

Detecting Guns Using Parametric Edge Matching

Aaron Damashek and John Doherty

Stanford University, Stanford, CA, USA

{aarond, doherty1}@stanford.edu

Abstract

Weapon detection is a difficult problem with numerous applications, particularly in the world of airport security. We present an attempt to identify pistols in x-ray images using Chamfer Matching. Our approach builds upon the basic Chamfer method to address issues with occlusions by combining results from sub-polygon templates using voting and machine learning. Experiments show that the underlying Chamfer method does not produce results with significant accuracy to replace airport security personnel.

1. Introduction:

Security is a big business. Across the world, millions of security professionals are paid to watch video streams from cameras, ensuring that any criminal activity that occurs is detected. If computer vision can be used to take over even a fraction of the billions of hours spent a year in these endeavors, the savings would be enormous. In addition, there is a great likelihood that computer vision could detect many incidences that go unnoticed by human supervisors.

One particular security issue that is of great concern is the presence of guns. Often, the presence of a gun is a good indicator that violence or criminal activity will soon follow. While important everywhere, detecting the presence of guns is of particular concern for security at airports. The TSA employs officers to watch X-Rays of all baggage going on to a plane to detect guns and other potential weapons. It would be incredibly useful if computer vision could be used to detect guns in these X-Rays.

To solve this problem, we attempted to use Chamfer Matching - a method that tries to match a template image of the object to be identified (in this case a gun) with a distance map based on the edges of an image in which the object potentially resides. We expanded upon this method to address the issue of occlusions, particularly superimpositions, by breaking the template into sub-polygons and attempting to match the sub-polygons. We attempted to further improve these results by using machine learning on the results we discovered testing the full template and sub-polygons.

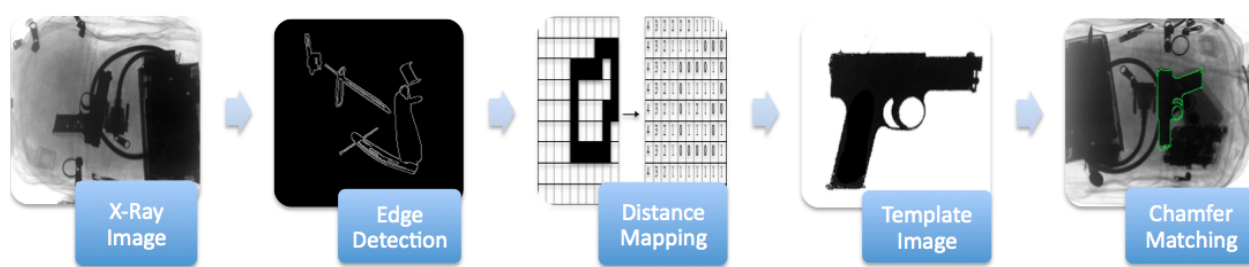


Figure 1: Block Diagram of the Chamfer Matching algorithm

2.1 Previous Work

Detection of guns and other dangerous objects in airport X-Ray imagery is an active area of research for the reasons mentioned above. Finding dangerous objects in X-Ray images is a detection problem at its root, but there are a number of unique challenges that make this an interesting area of research. First, luggage can be packed several layers deep, and target objects are generally covered by other objects. As a result, images of other objects are often superimposed onto the image of the target object. This superimposing effect is a type of occlusion unique to X-Ray images. The basic shape of the object is still present, but the features of the object may look quite different depending on what is covering it. Additionally, any detection algorithm must be invariant to rotation, skew, and scaling of the target object in X-Rays. Furthermore, all of these possibilities must be handled relatively quickly since the algorithm will most likely be deployed in crowded airports with serious time constraints.

Currently, most X-Ray scanners at airports do not use computer vision object detection techniques, instead relying on more basic image processing to make detection easier for security officers. These scanners generally employ basic segmentation algorithms to find different objects and color them differently in the output image. While there is work being done to apply modern machine learning and object detection algorithms to this problem, it is yet to achieve widespread distribution.

