International Journal of Computer Applications (0975 - 8887)
Volume 62 - No. 9, January 2013

Enhanced Gesture Recognition Performance through Improved Pre-Processing

Jacob Grosek
Department of Applied Mathematics
University of Washington

Peizhe Shi
Department of Applied Mathematics
University of Washington

J. Nathan Kutz
Department of Applied Mathematics
University of Washington

ABSTRACT

Gesture recognition is analyzed on a set of static hand gestures in the context of designing robust, real-time pre-processing techniques for applications in hand-held electronics. A comparative case study that uses various combinations of algorithms across the steps of the recognition process is made, revealing the fact that many method combinations can produce highly accurate results, even at low resolutions, given the right kind of pre-processing. The pre-processing includes the hand segmentation and normalization done before feature extraction. Indeed, pre-processing has by far the greatest effect on the overall accuracy, robustness, and speed of the gesture recognition process, significantly outweighing the influence of feature-extraction and classification. Even at image resolutions as low as 8×8 pixels, accuracies of 99% are achieved using a simple PCA feature selection scheme and a LDA classification method. These results suggest the priority and advantages of focusing on developing robust and efficient pre-processing methods.

Keywords:

Pre-processing, Hand, Gesture, Recognition

1. INTRODUCTION

Robust, real-time gesture and eye recognition methods and algorithms are of growing importance and interest in the consumer electronics market place and industry. Indeed, technologies are now being proposed for future integrated eye, face and gesture recognition features in many hand-held and portable electronic devices such as smart phones, laptops and laser projectors. Intelligent methods for performing such recognition tasks in real-time are in high demand in order to handle the tremendous growth projected by such revolutionary technologies and devices (see Fig. 1). Algorithms need to be robust, and preferably low-dimensional in nature, in order to do real-time processing in the fraction of a second. A great number of recognition techniques have been proposed [32, 3, 33], which mostly focus on the image features being extracted and/or the development of the statistical learning techniques involved in the recognition process. The aim is to illustrate how the performance of recognition algorithms can be enhanced, using minimal processing overhead, through improved pre-processing techniques.

The pre-processing step of the recognition process is the central focus of this manuscript. With a well-designed pre-processing algorithm, even simple, well known feature extraction and classification procedures can produce excellent results, and do so with very low overhead and processing times. The objective here is to demonstrate that improving the pre-processing procedure is a competitive alternative, or even a complement, to developing more sophisticated feature selection and statistical discrimination methods for applications in the consumer electronics arena.

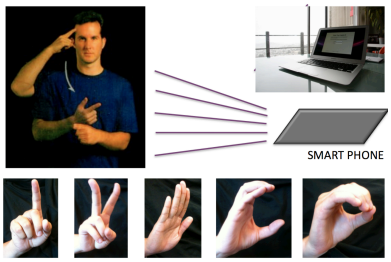

Fig. 1. Potential consumer electronics application of gesture recognition for mouse control such as one-click, double-click, cursor movement and control, scroll up and scroll down.

The hand gesture algorithms implemented here perform the following tasks: (i) pre-processing the images of hand gestures (scaling, rotation, and arm/wrist removal are key aspects of the pre-processing), (ii) extracting features from images that are distinct enough to separate different gestures, (iii) learning a classification rule from the training data, and (iv) classifying unknown images as one of the known gestures. Not implemented here, given that it is still an open area of active research, is a robust hand identification and detection algorithm that can extract hand images from complex and cluttered backgrounds. There are a variety of methods that can be envisioned for performing tasks (i)-(iv). In fact, state-of-the-art techniques exist for modeling the hand shape itself [8] and the dynamic gesture sequence from the frames of real-time video [17, 20]. However, many of these new methods come at the cost of increased processing times when compared to the simple, robust, algorithm combinations implemented here.

In this paper, the robustness and performance of tasks (ii)-(iv) are demonstrated under different pre-processing scenarios. Combining these tasks in clever ways is critical for allowing for the real-time processing and control of consumer electronic devices. This work allows for a direct performance and robustness comparison of various feature extraction and statistical testing techniques that hitherto have been deficient in the general literature of the recognition field. Thus the objective is not to bring the most sophisticated methods to bear on this problem, but to demonstrate that pre-processing of images in and of itself, using very little processing power and low-resolution, is the critical step in achieving accuracies near 100%.

