Ellipses Report

Ellipse seeking and drawing

Finding an ellipse, despite being a well configured task, consists in a few complicated and intricate smaller labors. Part of are curvature, filtering, finding edges as well as masks, different algorithms and using some mathematical shape-oriented operations.

Our algorithm offers a pre-processing pipeline for calculating the image edge map, by emphasizing the ellipse curvature and edge points. As in addition calculates the suspected ellipses point's tangent and its slope, calculates the TM lines in a smart manner and calculates the ellipse hypothesis center by encountering some of the points, and pairing them, in a controllable manner.

Here we specify the overall process. Our algorithm, as described above, consists of two main stages,

Ellipse Edge and Gradient orientation map:

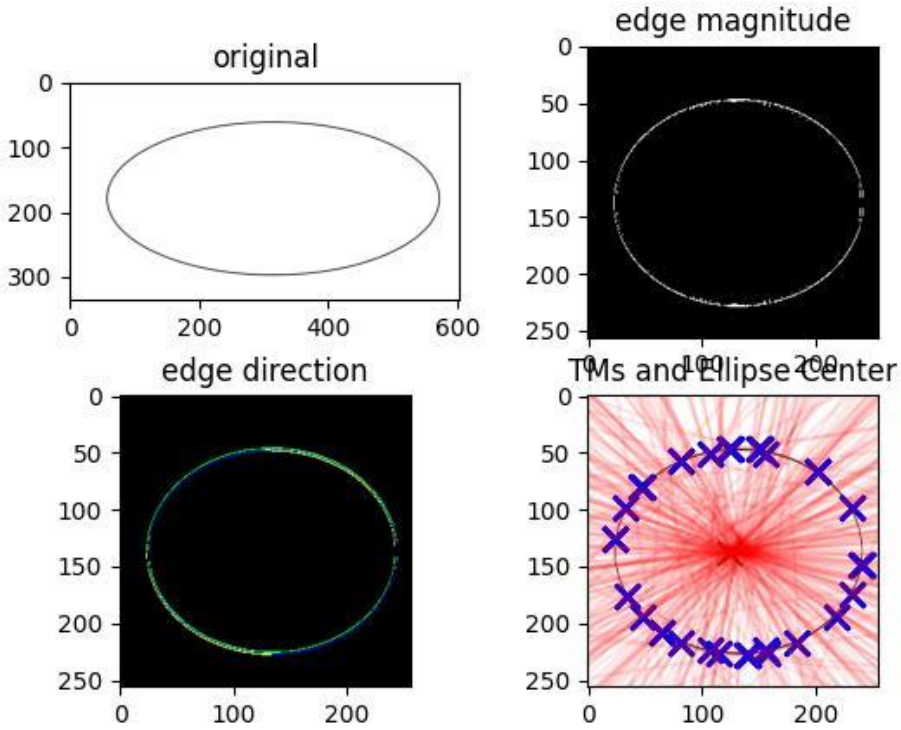
1. Resizing the image to a $(255\ x\ 255)$ grid.
2. Blurring the image by a manual maid Gaussian alike mask.
3. Edge map of the image (a Sobel edge map).
4. Gradient orientation map.
5. Second phase filtering – ellipse curvature seeking and thinning (ellipse outside edge points maintaining). Helps for better results as well as for pairing ellipses points with same orientation.
6. Third phase filtering – Canny thresholding. Helps keeping continuity of edge lines while neglecting other contributions of edge points.