Researchers have explored a number of computer vision approaches to detecting dangerous objects in airport images, reporting varying levels of success. These approaches range from using basic SIFT descriptors[2] to using an estimate of the 3D structure of the objects in the X-Ray to find potentially dangerous objects[3]. While these approaches have achieved fairly high detection accuracy, they are not accurate enough to be useful in a real security setting where there is no room to miss a detection. Additionally, most of these approaches are too slow for use in a fast paced security setting. For example, the 3D reconstruction approach mentioned earlier took about 30 seconds to perform a single detection. Another, slightly less important concern, is that many of these algorithms produce a high number of false positives. While false positives are much better than false negatives, they can still slow down the security screening process.

One object detection method that has been used in non-security settings is the Chamfer Matching Algorithm (HCMA). Chamfer Matching is an edge matching algorithm that tries to find the optimal alignment between the edges of a template and edges of the image in which we are searching. HCMA was successfully applied to the problem of locating tools in a toolbox[1], which is similar to the gun detection problem we are looking to solve. HCMA works well on objects with unique edge patterns, making it promising for identifying guns in images. The algorithm is extremely effective at identifying the object in an image where the only disturbances to the edges are noise. On the other hand it can struggle when dealing with images where either part of the object is missing or part of the edge is missing. Despite a number of potential drawbacks, Chamfer Matching has characteristics that make it promising for solving the pistol detection problem.

2.2 Contributions

Our method improves upon previous methods of gun detection as well as the Chamfer Matching

algorithm. First, our method is able to handle superimpositions far better than previous methods. While superimposed outlines of other objects in luggage stymie some methods because of the difficulty they present the segmentation problem, because such superimposed outlines do not affect the outline of the gun in the edge image, the full correct gun is still present in the distance map created from the edge map. As a result, despite the superimposed outlines of other objects in the image, there is still a perfect matching possible between the template and the outline of the gun in the distance image. Thus, the superimposed outlines of extraneous objects in X-Ray images do not hurt the results of our method to the same extent as others.

Our method improves upon the basic chamfer method in three major ways. First, while the opencv version of Chamfer Matching is position and scaling invariant, it does not test for different rotations of the template. Given the unknown rotation of guns in X-Ray images, it was essential for our method to be rotation invariant, so we have improved the method to take rotation into account. Second, we explored the possibility posited by the algorithm's inventors of raising accuracy through sub-polygon Chamfer Matching. While the Chamfer Matching method deals with occlusions much better than other methods, if the edges of the object are completely occluded (and not merely superimposed over by other edges), the method will not have the appropriate edges to match to and will thus be unable to find the object. However, if part of the object is not occluded, it is possible to match a subsection of the template to tell that the whole object, though not visible, is present. We expanded upon the basic chamfer method to subdivide the template into parts, which we then ran Chamfer Matching on to determine whether the object as a whole was present. Finally, we improved upon the method used in normal Chamfer Matching to determine the presence or absence of an object by using machine learning. Instead of comparing the smallest distance found from the template to the distance map to an arbitrary threshold, in the case of the whole template we use machine learning to examine the cost and determine the presence of a gun. In the sub-polygon case, we use machine learning to examine the costs of all the sub-polygons and then determine the presence of the gun as a whole.

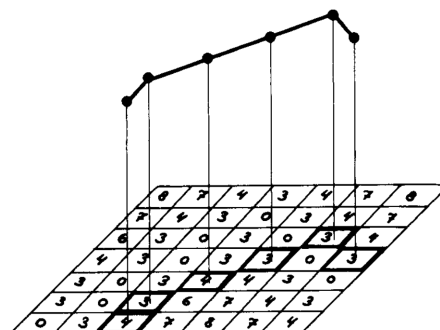
3.1 Technical Solution Summary

We tackled the problem of determining the presence of guns in X-Ray images by using a parametric edge matching algorithm called Chamfer Matching. Chamfer Matching finds the position in which a template image of a gun placed on a distance map (created from an edge map of the target X-Ray image) minimizes the sum of the distances on which it is placed. Based on this distance, it determines whether or not the gun is present in the image. We improved upon this basic method by splitting the template image up into many sub-images, and running Chamfer Matching on them. We used two methods for determining whether the gun as a whole was present based upon the results of running the chamfer algorithm on the sub-polygons: voting and machine learning.

