



The author(s) shown below used Federal funding provided by the U.S. Department of Justice to prepare the following resource:

Document Title: Quantitative Algorithm for the Digital Comparison of Torn Duct

Author(s): William Ristenpart, Ph.D., Frederic Tulleners, M.A., Alicia Alfter, B.S.

Document Number: 250661

Date Received: March 2017

Award Number: 2013-R2-CX-K009

This resource has not been published by the U.S. Department of Justice. This resource is being made publically available through the Office of Justice Programs' National Criminal Justice Reference Service.

Opinions or points of view expressed are those of the author(s) and do not necessarily reflect the official position or policies of the U.S. Department of Justice.



National Institute of Justice

The Research, Development, and Evaluation Agency of the U.S. Department of Justice

Final Report

Organization: University of California at Davis

Course Title: Quantitative Algorithm for the Digital Comparison of Torn Duct

Award # 2013-R2-CX-K009

Grantee EIN 94-6036494

Grantee Vendor Duns # 0471200840000

Reporting Period: July 1, 2015 – December 31, 2015

Principal Investigator: William Ristenpart, PhD Principle Investigator
UC Davis Department of Chemical Engineering

E-mail: wdristenpart@ucdavis.edu

Phone: (530) 752-8780

Grant Start Date: December 1, 2013

Grant End Date: March 31, 2016

Total Award Amount: \$150,000

William Ristenpart, PhD

February 2, 2016

Date

Quantitative Algorithm for Digital Comparison of Torn and Cut Duct Tape

William Ristenpart^{1, 2}, Ph.D., P.I.
Frederic A. Tulleners², M.A., Co-P.I.

Graduate Student Researcher
Alicia Alfter, B.S.²

This research is funded by a grant from the Office of Justice Programs, National Institute of Justice award number # 2013-R2-CX-K009. The opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect those of the Department of Justice.

1. Department of Chemical Engineering, University of California Davis,
One Shields Avenue, Davis, CA 95616.
2. Forensic Science Graduate Group, University of California Davis
1909 Galileo Court, Suite B, Davis, CA 95618

Abstract

Duct tape is often associated with criminal activity, including abductions, homicides, and the construction of explosive devices. As such, forensic scientists are asked to analyze and compare duct tape samples to assess possible link associations between duct tape found at the crime scene and the duct tape roll found with a suspect. Currently, physical end matching of duct tape is based on human judgment by visual examination, with no quantitative or statistical criteria for identification. Our research aims to minimize human contextual bias by combining digital image analysis and an objective, quantitative algorithm to assess likelihood of a match. We performed edge detection and morphological smoothing operations on high resolution images (1200 dpi) of torn duct tape edges to extract the torn edge coordinates. The coordinates of a given exemplar and a suspect sample tear were then compared by calculating the sum of square residuals (SSR) of the two sets of coordinates, yielding a single quantitative number representing the “closeness” of the match. Our analysis of 11 cohorts of 200 torn pairs, yielding 2,200 total pairs with 440,000 quantitative inter-comparisons, indicates that SSR values on the order of or less than 10^5 mm² have high probability of being a match. In 97% of all examined tears, the true match had the lowest observed SSR. The analysis also revealed, however, that non-matching samples could also yield low SSRs, with “false positive” rates ranging from 0.5% for some types of hand-torn duct tape to 62% for scissors-cut duct tape. The work presented here provides a starting point for quantitative assessment of the likelihood of physical end matching of duct tape without human contextual bias.

Contents

Abstract.....	iii
Contents.....	iv
Introduction.....	1
Figure 1: Example of Duct Tape End Matching	3
Methodology.....	6
Duct Tape Samples.....	6
Figure 2: Representative Sample Pair of Hand Torn Duct Tape	6
Table 1: Duct Tape Samples and Method Preparation.....	7
Imaging Procedure.....	7
Image Analysis Methodology.....	7
Figure 3: Overview of Morphological Image Processing.....	8
Comparison of Duct Tape Tears.....	10
Results.....	12
Figure 4: Illustration of Non-match, and Match Duct Tape Tear Comparisons.....	12
Figure 5: Representative 10 by 10 Matrix of SSR Values.....	14
Figure 6: Histogram of SSR Values for Non-match and Match Comparisons.....	15
Figure 7: Box plot of SSR Comparisons.....	16
Figure 8: Color map Matrix.....	17
Table 3: Average SSR Values for Match and Non-match Comparisons.....	18
Figure 9a: Box plot of Match and Non-match SSR Values.....	19
Figure 9b: Continuation of Fig. 9a.....	20
Table 4: Inaccuracy rate	21
Figure 10: Scatter plot of False Positive Rates versus Average SSR Value.....	23
Figure 11: Illustration of Morphological Image Processing.....	25
Figure 12: Scrim Patter differences in Tapes.....	26
Figure 13: Example of Image Analysis Artifact.....	27
Conclusion	28
Acknowledgments.....	28
References.....	29
Appendix A: Algorithm.....	30
Appendix B: Statistical Analysis.....	45
Appendix C: Histograms.....	49

Introduction

Duct tape is often found in association with criminal activity. For example, abductions, homicides, and improvised explosive devices all frequently feature duct tape, and forensic scientists are asked to compare duct tape samples and establish commonality. The standard procedure, however, involves visual and stereomicroscopic inspection of both sides of a suspected match (SWGMAT 2013). This approach is hence subjective and prone to contextual bias of the type that was criticized in the in 2009 National Research Council (NRC) report on the state of forensic science.

Several past studies of duct tape end matching (Bradley 2006, Tulleners 2010, and McCabe 2013) have recognized this problem, and used various visual assessment methods of multiple torn duct tape specimens to provide estimates of error rates for matches. Bradley *et al.* (2006) looked specifically at the process of comparing duct tape; they determined whether two or more samples could be linked to a common source by physical end matching, and measured the error rates associated with that analysis. Bradley asked four analysts to assess 50 pairs of hand torn duct tapes and 31 pairs of scissor cut tapes and asked the analysts to determine if they could match pairs or not. The analysts looked at both the backing and the adhesive sides of the duct tape. The initial results indicated 46 of 50 identified matches (four inconclusive) for hand torn end, and 25 of 31 identified matches (six inconclusive) for scissor cut end. Here, an “inconclusive” means that an end match was not identified (as opposed to being misidentified). Test sets with inconclusive results in the original administration of the sets were reevaluated by the other three analysts, who were not informed of the original results. Although no false positive errors were reported in the initial administration of the test, reevaluation of the test sets resulted in two misidentifications (false positives) of scissor cut ends, by two different analyst on different pairs of tape. Overall,

the rate of inconclusive matches and incorrect matches demonstrate the difficulty in visually assessing possible end matches.

A subsequent study by Tulleners *et al.* (2010) examined if duct tape end matches are distinguishable among a single roll of duct tape. The study was designed to obtain statistical inferences on the uniqueness of duct tape tears. Two analysts in Chan's study inter-compared 100 pairs of duct tape independently, in which 95 tape pairs matched, while 5 tape pairs did not match. The study involved a total of 10,000 possible end match inter-comparisons. Using this finite set, the analysts correctly matched the true match tape pairs, while correctly excluding the 5 mismatched pairs. Also, neither analyst reported multiple matches for a single tape specimen, suggesting each tear is unique. This work confirmed that analysts can correctly identify hand-torn end matches when there is a 95% probability that the true match is actually present within a small set of possible duct tape ends. In casework, however, there is no guarantee that a true match is present with such high probability; analysts instead face the much more challenging task of assessing the likelihood that a given unknown sample is indeed a match or not.

A study by McCabe *et al.* (2013) expanded from Bradley and Chan's experiments by including a higher number of samples to obtain statistical information on a larger data set. Three independent analysts examined 2,200 pairs of duct tape samples. Of the 2,200 pairs, half were matches, and half were not. The three analysts in McCabe's study obtained inconclusive rates ranging from 0% to as much as 23%. The analysts obtained false positive rates ranging from 0% to 8%, and false negative rates ranging from 0% to 1.5%. In the 2,200 comparisons, there were more false negatives than false positives, i.e., the analysts were more likely to say a pair was not a correct match when the pair was indeed a correct match. The mean match accuracy observed in McCabe's study ranged from 98.58% to 100% for torn tape (hand torn and Elmendorf torn), and

98.15% to 99.83% for cut tape (scissor cut and box cutter cut). The study also indicated that peer review lowered the error rate to 0% for hand torn pairs. Note that these mean accuracies did not include inconclusive, false positive, or false-negative comparisons; only the combined amount of correct matches for all researchers. Importantly, in McCabe's study the analysts compared one duct tape to another duct tape, and did not make multiple inter-comparisons with other tapes. McCabe *et al.* concluded that differences between analysts, brands, tape grades, tape color, and separation by tearing or cutting have varying contributions to misidentifications (false positive and false negative), or inconclusive results.

An example of the difficulty involved in visual assessment is provided in Fig. 1, which shows a false positive result obtained in the study by McCabe *et al.* The exemplar (at left of both

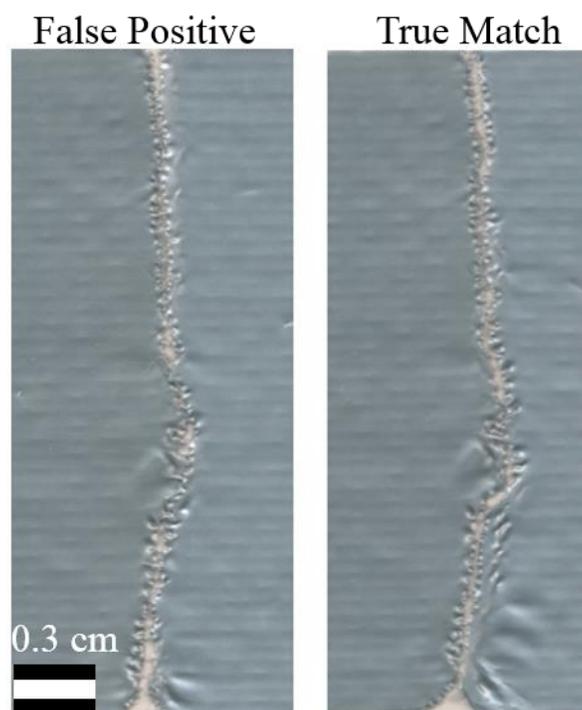


Figure 1. Example of duct tape end matching. The exemplar (at left in each pair) is identical in each image pair; the suspect samples (at right in each pair) are different. Two separate analysts in McCabe *et al.* (2013) incorrectly identified the pair at left as the true match, yielding a false positive. Scale bar is 0.3 cm.

pairs) is the same, but the suspect sample (at right) differs. Both samples “look” like a good match, and both analysts incorrectly chose the pair at left as the true match. The main issue with visual physical matching is that the conclusion of such a match is based on the subjective opinion of the examiner. The forensic science community needs a way to compare duct tape specimens quantitatively, and rigorously assess the validity of the forensic analysis.

The above-mentioned studies (Bradley 2006, Chan 2010, and McCabe 2013) used the visual observation efforts of different individuals along with the evaluation of manufacture class characteristics. None of these studies used any sort of quantitative methods for matching the duct tape samples. In other areas of forensic pattern matching analysis, such as tool mark analysis (cf. Gambino 2011), more quantitative methodologies have been explored. For example, Justino *et al.* (2006) discussed the surface profile of torn paper documents in an attempt to automate the matching of fragments. Justino’s method involved digital imaging of the paper fragments, and then application of a polygonal approximation to reduce the complexity of document boundaries and extraction of relevant features of the polygon for reconstruction. Justino *et al.* was able to reconstruct documents known to be derived from a single source, and they concluded that the performance of the algorithm dropped as the complexity (i.e., number of paper fragments) increased. Similarly, DeSmet *et al.* (2008) and Lin *et al.* (2011) studied the reconstruction of ripped-up or shredded paper documents using fragment stack analysis procedures and graph based algorithms respectively. The algorithms correctly merged the majority of shredded documents and helped to reduce the workload of a manual document reconstruction process. In these studies, however, the analysts reconstructing the documents assume most or all the pieces are readily available. In the case of duct tape end matching, in contrast, the major challenge for analysts is that the “missing piece” may or may not be located; a piece of duct tape found on a suspect might

or might not be a true match with a piece of duct tape found at a crime scene. The key point here is that while quantitative algorithms have been developed to evaluate end matches for paper documents, none have specifically addressed torn or cut duct tape.

The main goal of this work is to develop a quantitative methodology for assessing the likelihood of a duct tape match that avoids human subjectivity and contextual bias. Our approach involves quantitative image analysis, using edge detection and morphological smoothing operations to extract the coordinates of a duct tape tear edge from a high resolution digital photograph. The edge coordinates of a given exemplar and a suspect sample tear are then compared by calculating the sum of square residuals (SSR) of the two sets of coordinates. This mathematical operation yields a single quantitative number representing the “closeness” of the match. We analyzed the same large set of duct tape tears generated by McCabe *et al.* (2013), composed of 11 cohorts of 200 torn samples or a total of 2,200 samples. Our analysis of this large sample set yielded $200 \times 200 \times 11 = 440,000$ quantitative inter-comparisons. A key result from our work is that SSR values less than a critical value of approximately 1.6×10^6 pixels² (equivalent to 1.1×10^5 mm²) have high probability of being a match. In 97% of examined tears, the true match had the lowest observed SSR. The analysis also revealed, however, that non-matching samples could also yield low SSRs, with “false positive” rates ranging from 0.5% for some types of hand-torn duct tape to 62% for scissors-cut duct tape. Our results provide the first quantitative methodology for assessing likelihood of a true duct tape end match for forensic crime scene reconstruction.

Methodology

Duct Tape Samples

We used the previously generated duct tape samples previously analyzed by McCabe *et al.* (2013). McCabe *et al.* created the sample set by tearing or cutting the duct tape and placing the matching ends adjacent to one another on separate transparent acetate sheets. McCabe denoted the left side as the exemplar, and the right side as the sample; in casework the exemplar is analogous to the tape found with the victim and the sample to the roll found at a crime scene.

The duct tape sample set included two commercially available duct tape brands (Nashua and 3M) with two grades (general and professional), and two colors (black and gray). Figure 2 displays a representative black Nashua general grade duct tape pair. Four separation methods were used to generate the samples: hand torn (8 sets), an Elmendorf tear tester (1 set), scissor cut (1 set), and box cutter (1 set). The details of each set are listed in Table 1., in which each row represents a set of 200 pairs of duct tape with specific brand, grade, and method of separation. For reference the physical characteristics for each duct tape are listed in columns by vendor number, tensile strength, and thickness.