Center calculation:

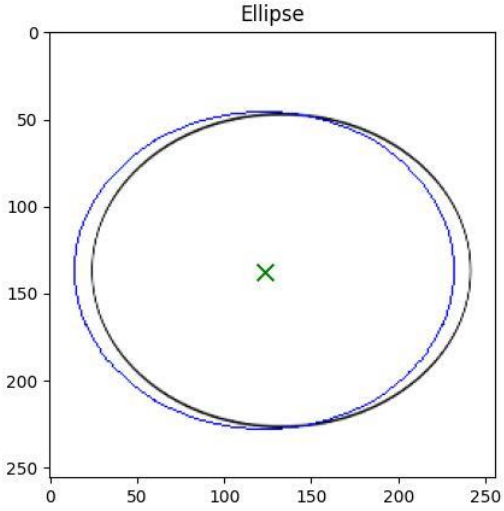
1. Neglecting Euclidean close points (for better calculation accuracy).
2. Choosing pairs of points in a controllable manner, such that opposite points will be picked in a high probability.
3. Calculating tangent radians, slops and from there calculated the TM lines for each pair of points according to the research paper.
4. Calculate the center by a voting algorithm (and according to a grid division).
5. Calculating the ellipse approximate shape and radiuses according to the points which contributes to the centers which last been encountered by the previous algorithm.

As an example and illustration for our algorithm, we present the following long self-escorted test images and its results,

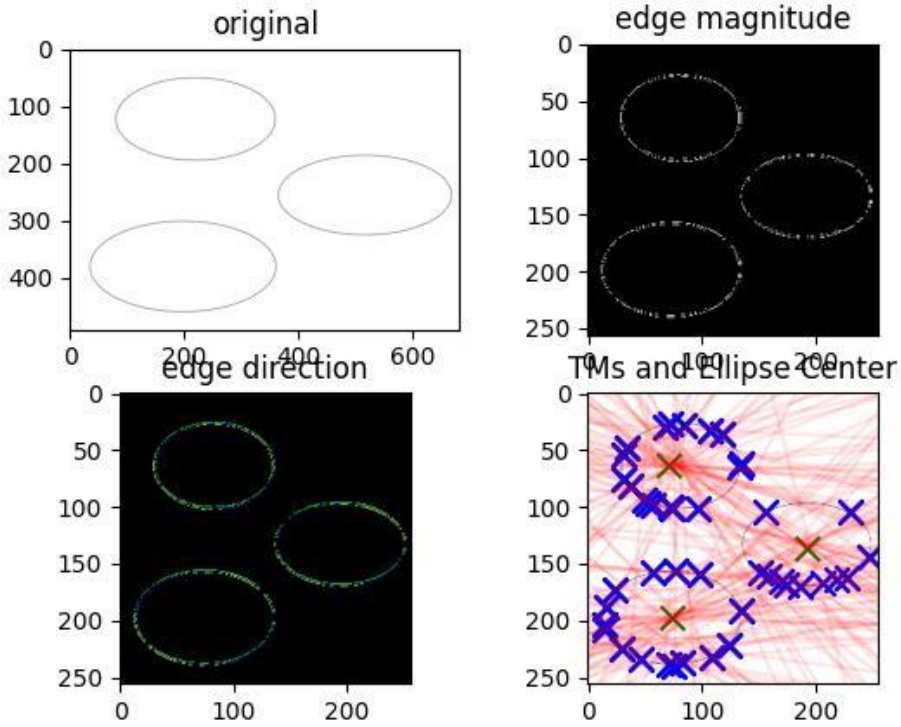
One Ellipse:

