Image Analysis and Computer Vision

Homework report

Riccardo Bertoglio

Politecnico di Milano

10-01-2019

Contents

1 Assignment

A. Scene information: The observed scene contains a car. The car chassy is symmetric about a symmetry plane. The symmetry plane is vertical; several pairs of symmetric point features can be identified, such as vertexes of the rear lights or vertexes of the license plate etc. Two wheels can also be observed: in each of the two wheels, the circular border between the tire and the rim can easily be identified on the image; the circular borders of the two wheels have the same diameter, and they lie on a common plane, at a same height above the street level. The plane containing the two circular borders is parallel to the symmetry plane of the car chassy (thus it is also vertical).

B. Image information: an image of a car is taken by a zero-skew camera. Natural camera can not be assumed. Use either Image1 or Image2

Assignment: write and test a Matlab program that analyzes either Image1 or Image2 to extract the information described below

1. **Image feature extraction and selection:** Use the learned techniques to find corner features and ellipses in the image. Then manually select those features and those parts of the detected ellipses, that are useful for the subsequent steps.
2. **Geometry:**
   - 2.1. Using constraints on the wheel circles mentioned on point A, determine the ratio between the diameter of the wheel circles and the wheel-to-wheel distance.
   - 2.2. Using also some of the detected pairs of symmetric features, calibrate the camera by determining the calibration matrix K. Assume the camera is zero-skew (but not natural).
   - 2.3. Fix a reference frame at a suitable position on the symmetry plane of the car, and reconstruct the 3D position of some of the symmetric pairs of features relative to the above reference.
   - 2.4. Localize the camera relative to the car reference.

Hint: use normalized image coordinates to reduce numerical errors (e.g., set image size to 1)

Figure 1: Image used in the program.

2 Image features extraction and selection

2.1 Keypoints detection

Harris detection To detect keypoints I used the Harris–Stephens algorithm in the Computer Vision Toolbox of Matlab. I selected the parameters after several experiments. The most important thing was to set the 'ROI' (Region Of Interest) parameter to select a smaller area than the entire image. In this way I could obtain the features I was interested in because they were unique only in small region of the image. As hardcoded points (Figure 2) I used:

- 2 upper vertexes of the license plate
- 2 lower vertexes of the license plate

2.2 Ellipses detection

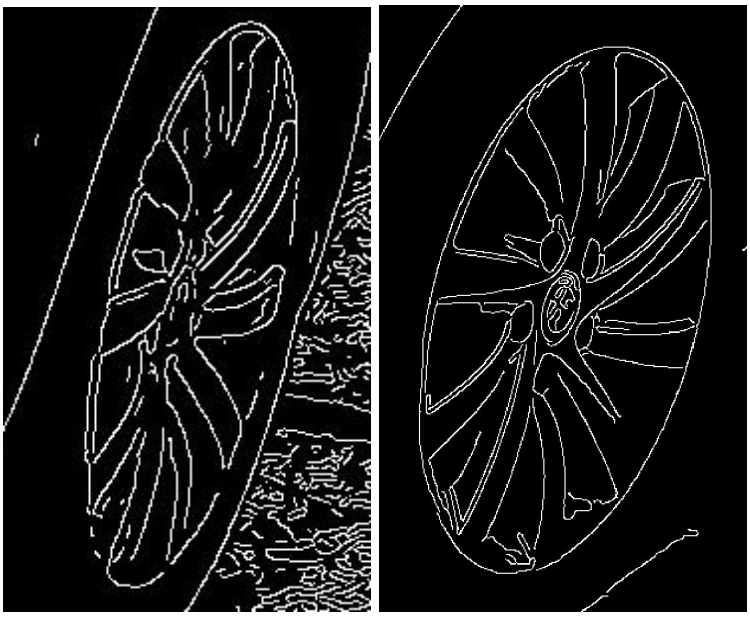
Ellipses detection To find the ellipses corresponding to the car wheels I used the MATLAB script "ellipseDetection" (Author: Martin Simonovsky; e-mail: mys007@seznam.cz; Release: 1.1; Release date: 25.7.2013) based on the