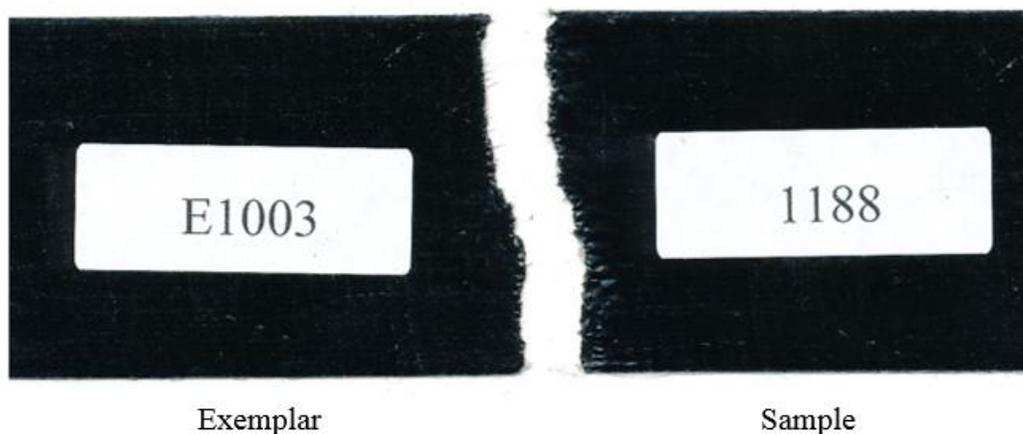


Figure 2. Representative example of a matching pair of hand torn Nashua General Black duct tape ends, each 4.5cm wide. In this set the duct tape labeled E1003 is the exemplar (representing a specimen found at a crime scene), and the duct tape labeled 1188 is the sample (representing a specimen found with a suspect).

Table 1. Duct Tape Samples and Method of Preparation

Separation Method	Grade	Color	Abbrev.	Tape Vendor Number	Tensile Strength (lb./inch)	Thickness (mils)
Hand Torn (Nashua)	General	Black	NGB	394	19	9
	General	Gray	NGG	394	19	9
	Professional	Black	NPB	398	27	11
	Professional	Gray	NPG	398	27	11
Hand Torn (3M)	General	Black	3MGB	L255	23	6.8
	General	Gray	3MGG	L255	23	6.8
	Professional	Black	3MPB	6969	32	10.7
	Professional	Gray	3MPG	6969	32	10.7
Elmendorf Scissor Box cutter	General	Gray	3MGG	L255	23	6.8
	Professional	Gray	3MPG	6969	32	10.7
	Professional	Gray	3MPG	6969	32	10.7

Imaging Procedure

We used a high resolution scanner (CanoScan 5600F) at 1200 DPI, using Photoshop version CS4, to obtain digital images of each exemplar and sample. All 4,400 duct tape ends were scanned individually, with the digital images stored as JPEGs. The entire library of images is available for interested researchers.

Image Analysis Methodology

We used Matlab to write an image analysis algorithm to extract the coordinates of the edge of the duct tape. The code is available as supplementary material; here we provide a basic overview of the algorithm methodology. Figure 3 provides a graphical overview.

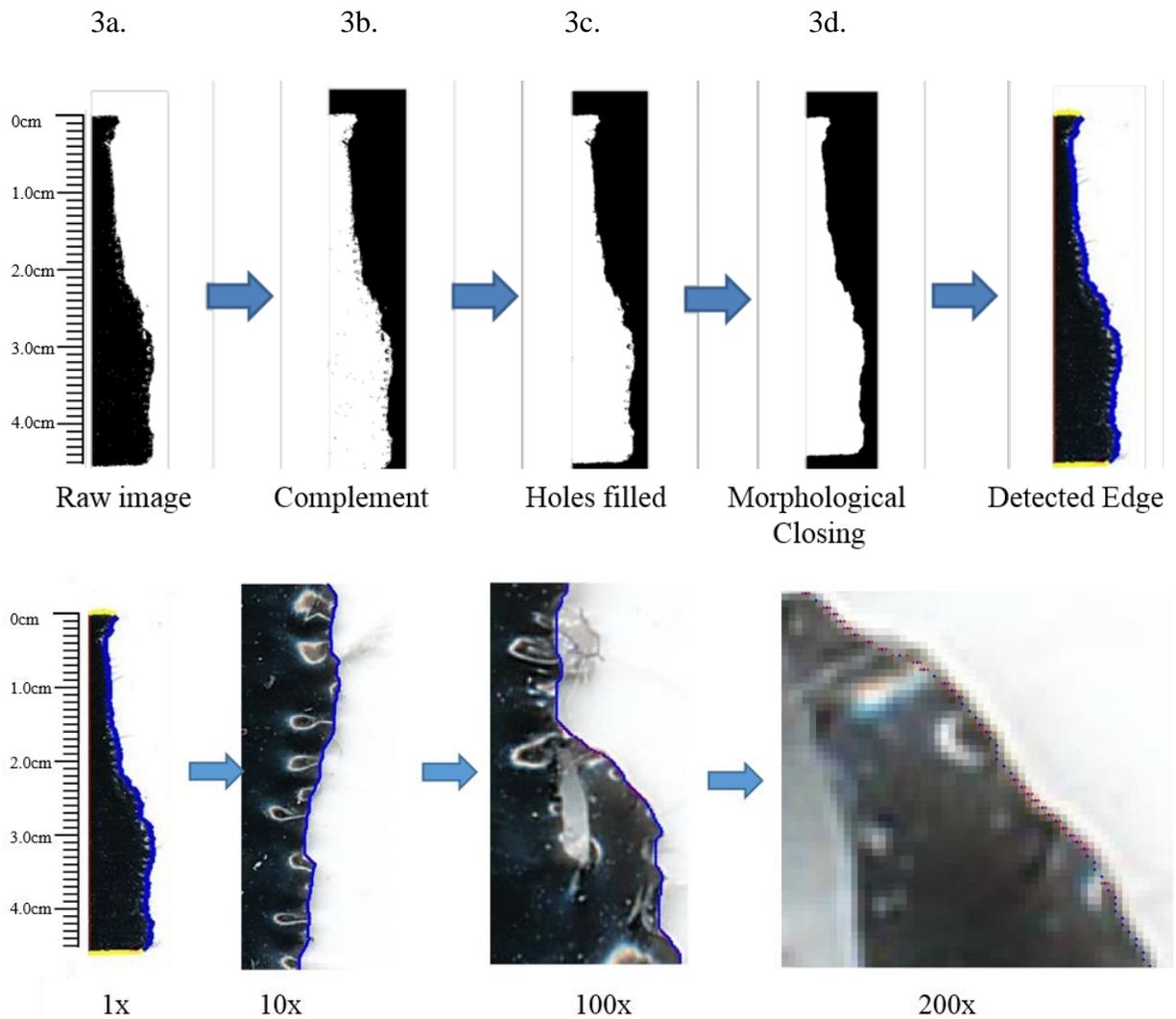


Figure 3. Overview of the morphological image processing to detect the edge of a duct tape sample (top row). The raw image is converted to binary, and then converted to the complement. Small holes in the complement image due to noise are filled in, and then morphological image processing smooths the tear edge via dilation and erosion. Standard edge detection then yields the coordinates of the tear edge, which are then averaged in the roll direction in the case of multiple detected edge values. The magnified images (bottom row) show the detected coordinates in blue, and the average of the detected coordinates in red, superimposed over the raw image.

First, the digital image is loaded into memory and converted to a binary image (Fig. 3a) via standard thresholding (cf. the *im2bw* command in Matlab). The choice of specific threshold value depends on the color of the duct tape, and average illumination of the image. We set the threshold for black duct tape at 0.5, and gray duct tape at 0.725. We determined the threshold level by inspection of several representative images and determined which threshold value yielded

binary images with the most fidelity to the original image while excluding as many loose threads as possible, since these interfered with the edge detection. Next the complement of the binary image is calculated (Fig. 3b), to identify the duct tape as the object of interest (i.e., white foreground rather than black background) The next step removes noise associated with either minor fluctuations in light reflection from the duct tape by filling in black pixels otherwise surrounded by white (cf. the *imfill* command in Matlab). Likewise, noise associated with small pieces of dirt, adhesive, or loose threads outside of the duct tape were removed based on a size threshold of 100 pixels (cf. the *bwareaopen* command). This procedure yielded a “cleaned up” image (cf. Fig 3c).

Many of the duct tape samples were not perfectly horizontal when scanned, so it was necessary to digitally rotate the images to provide a uniform coordinate system. Using the resulting cleaned up image, the top and bottom boundaries of the duct tape were then found via edge detection (excluding the region including the tear). Linear regression yielded the slopes of each boundary in the original image reference frame, and the average of the two slopes was used to determine how far the image should be rotated so that the duct tape image is perfectly horizontal (cf. *imrotate* command)

A major challenge with the image analysis was the frequent presence of loose threads protruding from the edge of the duct tape; these threads are partially pulled out from the duct tape backing during the tearing event. Because the threads are attached to the duct tape and sufficiently dark enough to survive the thresholding and cleaning operations, the edge detection procedure would incorrectly identify them as part of the duct tape edge. To remove these loose threads digitally, we used a standard morphological “closing” operation, which consists of an erosion operation followed by a dilation (cf. the *imclose* command in Matlab). This procedure effectively

removes small connected objects from a larger object, while preserving the overall shape of the object. We used a disk-shaped structuring element (of radius 15 pixels), which was found to effectively remove the majority of threads (cf. Fig. 3d and bottom row). Finally, standard edge detection using the Sobel approximation was used to extract the coordinates of the tear edge (cf. *edge* command in Matlab).

Comparison of Duct Tape Tears

The above image analysis procedure yielded a set of coordinates describing the tear edge, here denoted as (x, y) , where x is in the direction along the tear and y is the direction orthogonal to the tear (i.e., the roll direction). The central idea proposed here is that the tear coordinates of the exemplar and sample will be extremely similar if they are a true match. Since the coordinates are extracted from different images, however, a uniform reference frame is necessary. We define $x = x_0$ as the midpoint between the top and bottom edges of the tape. The average width of the duct tape is approximately 4.6 centimeters wide, and with 1,200 DPI scans, (approximately 472 pixels per centimeter) the duct tape is about 2,160 pixels wide. Because duct tape is flexible and stretchy, however, the post-tearing width for a given sample may be wider than the pre-torn width. To account for this, we exclude the very top and bottom of the tear coordinates, retaining the central 2,050 pixels around the midpoint for comparison. With the extreme edges excluded, $y = y_0$ is defined as the average of the tear y coordinates.

A final complexity is that the edge detection algorithm, even following the morphological smoothing and noise removal, can occasionally yield non-unique $y(x)$ coordinates, i.e., multiple values of y are observed for the same value of x . This occurs for example if a bit of loose thread survives the image analysis procedure, and it is curved over parallel to the tear edge. We attempted several procedures for removing this artifact, but most resulted in unacceptable loss of fidelity with

the apparent tear edge. As an approximation, therefore, we simply took the average of any non-unique y coordinates. In most cases this successfully resolved the edge, but occasionally caused problems (as discussed in more detail below).

To quantify how similar a given exemplar is to a proposed sample match, we calculated the “sum of squared residuals” (SSR) between the two sets of coordinates. Specifically, the SSR is given by

$$SSR = \sum_{i=1}^n (y_{ex,i} - y_{s,i})^2$$

where $y_{ex,i}$ is the i^{th} y coordinate of the exemplar (evaluated at x_i) and $y_{s,i}$ is the corresponding sample coordinate. (Note that the SSR here quantifies the difference between two sets of empirical data, and should not be confused with the “sum of squares due to a regression” that compares one set of data to a model.) If the two sets of edge coordinates are exactly identical, then the SSR is zero; if they are wildly dissimilar, then the SSR is large. The SSR is commonly used in linear and nonlinear regression analysis to determine “best fit” parameters of a proposed model against experimental data; in contrast, here we are using the SSR to evaluate two experimentally measured sets of values against one another. Note that, as in the case of regression analysis, a single SSR by itself holds little information; SSRs can only be judged by comparison to other SSRs. Nonetheless the key advantage of the SSR is that it yields a single number that characterizes how similar (or dissimilar) a proposed set of duct tape tears are in comparison to other sets.

Results

Representative examples of edge coordinates and their corresponding “residual plots” and SSR values are shown in Fig. 4. The top set of plots is an example of a non-match comparison.

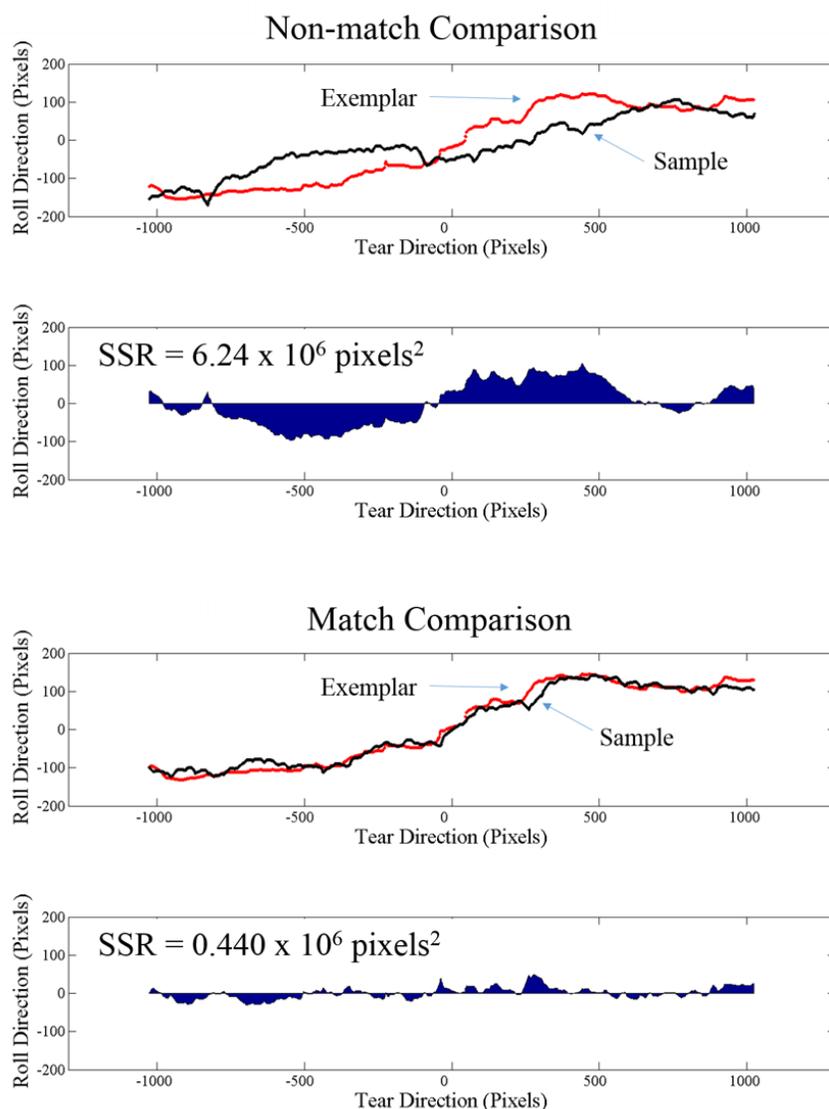


Figure 4. Representative sets of coordinates of duct tape edges and corresponding residual plots. On top, a non-matching set of edges yields a large SSR. At bottom, the correct match yields a much smaller SSR. Exemplar coordinates are red, sample coordinates are black.