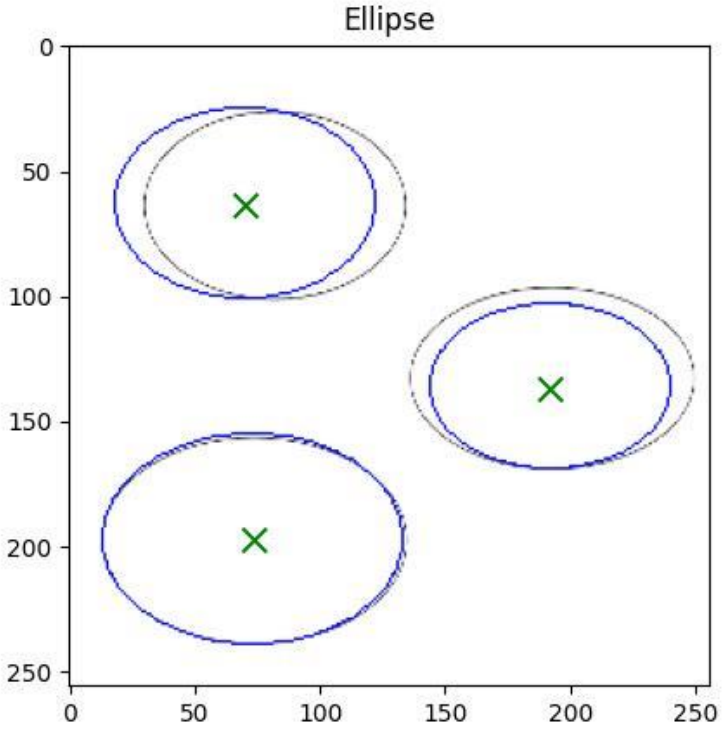


Here we present the original shaped image, its edge map, gradient orientation, and all the ellipse edge points which were picked by the algorithm, its contribution, and the center calculation.



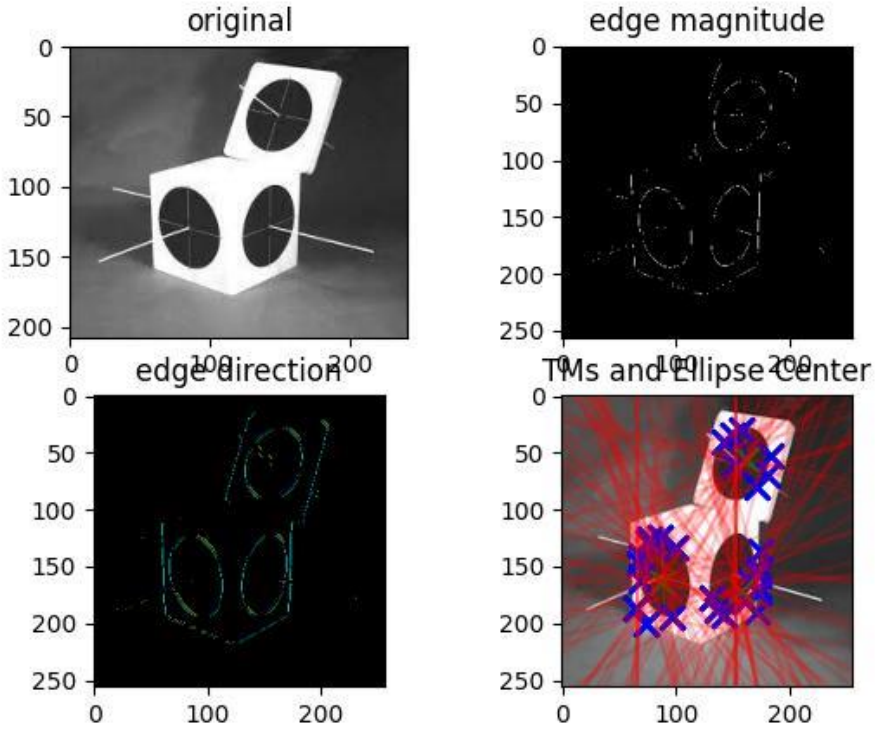
Finally, we present the ellipse drawing according to its center and encountered contributing point pairs.
Another, more generic example,

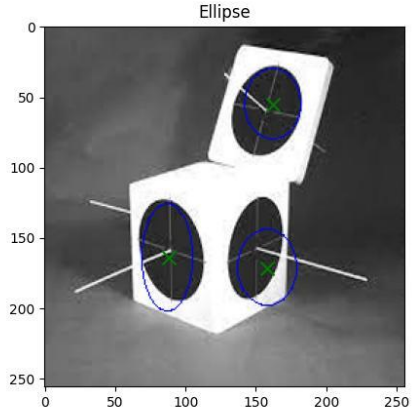




We can notice the good and relatively accurate results here (though it’s a somehow object-background separated general image).
For a more complex challenge, we picked up a few different images. We'll discuss each separately.

