertical position. We were able to find out the correspondences between image points and world cube corners, so that the first set of image coordinates corresponds to the first set of world coordinates, and so on.

1. Draw the image points, using small circles for each image point.

[1] The e-version of this homework solution is posted at http://www.cfar.umd.edu/~hyliu/CMSC828D/hw6.zip.

Solution: The following image shows the image points, and also the camera center found by computing K matrix at the end of this homework. Note that the image is upside down, as the origin for the pixel positions is actually at the top left of the image.



2. Write a Matlab function that takes as argument the homogeneous coordinates of one cube corner and the homogeneous coordinates of its image, and returns 3 rows of the matrix A (slide 25 and slide 26 of the Camera Calibration pdf document). This matrix A will be used to compute the 12 elements of the projection matrix P.

NOTE: The slides 25 and 26 showed zero vectors with 3 elements (03) in matrix A. This was wrong. The zero vectors should have 4 elements (04). This has been corrected (but your browser may still be caching the older version).

See web page for following questions.

Solution: The Matlab script (`hw6.m`) and results are:

```
function hw6
% Syntax: hw6
%
% Description: CMSC828D HW6
%
% Author: Haiying Liu
%   Date: Oct. 5, 2000
%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

dbstop if error

msg = nargchk(0, 0, nargin);
if (~isempty(msg))
  error(strcat('ERROR:', msg));
end

clear msg;

%======================================================================
%= Turn on the diary to save the result.

diary off;

filename = 'hw6.txt';
if (exist(filename, 'file'))
  delete(filename);
end

eval(['diary ', filename]);

disp('» hw6');

%======================================================================
%= Initialization.

world_coord = [ ...
     2,  2,  2; ...
    -2,  2,  2; ...
    -2,  2, -2; ...
     2,  2, -2; ...
     2, -2,  2; ...
    -2, -2,  2; ...
    -2, -2, -2; ...
     2, -2, -2; ...
  ];

image_coord = [ ...
    422, 323; ...
    178, 323; ...
    118, 483; ...
    482, 483; ...
    438,  73; ...
    162,  73; ...
     78, 117; ...
    522, 117; ...
  ];

%=======================================================================
%= Draw the image points.

figure;
plot(image_coord(:, 1), image_coord(:, 2), 'o');
title('Image points of cube corners');

%=======================================================================
%= Compute A.

disp(' ');
disp('::::::::::::');
disp(':: Part 3 ::');
disp('::::::::::::');
disp(' ');

nPoints = size(world_coord, 1);
A       = zeros(3 * nPoints, 12);
for index = 1:nPoints
  worldP            = world_coord(index, :)';
  imageP            = image_coord(index, :)';
  Ai                = gen3rows(worldP, imageP);
  row               = 3 * index - 2;
  A(row:row + 2, :) = Ai;
end

A

%=======================================================================
%= Compute P.

disp(' ');
disp('::::::::::::');
disp(':: Part 4 ::');
disp('::::::::::::');
disp(' ');

[U, S, V] = svd(A);
nCol_V    = size(V, 2);
P_col     = V(:, nCol_V);

P     = reshape(P_col, 4, 3)'

%=======================================================================
%= Compute T.

disp(' ');
disp('::::::::::::');
disp(':: Part 5 ::');
disp('::::::::::::');
disp(' ');

[U2, S2, V2] = svd(P);

nCol_V2      = size(V2, 2);
```

```
C            = V2(:, nCol_V2);

camera_center = C(1:3) ./ C(4)

%======================================================================
%= Compute M.

disp(' ');
disp('::::::::::::');
disp(':: Part 6 ::');
disp('::::::::::::');
disp(' ');

IC = [eye(3), camera_center];

M = P * IC' * inv(IC * IC');
M = M ./ M(3, 3)

%======================================================================
%= Compute Rx.

disp(' ');
disp('::::::::::::');
disp(':: Part 7 ::');
disp('::::::::::::');
disp(' ');

cosx =  M(3, 3) / sqrt(M(3, 3) * M(3, 3) + M(3, 2) * M(3, 2));
sinx = -M(3, 2) / sqrt(M(3, 3) * M(3, 3) + M(3, 2) * M(3, 2));

Rx   = [ ...
    1,    0,     0; ...
    0, cosx, -sinx; ...
    0, sinx,  cosx; ...
  ]

ax   = atan(sinx ./ cosx) * 180 / pi
disp('(in degree)');

N    = M * Rx

%======================================================================
%= Compute Rz and K.

disp(' ');
disp('::::::::::::');
disp(':: Part 8 ::');
disp('::::::::::::');
disp(' ');

% Since N(3, 1) is small enough, we directly compute Rz.

% Compute Rz.
cosz =  N(2, 2) / sqrt(N(2, 1) * N(2, 1) + N(2, 2) * N(2, 2));
sinz = -N(2, 1) / sqrt(N(2, 1) * N(2, 1) + N(2, 2) * N(2, 2));

Rz   = [ ...
```

```
    cosz, -sinz, 0; ...
    sinz,  cosz, 0; ...
       0,     0, 1; ...
  ]

% Compute az.
az   = atan(sinz ./ cosz) * 180 / pi
disp('(in degree)');

% Compute K.
K    = N * Rz;
K    = K ./ K(3, 3)

% Compute focal lengths in pixels.
disp(' ');
disp('Focal lengths in pixels');
disp(' ');

fx = K(1, 1)
fy = K(2, 2)

% Compute pixel coordinates of the image center of the camera.
disp(' ');
disp('Camera center in image (pixel coordinates)');
disp(' ');

u0 = K(1, 3)
v0 = K(2, 3)

hold on;
plot(u0, v0, '*');
text(u0 - 70, v0 - 20, 'camera center in image');

print -djpeg hw6_1;

%======================================================================
%= Stop diarying.

diary off;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [Ai] = gen3rows(world_coord, image_coord)
% Syntax: Ai = gen3rows(world_coord, image_coord)
%