By inspection, one can immediately see that the exemplar coordinates are extremely dissimilar to the sample coordinates.

The corresponding SSR value for the non-match comparison is 6.24×10^6 pixels². In contrast, the bottom set of plots is the for the true match comparison. Clearly these sets of coordinates are much more similar than the first comparison, and the corresponding SSR value for the true match comparison is 0.440×10^6 pixels² – more than an order of magnitude smaller for the non-match. From such a comparison an analyst would conclude that the bottom comparison in Fig. 4 is much more likely to be the true match than the top comparison (as is the case here).

A key advantage of the digital image analysis and SSR approach, however, is that such comparisons can be rapidly conducted for many possibly matching sets of torn duct tapes. The analysis procedure and SSR calculation was repeated for every exemplar and sample for each set of 200 exemplars and corresponding samples, yielding 40,000 comparisons. One way to visualize the results of this analysis is to examine a 200 by 200 matrix of SSR values for every comparison. Ideally, each SSR on the diagonal of such a matrix (corresponding to the true matches) will have the lowest SSR value for every row and column. An example of a representative 10 by 10 matrix is illustrated in Fig. 5. Here we see that non-matches (such as comparing exemplar 3 to sample 8) have large SSR values, while true matches (on the diagonal) have comparatively small SSRs. In fact, within this particular subset matrix, the diagonal values each have the lowest SSR for their respective rows and columns.

Tape #	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10
S1	0.305	5.725	6.584	5.481	12.980	9.576	4.031	6.346	3.966	14.498
S2	5.073	0.352	1.934	2.192	3.486	3.201	5.328	9.105	3.361	23.449
S3	5.576	1.116	0.439	4.055	2.986	2.161	5.787	13.250	4.879	26.328
S4	5.396	3.546	6.242	0.339	6.506	5.757	5.357	7.673	3.219	15.901
S5	14.727	3.282	4.095	5.307	0.285	1.523	9.618	21.202	7.937	30.802
S6	12.332	3.129	3.104	5.124	0.843	0.368	8.151	21.584	8.453	29.221
S7	4.322	5.759	7.171	5.377	9.867	7.225	0.510	10.481	3.409	10.283
S8	5.109	7.462	10.877	6.493	16.459	15.908	10.204	0.691	7.666	24.194
S9	4.210	4.342	6.273	3.360	8.233	7.180	2.989	8.071	0.519	12.124
S10	13.213	22.737	26.056	15.964	28.703	24.448	9.464	18.758	11.316	0.652

Figure 5. Representative 10 by 10 matrix of the SSRs, in millions of pixels², calculated by comparison of exemplars (on the horizontal axis) and samples (on the vertical axis) for Nashua general grade black tape. Each number corresponds to a specific comparison; for example, the SSR associated with exemplar #4 compared to sample #7 was 5.377×10^6 pixels². Note that all of the lower SSR values in this representative subset are on the diagonal (shaded gray), which corresponds to the true matches.

Inspection of the SSRs in Fig. 5 reveals that the SSRs differ wildly in magnitude: the lowest SSR in this subset was 0.285×10^6 pixels², while the largest SSR was more than two orders of magnitude larger at 30.8×10^6 pixels². Some of the non-match SSRs, however, are “close” to the true match SSRs. For example, exemplar 5 compared to sample 6 yielded an SSR of 0.843×10^6 pixels², perhaps worryingly close to the true match SSR of 0.285×10^6 pixels². The relative distributions of the SSRs for the matching and non-matching populations are illustrated in histograms in Fig. 6. Consistent with Fig. 5, the vast majority of SSR values for the true matches are much smaller than the vast majority of the non-matching SSRs. Fig. 6 also reveals, however, that the SSR values for the true matches and the non-matches both follow lognormal distributions. Most importantly from a forensics point of view, it is clear that there is some partial overlap in the distributions. These general trends – of widely separated lognormal distributions, albeit with regions of overlap – were observed in all 11 types of duct tapes examined (cf. Appendix C).

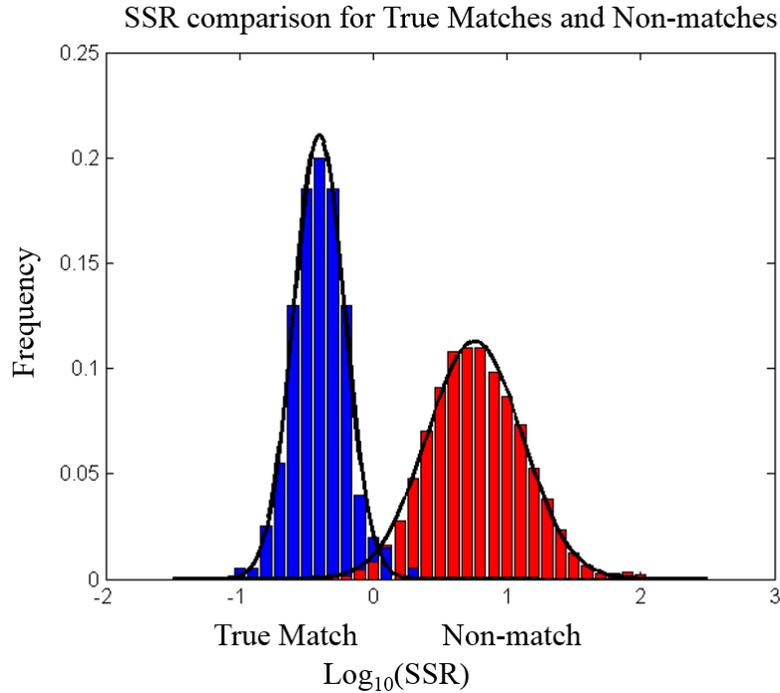


Figure 6. Histogram of the SSR values, in logarithmic scale based on 10^6 pixels², for true match (blue) and non-match (red) comparisons for hand-torn NGB tapes. There are 200 true match and 39,800 non-match comparisons, yielding 40,000 comparisons in total. Solid lines are Gaussian fits. The overlap of the blue and red is the area of interest. This overlap represents the SSR values that are similar in value for both true match and non-match.

The histograms in Fig. 6 compare the distributions of true match SSRs versus non-match SSRs, but they don't reveal whether or not “false positives” occurred, i.e., whether or not a non-matching SSR was actually lower than the specific corresponding true match. To check this more directly, we normalized the SSR values for each comparison by taking the logarithm of the observed SSR divided by the true match SSR value, i.e.,

$$\text{Normalized SSR} = \log_{10} \frac{\text{Observed SSR}}{\text{True match SSR}}$$

With this definition, normalized SSRs less than zero (because $\log(1) = 0$) indicates that a false positive occurred specifically for that exemplar. A representative set of normalized SSRs is shown in Fig. 7 as boxplots.

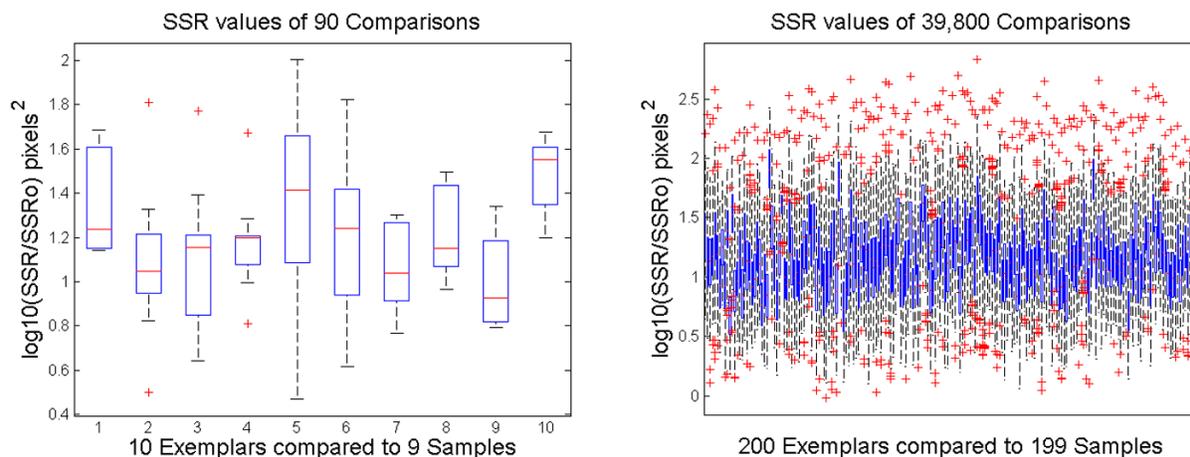


Figure 7. Box plot represents a comparison of SSR values among comparisons. The first figure is an example of 10 exemplars compared to 9 samples, making 90 comparisons. The second figure is an example of 200 exemplars compared to 199 Samples. The central red mark is the median, the edges of the box are the 25th and 75th percentiles, and the whiskers extend to the most extreme data points not considered outliers. The outliers are plotted individually with red plus signs.

The left-hand box plot in Fig. 7 shows the same representative subset of 10 exemplars as in Fig. 5, each normalized to the true match SSR and compared to 9 samples (excluding the true match for the normalized SSR). In this subset, none of the SSR values are below zero, indicating the true match always was the lowest SSR. Inspection of the whole set of 200 exemplars, however, shows that sometimes there are negative normalized SSR values (Fig 7, right-hand plot). In this set of 200 exemplars (each compared to 199 samples), there was exactly 1 comparison that received a lower SSR value than the true match SSR, i.e., there was one false positive.

Another way of showing the same information is in a colored matrix, analogous to Fig. 7, but with different colors representing numeric values (Fig. 8). The 200 exemplars are arranged horizontally and the samples vertically. The values lowest in the matrix are colored dark red. The SSR values in the critical zone are colored a lighter red, and the values in-between are colored yellow to cyan. The SSR values much higher than the true match SSR are colored blue to navy.

From this colorized matrix, it is understandable that both high and low SSR values are not randomly distributed, but that certain exemplars or samples tend to repeatedly yield extreme values (as indicated by the lines). We can also see from the matrix that out of 40,000 comparisons, most SSR values fall between 0.5 and 2 million pixels squared, consistent with Fig. 8.

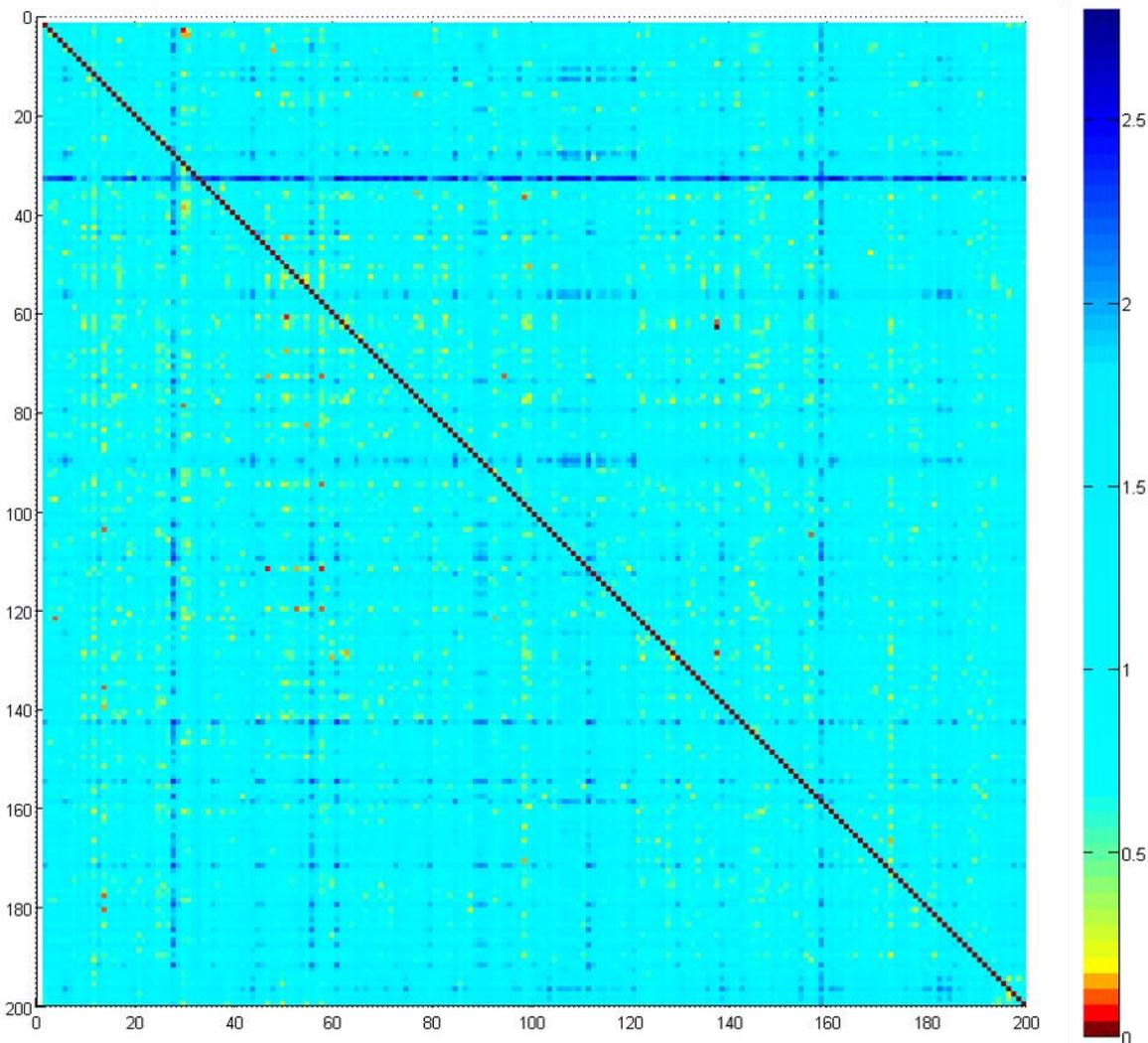


Figure 8. Color map matrix for hand torn Nashua general grade black duct tape. This is a 200 by 200 matrix with the exemplars arranged horizontally, and the samples vertically. The dark red color is the true match SSR values, the gradient above red are SSR values increasing larger than the true match SSR value.

The preceding figures illustrated the results for one representative type of hand torn duct tape. We repeated the procedure for all 11 sets of duct tape. Appendix C contains the raw SSR histograms for each set, and the mean values for each type are tabulated in Table 3. Here we summarize the distributions in Fig. 9, which compares the distributions of the true match and non-match SSRs as a boxplot. For each color box, the left box is based on the SSR of the true matches, and the right box is based on the SSR of non-matches. The results clearly show that the SSR for the true matches is consistently about an order of magnitude smaller, on average, than for non-matches – regardless of duct tape type or how it was torn or cut. At the same time, however, it is clear that there is always a tail end of the non-match distribution where SSR values comparable to those for the true match are obtained – again, regardless of duct tape type or how it was cut or torn. In other words, the results indicate that it is possible for false positives to occur no matter what type of duct tape is used or how it is cut or separated. Under no conditions did we see a complete separation of SSR values between true matches and non-matches. However, we do see that the SSR values for hand-torn true matches were invariably less than 1.6×10^6 pixels². This result suggests that a larger SSR value obtained for an unknown sample comparison in forensic case work could be safely identified as a non-match.