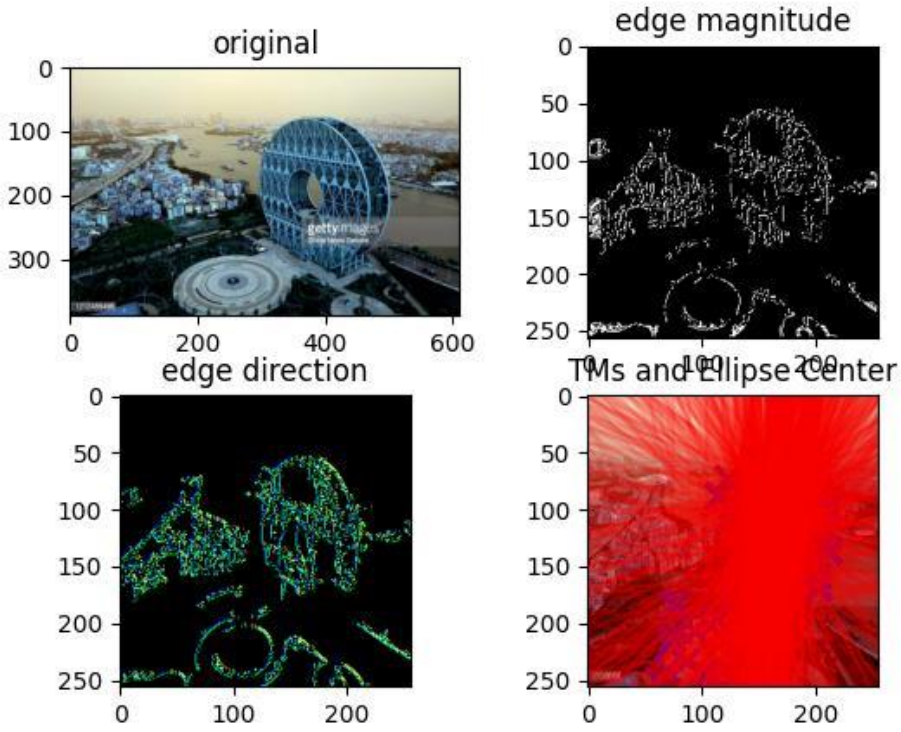
Box Image:

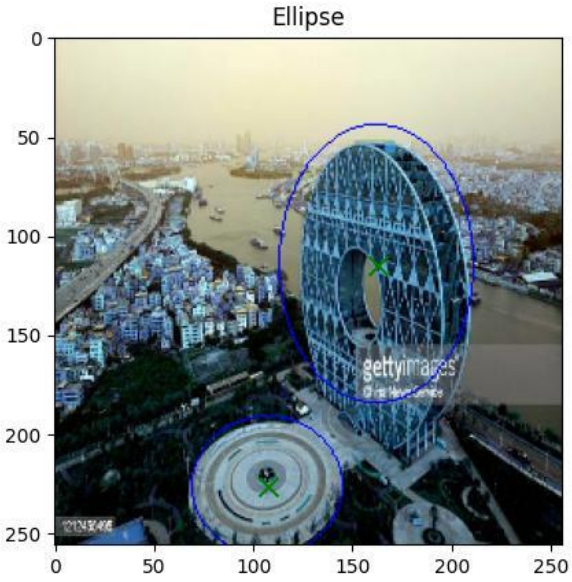




We can see that the algorithm managed to find closely approximate ellipses within the image's given object. Can also notice false positive ellipse centers. We estimate that the previous is due to the small ellipses Euclidean distance relatively to the box's edges. This contribute to the false, and biased, position of the ellipses center. Thus, also contributing to the ellipses final drawl shape.
For handling this problem, we offer to try and fine tune some of the filter parameters, in the pre-processing phase of the algorithm. Thus leading to disappearing of some of the edges boundaries and emphasizing others of the object ones.