Figure 2: Harris keypoints of a particular ROI and the four hardcoded points

paper "A New Efficient Ellipse Detection Method"[2].
The first step is to crop the image because the above algorithm is memory intensive. The elaboration on the entire image requires too much memory. The algorithm is optimized to work with edge images and it uses a sparse representation throwing away the pixel equal to zero. The edges image is built by using Canny edge detection (Figure 3) with a proper setting of min and max thresholds. Then I set some parameters of the algorithm "ellipseDetection" such as the minimum and maximum length of the major axis of the ellipse to speed up the computation. I turn off the randomization to obtain better results and I indicate to return only the best ellipse found. The algorithm return a matrix of best fits (in my case with just one element). Each row (there are params.numBest of them) contains six elements: [x0 y0 a b alpha score] being the center of the ellipse, its major and minor axis, its angle in degrees and score. To obtain the right conic matrix I had to calculate a rototranslation matrix H to apply to the conic matrix C (Figure 4) .

Lines tangent to the ellipses and horizontal vanishing point I found the four bitangent lines to the ellipses:

- 2 lines that are parallel in the real scene
- 2 lines that cross each other



(a) Front wheel. (b) Rear wheel.

Figure 3: Canny edges images of the cropped front and rear wheels.

Figure 4: Ellipses detected (in red).

In order to find the four lines l_1, l_2, l_3, l_4 that are bitangent to two conics $\mathbf{C}_1$, $\mathbf{C}_2$ I impose that these lines are tangent to both the conics.

$$\begin{cases} \mathbf{x}^\intercal \mathbf{C}_1 \mathbf{x} = \mathbf{0} \\ \mathbf{l} = \mathbf{C}_1 \mathbf{x} \\ \mathbf{y}^\intercal \mathbf{C}_2 \mathbf{y} = \mathbf{0} \\ \mathbf{l} = \mathbf{C}_2 \mathbf{y} \end{cases} \tag{1}$$

After some algebra we have:

$$\begin{cases} \mathbf{l}^\intercal \mathbf{C}_1^{-1} \mathbf{l} = \mathbf{0} \\ \mathbf{l}^\intercal \mathbf{C}_2^{-1} \mathbf{l} = \mathbf{0} \end{cases} \tag{2}$$

Solving system (2) I obtain the four lines.
The intersection of the lines parallel in the real scene is the horizontal vanishing point $\mathbf{vp}_1$ (Figure 5). For robustness the horizontal vanishing point is fitted with a third line, that is, the line passing through the image of the centers of the circumferences (see next paragraph).

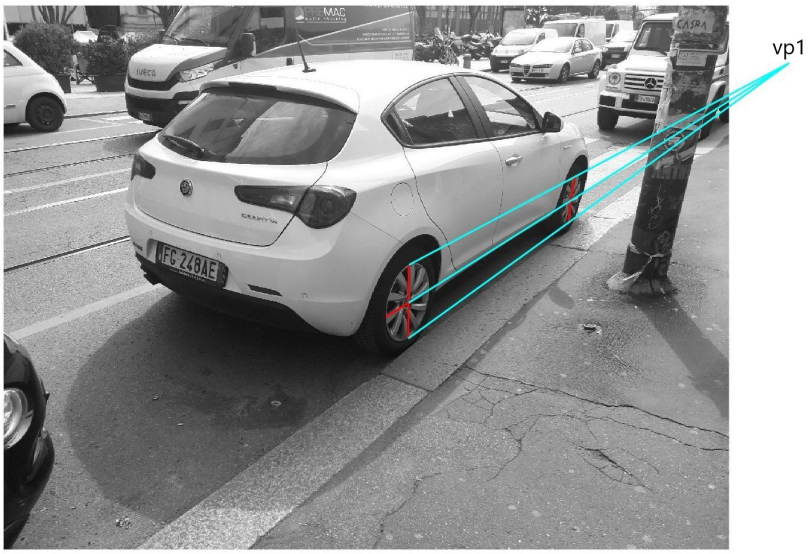


Figure 5: Horizontal vanishing point $\mathbf{vp}_1$.