Table 3. SSR values for true match and non-match comparisons (10⁶ pixel²)

Separation Method	Type of Tape	Average of true matches	Average of non-matches	Standard deviation of true matches	Standard deviation of non-matches
Hand Torn (Nashua)	NGB	0.4405	8.6105	0.2314	10.2146
	NGG	0.6950	11.4013	0.3823	12.6619
	NPB	0.4551	7.9443	0.3147	10.5715
	NPG	0.5139	10.0471	0.2614	9.7296
Hand Torn (3M)	3MGB	0.6558	9.0921	2.6301	27.0126
	3MGG	0.5424	6.3541	0.5186	10.4013
	3MPB	0.1137	4.7510	0.0454	4.2027
	3MPG	0.3265	5.8717	0.3012	12.4600
Elmendorf torn (3M)	3MGG	0.7129	10.8283	0.5281	26.4408
Scissor cut (3M)	3MPG	0.1587	2.6362	0.2180	3.6299
Box cut (3M)	3MPG	0.3157	21.9628	0.5396	30.2521

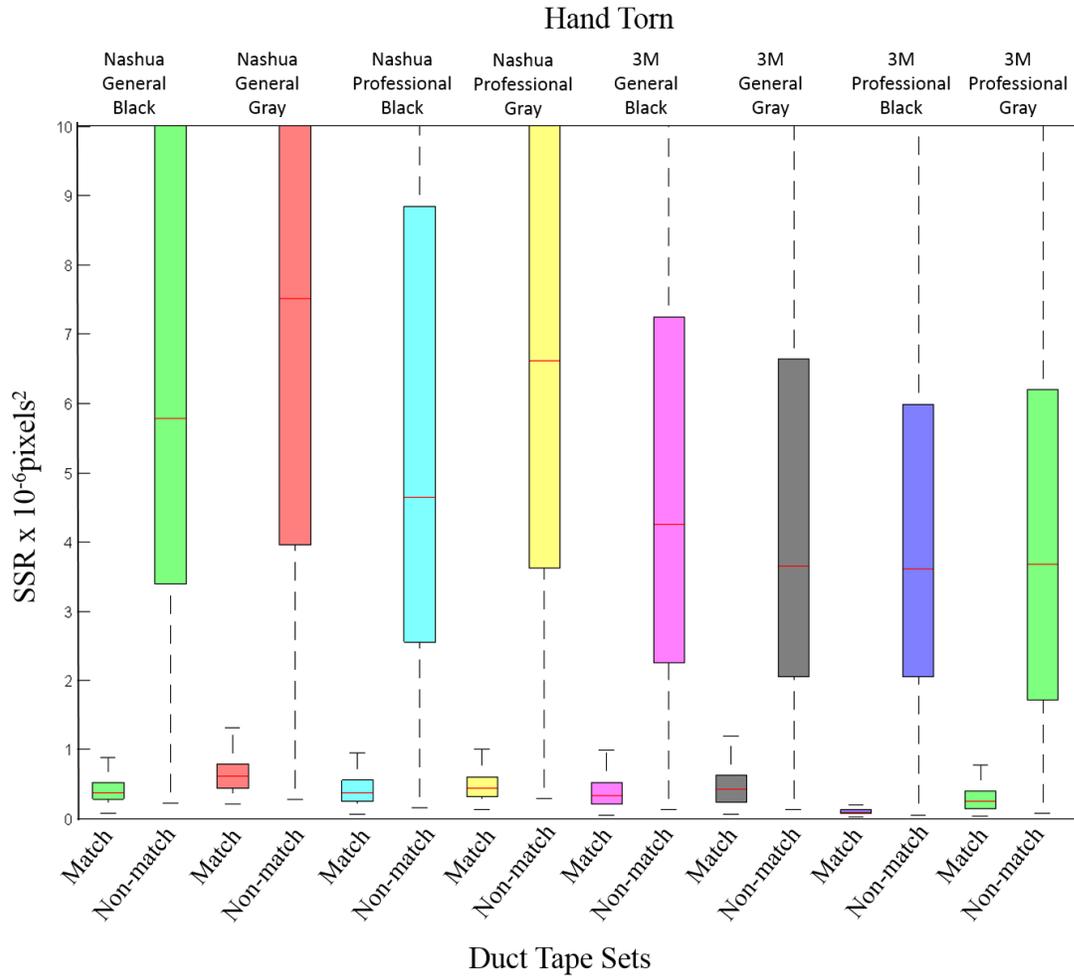


Figure 9a. Box plot of the SSR of true matches and SSR of non-matches for hand torn tapes. The true matches are the duct tape comparisons that we know to be the true correct match (left box for each set, based on $N=200$). The non-match comparisons are the duct tape comparisons for non-matches (right box for each set, based on $N=39,800$). The central red line in each box is the median, the edges of the box are the 25th and 75th percentiles, and the whiskers extend to the most extreme data points not considered outliers. The outliers and tops of some of the non-match boxes extend to values as high as 13×10^6 pixels² (not shown here for clarity).

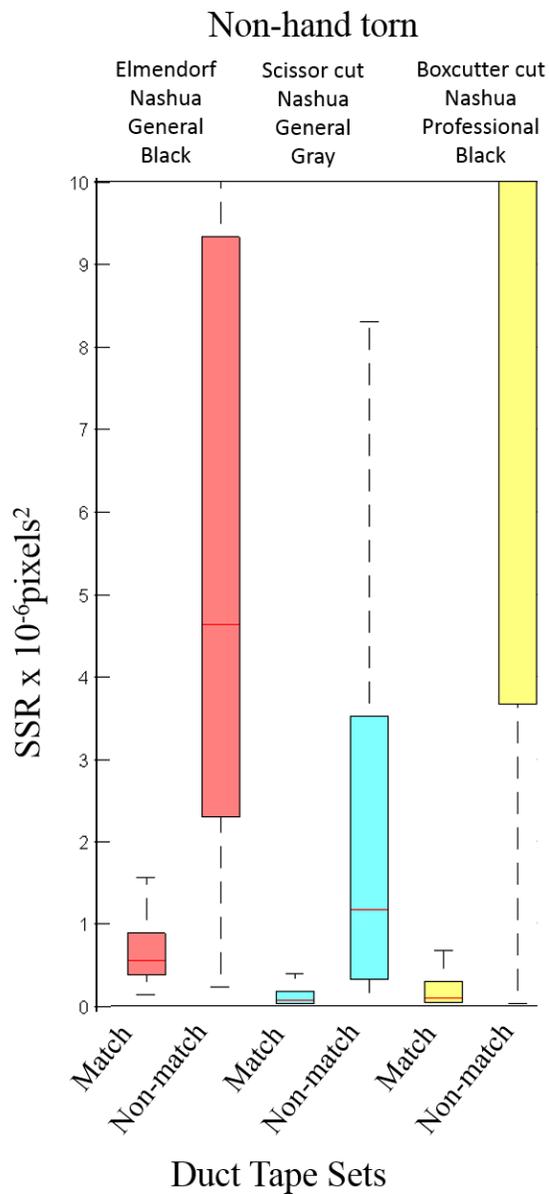


Figure 9b. Continuation of Fig. 9a.

Table 4. Inaccuracy rates

Separation Method	Abbrev.	Number in the Critical Zone (per 39,800)	Number of lower SSR than true (per 39,800)	Algorithm Inaccuracy Rate (%)	Number of lower SSR than true (per 199)	False Positive Rate (%)
Hand Torn (Nashua)	NGB	1	1	0.0025	1	0.5
	NGG	152	117	0.2925	37	18.6
	NPB	105	81	0.2025	15	7.5
	NPG	45	34	0.0850	14	7.0
Hand Torn (3M)	3MGB	261	194	0.4850	35	17.5
	3MGG	1092	720	1.8000	59	29.5
	3MPB	7	5	0.0125	3	1.5
	3MPG	795	528	1.3200	54	27.0
Elmendorf	3MGG	1237	848	2.1200	79	39.5
Scissor	3MPG	2087	1255	3.1375	123	61.5
Box cutter	3MPG	38	24	0.0600	33	16.5

Given that the tail end of the distribution of non-matches overlaps with the distribution of true matches, a key question is how often “false positives” occurred in the sense that a non-match SSR for a particular exemplar was lower than for the corresponding true match. There are two ways of addressing this question: by comparing the SSR values over all 40,000 inter-comparisons within a set, or the more applicable question of whether any false positives occurred within the intra-comparisons (1-to-1 comparisons) in a set of 200. Table 4 lists the number “near false positives” in the “critical zone,” which we defined here as SSR values within a factor of 2 of the true match SSR, and the number of “false positives,” defined as SSR values actually lower than the true match SSR, for both the inter-comparisons (per 39,800) and the intra-comparisons (per 199); the “algorithm inaccuracy rate” is the number of such false positives per 400 samples. The data in the table reveal a couple key trends. First, the inter-comparison inaccuracy rates indicate that the likelihood of any SSR being lower than the observed SSR is small, with values ranging from 0.0025% up to 3.1% at most. In other words, somewhere between 97 to 99.9% of measured SSRs were larger for non-matches than any true matches. Simultaneously, however, the second

key point is that there still can be a significant number of false positives within that small fraction of SSRs. The false positive rates within the intra-comparisons (based on 199) displays large variations between types of duct tape and type of tear, ranging from 0.5% for hand-torn Nashua general grade black, to as high as 61.5% for scissor-cut 3M professional grade gray. In other words, of 200 scissor-cut exemplars, 123 of them had at least one non-matching SSR that was lower than the correct match. The hand-torn tapes, on average, yielded less false positives than the machine or blade cut tapes. The average false positive rate was about 14% for the hand-torn (8 types), but about 39% for the machine or blade cut tapes (3 types).

Discussion

The results described above reveal a very promising feature: true matching SSR values were always below a specific critical value, and the vast majority of non-matching SSR values were an order of magnitude larger. This result points to a quantitative method of “ruling out” as matching an unknown duct tape sample pair based on a sufficiently large SSR.

The results also showed, however, that it is possible for the detected edge coordinates of two different tears to be extremely similar – so similar, in fact, that the SSR for a non-matching pair could be even smaller than for the true matching pair. This quantitative finding is analogous to the qualitative challenge illustrated by Fig. 1: the human analysts occasionally made incorrect false positive identifications, and the quantitative algorithm also yielded false positives based on lower SSRs. A key question is: why?

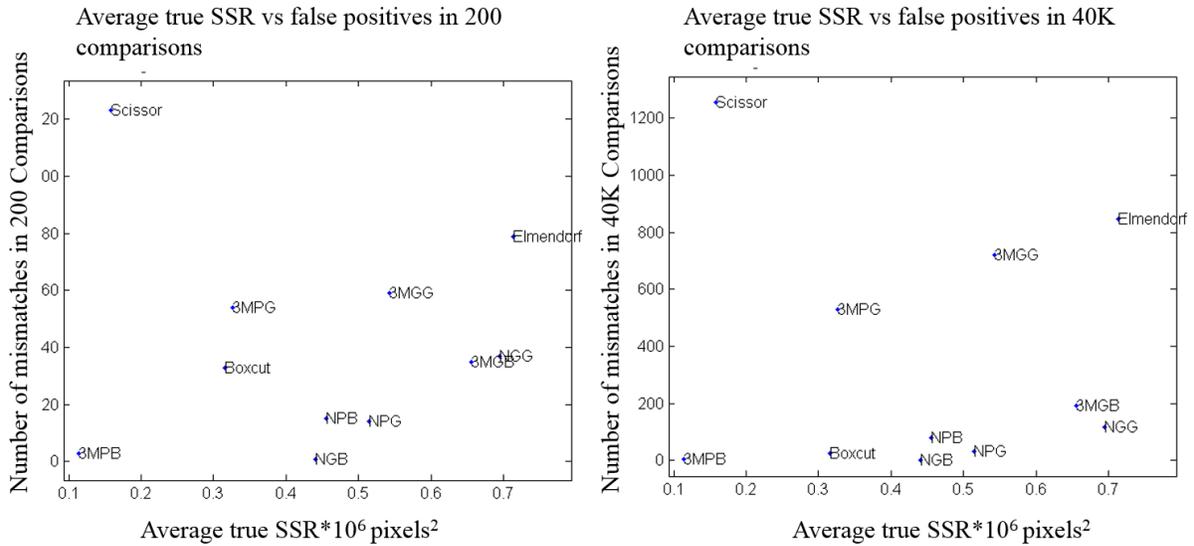


Figure 10. Scatter plot of the false positive rates versus average true SSR for each set of duct tapes, for both intra- and inter-comparisons.

Given the large variations in SSR false positive rates between types of tape and tears, a natural question is whether the error rate was correlated with the average SSR value for the true matches. Fig. 10 shows the false positive rate for intracomparisons and intercomparisons versus the average true SSR value, for all 11 types of duct tape tested. No clear correlation is observed; the scissor-cut tape for example had the highest error rate, and some of the lowest SSR values (which tends to occur if the cuts are very straight). Similar tests for correlations with other physical properties of the tapes (cf. Table 1) likewise yielded no discernible correlations (data not shown).

The duct tape samples used in this study consisted of two brands, with each brand having four different physical properties: color, thickness, tensile strength, and scrim pattern. In the following discussion of error rates, the analyst must keep in mind that to make a judgment based strictly on matching surface contours the analyst may conclude the mismatched tapes with SSR values below the true SSR value to be inconclusive. An inconclusive result for mismatched tapes means that the matching patterns are so similar that an analyst cannot make a conclusive response

based strictly on automated surface matching criteria with no other input. The algorithm does not include the broken threads, therefore significant reduction in the error rate may be made.

We looked at the number of false positives in regards to black and gray tapes. The gray tapes had more false positives than the black tapes, except for hand torn Nashua professional gray duct tape set. With the same brand and grade, and the only difference being the color, we set out to determine why the gray tapes have more mismatches. To determine this we performed a sensitivity study with respect to the threshold parameter of the algorithm. We originally set the threshold to 0.725 for gray tapes, and 0.500 for black tapes. After analyzing more duct tapes at different thresholds, we determined that the number of mismatches did change with different levels of threshold. Thresholds that were too high led to higher numbers of false positives, because the comparison points included disturbances like strings and smudges; thresholds that were too low also yielded higher numbers of false positives, because the comparison points included other disturbances like bright reflections on the duct tape image. The gray tape also had more interference, because the adhesive (glue) is similar in color. Any delamination of the tape or adhesive smudges from the adhesive interferes with the threshold image manipulations. The black tape is more distinguishing from the adhesive so the edge detection is closer to the true edge than

the gray tape. Figure 11 displays representative examples of the morphological image processing of black and gray tapes with different threshold values. Future analyses should note the importance of threshold value on the edge coordinates.