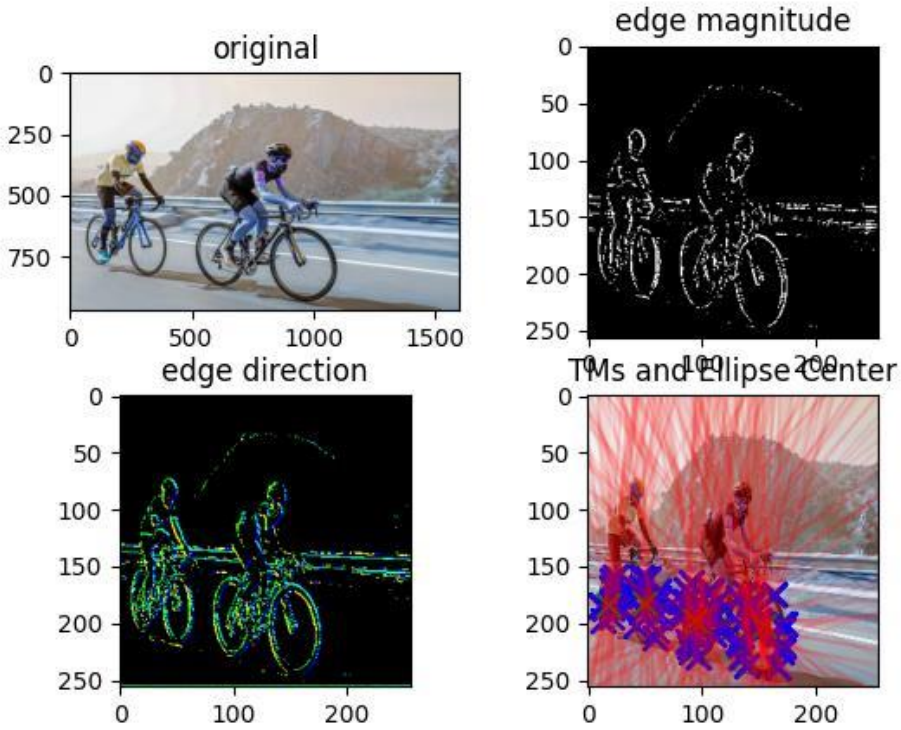
Round About:

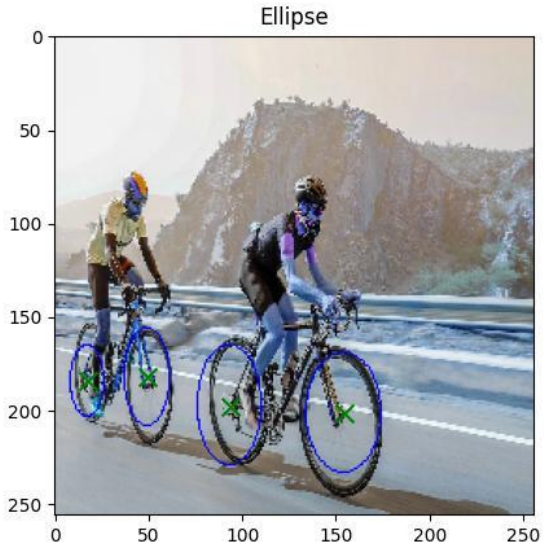




We can notice that we could find the main ellipses of the object round about. Regarding the ellipse un-accuracy, this is due to the amount of edges created by the vast smaller bigger wheel edges after the filtering process (it can also be seen in the edge map above). This could be resolved by a fine tuning of the parameters in the blurring phase before the edge map (close objects could be spread across the region and hence not contributing to our center).

