

## Horizon and affine rectification

Identify pairs of parallel lines in the image of a plane and get their vanishing points. Compute the vanishing line and use it to perform an affine rectification of the plane.

Credits: Giacomo Boracchi Edits: Luca Magri, October 2023, for comments and suggestions write to [luca.magri@polimi.it](mailto:luca.magri@polimi.it)

### Contents

- [load an image of a plane](#)
- [compute the lines passing through the points](#)
- [we can get vanishing points from pairs of parallel line in closed form](#)
- [Compute the horizon \(the vanishing line of the plane\)](#)
- [Given the horizon we can rectify the image](#)
- [rectify the image and display it](#)

```
clc;
clear;
```

### load an image of a plane

```
im = imread("E2_data/floor.jpg");
im = imresize(im,0.5);

% select four points
% important select the point clockwise or counterclockwise

figure;
imshow(im);
hold on;
[x, y] = getpts();
a = [x(1); y(1); 1];
b = [x(2); y(2); 1];
c = [x(3); y(3); 1];
d = [x(4); y(4); 1];

FNT_SZ = 20;

text(a(1), a(2), 'a', 'FontSize', FNT_SZ, 'Color', 'b')
text(b(1), b(2), 'b', 'FontSize', FNT_SZ, 'Color', 'b')
text(c(1), c(2), 'c', 'FontSize', FNT_SZ, 'Color', 'b')
text(d(1), d(2), 'd', 'FontSize', FNT_SZ, 'Color', 'b')
```



#### compute the lines passing through the points

```

lab = cross(a, b);
lbc = cross(b, c);
lcd = cross(c, d);
lda = cross(d, a);

```

#### we can get vanishing points from pairs of parallel line in closed form

```

v1 = cross(lab, lcd);
v2 = cross(lbc, lda);

% remember these have to be normalized before plotting them
v1 = v1/v1(3);
v2 = v2/v2(3);

```

#### Compute the horizon (the vanishing line of the plane)

the horizon is just the line that passes through the vanishing points

```

horizon = cross(v1, v2);

% display the result
plot([a(1), v1(1)], [a(2), v1(2)], 'b');
plot([d(1), v1(1)], [d(2), v1(2)], 'b');
plot([b(1), v1(1)], [b(2), v1(2)], 'b');
plot([c(1), v1(1)], [c(2), v1(2)], 'b');

plot([a(1), v2(1)], [a(2), v2(2)], 'b');
plot([c(1), v2(1)], [c(2), v2(2)], 'b');
plot([b(1), v2(1)], [b(2), v2(2)], 'b');
plot([d(1), v2(1)], [d(2), v2(2)], 'b');

plot([v1(1), v2(1)], [v1(2), v2(2)], 'b--');
text(v1(1), v1(2), 'v1', 'FontSize', FNT_SZ, 'Color', 'b')
text(v2(1), v2(2), 'v2', 'FontSize', FNT_SZ, 'Color', 'b')

hold off