This paper is organized as following: In section II, the pre-processing methods that normalize the gesture images are introduced. Section III presents several feature selection algorithms based on principal component analysis (PCA), generalized projection, and moment finding techniques. Section IV describes two classification algorithms and comparison styles for statistical decision making. Section V outlines and analyzes a comparative case study that ultimately illustrates why precedence ought to be

placed on developing efficient pre-processing methods over improving the other steps in the gesture recognition process. Also, particular interest is placed on the results that reveal the ability of the simple, developed algorithms to accurately detect the gesture set using very low-resolution images. Section VI further highlights the effects of various pre-processing steps on the subsequent feature selection and classification algorithms. Finally, section VII contains conclusions and a discussion about future work.

2. IMAGE PRE-PROCESSING

The main purpose of image pre-processing is to develop a representation of the images that preserves both the intra-class similarity and the inter-class distinctions between gestures. This goal is achieved by a consistent normalization of all the images once the hand has been properly segmented from its background, while maintaining, if not highlighting, the main features and shape of the articulated gestures. In this paper, hand identification and extraction are treated as separate processes from pre-processing, even though one can technically consider these part of pre-processing. In this way, any hand identification and localization algorithm can be added to the front end of the procedures and techniques developed in this paper to a get a complete gesture recognition scheme.

2.1 Pre-processing steps

The image processing can be broken down into the following steps that do not necessarily need to adhere to the given ordering: (1) grayscale conversion, (2) image resizing, (3) intensity normalization, (4) segmentation/background removal, (5) cropping, (6) arm/wrist removal, (7) centering, (8) orientation detection, and (9) rotation. To be more clear, these steps convert each raw image from color to grayscale. At some point, the image is down-sampled, meaning that it's resized in order to reduce the image resolution, usually to a square of the size $n \times n$, where n is an integer. The pixel intensities are normalized such that the brightest pixel is set to be pure white. The hand is selected from its background, which is set to be pure black. One may chose to eliminate the excess background pixels around the hand by cropping them out, which also effects the size (scale) of the hand within its frame. The excess arm and wrist regions of the hand image are removed, being cut-off at the bottom of the palm and set to pure black. The resulting hand region is set in the centroid of the image frame. The hand center is calculated by finding the average pixel position, weighted by the pixel intensities, of the hand. Then the hand orientation, or principal direction of the hand, can be determined by the best-fit line that satisfies the linear least squares of the hand pixel positions, again weighted by the pixel intensities. Finally, the hand is rotated by a simple linear transformation so that the hand points in a consistent direction within the frame; consistent to other like-gestures of the same class. The angle of rotation is calculated based off of the angle that the best-fit line makes with a horizontal line through the center of the hand. An example pre-processing scheme is depicted in Fig. 2, which shows the progression from the original color image to a final normalized gesture image (See images (a)-(g) in Fig. 2).
Erroneous black spots, which appear due to the rotation algorithm and rounding to the nearest pixel position, are filled in within the hand region. One could use morphological operations like dilation to handle these erroneous black spots [27]. However one needs to be careful at low image resolutions where dilation may start to erase the prominence of hand features and shapes. Even when one follows the dilation with an erosion operator in an attempt to restore the original features of the hand, the subsequent erosion only rarely can restore loss of shape or feature to very low resolution images.
Erosion morphological operations could also be implemented in attempt to suppress and eliminate background objects. However, these operations are best used at high image resolutions because, at low resolutions, they're only mildly successful in segmenting and removing the background and suffer from the same problem of erasing the prominence of hand features and shapes as was explained previously. One clever way to eliminate spurious, non-black, background pixels that sometimes appear away from the hand is to complete as many of the pre-processing steps as possible, with consideration to the computational time, before down-sampling the image. In the down-sampling process, the interpolation can often eliminate rogue pixels since their influence on the interpolation is minimal.
The algorithm for finding the principle direction can be confounded by a variety of hand postures, such as when the hand articulations don't have an obvious principal direction, and/or the arm/wrist region is masked and separated from the image frame by a dark sleeve, watch, or bracelet. In these cases the goal would be to get a consistent rotation of like-gestures.
Note that when doing gesture recognition, there are a variety of ways in which one can optimize the pre-processing in order to favor a certain image resolution. For instance, there are many parameters that control the automatic segmentation and background removal, which can be tailored for the best results at the desired image resolution. When one creates the training sets for real applications, it is best to choose and label the training set images after all the pre-processing and resizing has been accomplished, so that pre-processing errors do not skew the purity of the training set. Usually one desires only the best articulated gestures of each class to be included in the training set, thus labeling at a specific image resolution helps ensure the quality of the training set for that resolution.
Occlusion of the hand is dismissed in this paper as being a problem tangential to the primary trust of this paper, and so no steps are taken in the pre-processing to mitigate or overcome this effect.