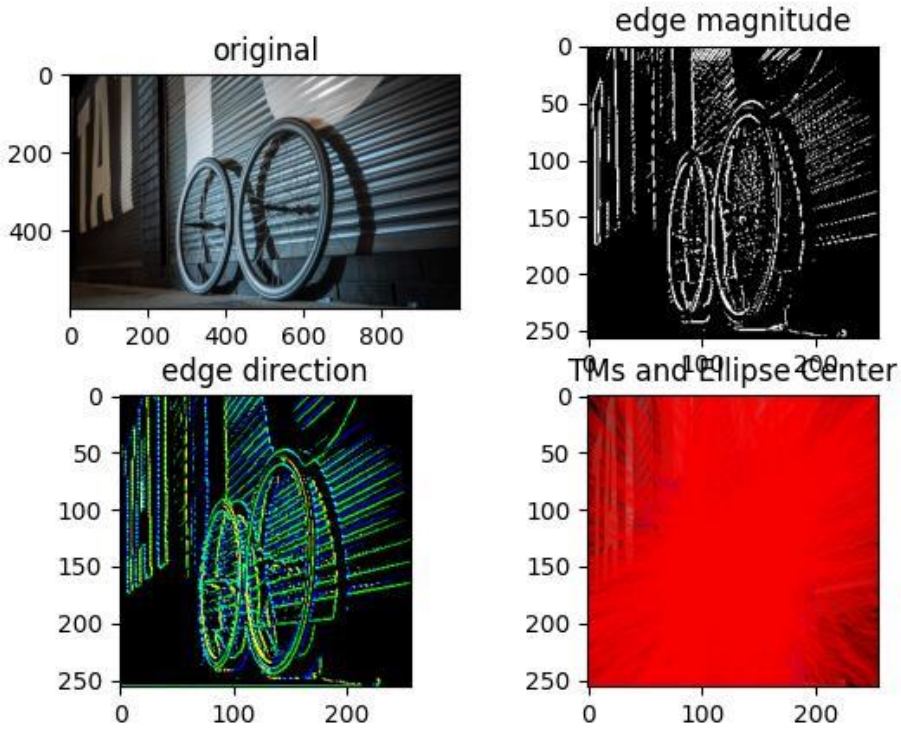
Bicycles:

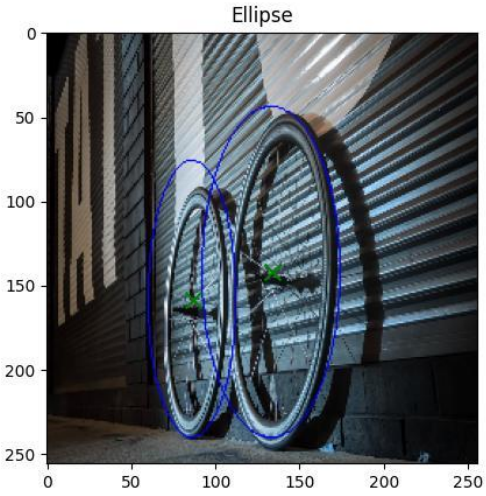




We can see that the results are pretty good regarding finding the image ellipses. In a contract side of manner, we can see, via the edge map magnitude, that after filtering the image we still, unwillingly, maintained some amount of the men edges and also some of the bicycle parts. This where our second phase filtering algorithm comes to an action, in which we seek to find an ellipse-shaped curvature while neglecting other edges which are not.
Hence, maintaining the main mass contributions for the ellipses edges points.

