Kyle Xiong
CIS II
Group 11

Paper Critical Review: A New Edge Detection Method Based on Contrast Enhancement

Paper Selection and Relevance to Project

Our project is focused on detecting occluding contours from sinus surgery videos to facilitate tool tip positioning with a magnetic tracker. We aim to extract the edges from each video frame and register them to CT data. By doing so, tool tip positioning can become more accurate and can overcome the limitations inherent in magnetic tracking. The paper I've selected introduces a new rigorous method of contour detection based on calculating the contrast of a region around each pixel of an image. The paper's method also includes a process called non-maximum suppression to filter out false positives (false edges), return as many true edges as possible, store these edges in two edge images, and connect these edge images together to form an ideally noise-free and precise edge detection image.

Although the paper addresses the problem we're currently having, there are potential problems with its method specific to our project. For example, the regions it selects may be too small to efficiently detect all the edges in a sinus surgery image; or the regions may be so small that we skew the edges we detect due to faulty non-maximum suppression. Even so, I am interested in the ideas that the paper presents and believe it will be useful in the upcoming steps of our project.

Problem Summary

Current edge detection methods for sinus surgery are limited by accuracy and efficiency. While there are accurate existing methods (i.e. a machine-learning algorithm developed by a group at Berkeley), they are too slow for our purposes – we require real-time edge detection,

and their method takes up to a few minutes to run and frequently takes up too much memory for MATLAB to handle. More efficient edge detection methods are either inaccurate or return too much noise (extraneous texture). The paper aims to accurately and efficiently detect edges by using new criteria. It makes few intensity comparisons and calculations for each pixel of the image, and uses a set of predetermined classifications to match each pixel to a certain edge shape. Edges inherently have higher contrast than non-edge points so it's possible to distinguish the two by detecting sharp changes in pixel intensity.

Currently, we have used various computer vision algorithms in attempts to accurate detect edges, including: Canny edge detection, Sobel edge detection, Horn-Schunk optical flow, Lucas-Kanade optical flow, our own smoothness filtering algorithm, and our own intensity filtering algorithm. We applied some combinations of these methods to each frame of the sample data we were given and achieved reasonably good results for a few of the frames. Our best result was generated from using Sobel edge detection, Horn-Schunk optical flow, smoothness filtering and intensity filtering. However, we used many parameters for each method, and each parameter was static. For these parameters, a few frames looked very good but for most frames our detection was subpar. The method introduced in the paper requires much fewer parameters and dynamically assesses whether an edge exists or not based on criteria inherent in every image in the world.

Therefore, our expected results from using this paper would be a minimum-noise and accurate edge detection image for every single frame of our sample data. In other words, we hope to maintain the accuracy of the first frame in every other frame that we process.

Kyle Xiong
CIS II
Group 11

Method Overview

The method introduced in the paper is called contrast enhancement edge detection (CEED). CEED calculates contrast for each pixel according to the intensity of its surrounding pixels. It looks for a neighborhood of pixels around each pixel and stores the calculated contrast for each pixel in a matrix. Edges can then be detected through threshold computation (edge connection step). The CEED workflow is as follows:

First the method takes the original image as its input and applies a Gaussian filter to the image. The result is a smooth image with less apparent textures and noise. The smooth image is input into an image enhancement (contrast calculation) algorithm and the algorithm outputs a contrast image, or a contrast matrix. Using the contrast matrix the method matches 3x3 windows of pixels to a set of 16 edge classifications and decides which classification best fits the window. The matrix is then used for non-maximum suppression, which removes any edges that aren't a local maximum. Edge connection then applies two thresholds to get two edge images, one that's heavily filtered and one that's lightly filtered. CEED first connects edges in the heavily filtered image and then uses the lightly filtered image to interpolate any missing edges. The end result is a connection edge detection image.

Analysis of Results

The results the authors saw were very encouraging. Compared to both LoG and Canny edge detection, this method had less noise and more comprehensive detection. They used three images: an x-ray of the knee, various items on a desktop, and Mount Rushmore. In the x-ray image, CEED did not have a significant improvement over Canny. Both methods found all

the true edges and CEED found more textures. Both Canny and CEED returned more true edges than LoG and showed much less noise. However, CEED was noticeably better for the desktop items and Mount Rushmore. In the desktop image, it showed more true edges and displayed more ridges on the key than Canny did, suggesting that it's more accurate than Canny. This makes sense as CEED scans each pixel rigorously for intensity variation and keeps local maxima. In the Mount Rushmore image, CEED showed complete contours around the presidents' pupils whereas Canny failed to do so.

Thus, from the authors' results I concluded that CEED might not necessarily reduce the noise we saw in Canny edge detection, but it is more accurate and we should build off of this accuracy in the upcoming steps.

Evaluation of Paper

This paper was useful to me because it provided a new edge detection method that seems more rigorous than our current detection method. CEED can eliminate false edges through non-maximum suppression. In addition, it can accurate display true edges by using a lower and higher intensity threshold when connecting edges. It's also relatively fast compared to our current method, using fewer comparisons and storage calls and less image processing. Our current method requires Sobel edge detection, optical flow edge detection, using motion vectors to calculate smoothness, filtering out smoothness based on intensity, and then using the filtered smoothness matrix to remove noise from our Sobel result. That's a lot more processing and I'm hopeful that this method can both reduce our runtime and improve accuracy of detection.

Possible problems with this method include inaccuracies with blurring and the ever-present danger of textures being detected. After applying the Gaussian filter, most edges become wider than 3x3, so it's easy to see that the method may detect a contrast peak at a pixel not on a true edge. After blurring, there is the question of whether contrast is highest at the edge of the blurring or at the true edge of the tissue. The paper doesn't address this issue and I think that's where it might fall apart when applying it to a high-resolution sinus surgery image.

In terms of extraneous textures, the method cares about sharp disparities in contrast during the non-max suppression step, which textures still have even after Gaussian blurring. Therefore the method isn't guaranteed to filter out textures, even if we applied the double threshold method in edge connection.

This paper's method has some potential problems if we apply it to sinus surgery, but I want to use the math and its categorization of edges to help our contrast profiling method. This method measures consistency intensity across detected edges to differentiate between textures and edges. By using the general shapes of edges in the paper, we can then expand the window size to look for intensity plateaus or valleys/peaks. True edges correspond to a plateau – that is, pixel intensity is uniform before and after the edge, and only sharply varies at the edge. Textures corresponding to a peak/valley because pixel intensity most likely returns to near the same value after a texture is detected.