2.2 Segmentation and background removal

Segmentation and background removal is an important step in pre-processing. It is understood that most modern applications to the hand gesture recognition field will want to take advantage of the movement of gestures that can be obtained from video sequences. This hand movement, along with well-established hand identification or localization techniques like the Viola-Jones detector [31], the Gaussian mixture model (GMM), kernel density estimation (KDE), and the max/min inter-frame differences technique [1], provide excellent ways to identify the hand in the background. Further, the use of passive infrared cameras in consumer electronic devices may afford a great deal of flexibility in automatically removing the *hot* hand from its background. No doubt, a noisy, complex, and/or cluttered background will make any detection algorithm less robust. However, background subtraction techniques and hand identification schemes are still a highly active area of current research and constitute an exceptionally challenging task in the image recognitions community. Pursuing and implementing hand identification schemes, although of tremendous importance, are somewhat tangential to the scope of the current interest and focus of this paper, and thus are omitted for the convenience of working with static hand gesture image sets that can provide evidence toward the strengths of quality pre-processing over elaborate feature extraction and classification schemes.
Some hand gesture databases with cluttered backgrounds, like the RWTH German Fingerspelling Database [6, 7], have been studied by others. However, some of the established hand gesture databases that are available on-line, such as the Cambridge hand gesture database [14] or the Sébastien Marcel datasets [5], also avoid the background subtraction issue by providing gesture sets

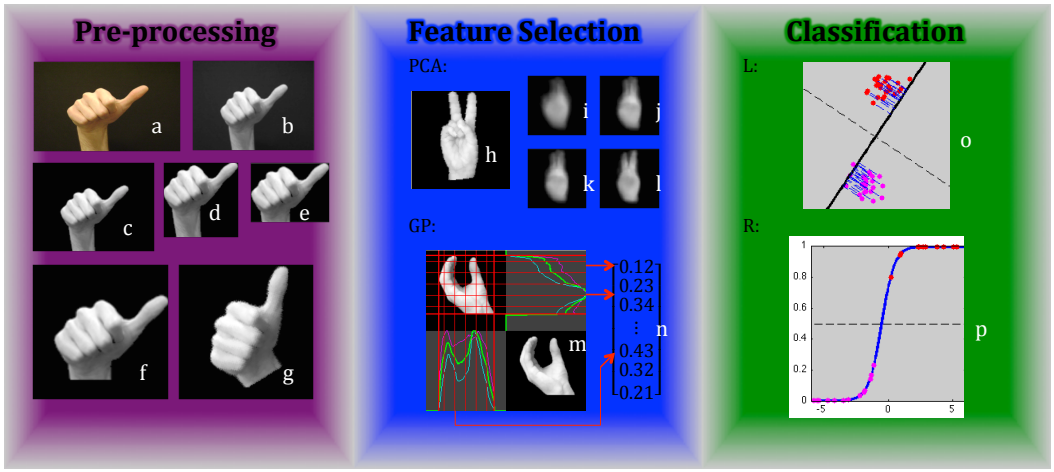

Fig. 2. This diagram exemplifies the recognition process starting with (left panel) an example pre-processing progression including: (a) original image, (b) grayscale conversion and intensity normalization, (c) segmentation and background removal, (d) crop, (e) arm/wrist removal, (f) downsample and center, and (g) rotation. Two feature selection methods are depicted (middle panel): PCA and GP. Using PCA, the image (h) can be reconstructed using (i) 1 feature, (j) 3 features, (k) 5 features, and (l) 7 features. Using GP, a feature vector (n) is formed by extracting horizontally and vertically a discrete number of the projection values (m). The classification methods used are (right panel) linear discrimination analysis (L) and logistic regression (R). Feature vectors (two classes) are illustrated by red and magenta dots. The optimal threshold (black dotted line) is determined in the training process.