Polar lines and other vanishing points The tangency points of the two bitangent lines that are parallel in the real scene gave me two points for each ellipse. These two points identify a line passing trough the diameter of the circumference (red lines in Figure 6). These two lines are parallel in the real

scene so their intersection constitute the vertical vanishing point $\mathbf{vp}_2$. Then, I want to find the images of the centers of the circumferences. The center of a circumference is identified by two perpendicular polar lines passing through the diameter. I used the vertical polar line previously found and an horizontal one (blue lines in Figure 6). The horizontal polar line is identified by two tangency points. These tangency points are identified by the intersection of the set of lines forming the dual conic and the lines passing through the vertical vanishing point. The two tangent lines l_5 and l_6 are found solving the following system:

$$\begin{cases} \mathbf{l} = \mathbf{C}_1\mathbf{x} \\ \mathbf{l}^\intercal \mathbf{vp}_2 = \mathbf{0} \\ \mathbf{x}^\intercal \mathbf{C}_1 \mathbf{x} = \mathbf{0} \end{cases} \tag{3}$$

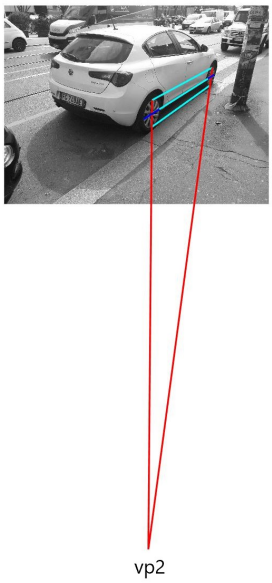


Figure 6: Vertical vanishing point $\mathbf{vp}_2$. red: vertical polar lines. blue: horizontal polar lines.

3 Wheels diameter and wheel-to-wheel distance ratio

To find the requested ratio I used a cross-ratio approach. Due to the fact that the cross-ratio is preserved through homographies I can exploit this property to find the requested ratio of the real lengths. I used 4 points on the line passing through the image of the centers of the circumferences. These 4 points are (Figure 7):

- the images of the 2 centers (b and c in figure)
- the most left point identified by the intersection of the line l_p and the ellipse (a in figure)
- the most right point identified by the intersection of the line l_p and the ellipse (d in figure)