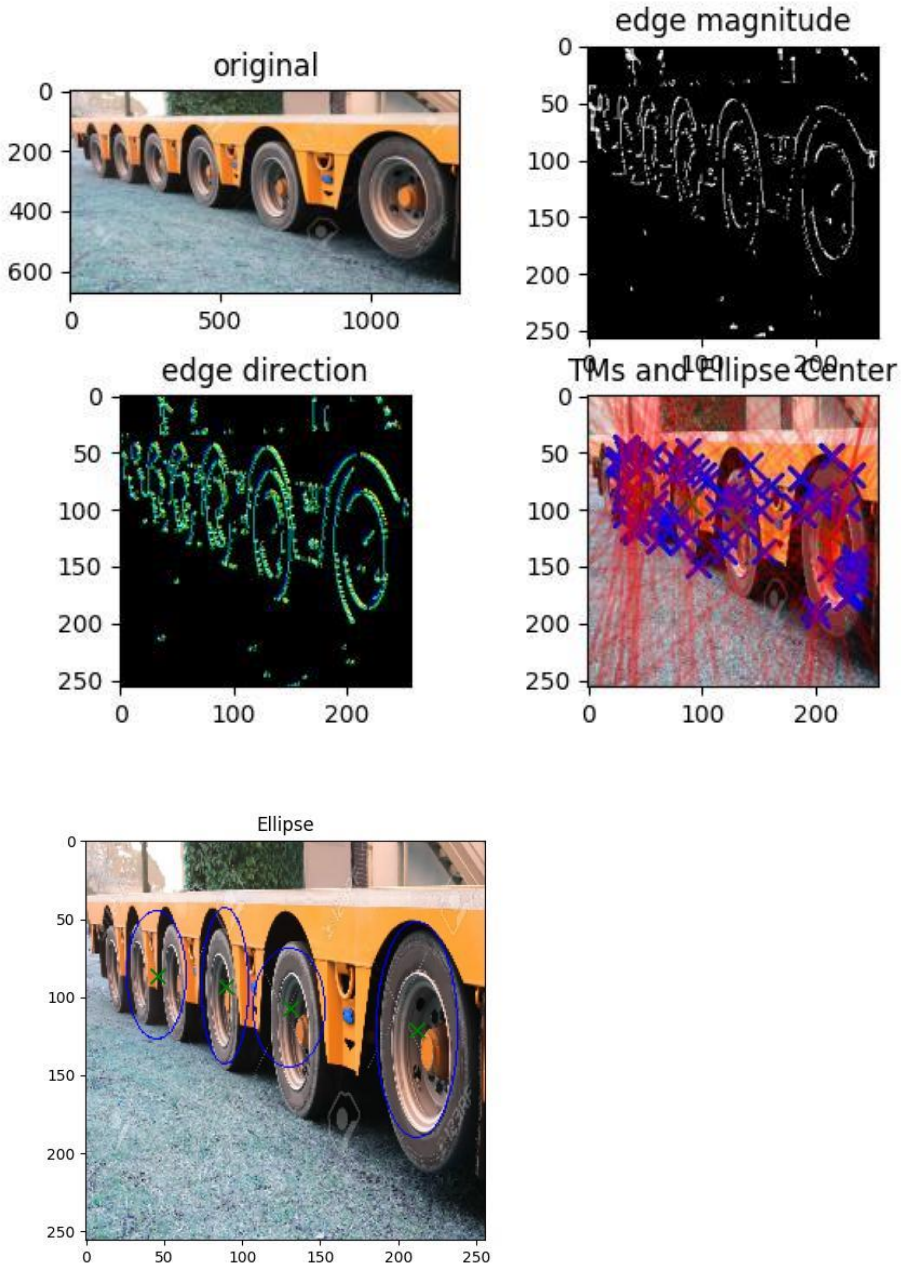
Two Tires:





We can notice that the background in this image differ from others a bit. We did struggle a bit with this at first, but tried to encounter a large amount of points that might contribute to the ellipse center voting (can be seen via the TM and Ellipse Center image above), and also fine tuning in the pre-processing phase. Leading to a vast amount of contribution points, but yet one center mass TM lines which intersect approximately within the ellipse centers.