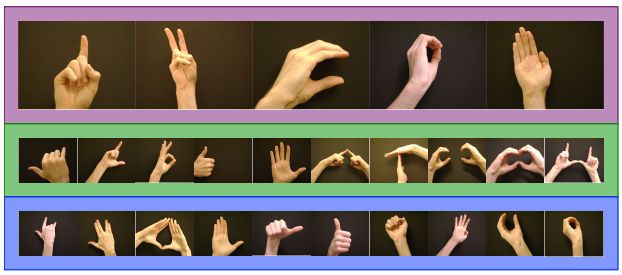

Fig. 3. These twenty-five hand images comprise a gesture lexicon that is implemented in this paper. The first row of five gestures were considered previously for a potential lexicon, which can be used as a computer mouse control application. Note the various lighting conditions, amount of arm/wrist region present in the frame, translations, scales, rotations, and slight occlusions contained within the dataset. There are even similar gestures that are articulated slightly differently in order to test the robustness of the recognition process. Other images in this dataset come from other people, sometimes with long sleeves of various colors and patterns.

with uniform backgrounds with well-centered and pre-oriented gestures. Indeed, these datasets remove almost all pre-processing from the gesture recognition process. In contrast, this paper's datasets use a fairly uniform, dark background under different lighting conditions, but leave the centering, orientation and pre-processing to the algorithms developed (see Fig. 3). Even with these significant liberties taken, a hand identification and background removal is still needed.

In order to identify the hand from the dark background, a distribution of the grayscale pixel intensities of each image is made. An advantage of using grayscale images over color images is that one doesn't need to consider the various hand colors of the general population, and low lighting levels can be normalized out in the pre-processing. In a histogram of the pixel intensity distribution, the dark background appears as a high frequency peak of low pixel brightness values. A threshold is automatically determined, based on the position of the minimum that separates these low intensity pixels from the brighter pixels of the hand. Then, all the pixels that are darker (of lower intensity) than the threshold are considered to be background, and are made to be pure black.

2.3 Arm/wrist removal

Creating an automatic algorithm for detecting and removing arm and wrist regions from an image can be quite difficult, yet makes an important impact on the performance of gesture recognition algorithms. Since this paper focuses on hand images that fill most of the frame of the image, one can safely deduce that the arm/wrist must exit the frame of the image. Assuming that the background removal is done well, sleeves or any other article of clothing or apparel may naturally cut the hand image off near the bottom of the palm or top of the wrist, as is desired. Otherwise, one can find the arm/wrist region on the frame of the image and follow it inward towards the hand, erasing the rows/columns of pixels that belong to the arm/wrist region as one goes. If one is viewing the hand from the front or back (not the side) there is a distinguishing feature that separates the arm from the hand; namely, a sudden increase in width of the arm/wrist/hand region as one transitions from the arm to the hand. Using this distinctive feature and tracking the arm/wrist widths as one moves toward the image center, an automatic algorithm for removing the arm/wrist region is created. Since this paper focuses on hand gestures as viewed from the front or back, removing arm/wrist regions from side-viewed hands is rendered moot. As with many automatic processes, there is a trade-off between robustness and computational processing time.

3. FEATURE SELECTION

There are many types of features that can be extracted from hand gesture images. In order to achieve high accuracies in gesture recognition, the features that are extracted need to be consistent within each gesture class of images but different between classes. It would impossible to create an exhaustive list of all the possible values associated with images that could be used as features. In this paper principal component analysis (PCA), generalized projections (GP), and image moment methods and some variations of those algorithms, are implemented as feature selection algorithms of the analysis that follows. All of these meth-

ods have been used previously in the context of gesture recognition [32, 2, 21, 2, 21, 28, 29, 34, 9, 16, 10, 11]. However, there seems to be lack of direct, head-to-head comparisons between both feature selection techniques and statistical testing methods as a function of image resolution and pre-processing methods in the recognition field.

Feature selection doesn't necessarily have to proceed the hand segmentation and preprocessing steps. For instance, it has been found that the aspect ratio of the hand can be an important feature, especially in the context of severe image down-sampling, where similar hand postures, e.g. a frontal view of an open palm hand with the fingers close together vs. a closed fist, can become confused at low image resolutions. This aspect ratio can best be extracted as a posture feature before the hand image is resized to be square. Additionally, there are many hand detection algorithms that find features within the image in order to detect the hand, and these features can also be used for the posture recognition [31, 1].

