# Automated Canvas Analysis for Painting Conservation

By Brendan Tobin

## 1. Motivation

Distinctive variations in the spacings between threads in a painting's canvas can be used to show that two sections of canvas originated on the same bolt of fabric. A canvas spacing match between two works by the same artist can provide strong evidence toward the authenticity of a painting. Figure 1. shows a matching canvas spacing pattern between a set of three paintings by Nicolas Poussin. Analysis of the structure of a painting's canvas can also identify areas of a painting susceptible to damage, or that may have been modified in the past.

The individual threads in a canvas are visible in an x-ray radiograph of a painting, however additional processing is necessary to visualize and compare canvas structure data. Current canvas analysis techniques as described in Erdmann et. al. are not capable of resolving thread spacings at the level of individual threads. This work seeks to develop thread-level canvas analysis techniques to identifying the location of all thread crossings in a x-ray radiograph image using machine learning techniques to train a classifier capable of identifying thread crossings in a radiograph. The position of individual thread crossings can then be used to develop a thread level-map of canvas structure.

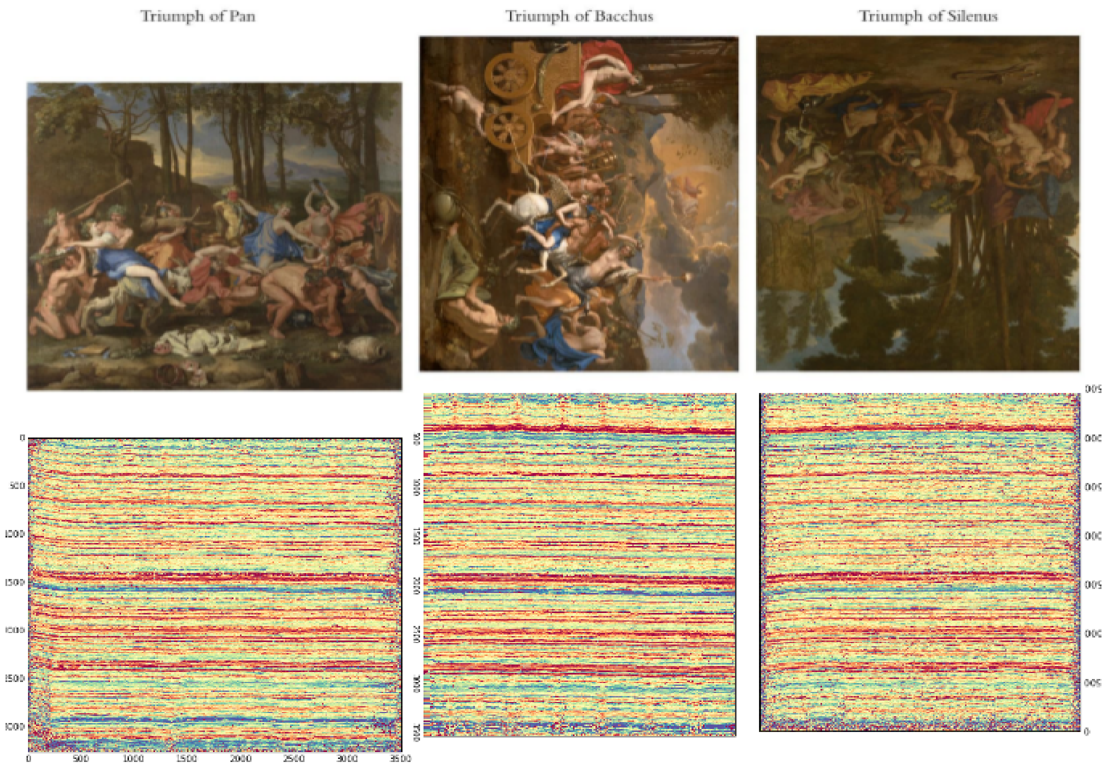

Fig. 1. Canvas spacing patterns used to prove the authenticity of The Triumph of Bacchus by comparing the pattern with two other paintings confirmed to be by Nicolas Poussin. Triumph of Pan and Silenus images courtesy of The National Gallery, London, and Bacchus image courtesy of the Nelson-Atkins Museum of Art.

## 2. Data collection

Raw data is extracted from a grayscale radiograph image of a painting. The image is preprocessed by a low-pass filter to remove large scale features captured in the radiograph, and histogram equalization is performed to normalize intensity values across the image. A large collection of relatively accurate thread crossing position data has been generated using template matching with a manually generated template for the radiograph of Nicolas Poussin's Triumph of Bacchus. Using this position data, patches of the image centered on a thread crossing are extracted. Thread crossings are spaced on average between 20 and 30 pixels apart. A square patch of 60x60 pixels centered on a thread crossing was selected in order to completely encompass a thread crossing. Data extracted from the Triumph of Bacchus radiograph produces 100,000 distinct crossing images. Negative training data is generated by selecting points randomly distributed across the canvas.