The Nashua brand duct tapes yielded a lower overall error rate than the 3M tapes. Nashua general grade tapes with black or gray colors had inaccuracy rates ranging respectively from 0.0025% to 0.29% for 40,000 intercomparisons, and 0.5% to 18.50% false positive rate for 200 intracomparisons. Nashua professional grade tapes with black and gray colors had inaccuracy rates ranging respectively from 0.085% to 0.20% for 40,000 comparisons, and 7.0% to 7.5% false positive rate for 200 comparisons. One possible reason for the error rate difference between the general and profession grade tapes may be because greater tape distortion and stretching occurs for the stronger professional tape; further tests are necessary to corroborate this hypothesis.

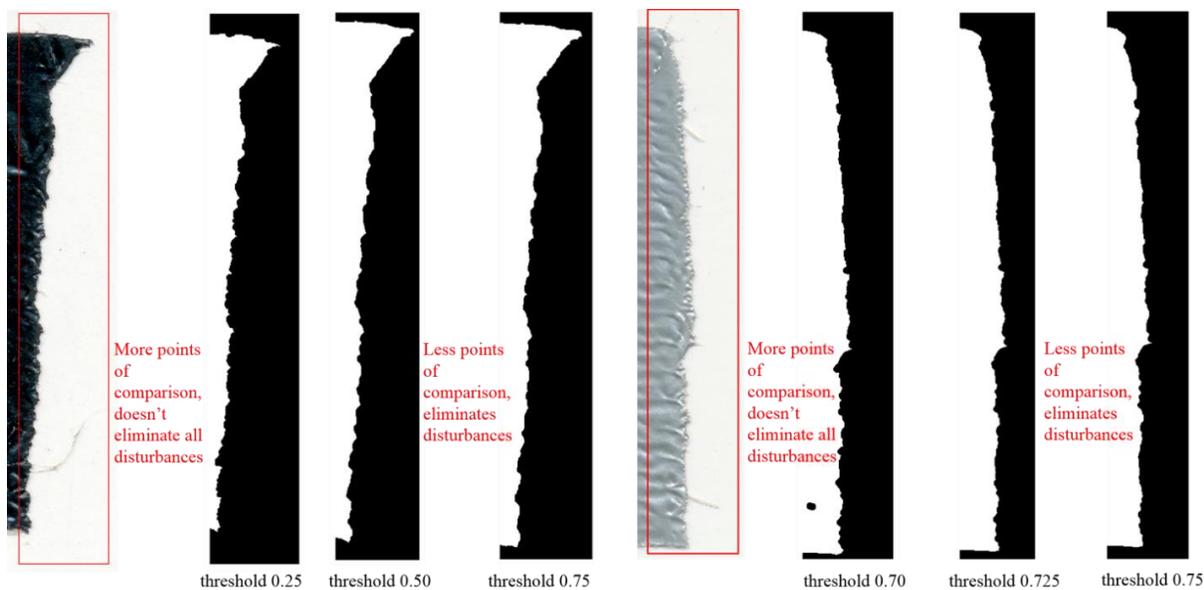


Figure 11. This illustrates the morphological image processing of black versus gray tapes. Both tapes are 3M General grade tape. The black tape shows three threshold levels; 0.250, 0.500, and 0.750. The gray tape shows three threshold levels; 0.700, 0.725, and 0.750. Lower level of threshold has more points of comparison, but doesn't eliminate all disturbances. Higher level of threshold has fewer points of comparison, but does eliminate most of the disturbances.

Another difference between the tapes is the scrim pattern (Fig. 12). We emphasize that our image analysis algorithm looked solely at the detected edge coordinates, and completely ignored the threads within the scrim. Portions of the tapes in these two sets were reviewed by looking at the scrim pattern on the back of the tape, as displayed in Fig. 12. While the tapes are listed as 3M professional black and gray with the same vendor number, in fact all the types of 3M duct tape have different scrim patterns. We manually evaluated the threads by removing the adhesive with hexane, and gently brushing the adhesive with a cotton swab to view the scrim pattern. In reviewing the scrim patterns, the black tape with the 1.5% false positive rate appears to have a much tighter weave than the gray tape with 27% false positive rate. As to why this difference in construction occurs is at this time unknown. The reason may be a case of mislabeled tape, or a change in vendors as most tapes appear to be sourced from China. In other words, some of the variation in error rates shown in Table 4 might be due to variations in the mechanical properties due to different fabrication procedures.

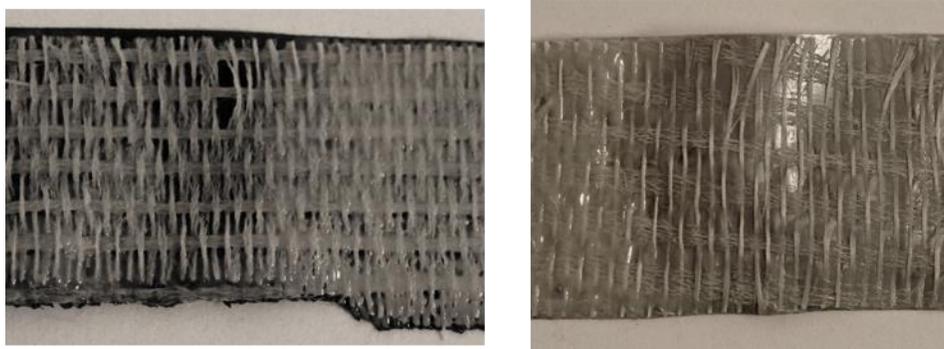


Figure 12. The images above display the scrim pattern differences between 3M professional black and gray tapes.

Another big challenge in our methodology was the existence of image analysis artifacts caused by loose fibers, creases in the tape, smudges, and/or scrim strings, all of which could interfere with accurate edge detection. These artifacts are not part of the duct tape edge, and in some cases are not even readily visible to the naked eye but are nonetheless detected within the high resolution scan. Fig. 13 shows one illustrative example of a loose thread that caused an artefactual ‘bulge’ in the detected edge. Our algorithm was unable to correct for all such artifacts, and given the large number of images in the data set (4,400) it was not feasible to manually “Photoshop” the artifacts out of the images. Future analyses will benefit from such work, or from more sophisticated image analysis algorithms that automatically correct such artifacts.

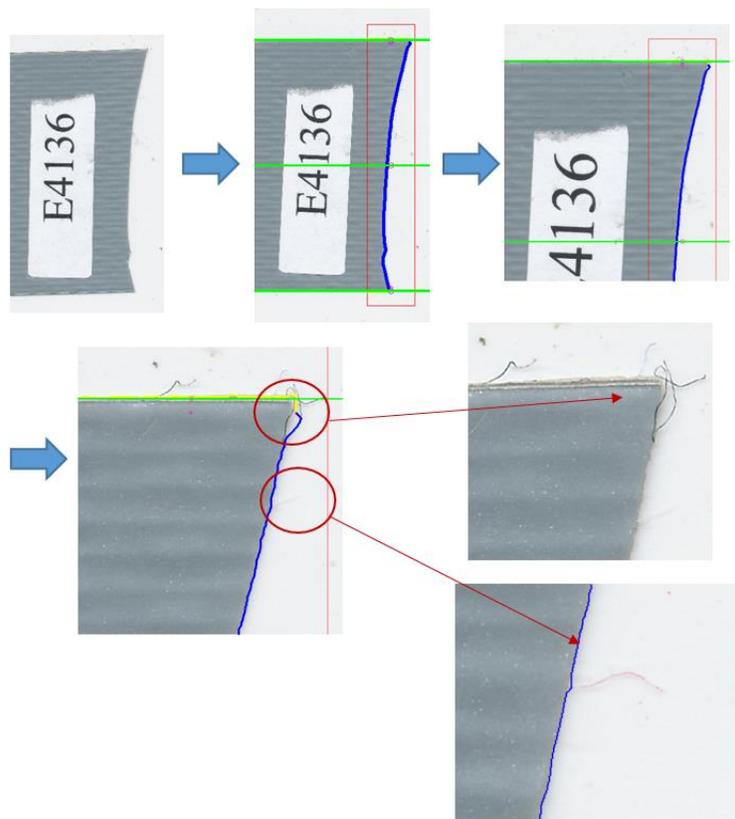


Figure 13. Example of an image analysis artifact that gives rise to an artefactually high true SSR value. Top left image shows the entire tear; the images at top right and bottom show magnification of the top right corner, which contains a thread barely visible to the naked eye.

Conclusions

We analyzed 11 sets of duct tape with 200 pairs of duct tape in each set of varying separation methods, brands, grades, and colors. Of the 11 sets, analysts tore 8 sets by hand, one set by Elmendorf tear tester, one set cut with scissors, and one set cut with a box cutter. Within each set of 200 pairs of duct tape, we made 40,000 inter-comparisons for a total of 440,000 comparisons. Our results indicate true matches do indeed have lower SSR values on average, but that false positives from non-matching samples readily occur with a non-negligible frequency. The highest false positive rate occurred with scissor cut tapes, suggesting that great caution should be exercised when performing tape end comparisons with apparently scissor cut tapes in casework.

An important caveat is that the methodology presented here only examined one side of the duct tape. Improved algorithms would also examine the underlying scrim and thread patterns. Likewise, not all artifacts could be removed by the morphological processing, so improved algorithms are necessary to detect and remove such artifacts.

The results presented here are useful to the forensic science community, because they provide the first quantitative methodology for end-matching of duct tape tear patterns. Instead of attempting to perform the end matching visually, where human error may affect the results, this approach offers a path toward quantitative assessments free of possible contextual bias.

Acknowledgments

The authors wish to thank Jessica Mally, Tiffany Coffin, and Neil Willits for their assistance in preparing the previous work; Jessica Mally for the research prototype; Tiffany Coffin for scanning duct tape samples; Neil Willits for statistics work with the SSR matrices.

The authors also would like to thank the Office of Justice Programs, National Institute of Justice for financial support with the award # 2013-R2-CX-K009.

References

- SWGMAAT: Guideline for Assessing Physical Characteristics in Forensic Tape Examinations. (2013, October). Retrieved February 02, 2016, from <http://www.swgmat.org/#!tape/c1w2c>
- Bradley, MJ, *et al.*, *A validation study for duct tape end matches*. J. Forensic Sci., 2006. **51**(3): p. 504-8.
- Tulleners, F. and Chan, K., *The statistical evaluation of torn duct tape fracture match as physical evidence*, AAFS Meeting, February 22-27, 2010 Abstract A157 p. 129
- De Smet, P., *Reconstruction of ripped-up documents using fragment stack analysis procedures*. Forensic Science International, 176, (2008) 124-136
- Gambino, C. et al., *Forensic Surface Metrology: Tool Mark Evidence*, Scanning 2011. **33**, 5, 272-278
- Lin, HY, *Reconstruction of shredded document based on image feature matching*, Expert Systems with Applications 39 (2012) 3324-3332
- Justino, E. Oliveira, L., Freitas, C. *Reconstructing shred documents through feature matching*. Forensic Science International, 160 (2006) 140-147
- McCabe, K. R., Tulleners, F. A., Braun, J. V., Currie, G. and Gorecho, E. N. (2013), *A Quantitative Analysis of Torn and Cut Duct Tape Physical End Matching*. Journal of Forensic Sciences, 58: S34–S42.
- NRC, *Strengthening Forensic Science in the United States: A Path Forward*, National Research Council, National Academic Press, 2009.

Appendix A

Duct Tape Code 1: Script to open data set

```
% DuctTapeScript_1.m
% by W. D. Ristenpart and Alicia Alfter
% Jan 2014 / June 2014
% Examines the edges of torn duct tape and compares the residuals
% Instructions:
% The program first asks you to load the exemplar, and then you will draw
% three boxes: one along the top (top) edge, one along the bottom (bottom)
edge,
% and one around the actual tear itself. Double click on each box when you
% are satisfied with its size and position. Then, repeat this process for
% as many test samples you have.

% Each duct tape is analyzed just once and an image specific .mat file is
saved.

% first, start with a blank slate
clc;
clear;
close all;

% important: set how many you want to analyze
num_exemplars = 200;
num_samples = num_exemplars; % the code now assumes there are equal numbers

% initialize empty data structures
exemplar_data = struct('FileName', [], 'Xcoords', [], 'Ycoords', []);
sample_data = struct('FileName', [], 'Xcoords', [], 'Ycoords', []);

% get the filenames of the exemplars
[ExFileNames, ExPathName] = uigetfile('*.jpg', ['Select all
', int2str(num_exemplars), ' exemplars '], 'multiselect', 'on');

% get the filenames of the samples
[SampFileNames, SampPathName] = uigetfile('*.jpg', ['Select all
', int2str(num_samples), ' samples '], 'multiselect', 'on');

%% analyze the exemplar image(s)
for j = 1:num_exemplars
    [smooth_xcoords, smooth_ycoords, mean_Ex_smooth, stddev] =
DuctTapeAnalyze_2(ExPathName, ExFileNames{j}); %x and y coordinates of tear
    exemplar_data(j).FileName = ExFileNames{j};
    exemplar_data(j).Xsmooth = smooth_xcoords;
    exemplar_data(j).Ysmooth = smooth_ycoords;
    exemplar_data(j).mean_xsmooth = mean_Ex_smooth;
    exemplar_data(j).stddev = stddev;
end
% analyze the sample image(s)
for j = 1:num_samples
    [smooth_xcoords, smooth_ycoords, mean_Sam_smooth, stddev] =
DuctTapeAnalyze_2(SampPathName, SampFileNames{j}); %x and y coordinates of tear
    sample_data(j).FileName = SampFileNames{j};
```

```

sample_data(j).Xsmooth = smooth_xcoords;
sample_data(j).Ysmooth = smooth_ycoords;
sample_data(j).mean_xsmooth = mean_Sam_smooth;
sample_data(j).stddev = stddev;
end
%% Compare residuals
%[ finalSSE_matrix ] = DuctTapeComparison_Plot3()
 [ finalSSE_matrix ] = DuctTapeComparison_3();