4. CLASSIFICATION

After feature selection, there are many proposed supervised pattern recognition algorithms that can statistically learn the parameters from labeled gesture images. Classification algorithms often learn the best parameters for separating gesture classes through an optimization on a training set. After the best parameters have been found, classification routines will project image features onto the classification value space where a decision is made as to what class the image belongs. Two standard methods are applied here: linear discriminant analysis [23] and logistic regression [22]. Other, more sophisticated, methods exist and are well known, such as neural networks [15], support vector machines (SVM) [30], adaptive boosting [12, 13, 4], Hidden Markov Models (HMM) [25, 26, 24], and conditional random fields (CRF) [19], and these methods have been shown to perform well even in real-time scenarios.

In most gesture recognition problems there are multiple classes present. In this paper, two classification styles are considered: pairwise testing and a one-versus-the-rest strategy. Pairwise classification checks every possible pairing of classes, deciding in which class a gesture is most likely to be. A one-versus-the-rest classification style checks each individual class against all other classes, which are clumped together and treated as a single class. A tree-based hierarchy classification [18], which narrows the search algorithm and eliminates the need for comparing all pairs of classes, can improve the scalability of having many gesture classes. Because only a few gesture classes are used in this paper, either gesture lexicons of five or twenty-five depending on the experiment, a tree-based hierarchy classification style is not implemented.

5. A COMPARATIVE CASE STUDY

In order to highlight the overall influence of pre-processing in the gesture recognition process, a case study is made using various recognition techniques at different image resolutions in order to discover what method combinations produce the best results. Only five gesture classes are explored in this study, with images mostly taken from two subjects, and all of the 189 hand images are well articulated, with similar sizes (a consistent scale), similar orientations within each gesture class (less than $\pm 10\,\mathrm{deg}$ rotation variance), and with little to no extra arm/wrist region showing in the image frame. There are 38 class 1 images (one-click), 48 class 2 images (double-click), 29 class 3 images (scroll-up), 45 class 4 images (scroll-down), and 29 class 5 images (move cursor). No outside gestures are inserted into the image set in this study. The raw images were gathered at several different times, with slightly different lighting conditions. This idealized data set is amassed and implemented in order to show that proper pre-processing can still have profound effects on the results. Without pre-processing, feature selection and classification schemes perform quite poorly (approximately 40% accuracy).

Table 1. Acronyms associated with feature selection, classification method and classification style.

Feature Selection Method	
S/PCA	singular-value decomposition/ principal component analysis
2DPCA	two-dimensional principal component analysis
GP	generalized projections
DGP	derivative of generalized projections
CGP	circular generalized projections
IM	invariant moments
Classification Methods	
R	logistic regression
L	linear discrimination analysis
Classification styles	
P	pairwise
OR	one versus the rest

In this study two different pre-processing methods are employed, which will be called reduced pre-processing (RPP) and cropped pre-processing (CPP). Both of these methods start by reading in a raw color image. The RPP method completes the pre-processing steps mentioned in the Sec. 2 in the same order as they are listed, with exception of not completing steps (5) and (6), i.e. the cropping and the arm/wrist removal. The CPP method crops the unnecessary spaces above and below the hand region, and makes sure the hand resides on the center with wrist against the bottom and highest finger tip touching the top. CPP does some extra work to ensure that the background is completely removed from the image by removing any extraneous, isolated bright dots in the image frame. Images are not resized into a square $n \times n$ image until after the background removal and cropping has been completed. Then CPP centers and rotates the image as is done in steps 7-9. Linear interpolation is then used to get the pixel value of the rotated hand. Finally, the CPP procedure removes the arm/wrist region from the hand image (step 6 is done last). Because CPP does more pre-processing and in a more perceptive and insightful way, one should expect the CPP to perform significantly better.

For this comparative study, only three feature selection methods are employed, namely, the traditional PCA method, generalized projections, and derivatives of generalized projections. These feature selection methods all extract features that are distinct enough from one another so as to allow one to determine if one method is significantly advantageous over the others. Moreover, the fact that these methods lack some of the invariance properties (scale, translation, rotation, and skew), allows for a better investigation of the pre-processing effects on the recognition process.

The two pre-processing and the three feature selection schemes are used in combination with the LDA and logistic regression classification methods. Both the pairwise and the one-versus-the-rest classification styles are also implemented in combination with the other strategies. Table 1 provides the acronyms for the various methods that will be implemented.

