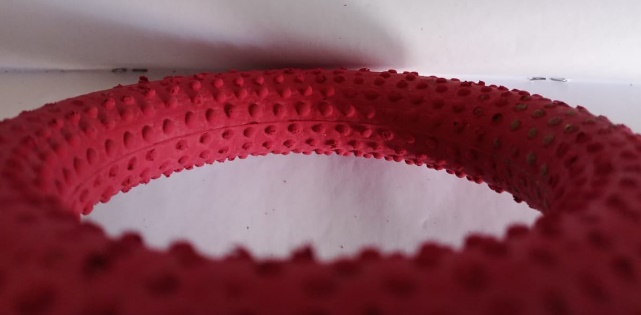
**Problem Statement:**

A system for finding surface defects in O-Rings.

**Image acquisition and hardware setup:**

I believe one of the biggest concerns in this problem is the lighting and reflections. As we have the control on the environment, it’s possible to adjust to just right amount of lighting to get a perfect exposure such that all the defects would be conspicuous and also there shouldn’t be too much reflection.

If all the possible surface defects are to detected, I think at least six views are to needed to cover the entirety of the surface area irrespective of the shape of the o-ring as long as it’s a ring.



Each of these views have an opposite side/view and thus these six views cover some portion that is unique to that view and can’t be visible from a different view.

The first view is the easiest to obtain but also the most significant one as the ring diameter and thickness can be obtained accurately from it.

For the first view, the camera has to be setup in such a way that it is perpendicular to the surface on which ring rests and placed at a perfect distance such that the field of view is limited as long as it is able to capture the biggest o-ring under test completely. The projection onto the image plane should be as orthogonal as possible. And the background or the surface on which the ring rests should be chosen to be completely plain without any marks and as less reflective as possible.

Reasons for projection to be orthogonal:

* It results in ring being projected as circles and not ellipses. Fitting a circle ( three parameters) is more time and space efficient than fitting an ellipse ( four parameters)
* It results in symmetric image and the reflection will be uniform
* Determining diameter and thickness would be easier in case of circles.

The next two views are only for defect detection. I can’t say much about the hardware setup for them.

For these views, maybe an arrangement be made such that the ring gets rotated/flipped by an angle of 180 degrees and the two views captured in-between. Then repeat the process once more to get the opposite of those views.

Another idea might be that the robotic arm picks up the ring at one end and places it onto a holder which has been adjusted to hold upright a ring of specific width and then capture the second view. For the third view maybe the ring be tilted a little.

But acquiring these two views doesn’t seem much feasible in the target time. Then we would have to just do with the first view and it’s opposite and face ***corner issues*** as inner and outer edges won’t be visible. The rings can be inspected in sequence in a conveyer belt and then flipped to get the opposite view.

We would have to be more careful about lighting. To highlight the edges to some extent, we might want to introduce some backlight and experiment with the proportions of direct light backlight to get the best possible exposure.

Some of the basic factors to be considered while choosing a camera for a problem are:

* Good frame rate because we want to inspect as many rings as we can in sequence.
* Large enough camera sensor to get high resolution images with bigger pixels (because we would
* Aperture, lens for the desired field of view

**Determining thickness and diameter of the ring:**

Once we make sure that the projection onto image plane is orthogonal and the rings are being projected as circle, it is easy to fit circles to both the boundaries of the ring. We may choose to go with Hough circle transform or some other algorithm.

Diameter of ring is simply twice the radius of the inner circle and the difference between the radius of the inner and outer circle. Thus, we can obtain the diameter and thickness in image coordinates in no time.

Next we find the diameter and thickness in world coordinates by mapping the image coordinates to the world coordinates.

Consider the following representation of image formation,

By simple law of similar triangles, we have:

ho / do = hi / f

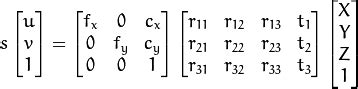
* ho = hi \* do / f

where ho = actual height of object, do = distance of object from the image plane , hi = height of the object in image, f = focal length



This is the simple case when projection is orthogonal and there’s no rotation or translation required.

This relation in terms of X, Y, and Z world coordinates(cm or mm) in homogeneous form can be written as:



Where (u, v) is the image coordinate, (cx, cy) is called principal point for aligning the centre of image with the origin of image coordinates, (fx, fy) are focal lengths in pixel values and s is scaling factor

The first matrix on right hand side has called intrinsic camera parameters

Second matrix has rotation and translation parameters and it depends on the camera position and orientation, (i.e) it differs from one view to another. This is one more reason we want the projection to be orthogonal, in which case, (r11 , r22, r33) are 1’s and the rest of the matrix is zeros.