```

Duct Tape Code 2 Function in Code 1: Analyze data set

```

function [smooth_xcoords,smooth_ycoords,mean_xsmooth,stddev] =
DuctTapeAnalyze_2(PathName,FileName)
%duct_tape_analyze_ver1.m
% By W. D. Ristenpart and Alicia Alfter
% March 10, 2011 / Jan 2012
% This function rotates the image after selecting top and bottom
% boundaries. Then the tear boundary is selected and the x and y
% coordinates are plotted along the tear (shown in red, 98% tear coorinates
% detected). The truncated
% portion of the tape is shown in yellow (2%).

I = imread([PathName,FileName]);
thold = 0.725; % thold is the threshold, adjust it for black (0.50) and
silver (0.725)
minsize = 100; % minsize is the smallest number of pixels, adjust it for
black (100) and silver ()
edge_thold = 0.05; % edge of top and bottom boundary threshold.
ythr = .98;% y threshold is region of x and y coordinates detected on the
tear boundary (red points)
% y threshold is 98% meaning it detects the y coordinates in the red portion
of the tape, 2% yellow is trunkated.
close all
figure('Color','w');
imshow(I), hold on,
%% TOP BOUNDARY
title('Enclose top boundary in a box over the horizontal edge,double click
inside boundary,press enter,and wait');
[Inrth, recttop] = imcrop;%imcrop fuction creates an interactive image
%cropping tool. double click on the tool to crop the duct tape image
line.RECT is a 4-element vector with the form [XMIN YMIN WIDTH HEIGHT];
bwtop =
edge(bwareaopen(imfill(imcomplement(im2bw(Inrth,thold)), 'holes'),minsize), 'so
bel',edge_thold);%morphological operations: dilation and erosion
%bwtop is the line detection variable, it draws the line on the cropped
%duct tape image.
[r,c]= find(bwtop); % returns the row and column instead of linear indices in
bwtop
pf = polyfit(c,r,1); %Finds the coefficients of the line y=mx+b
m = pf(1); % m is the slope of the top boundry
b = pf(2); %b is the y-intercept of top boundry
imsize = length(I(1,:));%this is the length of the image
badj = b + recttop(2) - m*recttop(1);%b intercept of duct tape adjusted to
the image
line([1 imsize],[badj,m*imsize+badj], 'Color','g','LineWidth',2)%blue line of
top boundry

```

```

angle1 = (atand(m)); % angle of the top boundary
%% BOTTOM BOUNDARY
title('Enclose bottom boundary in a box over the horizontal edge,double click
inside boundary,press enter,and wait');
[Ibottom, rectbottom] = imcrop;
bwbottom =
edge(bwareaopen(imfill(imcomplement(im2bw(Ibottom,thold)), 'holes'),minsize), '
sobel',edge_thold);%morphological operations: dilation and erosion
[r,c]= find(bwbottom);
pf = polyfit(c,r,1);
ms = pf(1);% slope of bottom boundary
bs = pf(2);%y intercept of the bottom boundary
badjs = bs + rectbottom(2) - ms*rectbottom(1);%b intercept of duct tape
adjusted to the image
line([1 imsize],[badjs,ms*imsize+badjs], 'Color','g','LineWidth',2)%blue line
of slope in imcrop rectangle
angle2 = (atand(ms));% angle of bottom boundary
%% find center of duct tape
bcenter = (badj + badjs)/2;
mcenter = (m + ms)/2;
line([1
imsize],[bcenter,mcenter*imsize+bcenter], 'Color','g','LineWidth',2)%blue line
of center
%% ROTATE IMAGE:Average the top and bottom angles
angle = [angle1 angle2];
angle = mean(angle);
Irot = imrotate(I,angle,'bilinear','crop');
imshow(Irot), hold on
%% blue lines with angle rotation
baftertop = badj + (imsize/2)*(tand(angle1)-(tand(angle1-angle)));
bafterbottom = badjs + (imsize/2)*(tand(angle2)-(tand(angle2-angle)));
baftercenter = (baftertop+bafterbottom)/2;
maftertop =(tand(angle1-angle));
mafterbottom = (tand(angle2-angle));
maftercenter = (maftertop + mafterbottom)/2;

line([1 imsize],[baftertop,maftertop*imsize +
baftertop], 'Color','g','LineWidth',2)
line([1 imsize],[bafterbottom,mafterbottom*imsize +
bafterbottom], 'Color','g','LineWidth',2)
line([1 imsize],[baftercenter,maftercenter*imsize +
baftercenter], 'Color','g','LineWidth',2)
%% TEAR BOUNDARY
title({'Enclose tear boundary in a box over the vertical edge,double click
inside boundary,press enter,and wait.'...
'Exemplar tear region on right side of image, Sample tear region on left side
of image.'});
[Itear, recttear] = imcrop(Irot);
rectangle('Position',recttear,'EdgeColor','r');%red box for cropped boundary
area

[y,x] = DuctTapeMorphological_2(thold,Itear,minsize);%This function reads the
rotated image and fills in the spaces or holes
% that aren't needed for edge detection, and outputs the coordinates of
pixels on edge of duct tape image.

```

```

plot(x+recttear(1)-1,y+recttear(2)-1, '.y')%adds yellow line along the tear in
the cropped image
plot(recttear(1)+recttear(3)/2,recttear(2)+recttear(4)/2, 'oy')%yellow circle
in the center tear edge of duct tape, done by taking the average
% of the top and bottom boundary.

xtear = recttear(1)+recttear(3)/2;%xmin and half of width
ytop = maftertop*recttear(1)+ baftertop;%ymin of tear boundary
ybottom = mafterbottom*recttear(1)+ bafterbottom;%ymax of tear boundary
ycenter = ytop+(ybottom-ytop)/2;
plot([xtear xtear xtear],[ytop ybottom ycenter], 'om')%magenta circles at edge
of yellow line (corners of tape) and center

ymin = ycenter-(ycenter-ytop)*ythr;%bottom of tear boundary
ymax = ycenter+(ybottom-ycenter)*ythr;%top of tear boundary
plot([xtear xtear],[ymin ymax], 'pm')%magenta pentagram at the start and end
of the red line (comparison region)

ypick = y(y+recttear(2)>ymin & y+recttear(2)<ymax);%ypick is the y
coordinates between the ignored corners. raw data
xpick = x(y+recttear(2)>ymin & y+recttear(2)<ymax);%xpick is the y
coordinates between the ignored corners. raw data
plot(xpick + recttear(1)-1, ypick + recttear(2)-1, '.r')% red points for x and
y coordinates are plotted

xcoords = round(xpick +recttear(1)-xtear); %x is shifted to be centered in x
direction. raw data shifted
ycoords = round(ypick +recttear(2)-ycenter); %y is shifted to be centered in
y direction. raw data shifted

%% Smooth the x and y coordinates please.
smooth_ycoords=unique(ypick);%smooths the raw coords in y direction (ypick)
smooth_xcoords=zeros(1,length(smooth_ycoords));
for j=1:length(smooth_ycoords)
    smooth_xcoords(j)=mean(xpick(ypick == smooth_ycoords(j)));
end
smooth_xcoords = smooth_xcoords';
%smooths the raw coords in x direction (xpick) that correspond to the
%smoothed y coords (ypick)
plot(smooth_xcoords + recttear(1)-1, smooth_ycoords + recttear(2)-1, '.b')
% plot the smoothed coordinates in blue on the tear boundary
smooth_xcoords= round(smooth_xcoords +recttear(1)-xtear);%shifted the smooth
coordinates
smooth_ycoords= round(smooth_ycoords +recttear(2)-ycenter);%shifted the
smooth coordinates
% shifts the tear to be centered at (0,0)

mean_xsmooth = mean(smooth_xcoords);
stddev = std(smooth_xcoords);
%% green lines for top, bottom, and tear boundaries
line([1 imsize],[baftertop,maftertop*imsize +
baftertop], 'Color', 'g', 'LineWidth', 2)
line([1 imsize],[bafterbottom,mafterbottom*imsize +
bafterbottom], 'Color', 'g', 'LineWidth', 2)
line([1 imsize],[baftercenter,maftercenter*imsize +
baftercenter], 'Color', 'g', 'LineWidth', 2)

```

```

%% Saving new image
title([PathName,FileName])
saveas(gcf,[PathName,FileName(1:end-4),'.fig'])
save([PathName,FileName(1:end-4),'.mat']);
end

```

Duct Tape Code 3a: Compare Exemplar and Sample

```

%% DuctTapeScript_3.m
% by W. D. Ristenpart and Alicia Alfter
% Jan 2014 / January 2015
% Compares the smooth coordinates of exemplar and sample and calculates the
% sum of square residual (SSR or SSE)
% Instructions:
% The program first asks you to load the exemplar, and then it will open
% the smooth coordinates data and calculate the SSR
% Colormap is a diagram that has set colors for the range of numbers in a
% matrix
% Histogram and Boxplot are visual tools to see how many false positives in a
matrix

% now, analyze residuals
close all;

% get the filenames of the exemplars
[ExFileNames,ExPathName] = uigetfile('*.mat','Select all analyzed exemplars
'),'multiselect','on');

% get the filenames of the samples
[SampFileNames,SampPathName] = uigetfile('*.mat','Select all analyzed
samples'),'multiselect','on');

% count how many you want to analyze
num_exemplars = length(ExFileNames);
num_samples = length(SampFileNames); % the code now assumes there are equal
numbers

if num_exemplars~=num_samples;
    disp('Warning! Unequal numbers of exemplars and samples')
    return;
end

% initialize empty data structures;
comparison_data = struct('Exemplar_FileName',[],
'Sample_FileName',[],'ex_num',[],'sample_num',[],'Y',[],
'X_exemplar',[],'X_sample',[],'Residuals',[],'SSE',[]);
[ finalSSE_matrix ] = zeros(num_exemplars,num_samples);%This finalSSE matrix
forms the matrix of SSE per exemplar and sample comparisons
[ Rvalue_matrix ] = zeros(num_exemplars,num_samples);
[ Pvalue_matrix ] = zeros(num_exemplars,num_samples);

cntr = 1;
for j = 1:num_exemplars

```

```

disp(j)
Ex_Data =
load([ExPathName,ExFileNames{j}], 'smooth_xcoords', 'smooth_ycoords', 'mean_xsmooth');
Ex_Xsmooth = [Ex_Data.smooth_xcoords];%opens exemplar smooth coords data
Ex_Ysmooth = [Ex_Data.smooth_ycoords];%opens exemplar smooth coords data
Ex_mean_xsmooth = [Ex_Data.mean_xsmooth];%mean of the smooth coords
std_Ex_Xsmooth(j) = std(Ex_Xsmooth);%standard deviation of smooth coords

for k = 1:num_samples
Sample_Data =
load([SampPathName,SampFileNames{k}], 'smooth_xcoords', 'smooth_ycoords', 'mean_xsmooth');
Sample_Xsmooth = [Sample_Data.smooth_xcoords];%opens sample smooth
coords data
Sample_Ysmooth = [Sample_Data.smooth_ycoords];%opens sample smooth
coords data
Sample_mean_xsmooth = [Sample_Data.mean_xsmooth];%mean of the smooth
coords
std_Sam_Xsmooth(k) = std(Sample_Xsmooth); %standard deviation of
smooth coords

[sse, x_ex, x_sample, Rvalue, Pvalue] =
DuctTapeResiduals_4(Ex_Xsmooth, Sample_Xsmooth, Ex_mean_xsmooth, Sample_mean_xsmooth);%function calculates sse
comparison_data(cntr).X_exemplar = x_ex;%exemplar smooth coords
comparison_data(cntr).X_sample = x_sample;%sample smooth coords
comparison_data(cntr).SSE = sse;
comparison_data(cntr).Rvalue = Rvalue;%correlation coefficient
comparison_data(cntr).Pvalue = Pvalue;%pvalue of correlation
coefficient

cntr = cntr + 1;
finalSSE_matrix(k,j) = sse/10^6;%final sse
Rvalue_matrix(k,j) = Rvalue;%correlation coefficient
Pvalue_matrix(k,j) = Pvalue;%pvalue of correlation coefficient
end
end

%% color map matrix
sum_fp_col = 0;
colormap_matrix = finalSSE_matrix*0;
for j = 1:length(finalSSE_matrix);
colormap_matrix(j,j) = 1;
diagval = finalSSE_matrix(j,j);%diagval equals the correct matched pair
sse
col_false_positives = find(finalSSE_matrix(:,j)<diagval);%finds the sse
values lower than the diagval or true sse
if length(col_false_positives)>0;
disp(['False Positive Found! at exemplar ', int2str(j), ', sample(s)
', int2str(col_false_positives)']);%transpose
colormap_matrix(col_false_positives,j) = 0.5;
sum_fp_col = sum_fp_col+length(col_false_positives);
end

```

```

sse_col_normal(:,j) = finalSSE_matrix(:,j)/finalSSE_matrix(j,j);% SSE/SSEo
step 1 of normalization
end

sum_fp_row = 0;
for j =1:length(finalSSE_matrix);
    colormap_matrix(j,j) = 1;
    diagval = finalSSE_matrix(j,j);
    row_false_positives = find(finalSSE_matrix(j,:)<diagval);
    if length(row_false_positives)>0;
        disp(['False Positive Found! at sample ', int2str(j), ', exemplar(s)
', int2str(row_false_positives)]);
        colormap_matrix(j,row_false_positives) = 0.5;
        sum_fp_row = sum_fp_row+length(row_false_positives);
    end
sse_row_normal(j,:) = finalSSE_matrix(j,:)/finalSSE_matrix(j,j);% normalized
sse matrix
end
total_fp = sum_fp_col+sum_fp_row;

%% histogram
a=num_exemplars-1;
b = num_samples;

colornormal_col=log10(sse_col_normal);%normalize SSE/SSEo for colormap
colornormal_row = log10(sse_row_normal);

sse_col_normal(logical(eye(size(sse_col_normal)))) = [];% take out the true
SSE/SSEo = 1 diagonal values, step 2 of normalization
sse_row_normal(logical(eye(size(sse_row_normal)))) = [];

sse_col_log = log10(sse_col_normal);%normalize SSE/SSEo by taking log10 of
SSE/SSEo Step 3 of normalization
sse_row_log = flip(log10(sse_row_normal));

sse_reshape_col = reshape(sse_col_log,a,b);%reshapes matrix to be a 200x200
minus the diagonal true values
sse_reshape_row = reshape(sse_row_log,a,b);

%% boxplot
std_col_matrix = [std_Ex_Xsmooth' sse_reshape_col'];
std_row_matrix = [std_Sam_Xsmooth' sse_reshape_row'];

sort_std_row = sortrows(std_row_matrix,(1));
sort_std_col = sortrows(std_col_matrix,(1));

sorted_row_matrix = sort_std_row(:,2:end);
sorted_col_matrix = sort_std_col(:,2:end);

sorted_row_without_firstcol = sorted_row_matrix';
sorted_col_without_firstcol = sorted_col_matrix';
%sorted sse by the std deviation in increasing order
figure, boxplot(sorted_row_without_firstcol);
xlabel('Exemplars','FontSize',15)
ylabel('RSS*10^-6 pixels^2','FontSize',15)

```

```

title('std dev SSE vs exemplars','FontSize',15)

figure, boxplot(sorted_col_without_firstcol);
xlabel('samples','FontSize',15)
ylabel('RSS*10^-6 pixels^2','FontSize',15)
title('std dev SSE vs samples','FontSize',15)

%% index color map image
colormap jet
cmap = colormap;
invjet = flipud(cmap);
colormap(invjet);
colormapeditor
figure, surf(colornormal_col);
colorbar;
set(gcf,'EdgeColor','none')
set(gcf,'Color','w')
axis square;
% colormap jet
cmap = colormap;
invjet = flipud(cmap);
colormap(invjet);
colormapeditor
figure, surf(colornormal_row);
colorbar;
set(gcf,'EdgeColor','none')
set(gcf,'Color','w')
axis square;
% save workspace
save('sample_data.mat');