In order to best measure the error rates in properly recognizing the hand gestures, 100 rounds of cross-validation testing is completed for all of the method combinations found in Table 1. In such tests, the training and testing sets are randomly chosen from the entire set of 189 images, ensuring that each gesture class has at least a few images in the training set. The target is to have the training set size be about 20% of the entire image set size. In each round, the images are randomly reshuffled, and new training and testing sets are obtained. After 100 rounds of testing, the

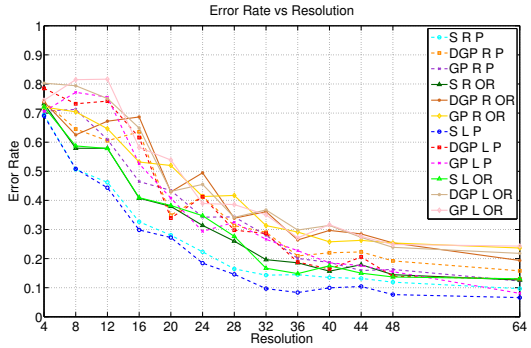

Fig. 4. The average error rate of various recognition algorithms as a function of the resolution of hand images using RPP pre-processing. Compare the trends of these plots with those of Figure 5, which only differ because of another pre-processing scheme implemented.

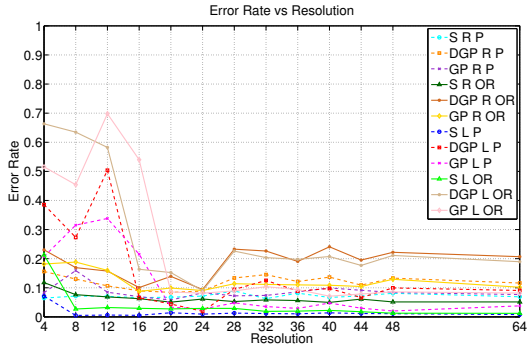

Fig. 5. The average error rate of various recognition algorithms as a function of the resolution of hand images using CPP pre-processing. Compare the trends of these plots with those of Figure 4, which only differ because of another pre-processing scheme implemented.

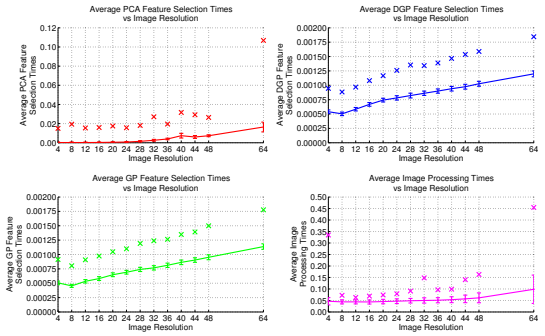

Fig. 6. The average computation times in order to complete the feature selection and pre-processing phases of the hand gesture recognition problem. All the "x"s mark the maximum computation times, and the error bars represent one standard deviation above and below the average times.

final error rate is produced by averaging over the 100 rounds of testing.

Figures 4 and 5 summarize the average error rate results. Associated with each line on the graph are three acronyms from Table 1 that highlight the feature selection, classification method and classification style. The resolution is the number of pixels in the x or y direction of the image, which is square. Figure 4 illustrates the use of the RPP pre-processing method while Fig. 5 shows the CPP pre-processing technique. Fundamental to these graphs is the accuracy of the gesture recognition as a function of the resolution of the images. A lower image resolution will guarantee a faster, real-time algorithm. Figure 4 starts with error rates between about 70% and 80% at the 4×4 pixel resolution and achieves about 7% to 25% error rates at the 64×64 pixel resolution. Figure 5 fairly consistently achieves error rates between about 1% and 20% at all the pixel resolutions. For well resolved images (64×64 pixel resolution), the difference in the average accuracy of the techniques shifts from $\approx$85% (RPP) to $\approx$95% (CPP). The difference is much more pronounced for lower resolutions (8×8 pixel resolution) where a number of the CPP methods can still achieve $\approx$99% average accuracy whereas any RPP technique is only 40% accurate. The most striking aspect of the CPP at such low resolutions is its ability to distinguish between classes when the gestures are no longer recognizable to the human eye. Note that the 8×8 resolution is especially attractive for rapid, low overhead detection applications on hand-held electronics.