3.2 Technical Solution Details

Chamfer Matching is an edge matching algorithm, so an edge detector is crucial to the success of the algorithm. The Chamfer Matching paper[1] does not go into details on a specific edge detection algorithm to use, so we decided to use the OpenCV Canny Edge Detector[9] for simplicity and accuracy. As for the implementation of Chamfer Matching, there is a version of the Chamfer algorithm in OpenCV that we are basing our implementation on.

After running Canny Edge Detection on an X-Ray image and template image of a pistol, the algorithm selects a set of points from the template edges to create the “polygon.” It then applies a distance transform to the X-Ray edges to create what is known as the “distance image.” The distance transform assigns a value to every pixel in the distance image equal to the pixel’s distance to the nearest edge pixel. In other words, in the distance image, edges have values of zero, and all other pixels have values equal to the distance to the nearest edge.



The goal of matching is to find the best way to place the polygon points on the distance image such that the sum of the distance pixels that polygon points fall on is minimized (Figure 2). Specifically, the algorithm tries to match the polygon points to the distance image at the location, scale, and rotation that minimizes the matching cost. The cost of a match is defined as:

$$\frac{1}{3} \sqrt{\frac{1}{n} \sum_{i=1}^n v_i^2}$$

Where v_i is the value of the distance image at a point i in the transformed polygon, and n is the number of points in the polygon. Our implementation used a brute force approach to run through possible transforms of the polygon. This gives us a runtime of $O(x*y*s*r)$ where x is the number of horizontal positions we search, y is the vertical positions, s is the number of scales, and r is the number of rotations. This runtime is not ideal, but there are techniques to speed up the process. One proposed method to improve the runtime of the Chamfer Matching algorithm is to make use of a resolution pyramid. This is a hierarchical approach to Chamfer Matching where the distance image is computed at several resolutions ranging from a very low resolution approximation to the resolution of the original image. The algorithm starts by running Chamfer Matching on the low resolution distance image and uses the results to guide matching on higher resolution distance images. Ideally, by the last step (at the original resolution), the matching algorithm only has to make minor adjustments to the scale, rotation, and position of the match. The idea is that Chamfer Matching runs significantly faster on smaller images, but can still reliably find the region of the most likely match. While this algorithm runs the risk of getting caught at local maxima at lower resolution, it could offer the dramatic runtime improvements that would be necessary to use our algorithm in a real world system.

Implementation

We use code from OpenCV[9] as the basis for our Chamfer Edge Matching implementation. We use the OpenCV Canny Edge Detection implementation to get the edges for the template and X-Ray image. The OpenCV Chamfer Matching code covers creating the polygon and distance images. It also handles iterating through scales and locations of the polygon on the distance image. However, it does not perform rotations, which are crucial to solving our problem. We implemented a template rotation transformation as well as a parallelized method to run Chamfer Matching on a template and a horizontally flipped version of the template.

Sub-Polygon Method

Chamfer Edge Matching can perform well when matching a template to simple edge images, but it can be easily thrown off by images with many overlapping edges. The algorithm struggles significantly with images in which the target object is partially occluded. Because occlusions are common, if not ubiquitous, in X-Ray images, we needed to improve the basic Chamfer Matching algorithm.

To improve resistance to occlusions, we decided to split the template image into sub-images to generate a set of sub-polygons. Ideally, running Chamfer Matching on these sub-polygons would detect important parts of the template without needing the entire target object to be visible in the image. We then combined these sub-matches in a variety of ways to more accurately match the template.