The training set is relatively accurate, but contains a small number of points that do not correspond to tread centers. Similarly, a portion of randomly assigned points will correspond to thread centers, resulting in false negatives in the training data. For an average thread spacing of 25 pixels, approximately 1 / 625 randomly selected points will correspond to 'true' thread centers.

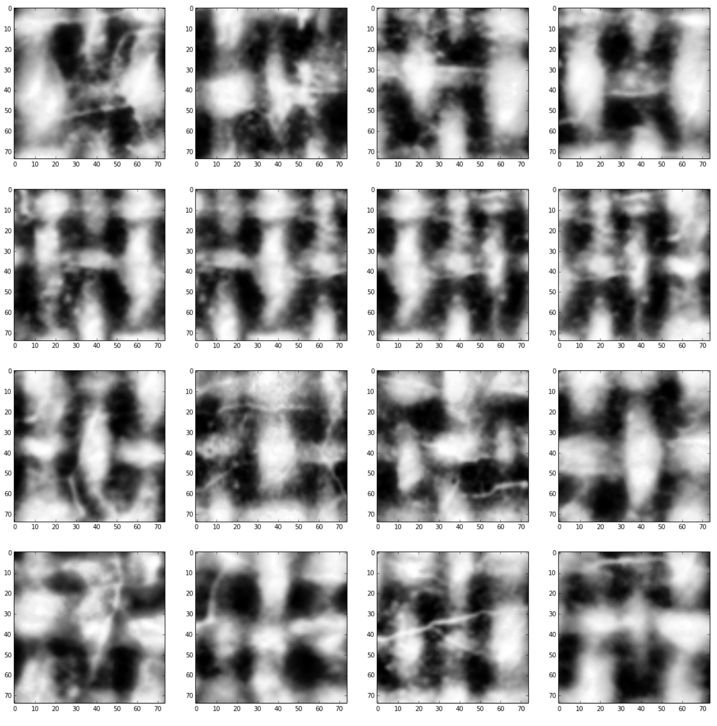

Fig. 2. A set of images containing thread crossings.

## 3. Features and preprocessing

I evaluated two methods of preprocessing to reduce the high dimensionality of input image sections. A 60x60 pixel image section used as a feature set results in 3600 features capable of assuming 256 discrete intensity values. In order to reduce the size of the feature space two methods for dimensional reduction were investigated: computing a Histogram of Oriented Gradient (HOG) classifier for the image section, and applying Principle Component Analysis (PCA) to reduce the dimensionality of the dataset. The accuracy of each preprocessing technique, along with the raw 3600 feature crossing descriptor was investigated for each technique.

A HOG classifier algorithm divides the image into subsections, and computes the direction and value of the primary gradient in each image section. [Dalal et. al.] This results in a classifier consisting of 392 values that describe the image section. The HOG descriptor captures the spatial relationship between adjacent pixels and is robust to changes in illumination and image intensity. Separately, I applied PCA [Martínez et. al.] to the set of raw pixel data in order to reduce the dimensionality of the data. Retaining 95% of the variance of the dataset results in 91 principle components.

## 4. Models

The algorithm's goal is to calculate the probability that an image patch centered on each point corresponds to a thread center. Support Vector Machines (SVM) and Logistic Regression are both well suited for this purpose. I implemented both models with two classes. One class consists of the patches centered on known thread crossings, and the other class is comprised of the randomly selected image patches.

I implemented Logistic Regression with the help of the Liblinear library. [Pedregosa et. al.] Logistic Regression is based upon maximizing:

$$\sum_{i=1}^{m} logp(y^i|x^i;\theta)\ -\theta^T\theta$$

with respect to $\theta$ where $p(y^i|x^i;\theta)$ is the Sigmoid function:

$$(1+exp(-\theta^T x)^{-1}(1-(1+exp(-\theta^T x)^{-1})^{1-y^i}$$

I also implemented a two class C-SVM [Chih-Chung Chang et. al.] with a linear kernel which is based upon minimizing:

$$1/2\ ||w||^2 + C\sum_{i=1}^{m}\varepsilon_i \qquad s.t.$$

$$y^i(w^T x^i + b)\ >\ 1-\varepsilon_i \qquad i = 1,2....\ m$$

Using both of these models it is possible to calculate the probability that a given set of features belongs to the positive image class.

In order to extract the location of thread centers from the data used, the following process is applied. For each pixel in the image under consideration, a 70x70 patch is extracted. It is preprocessed by computation of a HOG descriptor or PCA. A trained model is then used to calculate the probability the current image patch under consideration is a thread crossing. The probability it belongs to the thread crossing set is retained for that pixel. The processing is then repeated for each pixel in the image. This process results in a probability map describing the probability that each point in the image is a thread crossing. Local maxima of the probability map are then identified as the location of thread crossings.