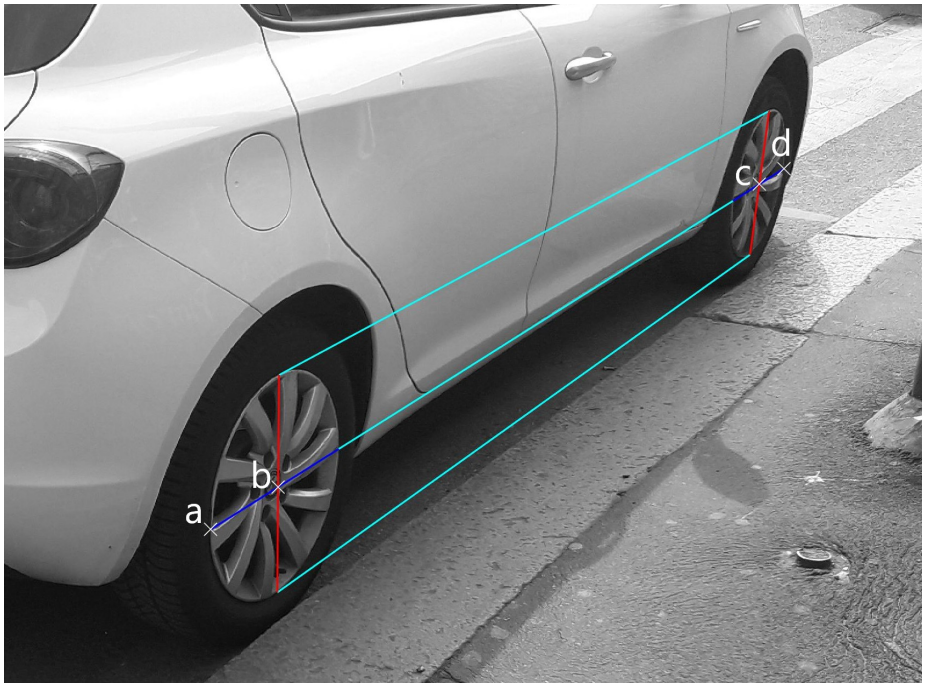


Figure 7: Four collinear points on the horizontal polar line.

I can write the expression of the cross-ratio of the real scene in this way:

$$\frac{\frac{A-B}{A-C}}{\frac{D-B}{D-C}}$$

being A, B, C, D the respective 3D points of the image points a, b, c, d.

Now, I have that

$$A - B = r$$

and

$$B - C = w$$

where r is the radius of the two circumferences and w is the wheelbase, that is, the distance of the centers of the circumferences. So I can rewrite the cross-ratio as:

$$\frac{\frac{r}{r+w}}{\frac{r+w}{r}} = \frac{r^2}{(r+w)^2} = k$$

where k is the cross-ratio calculated with the image points a, b, c, d.
So, doing some algebra:

$$r^2 = (r+w)^2 k$$

$$1 = (1 + \Big(\frac{w}{r}\Big)^2 + 2\Big(\frac{w}{r}\Big))k$$

if I put $r/w = R$ then

$$1 = (1 + (1/R)^2 + 2(1/R))k$$

R is the requested ratio and I have to solve a quadratic equation in one variable.

4 Camera calibration (intrinsic)

The intrinsic camera calibration consists in finding the matrix $\mathbf{K}$ that describes some physical properties of the camera. The intrinsic parameters represent the optical center and focal length of the camera. The camera coordinates are mapped into the image plane using them. To find the matrix $\mathbf{K}$ I firstly find the image of the absolute conic ω because they are related by the following equation:

$$\omega = (\mathbf{K}\mathbf{K}^\intercal)^{-1}$$

So, assuming a zero-skew camera, if $\mathbf{K}$ has the following structure:

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

Then, ω can be expressed in this way:

$$\omega = \begin{bmatrix} \alpha^2 & 0 & -u_0\alpha^2 \\ 0 & 1 & -v_0 \\ -u_0\alpha^2 & -v_0 & f_y^2 + \alpha^2 u_0^2 + v_0^2 \end{bmatrix}$$

To find the image of the absolute conic ω I need at least 4 constraints. In general ω is an homogeneous vector of 6 elements so with 5 DOF. Assuming a zero-skew camera reduces the DOF to 4.
To increase robustness I solve an overdetermined system with the following equations (constraints on ω):

- 3 orthogonal vanishing points: the two vanishing point previously found $\mathbf{vp}_1$ and $\mathbf{vp}_2$ plus a third orthogonal vanishing point $\mathbf{vp}_3$ that comes from the intersection of the images of two lines from the posterior of the car that are parallel in the real scene. These two lines are identified by the 4

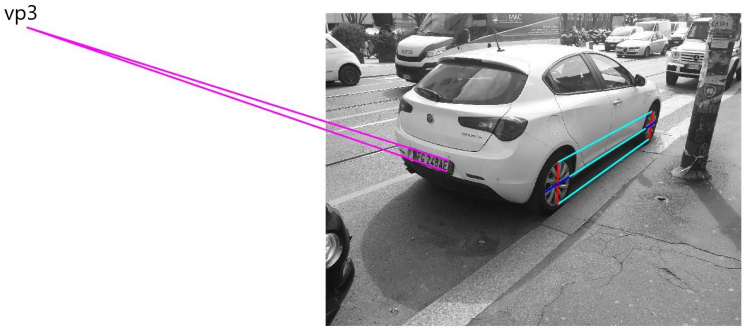


Figure 8: Vanishing point $\mathbf{vp}_3$ from the four detected keypoints.

keypoints detected at the beginning by the Harris algorithm, that is, the 4 vertexes of the license plate (Figure 8).