Our first attempt was to use a basic voting scheme to determine the presence of a pistol by checking for the presence of smaller pistol regions. We broke the polygon up into evenly spaced sub-polygons, discarding any sub-polygons that did not contain any edges. We ran this for 2x2, 3x3, and 4x4 divisions of the template as shown in Figure 3. The issue with this approach is that smaller sub-polygons can often find lower cost matches within the distance image because they contain more generic edges. Because it is often easy for the algorithm to find quality matches for many of the sub-polygons, the voting algorithm was not successful in determining whether or not a gun was present. We continued to improve this method with techniques discussed later.

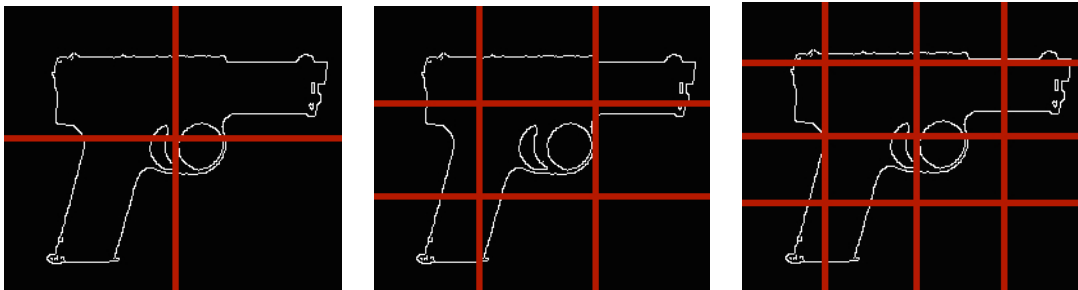


Figure 3: Template division into 4, 9, and 16 sub-polygons

Machine Learning

Machine learning was used to improve the accuracy of the basic chamfer detection method, as well as the sub-polygon Chamfer detection method. Since the problem was a classic linear classification problem with two classes, we decided to use a Support Vector Machine (SVM), and utilized the dlib C++ library[7]. In the full template test, we used the cost returned by the Chamfer Matching algorithm as the sole feature for the SVM. Testing at different levels of subdivision, we used the costs and gun presence decisions for each sub-polygon as features for the SVM. To train the SVM, we randomly shuffled our pre-labelled data, and used the first 70% of samples. To determine the appropriate parameters for the SVM, we looped through numerous possibilities, performing 10-fold cross validation for each to determine which settings produced the most accurate results classifying both positive and negative examples.

Dataset

A quality dataset was crucial to testing our basic Chamfer Matching algorithm as well as training and testing our machine learning component. While X-Ray image data from security agencies like the TSA is

not readily available to the public, we were able to obtain a large dataset of both gun and non-gun X-Ray images from researchers working on the same problem[8]. Domingo Mery et al.[3] were working on detecting dangerous objects from multiple views and created a large dataset of X-Ray images from multiple angles that they let us use.

4. Experiments

Figures 4 and 5 show the results of the first set of experiments that we ran. This was simply a test of the basic Chamfer Matching algorithm before applying any sort of sub-polygon matching. These results show that the algorithm performs fairly well in images where the pistol is unoccluded and not skewed. The detection performance falls off significantly in cases where the pistol is somewhat occluded or skewed. Overall, the basic algorithm correctly located the guns in 53 of the 257 images with guns.