```

Select another pair of parallel lines



```
disp('Select another pair of parallel lines');
figure(gcf),
hold on;
[x, y] = getpts();
e = [x(1); y(1); 1];
f = [x(2); y(2); 1];
g = [x(3); y(3); 1];
h = [x(4); y(4); 1];

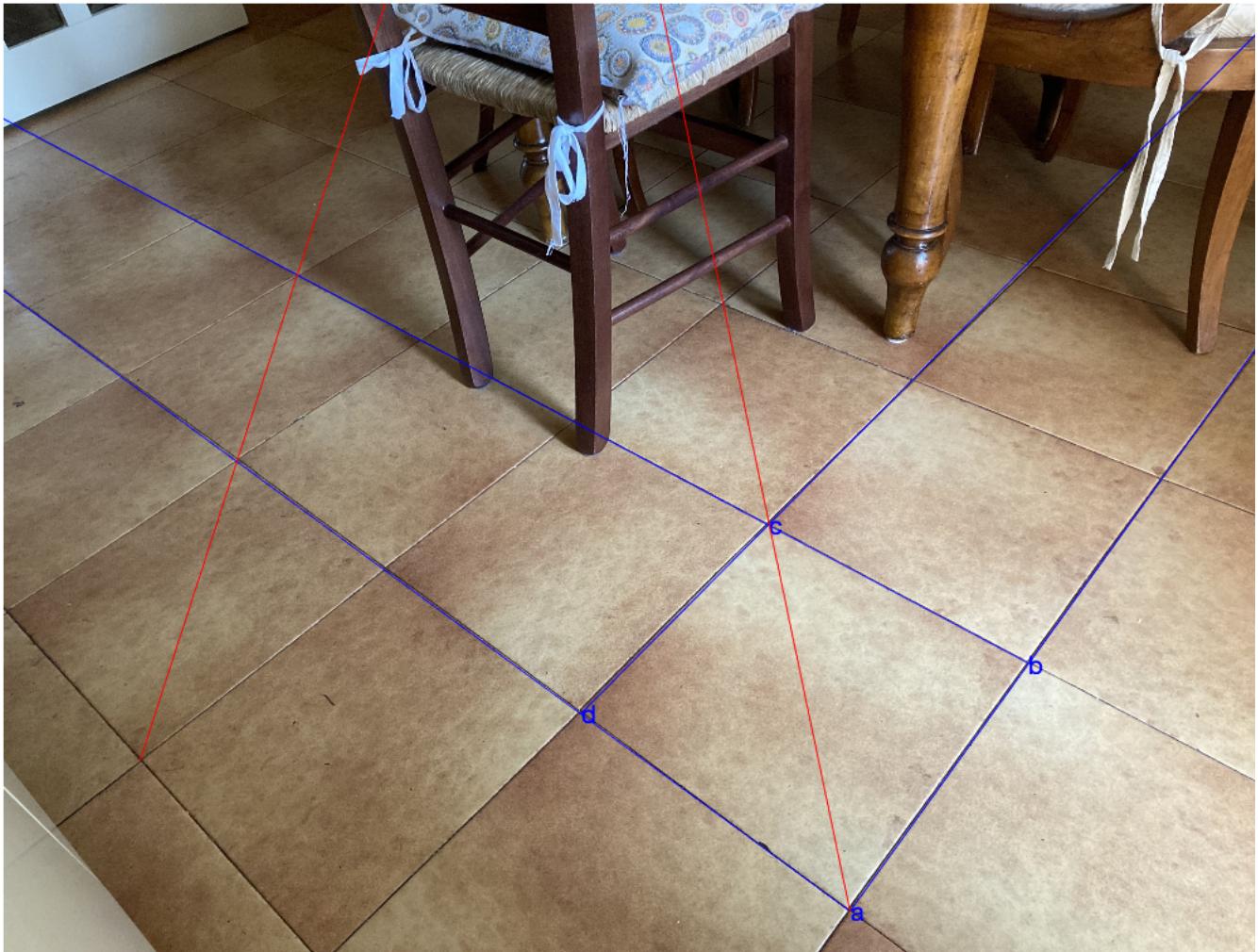
lef = cross(e, f);
lgh = cross(g, h);

v3 = cross(lef, lgh);
v3 = v3 / v3(3);

horizon' * v3
% it is ok to have an higher error, we have fitted the vanishing points
% from noisy estimates, moreover we are not taking into consideration
% radial distortion etc... we are not dealing with a perfect pinhole
% camera...
plot([e(1), v3(1)], [e(2), v3(2)], 'r');
plot([g(1), v3(1)], [g(2), v3(2)], 'r');
text(v3(1), v3(2), 'v3', 'FontSize', FNT_SZ, 'Color', 'b');

% what happens if the camera is not perspective but orthographic?

ans =
-3.0317e+05
```



### Given the horizon we can rectify the image

```

horizon = horizon./norm(horizon); % super important to regularize the homography
H = [eye(2),zeros(2,1); horizon(:)'];
H = det(H)*H;
% we can check that H^-T* imLinfty is the line at infinity in its canonical
% form:
fprintf('The vanishing line is mapped to:\n');
disp(inv(H)*horizon);

```

The vanishing line is mapped to:  
0.0000  
0  
-1.0000

### rectify the image and display it

```

t = maketform( 'projective', H');
J = imtransform(im,t);
figure;
imshow(J);
% how are the lines in the rectified images.
% Are the tile of the floor square? Why? Why not?
% What happen to the points that doesn't lie on ground plane?

```



Published with MATLAB® R2021b