These equations gives a linear constraint each and are of the following form:

$$\mathbf{vp}_a^\intercal \, \omega \, \mathbf{vp}_b = 0$$

where $\mathbf{vp}_a$ and $\mathbf{vp}_b$ are vanishing point corresponding to orthogonal lines.

- image of the circular points: the images of the circular points give two linear equations providing two constraints. As the circular points are part of the IAC (image of the absolute conic) we impose:

$$\mathbf{I}'^\intercal \omega \, \mathbf{I}' = 0$$

since $\mathbf{I}' = h_1 + jh_2$ we can put the real and imaginary parts equal to zero obtaining two linear equations

$$\begin{aligned} h_1^\intercal \omega h_2 &= 0 \\ h_1^\intercal \omega h_1 &= h_2^\intercal \omega h_2 \end{aligned}$$

where h_1 and h_2 are, respectively, the real part and the imaginary part of the imaged circular point $\mathbf{I}'$.

So, the final system of equations is:

$$\begin{cases} \mathbf{vp}_1^\intercal \, \omega \, \mathbf{vp}_2 = 0 \\ \mathbf{vp}_1^\intercal \, \omega \, \mathbf{vp}_3 = 0 \\ \mathbf{vp}_2^\intercal \, \omega \, \mathbf{vp}_3 = 0 \\ h_1^\intercal \omega h_2 = 0 \\ h_1^\intercal \omega h_1 = h_2^\intercal \omega h_2 \end{cases}$$

5 3D position of symmetric features

In this section I explain the 3D reconstruction of symmetric points, such as the vertexes of the license plate, with respect to a reference frame placed on the car symmetry plane. In order to do this I need to build a rototranslation matrix that maps points in the camera reference frame into this new one reference frame. Therefore, I firstly compute the rotation matrix R_{cw} that maps points of the world in the camera reference frame.
A 3D point $\mathbf{x}_w$ of the world maps to an image point $\mathbf{u}$ in the following way:

$$\mathbf{u} = \mathbf{P}\mathbf{x}_w = \begin{bmatrix} KR_{\text{cw}} & -KR_{\text{cw}}t_{\text{cw}} \end{bmatrix} \mathbf{x}_w$$

where $\mathbf{P}$ is called projection matrix. In order to compute this matrix I fix the world reference frame orientated with the x axis parallel to the rear vanishing point $\mathbf{vp}_3$ and the z axis parallel to the vertical vanishing point $\mathbf{vp}_2$. The y axis is just the cross-product of the other two such that I have a right-handed reference system. First I obtain the viewing ray of the vanishing points $\mathbf{vp}_3$, $\mathbf{vp}_2$ multiplying by the inverse of K and then I normalize them to obtain a versor.
So, if

$$R_{\text{cw}} = \begin{bmatrix} R^1_{\text{cw}} & R^2_{\text{cw}} & R^3_{\text{cw}} \end{bmatrix}$$

Then

$$R^1_{\text{cw}} = K^{-1}\mathbf{vp}_3$$
$$R^2_{\text{cw}} = K^{-1}\mathbf{vp}_2$$
$$R^3_{\text{cw}} = cross(R^2_{\text{cw}}, R^1_{\text{cw}})$$