```

Duct Tape Code 3b: Plot SSR

```

%% DuctTapeScript_Plot3.m
% by W. D. Ristenpart and Alicia Alfter
% Jan 2014 / January 2015
% Compares the smooth coordinates of exemplar and sample and calculates the
% sum of square residual (SSR or SSE)
% Instructions:
% The program first asks you to load the exemplar, and then it will open
% the smooth coordinates data and calculate the SSR
% Colormap is a diagram that has set colors for the range of numbers in a
% matrix
% Function Plot4 plots the comparisons

close all;

% get the filenames of the exemplars
[ExFileNames,ExPathName] = uigetfile('*.mat',('Select all analyzed exemplars
'),'multiselect','on');

% get the filenames of the samples
[SampFileNames,SampPathName] = uigetfile('*.mat',('Select all analyzed
samples '), 'multiselect', 'on');

```

```

% count how many you want to analyze
num_exemplars = length(ExFileNames);
num_samples = length(SampFileNames); % the code now assumes there are equal
numbers

if num_exemplars~=num_samples;
    disp('Warning! Unequal numbers of exemplars and samples')
    return;
end

% initialize empty data structures;
comparison_data = struct('Exemplar_FileName', [],
'Sample_FileName', [], 'ex_num', [], 'sample_num', [], 'Y', [],
'X_exemplar', [], 'X_sample', [], 'Residuals', [], 'SSE', []);
[ finalSSE_matrix ] = zeros(num_exemplars,num_samples);%This finalSSE matrix
forms the matrix of SSE per exemplar and sample comparisons

cntr = 1;
for j = 1:num_exemplars
    disp(j)
    Ex_Data =
load([ExPathName,ExFileNames{j}], 'smooth_xcoords', 'smooth_ycoords', 'mean_xsmo
oth');
    Ex_Xsmooth = [Ex_Data.smooth_xcoords];
    Ex_Ysmooth = [Ex_Data.smooth_ycoords];
    Ex_mean_xsmooth = [Ex_Data.mean_xsmooth];

    for k = 1:num_samples
        Sample_Data =
load([SampPathName, SampFileNames{k}], 'smooth_xcoords', 'smooth_ycoords', 'mean_
xsmooth');
        Sample_Xsmooth = [Sample_Data.smooth_xcoords];
        Sample_Ysmooth = [Sample_Data.smooth_ycoords];
        Sample_mean_xsmooth = [Sample_Data.mean_xsmooth];

        [sse, x_ex, x_sample] =
DuctTapeResiduals_Plot_4(j,k,ExFileNames, SampFileNames, Ex_Xsmooth,
Sample_Xsmooth, Ex_Ysmooth,
Sample_Ysmooth, Ex_mean_xsmooth, Sample_mean_xsmooth);
        comparison_data(cntr).Exemplar_FileName = ExFileNames{j};
        comparison_data(cntr).ex_num = j;
        comparison_data(cntr).Sample_FileName = SampFileNames{k};
        comparison_data(cntr).sample_num = k;
        comparison_data(cntr).X_exemplar = x_ex;
        comparison_data(cntr).X_sample = x_sample;
        comparison_data(cntr).SSE = sse;

        cntr = cntr + 1;
        finalSSE_matrix(k,j) = sse/10^6;
    end
end

%% index color map image
sum_fp_col = 0;

```

```

colormap_matrix = finalSSE_matrix*0;
for j = 1:length(finalSSE_matrix);
    colormap_matrix(j,j) = 1;
    diagval = finalSSE_matrix(j,j);%diagval equals the correct matched pair
sse
    col_false_positives = find(finalSSE_matrix(:,j)<diagval);
    if length(col_false_positives)>0;
        disp(['False Positive Found! at exemplar ', int2str(j), ', sample(s)
', int2str(col_false_positives)']);%transpose
        colormap_matrix(col_false_positives,j) = 0.5;
        sum_fp_col = sum_fp_col+length(col_false_positives);
    end
sse_col_normal(:,j) = finalSSE_matrix(:,j)/finalSSE_matrix(j,j);% normalized
sse matrix
end

sum_fp_row = 0;
for j =1:length(finalSSE_matrix);
    colormap_matrix(j,j) = 1;
    diagval = finalSSE_matrix(j,j);
    row_false_positives = find(finalSSE_matrix(j,:)<diagval);
    if length(row_false_positives)>0;
        disp(['False Positive Found! at sample ', int2str(j), ', exemplar(s)
', int2str(row_false_positives)']);
        colormap_matrix(j,row_false_positives) = 0.5;
        sum_fp_row = sum_fp_row+length(row_false_positives);
    end
sse_row_normal(j,:) = finalSSE_matrix(j,:)/finalSSE_matrix(j,j);% normalized
sse matrix
end
total_fp = sum_fp_col+sum_fp_row;

%% boxplot
a=num_exemplars-1;
b = num_samples;

sse_col_normal(logical(eye(size(sse_col_normal)))) = [];
sse_col = reshape(sse_col_normal,a,b);

sse_row_normal(logical(eye(size(sse_row_normal)))) = [];
sse_row = reshape(sse_row_normal,a,b);

sse_col_log = log10(sse_col);
sse_row_log = log10(sse_row);

figure, boxplot(sse_col_log);
xlabel('comparisons')
ylabel('log10 normalized','FontSize',15)
title('log10 of sse by column','FontSize',15)

figure, boxplot(sse_row_log);
xlabel('comparisons','FontSize',15)
ylabel('log10 normalized','FontSize',15)
title('log10 of sse by row','FontSize',15)

```

```

% color map
figure('Color','w');
axes('Position',[0.1 0.1 0.8 0.8]);
imshow(colormap_matrix);
axis square;
c1=[1 1 1]; %White
c2=[1 0 0]; %Red
c3=[0 1 0]; %Green
% c4 =[1 1 0]; %Yellow

cmap=[c1;c1;c2;c2;c3;c3];
colormap(cmap)

% save workspace
save('sample_data.mat');

```

Duct Tape Code 4a Function of 3a: Calculate SSR

```

function [sse, Ex_Xsmooth_adj, Sample_Xsmooth_adj, Rvalue,Pvalue ] =
DuctTapeResiduals_4(Ex_Xsmooth, Sample_Xsmooth,Ex_mean_Xsmooth,
Sample_mean_Xsmooth)
%DuctTapeResiduals_4.m
% By W. D. Ristenpart and Alicia Alfter
% December 2014
%Compares tear direction and roll direction coordinate points

%% SHIFT IN ROLL DIRECTION

w = 2000; %width of the tear boundary

difference = length(Ex_Xsmooth)-(w);
cutoff = difference/2;
Ex_Xsmooth_adj = (Ex_Xsmooth(cutoff+1:end-cutoff));

difference = length(Sample_Xsmooth)-(w);
cutoff = difference/2;
Sample_Xsmooth_adj = (Sample_Xsmooth(cutoff+1:end-cutoff));

if Ex_mean_Xsmooth < Sample_mean_Xsmooth
    dir = 1;
else
    dir = -1;
end
% Shifts takes the difference between the roll direction exemplar and
% sample points. The length of shifts is enlarged by 2, because the factor
% makes the vector of shifts go past the minimum.
shifts = 0:1:2*round(abs(Ex_mean_Xsmooth-Sample_mean_Xsmooth));
% sse_vector is the sse of each point in the roll direction
sse_vector = zeros(1,length(shifts));
for j = 1:length(sse_vector);% calculates sse
    sse_vector(j) = sum((Ex_Xsmooth_adj - Sample_Xsmooth_adj +
dir*shifts(j)).^2);
end

```

```

[R,P] = corrcoef(Ex_Xsmooth_adj, Sample_Xsmooth_adj);
Rvalue=R(1,2);%Correlation coefficient
Pvalue = P(1,2);

% the minimum index is the minimum of the sse_vector vs shifts plot
[~,minind] = min(sse_vector); % find the position at which the sse is the
minimum
sse = sse_vector(minind); % the lowest sse for the exemplar and sample roll
direction comparison is selected. To see the figure,
plot(shifts,sse_vector,'or')
Ex_Xsmooth_adj = Ex_Xsmooth_adj + dir*shifts(minind); % shifts the exemplar
closer to the sample
end

```

Duct Tape Code 4b Function 3b: Plot comparisons

```

function [sse, Ex_Xsmooth_adj, Sample_Xsmooth_adj ] =
DuctTapeResiduals_Plot_4(exemplar_index,sample_index,Ex_filename,Samp_filenam
e,Ex_Xsmooth, Sample_Xsmooth,Ex_Ysmooth, Sample_Ysmooth,Ex_mean_Xsmooth,
Sample_mean_Xsmooth)
%DuctTapeResiduals_Plot4.m
% By W. D. Ristenpart and Alicia Alfter
% December 2014
%Compares tear direction and roll direction coordinate points

%% SHIFT IN ROLL DIRECTION

w = 2000; %width of the tear boundary
difference = length(Ex_Ysmooth)-(w);
cutoff = difference/2;
Ex_Ysmooth_adj = (Ex_Ysmooth(cutoff+1:end-cutoff));

difference = length(Sample_Ysmooth)-(w);
cutoff = difference/2;
Sample_Ysmooth_adj = (Sample_Ysmooth(cutoff+1:end-cutoff));

difference = length(Ex_Xsmooth)-(w);
cutoff = difference/2;
Ex_Xsmooth_adj = (Ex_Xsmooth(cutoff+1:end-cutoff));

difference = length(Sample_Xsmooth)-(w);
cutoff = difference/2;
Sample_Xsmooth_adj = (Sample_Xsmooth(cutoff+1:end-cutoff));

if Ex_mean_Xsmooth < Sample_mean_Xsmooth
    dir = 1;
else
    dir = -1;
end
% Shifts takes the difference between the roll direction exemplar and
% sample points. The length of shifts is enlarged by 2, because the factor
% makes the vecotr of shifts go past the minimum.
shifts = 0:1:2*round(abs(Ex_mean_Xsmooth-Sample_mean_Xsmooth));
% sse_vector is the sse of each point in the roll direction
sse_vector = zeros(1,length(shifts));

```

```

for j = 1:length(sse_vector);% calculates sse
    sse_vector(j) = sum((Ex_Xsmooth_adj - Sample_Xsmooth_adj +
dir*shifts(j)).^2);
end
% the minimum index is the minimum of the sse_vector vs shifts plot
[~,minind] = min(sse_vector); % find the position at which the sse is the
minimum
sse = sse_vector(minind); % the lowest sse for the exemplar and sample roll
direction comparison is selected. To see the figure,
plot(shifts,sse_vector,'or')
Ex_Xsmooth_adj = Ex_Xsmooth_adj + dir*shifts(minind); % shifts the exemplar
closer to the sample

%% GRAPHICAL DIAGRAM OF TEAR
figure('Color','w')
subplot(2,1,1);
set(gcf,'Units','normalized','Position',[.1 .1 .8 .8])
plot(Ex_Ysmooth_adj,Ex_Xsmooth_adj,'.r')
hold on;
plot(Sample_Ysmooth_adj,Sample_Xsmooth_adj,'.g')

xlim([-1300 1300])%Adjust the x axis
ylim([-300 300]) %Adjust the y axis
box on
title([Ex_filename(exemplar_index),' vs. ',
Samp_filename(sample_index)],'FontSize',25);

xlabel('Tear Direction (Pixels)','FontSize',25);
ylabel('Roll Direction (Pixels)','FontSize',25);

%% GRAPHICAL DIAGRAM OF RESIDUAL COMPARISONS
resid = Sample_Xsmooth_adj- Ex_Xsmooth_adj;
a = -1000:999;
subplot(2,1,2);
area(a,resid);

xlim([-1300 1300])%Adjust the x axis
ylim([-300 300]) %Adjust the y axis
xlabel('Tear Direction (Pixels)','FontSize',25);
ylabel('Roll Direction (Pixels)','FontSize',25);
box on
title(['Sum Squared Errors =
',num2str(round(sse))], 'Color','g','FontSize',25);
end

```

Duct Tape Code 5: Calculate Unique SSR for exemplars and samples

```

%% now, analyze residuals
close all;

% get the filenames of the exemplars
[ExFileNames,ExPathName] = uigetfile('*.mat',('Select all analyzed exemplars
'),'multiselect','on');

% get the filenames of the samples

```

```

[SampFileNames,SampPathName] = uigetfile('*.mat',('Select all analyzed
samples '), 'multiselect', 'on');

% count how many you want to analyze
num_exemplars = length(ExFileNames);
num_samples = length(SampFileNames);

if num_exemplars~=num_samples;
    disp('Warning! Unequal numbers of exemplars and samples')
    return;
end
comparison_data = struct('Exemplar_FileName', [],
'Sample_FileName', [], 'ex_num', [], 'sample_num', [], 'Y', [],
'X_exemplar', [], 'X_sample', [], 'Residuals', [], 'SSE', []);
[ U_vector_Ex ] = zeros(1,num_exemplars);
[ U_vector_Sam ] = zeros(1,num_samples);
cntr = 1;
for j = 1:num_exemplars
    Ex_Data =
load([ExPathName,ExFileNames{j}], 'smooth_xcoords', 'smooth_ycoords');
    Ex_Xsmooth = [Ex_Data.smooth_xcoords];%open exemplar smooth coords
data
    Ex_Ysmooth = [Ex_Data.smooth_ycoords];%open exemplar smooth coords
data

    [U_sseEx,bestfit_Ex,U_Rvalue_Ex,U_Pvalue_Ex,resid] =
DuctTapeResiduals_5(Ex_Xsmooth,Ex_Ysmooth);%function that calculates U SSR to
best fit line for exemplars
    comparison_data(cntr).SSEEx = U_sseEx;
    comparison_data(cntr).BestEx = bestfit_Ex;
    comparison_data(cntr).resid = resid;
    comparison_data(cntr).RvalueEx = U_Rvalue_Ex;
    comparison_data(cntr).PvalueEx = U_Pvalue_Ex;

    U_vector_Ex(j) = U_sseEx/10^6;%unqiue SSR vector for exemplars
end
for k = 1:num_samples
    Sample_Data =
load([SampPathName,SampFileNames{k}], 'smooth_xcoords', 'smooth_ycoords');
    Sample_Xsmooth = [Sample_Data.smooth_xcoords];%open sample smooth
coords data
    Sample_Ysmooth = [Sample_Data.smooth_ycoords];%open sample smooth
coords data

    [U_sseSam,bestfit_Sam, U_Rvalue_Sam,U_Pvalue_Sam,resid] =
DuctTapeResiduals_6(Sample_Xsmooth,Sample_Ysmooth);%function that calcualtes
U SSR to best fit line for samples
    comparison_data(cntr).SSESam = U_sseSam;
    comparison_data(cntr).BestSam = bestfit_Sam;
    comparison_data(cntr).resid = resid;
    comparison_data(cntr).RvalueSam = U_Rvalue_Sam;
    comparison_data(cntr).PvalueSam = U_Pvalue_Sam;

    U_vector_Sam(k) = U_sseSam/10^6;%unqiue SSR vector for sample