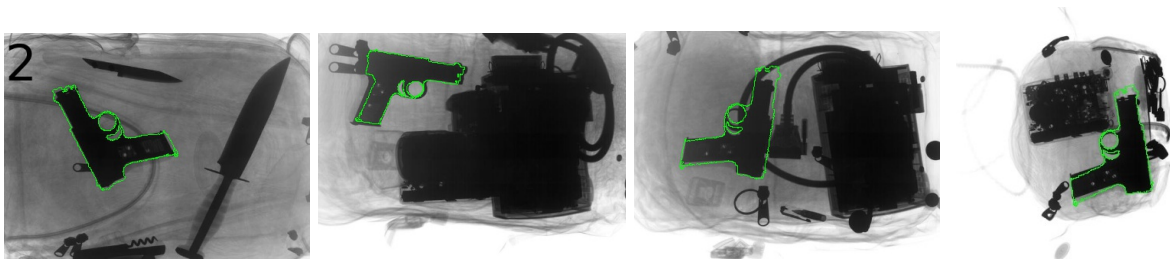


Figure 4: Basic Chamfer Matches

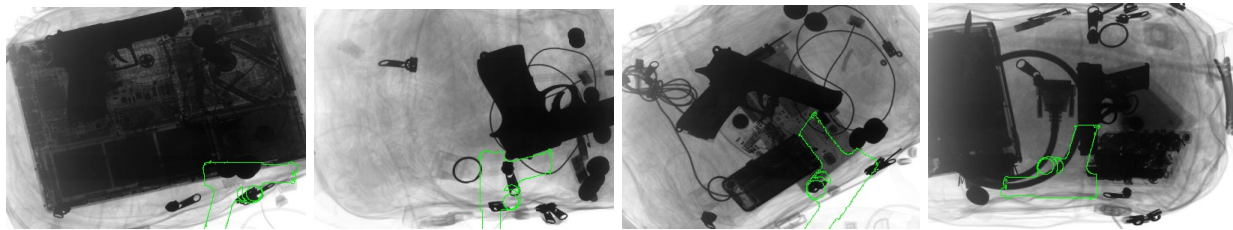


Figure 5: Basic Chamfer Misses

The images in Figure 6 show the mixed results of sub-polygon matching. In some cases the sub-polygon is exactly aligned to the correct edges in the X-Ray image. On the other hand, the sub-polygons also find quality matches in images without guns, as well as matches with costs lower than the correct match in images with guns.

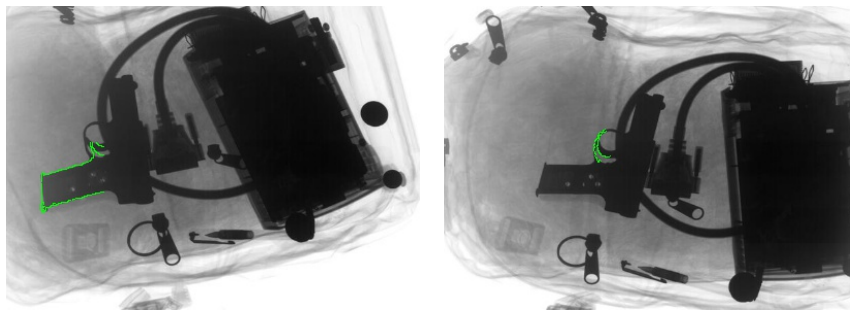




Figure 6: Sub-Polygon Matches

As a result, it was difficult to classify gun presence based on voting, since in almost every case the sub-polygon template was able to find a decent match. We hoped to use machine learning to develop a better threshold for the full template, and to identify and weigh more important sub-polygon features as well as their costs. However, given the nature of matches discussed above, it was not possible to easily discern a pattern from the costs. As can be seen in Figure 7, the distribution of costs for sub-polygons in both positive and negative cases was roughly the same. As a result, attempts to use machine learning were unsuccessful in differentiating images with and without guns. The machine learning chose to simply label all images as negative examples, since this produced a higher accuracy given the higher number of non-gun images in our database.