To summarize, the CPP method allows for high accuracy at low resolution. The actual accuracies presented in these graphs are not as significant as the general trends of the plots, which, in Fig. 4, show a nearly monotonic decline in error rates as the image resolution increases, and in Fig. 5, show mostly consistent accuracies for all resolutions. One would expect that higher resolution images would be easier to classify since the features become more salient and detailed. Of course this is only true to a certain point, eventually further detail does not contribute any new pertinent information about the features; this is why Figure 4 starts to level off near the 64×64 pixel resolution. However, Fig. 5 shows that even better than the expected performance can be achieved given the proper pre-processing, using the same simple, well-known feature selection and classification methods. In addition to the accuracy as a function of resolution, the computing time required to perform the recognition algorithm is considered. As stated previously, the bulk of the algorithm processing time is found in the pre-processing of the images. Figure 6 demonstrates the computed processing time, extracting 10 features, for the PCA based method (top left panel) as well as the generalized (bottom left panel) and derivative of generalized projection (top right panel) methods. All results are presented as a function of the image resolution. The bottom right panel of this figure shows the pre-processing times as a function of resolution for the RPP method.

6. ANALYSIS OF PRE-PROCESSING

In order to further elucidate the importance of efficient pre-processing methods, consider the results of Table 2 for which a variety of hand gesture recognition method combinations are used to detect the class of the hands. The timing results produced here and throughout the paper are generated on a 1.86 GHz Intel Core Duo processor, which is fairly standard among moderately powered laptops. All the images have a resolution of 32×32 pixels, and the results are averaged over 1349 images, which contain 20 more gestures other than the 5 gestures previously mentioned, and many non-gestures and/or poorly articulated gestures (See Fig. 3). This set of images also has a larger rotational variance among the images. Here $\bar{t}$ is the average time to complete the method, with a corresponding standard deviation σ_t. The minimum $t_{\min}$ and maximum $t_{\max}$ times are also given. The accuracy is measured using both LDA and logistic regression classification schemes, and are presented under the Success Rate column. These success rates are averages of the *within class success rate*, meaning when images are correctly labeled within their respective class, and the *out-of-class success rate*, meaning when images are correctly not labeled to belong to classes to which they do not belong. All of the 1349 images used in these performance tests have been pre-processed using all of the pre-processing steps listed in Sec. 2 as is illustrated in Fig. 2. The test

Table 2. Performance of various methods of the gesture recognition process at an image resolution of 32×32 pixels.

Feature Selection Method	$\bar{t}$ [msec]	σ_t [msec]	$t_{\min}$ [msec]	$t_{\max}$ [msec]	Success Rates
PCA	2.58	0.946	2.33	24.6	L $\rightarrow$ 0.889 R $\rightarrow$ 0.885
2DPCA	0.340	1.658	0.200	34.0	L $\rightarrow$ 0.940 R $\rightarrow$ 0.962
GP	1.709	0.531	1.482	15.90	L $\rightarrow$ 0.873 R $\rightarrow$ 0.861
DGP	1.799	0.0366	1.767	2.54	L $\rightarrow$ 0.857 R $\rightarrow$ 0.847
CGP	34.5	2.21	29.7	46.0	L $\rightarrow$ 0.897 R $\rightarrow$ 0.859
IM	10.80	1.218	10.47	50.3	L $\rightarrow$ 0.782 R $\rightarrow$ N/A
Classification Method	$\bar{t}$ [msec]	σ_t [msec]	$t_{\min}$ [msec]	$t_{\max}$ [msec]	
R	0.1008	0.0530	0.0908	1.090	
L	0.0621	0.0266	0.0554	0.737	
Pre-Processing Method	$\bar{t}$ [msec]	σ_t [msec]	$t_{\min}$ [msec]	$t_{\max}$ [msec]	
All	88.5	21.5	44.6	195.7	

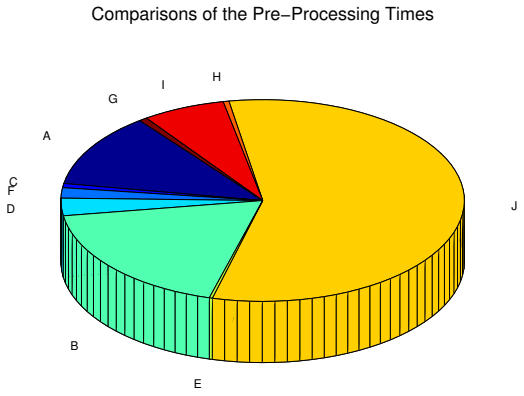

Fig. 7. The relative comparisons of pre-processing times averaged over 1349 images and 13 different image resolutions, ranging from 4×4 images to 64×64 images. The letter labels correspond to the pre-processing steps as follows: A. Grayscale Conversion' (11.87%), B. Image Resizing (18.31%), C. Intensity Normalization (0.67%), D. Background Subtraction (2.81%), E. Cropping (0.25%), F. Arm/Wrist Removal (1.67%), G. Centering (0.64%), H. Orientation Detection (0.47%), I. Rotation (6.63%), and J. Other (56.69%). In order to complete all the pre-processing steps, the average computation time was 0.0847 sec.

that included invariant moments (IM) and logistic regression (R) failed to recognize the hand images because of a failure in the optimization routine that trains the logistic regression scheme.