%         world_coord  - world coordinate of a point
%         image_coord  - image coordinate of the
%                        correspondent point
%         Ai           - the 3 rows of the A.
%
% Description: Generate 3 rows of matrix A to
%              compute projection matrix P
%
% Author: Haiying Liu
%   Date: Oct. 5, 2000
%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

dbstop if error

msg = nargchk(2, 2, nargin);
if (~isempty(msg))
  error(strcat('ERROR:', msg));
end

clear msg;

%======================================================================

zeros4 = zeros(4, 1);

u = image_coord(1);
v = image_coord(2);

Xi = [world_coord; 1];

Ai = [ ...
      zeros4',    -Xi',  v * Xi'; ...
         Xi', zeros4', -u * Xi'; ...
    -v * Xi', u * Xi',  zeros4'; ...
    ];
» hw6

::::::::::::
:: Part 3 ::
::::::::::::


A =

  Columns 1 through 6

           0           0           0           0          -2          -2
           2           2           2           1           0           0
        -646        -646        -646        -323         844         844
           0           0           0           0           2          -2
          -2           2           2           1           0           0
         646        -646        -646        -323        -356         356
           0           0           0           0           2          -2
          -2           2          -2           1           0           0
         966        -966         966        -483        -236         236
           0           0           0           0          -2          -2
           2           2          -2           1           0           0
        -966        -966         966        -483         964         964
           0           0           0           0          -2           2
           2          -2           2           1           0           0
        -146         146        -146         -73         876        -876
           0           0           0           0           2           2
          -2          -2           2           1           0           0
         146         146        -146         -73        -324        -324
           0           0           0           0           2           2
          -2          -2          -2           1           0           0
         234         234         234        -117        -156        -156
```

```
           0           0           0           0          -2           2
           2          -2          -2           1           0           0
        -234         234         234        -117        1044       -1044

  Columns 7 through 12

          -2          -1         646         646         646         323
           0           0        -844        -844        -844        -422
         844         422           0           0           0           0
          -2          -1        -646         646         646         323
           0           0         356        -356        -356        -178
         356         178           0           0           0           0
           2          -1        -966         966        -966         483
           0           0         236        -236         236        -118
        -236         118           0           0           0           0
           2          -1         966         966        -966         483
           0           0        -964        -964         964        -482
        -964         482           0           0           0           0
          -2          -1         146        -146         146          73
           0           0        -876         876        -876        -438
         876         438           0           0           0           0
          -2          -1        -146        -146         146          73
           0           0         324         324        -324        -162
         324         162           0           0           0           0
           2          -1        -234        -234        -234         117
           0           0         156         156         156         -78
        -156          78           0           0           0           0
           2          -1         234        -234        -234         117
           0           0       -1044        1044        1044        -522
       -1044         522           0           0           0           0


::::::::::::
:: Part 4 ::
::::::::::::


P =

   -0.1923   -0.0281   -0.0785   -0.7346
    0.0002   -0.2043    0.0001   -0.6121
    0.0000   -0.0001   -0.0003   -0.0024


::::::::::::
:: Part 5 ::
::::::::::::


camera_center =

   -0.0002
   -2.9988
   -8.2866


::::::::::::
```

```
:: Part 6 ::
::::::::::::


M =

 -755.8644  109.6389  299.9343
    0.9140 -619.5401  507.2018
    0.0029    0.3655    1.0000


::::::::::::
:: Part 7 ::
::::::::::::


Rx =

    1.0000         0         0
         0    0.9392    0.3433
         0   -0.3433    0.9392


ax =

  -20.0772

(in degree)

N =

 -755.8644    0.0130  319.3451
    0.9140 -756.0064  263.7003
    0.0029   -0.0000    1.0647


::::::::::::
:: Part 8 ::
::::::::::::


Rz =

   -1.0000    0.0012         0
   -0.0012   -1.0000         0
         0         0    1.0000


az =

    0.0693

(in degree)

K =

  709.9308   -0.8705  299.9388
```

```
         0  710.0652  247.6755
   -0.0027    0.0000    1.0000


Focal lengths in pixels


fx =

  709.9308


fy =

  710.0652


Camera center in image (pixel coordinates)


u0 =

  299.9388


v0 =

  247.6755
```