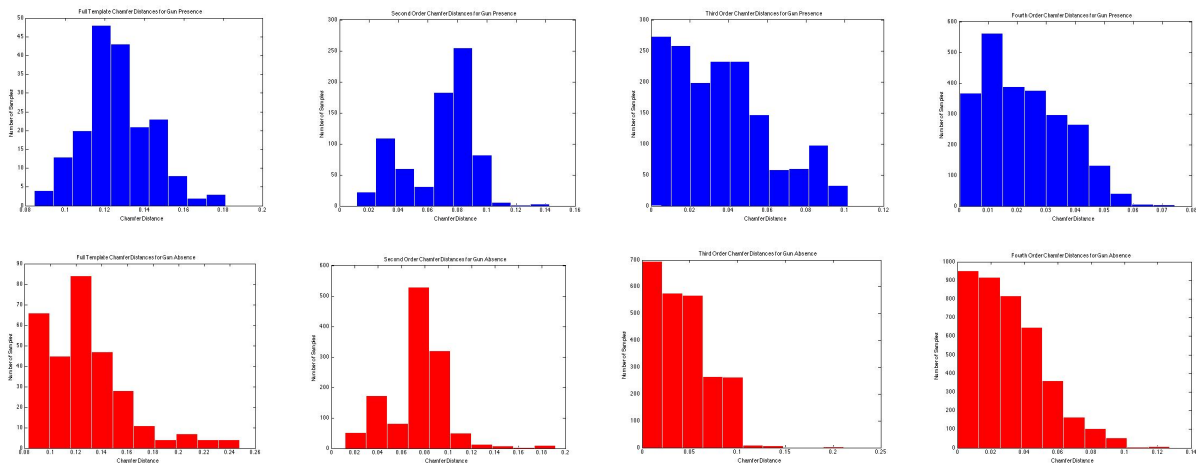


Figure 7: Distribution of Chamfer Distances

In response to our difficulties differentiating positive and negative cases due to the lack of differentiation in costs, we hypothesized that there must be something negatively impacting the way costs were calculated. To test this, we tried removing the weight on orientation in determining cost. As a result, we saw far more differentiation in costs, as can be seen in Figure 8.

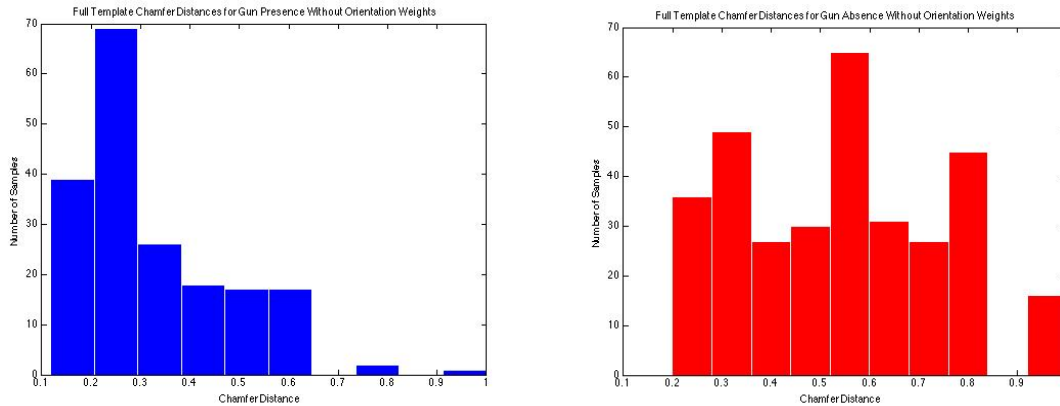


Figure 8: Distribution of Chamfer Distances Ignoring Orientation

However, despite this increased differentiation, we did not see improved accuracy. Though the distributions were now distinct, due to the extent of the remaining overlap, the accuracy obtained from splitting along the apparent boundary remained less than that identified by the machine learning algorithm - simply rejecting all cases.

Given the greater differentiation in the full template case, we made another effort to use sub-polygon Chamfer Matching, this time ignoring the orientation in our weighting. However, once again, the overlap of Chamfer distances between positive and negative samples was too great.

