

Pollution Patch

Aleksander Gerard-Ursin, ECE '20

Introduction:

Our project was to create a patch composed of panels, each made of different types of smart threads, that would change color when it reacted with specific toxic industrial chemicals (TIC) (Owyeung et al., 2019). We were also tasked with developing an app that would use an image processing algorithm to detect the different panels on the patch, extract their pixel data, and classify the small color changes of the different threads on the patch. This was in turn used that information to determine what TIC(s) the patch had been exposed to (Lim et al., 2009). This however brought up a big challenge. Namely, how can you algorithmically recognize different parts of an image corresponding to asymmetrical shapes? In this report I will do my best to outline, one component of our approach to this issue. My focus will be on how we can detect asymmetrical shapes on an image, from a highly unreliable system i.e. a smartphone's camera in a multitude of lighting conditions.

My focus will be split up into two sections, how do you detect a shape using an edge detector and how can we filter out the noise to allow for finding unclear borders between objects in an image. These elements are core to the Image Processing system that will be used by our app and represent a lot of the knowledge that we have gained over the semester through significant amount trial and error.

Edge Detection:

The way that we were able to detect the patch and its individual panels was by using something known as the Canny edge detection algorithm. What is meant by an edge is the border of a shape on an image. Let us take the following image of our patch for example:



Figure 1: Image of pollution patch

We want to be able to detect the different panels; to do so we need to find the panels' edge. The edge of a panel is the border between the panel and the outline. In other words, the edge can be defined as where there is a large color difference/gradient. In fact, that is how the canny edge detection algorithm works. By dividing up the light intensity into an upper and lower threshold, the algorithm finds areas where the two thresholds meet, and that is what is considered an edge (Gaurav & Ghanekar, 2018). Here is an example of that same image after the edges were detected:

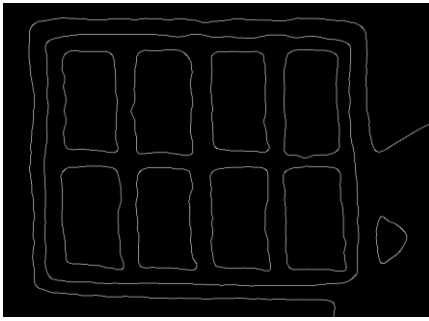


Figure 2: Pollution patch after Canny Edge Detection

The threshold is of course the most important aspect of this algorithm, since that is how we find the edges. The mechanism that is often used is finding the average intensity of every pixel of an image and using that average to find the two thresholds. This works a large amount of the time, but sometimes the camera quality might not be good enough to fully detect these gradients. This is where the next part comes in.

Filtering:

Typically, in an image there is something known as noise in it. Noise is another way of defining random distortion of an image in other words, the messy part of the image. Filtering is what it is called when you try and remove this messy part. The way that filtering works is as follows.

Most noise can be modelled by some mathematical function. Of course, this will not be a perfectly accurate model of the noise, but you can get close. With the model, you can do something known as convolving the model with the signal that you have. In our case the signal is the image of the patch. By convolving the signal with the model of the noise, you effectively remove all forms of that noise in the signal (Andreas Antoniou author, 2018). However, this is not a complete picture of what it means to convolve a signal.

To convolve a signal, you need to first change the function representing the noise and the signal itself into the frequency domain. What is meant by this is that using a mathematical function, you find specific patterns and record the number of

patterns per signal. This is what allows us to get rid of the noise using convolution, since we can find where the patterns of the noise overlap with our signal and remove it. Before converting is back into its original domain. (*Signal Processing: A Mathematical Approach - Tufts University, n.d.*)

In our case, we had to implement multiple types of filters. We had to try our best to remove as many forms of noise that we could so that our Canny Edge detection algorithm could work as well as possible. This reduction of noise would smooth the gradient, making edges easier to detect. This worked out well as can be seen by the how accurate the outlines of the panels are in the following image:

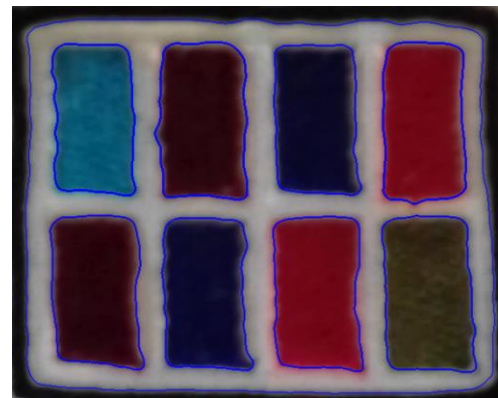


Figure 3: Pollution patch with panels found

Conclusion:

One of the most interesting parts about this project was how many concepts that seemed like they were only focused on one specific idea could be applied to a myriad of different fields. Filtering is a beautifully simple method to getting rid of some of the biggest annoyances in signals. The signals could be everything from images, music, language, or whatever else one could think of. However, the most exciting bit about this, is how it shows that often one can find uses for skills one has in areas one would not expect them to be useful in.

Bibliography:

- Andreas Antoniou author. (2018). *Digital Filters: Analysis, Design, and Signal Processing Applications* (2nd edition.). New York, NY: McGraw-Hill Education.
- Gaurav, K., & Ghanekar, U. (2018). Image steganography based on Canny edge detection, dilation operator and hybrid coding. *Journal of Information Security and Applications*, 41, 41–51. <https://doi.org/10.1016/j.jisa.2018.05.001>
- Lim, S. H., Feng, L., Kemling, J. W., Musto, C. J., & Suslick, K. S. (2009). An optoelectronic nose for the detection of toxic gases. *Nature Chemistry*, 1(7), 562–567. <https://doi.org/10.1038/nchem.360>
- Owyeung, R. E., Panzer, M. J., & Sonkusale, S. R. (2019). Colorimetric Gas Sensing Washable Threads for Smart Textiles. *Scientific Reports*, 9(1), 5607. <https://doi.org/10.1038/s41598-019-42054-8>
- Signal processing: A mathematical approach—Tufts University*. (n.d.). Retrieved May 4, 2020, from https://tufts-primo.hosted.exlibrisgroup.com/primo-explore/fulldisplay?docid=01TUN_ALMA51227931130003851&context=L&vid=01TUN&lang=en_US&search_scope=EVERYTHING&adaptor=Local%20Search%20Engine&tab=everything&query=any,contains,convolution%20filter
- Turin, G. (1960). An introduction to matched filters. *IRE Transactions on Information Theory*, 6(3), 311–329. <https://doi.org/10.1109/TIT.1960.1057571>