The scaling factor depends on Z, so we must know the distance between the object and image plane (camera is fixed while setup, so Z is easy to calculate by fixing the height of the ring to a certain level)

I’m assuming the matrix of intrinsic parameters to be calculated priori by camera calibration and distortion correction.

The final step is plugging into the equation all the known values and obtaining X and Y thus calculating the diameter and thickness in cm/mms.

Based on this ordering of the rings may be done.

**Defect Detection:**

**Preprocessing :**

No single preprocessing procedure can be generalized for all different setups and different defects. Once, the image acquisition and environment setup if fixed, we may try different filters and operations that gives satisfying results. Some of the generic suggestions that might work are CLAHE, bilateral filter, etc; although the parameters would vary from one setup to another.

**CLAHE:**

It stands for Contrast Limited Adaptive Histogram Equalization. It’s used to deal with the over-amplification of contrast which occurs in simple histogram equalization and over-amplification of noise that occurs in Adaptive Histogram Equalization. Instead of applying HE to the whole image, CLAHE divides the image into smaller regions(called tiles) of specified size (given as a parameter to the algorithm) and applies HE in those regions independently. A threshold for contrast limiting is to be specified so that if contrast is higher than threshold it gets clipped.

Overall, CLAHE helps in reducing extremely bright and dark areas ( which may occur due to lighting variations) . CLAHE is applied to a channel that has to do with the brightness/intensity aspect of the color, it is usually applied on a grayscale image or Value-channel in HSV space or L-channel in L\*a\*b\* color space, etc;



It is evident that the cuts are more clearly distinguishable in the right image(CLAHE) than in the left image(original). And the contrast is improved.

**Bilateral Filtering:**

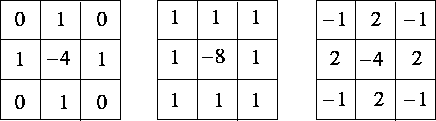
Median or Gaussian blur can’t be used to remove the noise here because they may smoothen the edges which may be defects in the ring.

So, bilateral filter serves our purpose as it’s an edge preserving, non-linear and noise-reducing filter. It replaces intensity of each pixel with weighted average of intensity values of its neighbourhood. Gaussian distribution can be used for these weights. Instead of a single sigma(spatial) for a Gaussian blur, there is an additional sigma for color space, so pixels with only similar intensity in the neighbourhood are considered for blurring.

**Edge detection:**

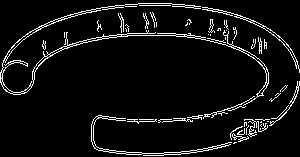
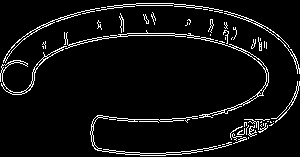
Cuts/fissures or any other defects that are distinguishable by color intensities can be identified as edges. Laplacian filter is a very useful filter that detects edges agnostic of lighting as long as the reflections are somewhat evenly distributed (after CLAHE).

Laplacian filter is a second order spatial derivate filter that identifies edges by computing second order derivatives. For applying convolutions, in discrete space, Laplacian operator is commonly approximated as these three masks.

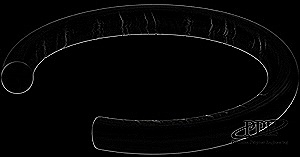
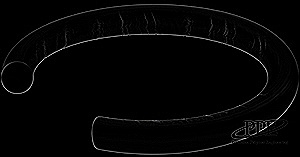


I obtained the following results after applying the Laplacian API function on the images mentioned in the previous section. I also applied a Canny operator next to highlight the edges(make edges more white and noticeable)

Left image was Laplacian followed by Canny on the original image, right one same application but with CLAHE.



Canny was only applied for visualization, the double layer that can be seen is because of that. For defect detection Canny is omitted and images would look like this:



**Defect detection:**

After obtaining the above image, we **mask out the boundary of the ring** (as we already know the two boundary circles). All that is left of the image are the defects. A perfect ring would have no edges and the resultant image would be completely blank. It’d detect deformations as well as they wouldn’t lie on the circle.

When applying the same steps to a different image with same parameters, horrendous results were produced. It required change in tile size of CLAHE, intensity halving, and additional bilateral filtering to produce similar results to that of previous image. This was because this image has too much reflection associated with the ring.



Some other ideas are fitting lines for cracks, fissures,

blob detection for foreign material, mould defects,

finding contours for path defects, etc;

**Conclusion:**

I believe any solution designed would be specific to that type of environment conditions- a fixed image acquisition and hardware setup. I think edge/contour detection would work good for any sharp deviations in colour(although a simple marking with a white marker would also be detected as a defect and what if some portion of the ring fades in colour and results in an edge?). A deep learning approach might be helpful to generalize the problem to some extent.