## 5. Results

Each preprocessing method was used in combination with each model. Each combination was trained using a varying number of training examples in order to check for convergence. Each model was validated using a set of 10,000 crossing images reserved for validation. A comparison of training and validation error indicates that the SVM based models are fully converging and have sufficient training data. The training and validation error do not fully converge for Logistic regression based models, indicating that 100,000 training examples is insufficient to fully train a classifier. Due to limited computing power it was impractical to train these models using a larger dataset. The datasets processed using the HOG classifier resulted in the lowest classification error.

| **Algorithm** | **Classification Error** |
|---|---|
| Log. Regression | 20.0% |
| Log. Reg. + HOG | 15.7% |
| Log. Reg. + PCA | 18.3% |
| SVM | 17.0% |
| SVM + HOG | 12.1% |
| SVM + PCA | 15.6% |

| **Number of training samples** | **100** | **1,000** | **10,000** | **50000** | **100,000** |
|---|---|---|---|---|---|
| **Training Error** | | | | | |
| Log. Regression | 0.00% | 0.00% | 8.70% | 10.67% | 13.20% |
| Log. Reg. + HOG | 0.00% | 0.00% | 3.00% | 9.19% | 16.00% |
| Log. Reg. + PCA | 0.00% | 6.00% | 17.50% | 16.42% | 17.80% |
| SVM | 0.00% | 0.00% | 5.00% | 8.46% | 14.20% |
| SVM + HOG | 0.00% | 1.00% | 6.60% | 7.86% | 11.30% |
| SVM + PCA | 0.00% | 9.60% | 14.10% | 13.86% | 14.90% |
| **Classification Error** | | | | | |
| Log. Regression | 41.00% | 26.90% | 21.10% | 20.50% | 20.40% |
| Log. Reg. + HOG | 35.00% | 25.50% | 21.70% | 18.61% | 15.70% |
| Log. Reg. + PCA | 33.10% | 24.70% | 18.70% | 17.78% | 18.30% |
| SVM | 30.00% | 27.10% | 19.30% | 18.10% | 17.00% |
| SVM + HOG | 31.00% | 25.40% | 18.00% | 13.63% | 12.10% |
| SVM + PCA | 27.90% | 21.20% | 16.20% | 15.65% | 15.60% |

## 6. Conclusion

The low level of training error indicates that a large number of feature points are necessary to properly train a classifier in either case. This is likely due to the high dimensionality of the image features. The convergence of the training and validation error with high sample sizes indicates that the model is converging. Preprocessing with PCA appears to significantly reduce the amount of training data necessary to fully train the model. However preprocessing with a HOG classifier achieves lower classification error using both Logistic Regression and SVM. This is possibly due to the fact that it is accounts for the spacial relationship of the image data and is designed to be robust with respect to varying image illumination.

While a HOG classifier based approach achieves a higher degree of accuracy in correctly separating labeled data, a PCA based approach is almost as strong when it comes to calculating the probability an image patch is a thread crossing. Both methods produce an effective probability map which can be used to determine thread crossing locations. The PCA projection necessary to evaluate an image patch is much less computationally intensive than the HOG algorithm, meaning the PCA based approach is more viable for operating on large datasets with limited computational power.

## 7. Future Work

I would also like to segment the set of thread crossings into two distinct subsets, based upon whether the vertical or horizontal thread crossed over the other. I experimented with K-means Clustering in order to separate the dataset without manually labeling a set of training examples, but I was unable to achieve consistent separation using K-means Clustering. In the future I will manually label a set of training data and train a model to separate vertical and horizontal thread crossings. Additionally I plan to experiment with Eigenfaces and Fisherfaces [Belhumeur et. al.] models applied to thread crossing identification. The Eigenfaces method also makes use of PCA, which showed promise as a less computationally intensive method than one based upon HOG classifiers.

## 8. References

1. Erdmann, R. G., C. R. Johnson, M. Schafer, J. Twilley, and T. Sawyer . "Re-uniting Poussins Bacchanals Painted for Cardinal Richelieu through Quantitative Canvas Weave Analysis." *AIC Contemporary in Conservation conference on* 31 May. 2013.

2. Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on* 25 Jun. 2005: 886-893.

3. Martínez, Aleix M, and Avinash C Kak. "Pca versus lda." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 23.2 (2001): 228-233.

4. Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

5. Pedregosa, Fabian et al. "Scikit-learn: Machine learning in Python." *The Journal of Machine Learning Research* 12 (2011): 2825-2830.

6. Belhumeur, Peter N. João P Hespanha, and David Kriegman. "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 19.7 (1997): 711-720.