```

end

Duct Tape Code 6 Function of 5: Unique coordinates for exemplars

```
function [U_sseEx,bestfit_Ex,U_Rvalue_Ex,U_Pvalue_Ex,resid] =
DuctTapeResiduals_5(Ex_Xsmooth,Ex_Ysmooth)
%duct_tape_get_residuals_ver1
% By W. D. Ristenpart and Alicia Alfter
% January 2015
%Calculates the Unique coordinates for every exemplar. Unqie is the
%difference of the Exemplar smooth coordinates and the best fit line.
%% SHIFT IN ROLL DIRECTION
pf2 = polyfit(Ex_Ysmooth,Ex_Xsmooth,1); %Finds the coefficients of the line
y=mx+b
m2 = pf2(1); % m is the slope of the top boundary
b2 = pf2(2); %b is the y-intercept of top boundary
bestfit_Ex = m2*(Ex_Ysmooth)+b2;

U_sseEx = sum((Ex_Xsmooth - bestfit_Ex).^2);

[R,P] = corrcoef(Ex_Xsmooth, bestfit_Ex);
U_Rvalue_Ex = R(1,2);%Correlation coefficient
U_Pvalue_Ex = P(1,2);

resid = Ex_Xsmooth-bestfit_Ex;
end
```

Duct Tape Code 7 Function of 5: Unique coordinates for samples

```
function [U_sseSam,bestfit_Sam, U_Rvalue_Sam,U_Pvalue_Sam, resid] =
DuctTapeResiduals_6(Sample_Xsmooth,Sample_Ysmooth)
%duct_tape_get_residuals_ver1
% By W. D. Ristenpart and Alicia Alfter
% April 2014
%Calculates the Unique coordinates for every sample. Unqie is the
%difference of the Sample smooth coordinates and the best fit line.

%% SHIFT IN ROLL DIRECTION
pf = polyfit(Sample_Ysmooth,Sample_Xsmooth,1); %Finds the coefficients of
the line y=mx+b
m = pf(1); % m is the slope of the top boundary
b = pf(2); %b is the y-intercept of top boundary
bestfit_Sam = m*(Sample_Ysmooth)+b;

U_sseSam = sum((Sample_Xsmooth - bestfit_Sam).^2);

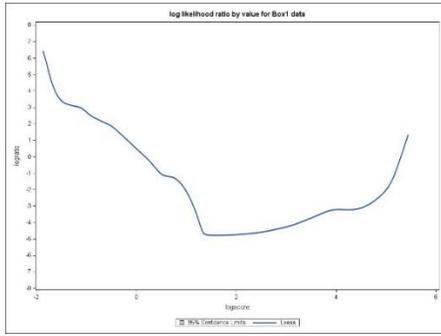
[R,P] = corrcoef(Sample_Xsmooth, bestfit_Sam);
U_Rvalue_Sam = R(1,2);%Correlation coefficient
U_Pvalue_Sam = P(1,2);

resid = Sample_Xsmooth-bestfit_Sam;
end
```

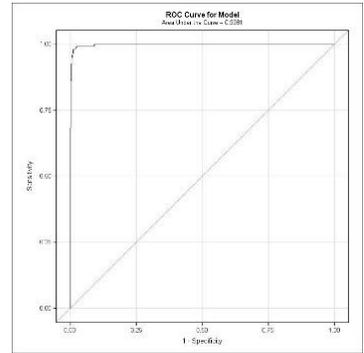
Appendix B: Statistical Analysis

The statistician Neil Willits performed receiver operating characteristic (ROC) curves on the resulting SSR matrices for each set of 200 pairs of duct tape. The ROC curves were plotted representing the sensitivity (the proportion of true matches correctly identified) and the specificity (the proportion of pairs that were identified as matching that were in fact true matches) among the 40,000 pairwise comparisons for a given type of duct tape and tear. In most instances these were able to identify nearly all true matches with a very low false positive rate. Graphs are also presented comparing the distribution of the log score for true and false matches within each data set. Log scores were used instead of the measured score so that the distributions wouldn't be squeezed up against the line score = 0. These distributions are estimated using a Gaussian kernel method, which produces smooth density estimates (as opposed to discrete histograms), but which will tend to cause a slight inflation in the tail probabilities. Finally plots are presented of the log likelihood ratio based on the density estimates, which give an indication of the fold-difference in likelihood between true and false matches for a given level of (log) score.

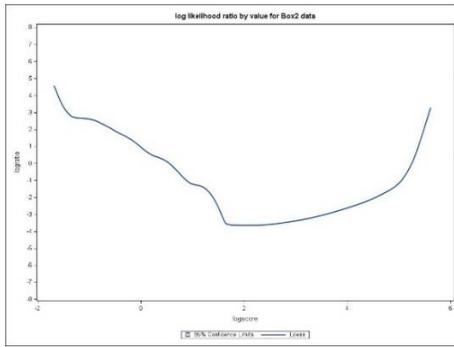
Hand torn NGB Likelihood Ratio



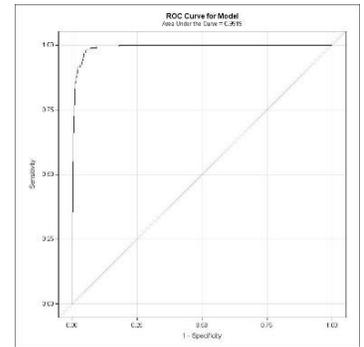
Hand torn NGB ROC Curve



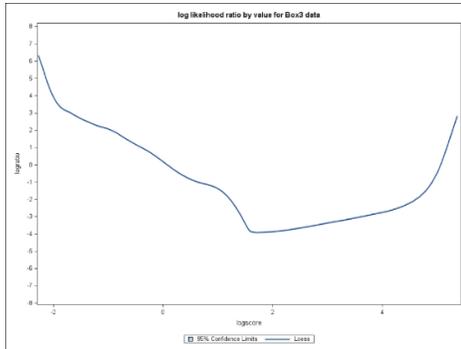
Hand torn NGG Likelihood Ratio



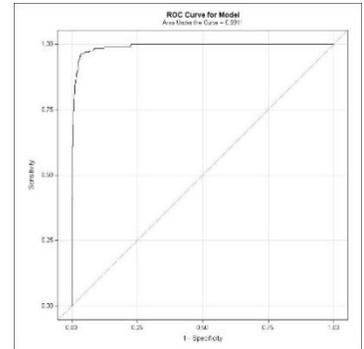
Hand torn NGG ROC Curve



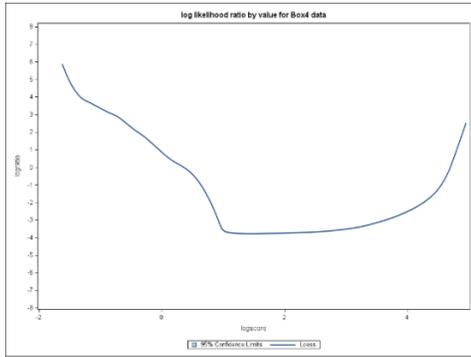
Hand torn NPB Likelihood Ratio



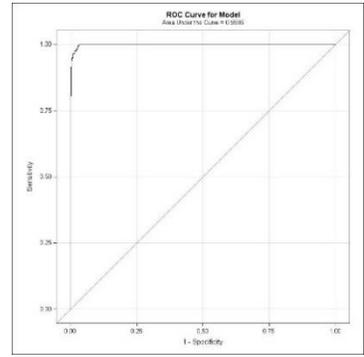
Hand torn NPB ROC Curve



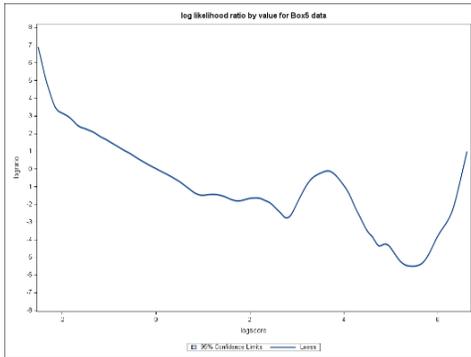
Hand Torn NPG Likelihood Ratio



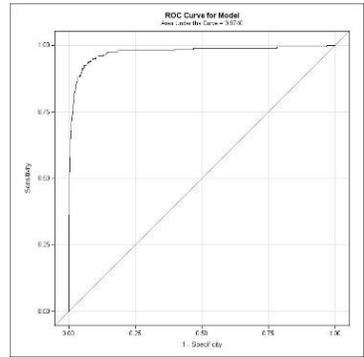
Hand torn NPG ROC Curve



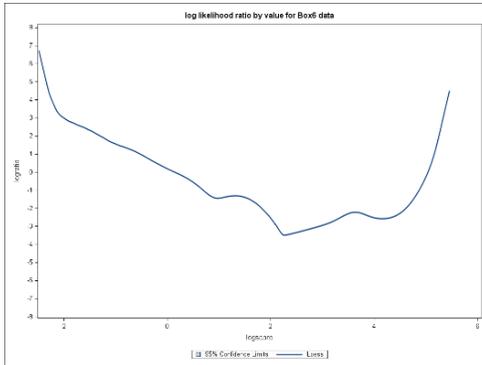
Hand torn 3MPG Likelihood Ratio



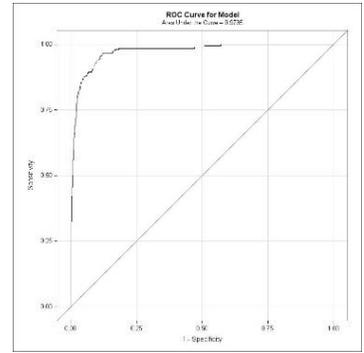
NPG ROC Curve



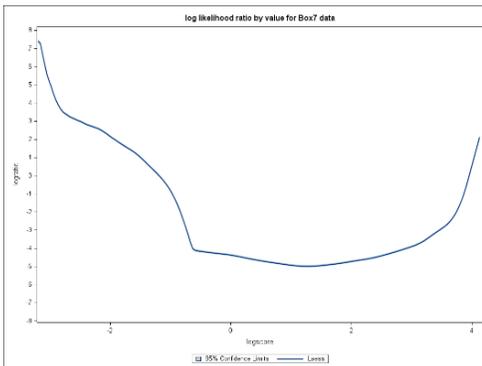
Hand torn 3MPG Likelihood Ratio



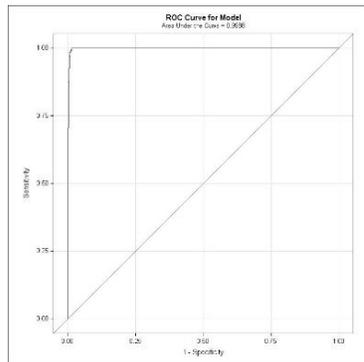
NPG ROC Curve



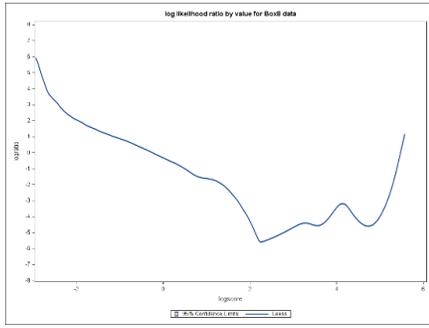
Hand torn 3MPG Likelihood Ratio



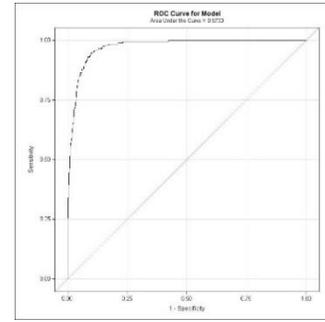
NPG ROC Curve



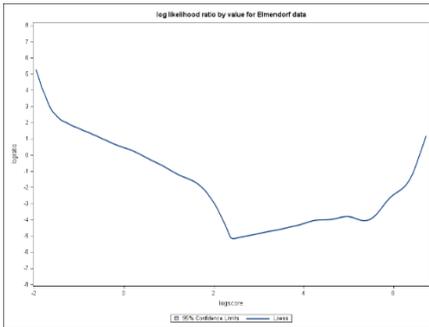
Hand torn 3MPG Likelihood Ratio



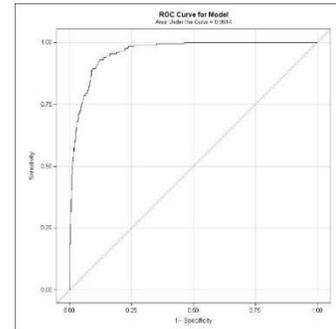
Hand torn 3MPG ROC Curve



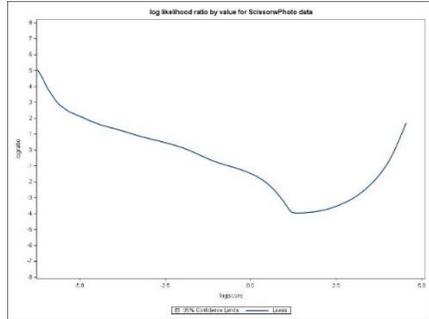
Elmendorf torn 3MPG Likelihood Ratio



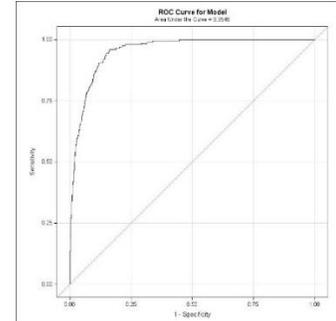
Elmendorf torn 3MPG ROC Curve



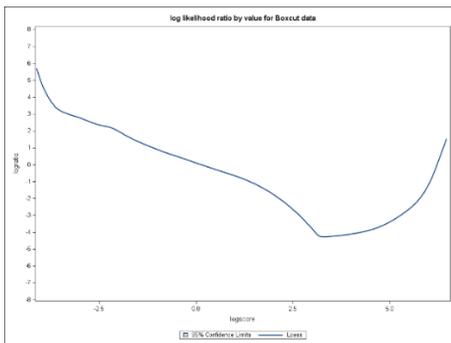
Scissor cut 3MPG Likelihood Ratio



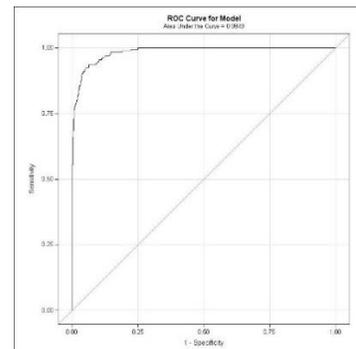
Scissor cut 3MPG ROC Curve



Box cut 3MPG Likelihood Ratio



Box cut 3MPG ROC Curve



Appendix C: Supplemental research

Histograms