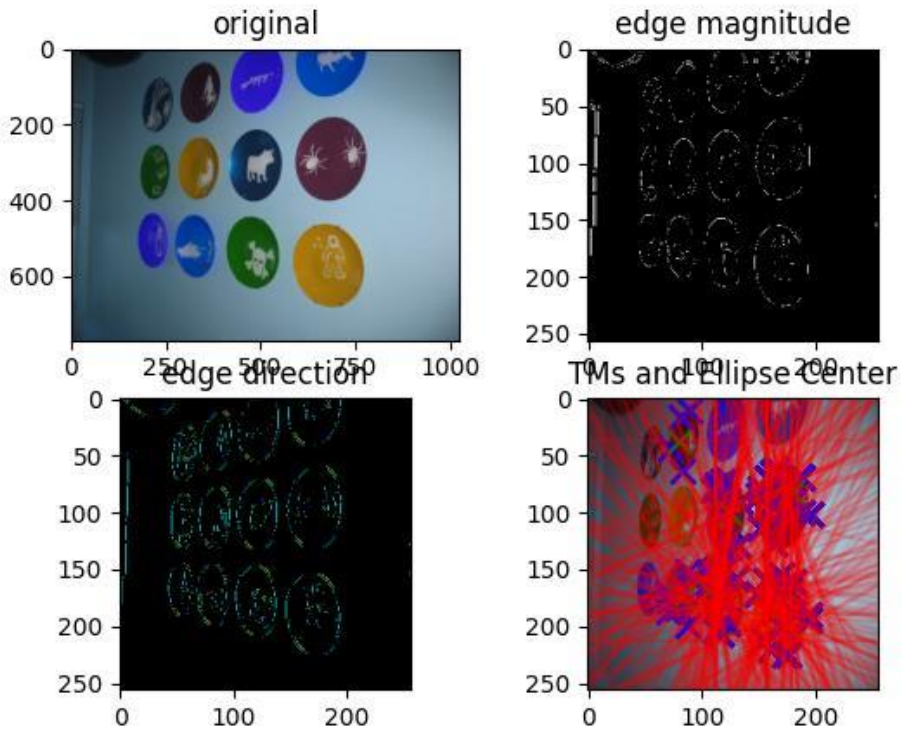
Long Truck:

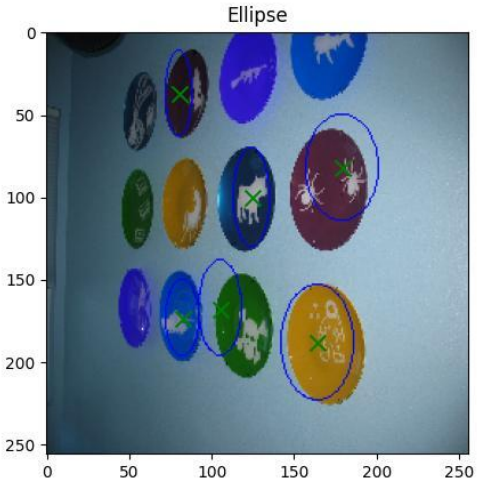


This one is an interesting one. We can see that our algorithm missed the tire edges (In the TM image above and also in the final result), this is due to the high contrast between the tire and the truck body. emphasizing false edges and lines in that region. We can notice that our second phase algorithm (ellipse curvature) also had some hard time recognizing those exact areas due to the ellipse shape curvature of this false region.

Other than being extremely accurate, we managed to find approximately the ellipse centers and draw its shape with some good accuracy. Regarding the last tire of the truck, the resolution was too small for recognizing it separately, and when tried to make a smaller grid contribution it affected the rest of the image (according to what've mentioned above).

Wall Plates:





Here we managed to find and draw some of the ellipses, yet with some, a couple of things to notice. The plates are close to each other in such a manner that their edges contributes to each other's center voting, and the plates (ellipses) inner textures still appeared in the edge map which also contributed to the center biasing problem. We could've had picked a smaller grid division to encounter better voting, but it required extra caution for maintaining the current results.