The second step is to obtain the translation vector t_{cw}. Since we want to place the world reference system on the symmetry plane of the car, the translation vector is equal to the middle point of two symmetric features of the car posterior. The idea is to use the backprojection rays of the two lower vertex of the license plate and find two points on them such that they have same coordinate y, z and with the x one the opposite of the other.
We call these two points dl and dr because they are, respectively, the down left vertex of the license plate and the down right one. I find the backprojections dl_{bp}, dr_{bp} in the following way:

$$dl_{bp} = (KR_{\text{cw}})^{-1}dl$$

$$dr_{bp} = (KR_{\text{cw}})^{-1}dr$$

I obtain the directions vectors normalizing them:

$$v_1 = dl_{bp}/norm(dl_{bp})$$

$$v_2 = dr_{bp}/norm(dr_{bp})$$

And I solve the system:

$$\begin{cases} k/2 = \lambda_1 v_{11} \\ y = \lambda_1 v_{21} \\ z = \lambda_1 v_{31} \\ -k/2 = \lambda_2 v_{12} \\ y = \lambda_2 v_{22} \\ z = \lambda_2 v_{32} \end{cases}$$

where λ_j is the length of the vector j, v_{ij} is the i-th component of the vector j and k is a fixed arbitrary distance between the two symmetric features in the real scene.
So, the points in the camera reference frame are (see Figure 9 and 10):

$$mid = \begin{bmatrix} 0 \\ \overline{y} \\ \overline{z} \end{bmatrix} = t_{\text{cw}}$$

$$dl = \begin{bmatrix} k/2 \\ \overline{y} \\ \overline{z} \end{bmatrix}$$

$$dr = \begin{bmatrix} -k/2 \\ \overline{y} \\ \overline{z} \end{bmatrix}$$

where $\overline{y}$, $\overline{z}$ are the variables solved with the above system.
In the world reference frame dl and dr are simply:

$$dl = \begin{bmatrix} k/2 \\ 0 \\ 0 \end{bmatrix}$$

$$dr = \begin{bmatrix} -k/2 \\ 0 \\ 0 \end{bmatrix}$$

6 Camera localization

Since we have the rototranslation matrix with respect to the car reference frame the camera is localized (see Figure 9 and 10).

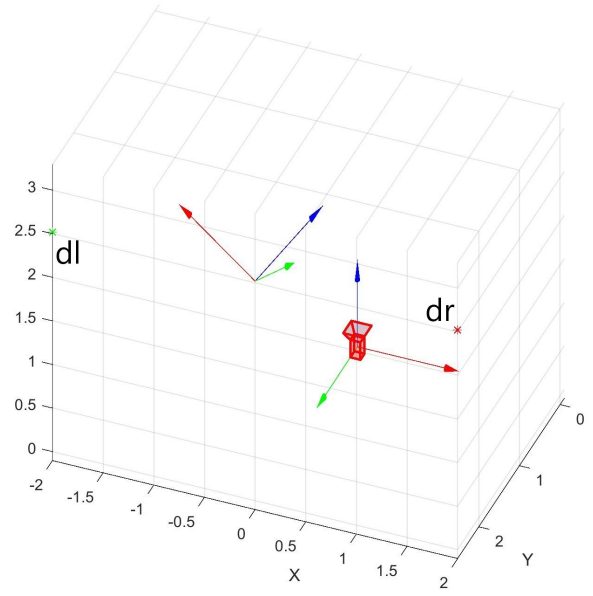


Figure 9: Camera reference, world reference and symmetric points.

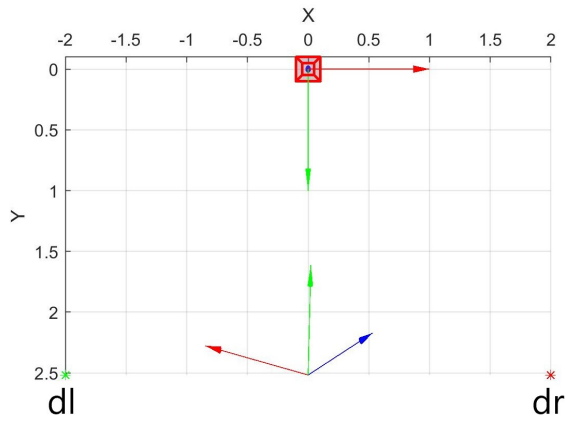


Figure 10: Camera reference, world reference and symmetric points.