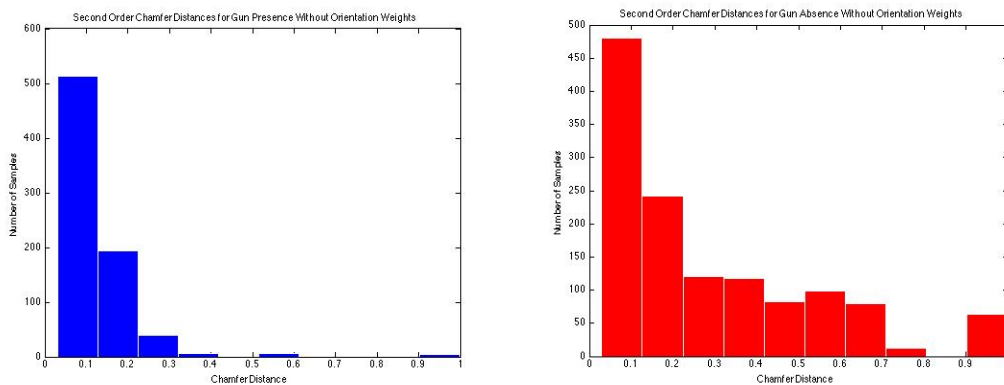


Figure 9: Distribution of Sub-Polygon Chamfer Distances Ignoring Orientation

As a whole, our attempts achieved only minimal success in identifying the presence of guns. As can be seen from Figures 10 and 11, success rates ranged from ~40-65%, while F1 scores taking into account accuracy and precision were stable near 0.6.