7 Computed results

Vanishing points

- Horizontal vanishing point $\mathbf{vp}_1$

```
fh_inf = [
4.470753873929544e+03;
3.278916849471632e+02;
1]
```

- Vertical vanishing point $\mathbf{vp}_2$

```
v_inf = [
2.456179181221011e+02;
9.259322238001098e+02;
0.109162323713662]
```

- Rear vanishing point $\mathbf{vp}_3$

```
h_rear_inf = [
-7.646927160000041e+06;
3.614047999999821e+05;
2.575899999999998e+03]
```

Wheels plane reconstruction

- Cross-ratio

```
CR = 0.005990695611017
```

- Wheels diameter/wheelbase ratio

```
diam_wheelbase_ratio = 0.167785712451972
```

Calibration matrix

- Image of circular points

```
I_im = [
3.964295209636065e+03 + 9.317763588461681e+02i;
2.187550864696845e+03 - 3.421377856412592e+03i;
1.000000000000000 + 0.000000000000000i]

J_im = [
3.964295209636065e+03 - 9.317763588461681e+02i;
2.187550864696845e+03 + 3.421377856412592e+03i;
1.000000000000000 + 0.000000000000000i]
```

- Image of the absolute conic

$$w = \begin{bmatrix} 1.675958318861366 & 0 & -4.904925197685752e+03 \\ 0 & 1 & -2.831084878983815e+03 \\ -4.904925197685752e+03 & -2.831084878983815e+03 & 3.089067399992709e+07 \end{bmatrix}$$

- Calibration matrix K

$$K = \begin{bmatrix} 3.007369252025376e+03 & 0 & 2.407103220352066e+03 \\ 0 & 3.131540655252108e+03 & 1.727722871681583e+03 \\ 0 & 0 & 1] \end{bmatrix}$$

3D symmetric points reconstruction

- Rotation matrix R_{cw}

$$R_{cw} = \begin{bmatrix} -0.887150557909817 & 0.014686839500633 & 0.461246337691944 \\ -0.312780550543189 & -0.754031538073834 & -0.577585289744690 \\ 0.339311383248780 & -0.656673995476821 & 0.673533257557719 \end{bmatrix}$$

- Middle point and symmetric features

```
pt_rear_mid = [0;2.259803578784244;2.259932673356913]

pt_rear_dl = [-2;2.259803578784244;2.259932673356913]

pt_rear_dr = [2;2.259803578784244;2.259932673356913]
```

8 How to use the program

Carefully follow these instructions:

- Start Matlab by clicking on the 'startup.m' file in the main folder so all the environment variables are properly set
- If you want to use the interactive version of the program you have to set to 'true' the global variable 'intercative' at the beginning of the program. In the 'hardcoded' version you won't see the drawings of the ellipses and the Canny edges
- Open the 'src' folder and launch the 'main.m' file

Only for interactive mode:

- In the Harris keypoints window zoom the image with the lens tool so you can better select the points
- Deselect the lens tool and choose 4 keypoints
- The 4 keypoint must be chose starting from the upper left one going clockwise, so the order should be: upper left, upper right, down right, down left
- For the last point make a double click or press 'Enter' after the last point
- Then it will appear a gray scale image and you have to crop a first rectangle around the posterior wheel. Beware to draw the rectangle as small as possible to reduce the computation time.
- The algorithm will find the posterior ellipse and then you can draw a rectangle around the anterior wheel to find the second ellipse

References

[1] John Canny; *A Computational Approach to Edge Detection*; IEEE Transactions on Pattern Analysis and Machine Intelligence, Nov. 1986.

[2] Yonghong Xie Qiang, Qiang Ji; *A New Efficient Ellipse Detection Method*; Proceedings 16th International Conference on Pattern Recognition, 2002.