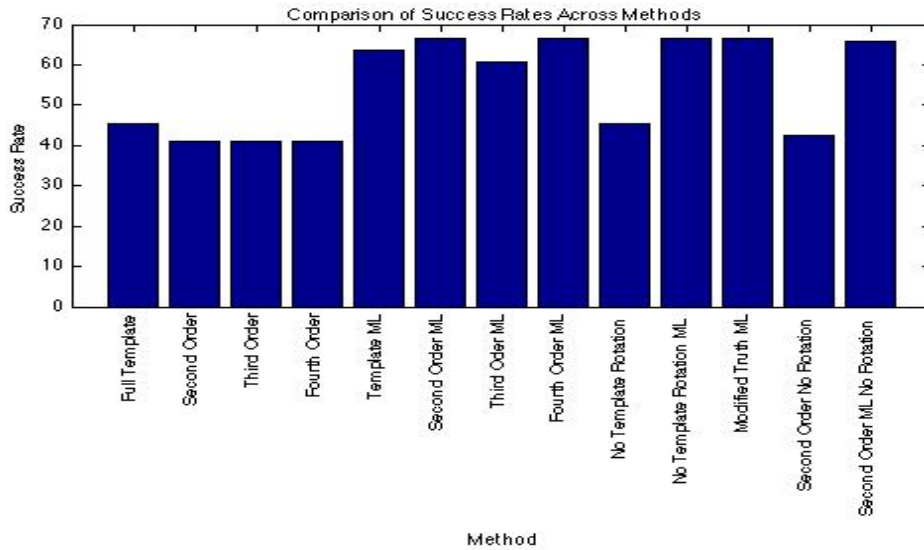


Figure 10: Chamfer Success Rates

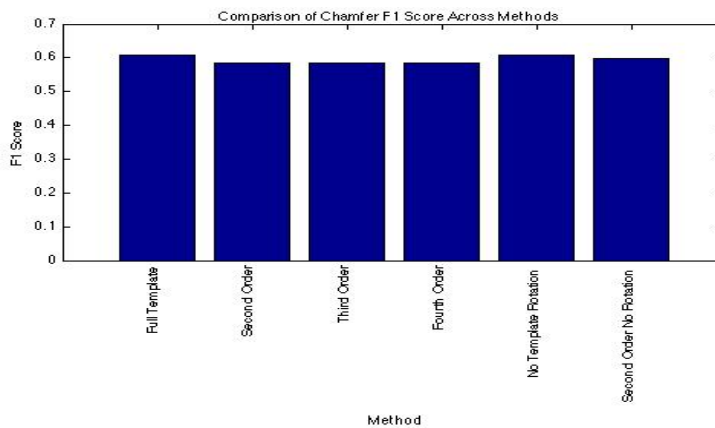


Figure 11: F1 Scores

5. Conclusions

Applying Chamfer Matching to the detection of pistols in X-Ray images has not provided the accuracy necessary for real world implementation. While the Chamfer Matching algorithm matches some images very well, it is quite finicky, and often finds guns where they are not. As a result, our classification results using the basic chamfer method were only moderately successful. While we had hoped using subdivisions would improve our accuracy by dealing with some of the effects of occlusions, we saw that subdivision at all levels did not have a positive effect on accuracy. Similar to the basic Chamfer Matching, subdivisions were often matched to parts of the image where there was no gun. In fact, subdivision matching often did worse because simple subdivisions could find low cost matches in any image. These flaws made our simple voting based algorithm largely unsuccessful. We hoped to see more success using a SVM to learn the relative importance of the different subdivisions. But because subdivision matching was largely unpredictable, this failed to produce consistent results as well.

Improvements

While there are some potential improvements that could be made to our algorithm, Chamfer Matching does not seem to provide a good foundation for solving the pistol detection problem. That said, one potential improvement to subdivision matching would be to give a weight to the relative locations and rotations of subdivision matches. Currently, the algorithm searches for each subdivision independently and identifies the best possible match for each. We could instead search for the subdivisions concurrently, giving an overall score to the combined matching of all subdivisions. In other words, if we were matching subdivisions for the handle and the trigger of a pistol, we would minimize the edge matching cost of each subdivision as usual. But additionally, the total match would be given a lower cost in cases where the matches for the handle and trigger were near each other and correctly aligned and a higher cost in cases where the subdivisions appear on opposite sides of the image.

Besides that possible improvement, it seems that Chamfer Matching has some fundamental flaws that limit its usefulness in pistol detection. Namely, there is not much that can be done to work around the algorithm's ability to cope with slight occlusions. Using SIFT feature descriptors as input to our machine learning algorithm would have likely seen more success.

6. References:

- [1] Borgefors, Gunilla. "Hierarchical chamfer matching: A parametric edge matching algorithm." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 10.6 (1988): 849-865.
- [2] Gesick, Richard, Caner Saritac, and Chih-Cheng Hung. "Automatic image analysis process for the detection of concealed weapons." *Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies*. ACM, 2009.
- [3] Mery, Domingo, et al. "Detection of regular objects in baggage using multiple X-ray views." *Insight-Non-Destructive Testing and Condition Monitoring* 55.1 (2013): 16-20.
- [4] Roomi, Mohamed Mansoor, and R. Rajashankari. "DETECTION OF CONCEALED WEAPONS IN X-RAY IMAGES USING FUZZY K-NN." *International Journal of Computer Science, Engineering & Information Technology* 2.2 (2012).
- [5] Anderson, SB. "TSA Carry-on Gun Confiscation Data 2013." *Medill National Security Zone*. TSA, 5 Jan. 2014. Web. 19 Mar. 2014.
<<http://nationalsecurityzone.org/site/tsa-carry-on-gun-confiscation-data/>>.
- [6] Burns, Bob. "TSA Week in Review." Web log post. *The TSA Blog*. TSA, 28 Feb. 2014. Web. 19 Mar. 2014. <<http://blog.tsa.gov/search/label/Week%2520In%2520Review>>.
- [7] Davis E. King. [Dlib-ml: A Machine Learning Toolkit](#). *Journal of Machine Learning Research* 10, pp. 1755-1758, 2009
- [8] Mery, Domingo. X-Ray Image Database. 2012. Raw data.
<Http://dmery.ing.puc.cl/index.php/material/gdxray/>, n.p.
- [9] Bradski, G. *OpenCV*. Computer software. *OpenCV Dev Zone*. Vers. 2.4.9. Dr. Dobb's Journal of Software Tools, 15 Jan. 2008. Web. 20 Feb. 2014. <<http://code.opencv.org/>>.