Table 2 reiterates the fact that pre-processing is the most time-consuming step of the recognition process, whereas the classification scheme is the least costly step. In fact, these pre-processing times most likely form a lower bound for the times that would occur in actual hand gesture detection applications, which would need to implement more elaborate and robust identification and localization routines not included in these results. Since real-time applications depend on fast processing, it's apparent that improving the pre-processing performance is paramount. Table 2 also shows that the both classification schemes, LDA and logistic regression, produce similar results in accuracy averaged over *within* and *out-of-class success rates*, both of which are needed for proper hand detection. This provides some evidence that if the pre-processing is done well, and the best features are extracted from the images, then the classification scheme need not be overly sophisticated.

Figure 7 further breaks down the pre-processing time into its constitutive components. A relative comparison of the various pre-processing steps are illuminated so that one can visually determine their importance. This is said with the caveat that the "Background Subtraction" appears to be less significant (2.81%) to the overall pre-processing time than it should be because identifying and localizing the hand in cluttered backgrounds is not considered. The "Other" pre-processing step (56.69%) is dominated by the process of importing the raw image.

With the importance of pre-processing in mind, consider a new performance study that aims to emphasize the contributions of the various steps of the pre-processing, and whose results are illustrated in Fig. 8. Only the PCA feature selection method is implemented in this test because of its sensitivity to variations in translation, rotation, and scale. Also, only the LDA classification scheme with a one-vs-the-rest comparison style will be used to determine the gesture class of the hand images. This study uses only one well articulated image for each gesture class, there being five different classes in total, as before. These images do have some significant portions of the arm/wrist region showing in the frame of the images.

Recalling the nine pre-processing steps from Sec. 2, these images are pre-processed in 12 different ways: The first six ways stem from completing all the possible pre-processing steps ("All"), and then, in the other five ways, a single pre-processing step is removed. The last six ways stem from completing as little pre-processing as possible ("None"), and then, in the remaining five ways, a single pre-processing step is added. However, even the images that are pre-processed under "None" are converted to grayscale, resized to the desired image resolution, and have the hand segmented from the background. In order to depict the effects of not having certain pre-processing steps, noise is added to the images in order emphasize the missing step(s); i.e. when no image cropping is done, the hand size is purposely rescaled in its frame to appear much smaller than the ideally articulated gesture. Other image noise includes de-centering the hand within the frame of the image, and a 90 degree rotation of the hand counter-clockwise.

The success rates are averaged over the five images of each pre-processing scenario and over 13 different image resolutions, ranging from 4×4 images to 64×64 images. Figure 8 illustrates the results of this test with both the within and out-of-class success rates for all 12 pre-processing scenarios.

As would be expected, the images that were pre-processed with "All" of the possible steps performed with perfect accuracy because the classification was trained (learned) using these images. The success rates of these fully pre-processed images serve as an upper bound to which the other pre-processing methods can be compared against. Likewise, success rates for the images that were pre-processed with "None" of the possible steps, serve as a lower bound to which the other pre-processing methods can be compared against. In this case, the lower bound is about 40% accuracy. Thus, the actual accuracies are not as meaningful as the relative differences in accuracy between each pre-processing scenario.

Figure 8 clearly shows that pre-processing is key to accurately recognizing hand gestures. The normalization of the pixel intensities seems to be the least important of the pre-processing steps in this study, while rotation seems to be the most important. Some pre-processing steps depend on other steps; for instance, centering the hand within the frame of the image is useless without first removing the excess arm/wrist regions. Not accounting for the PCA method's sensitivity to translation, scale, and rotation variances in the pre-processing causes the recognition rates to suffer.

One way of avoiding the costly pre-processing would be to use translation, scale, and/or rotation invariant feature selection methods. Even still, the pre-processing procedure can't be completely avoided. For instance, the arm/wrist detection is still

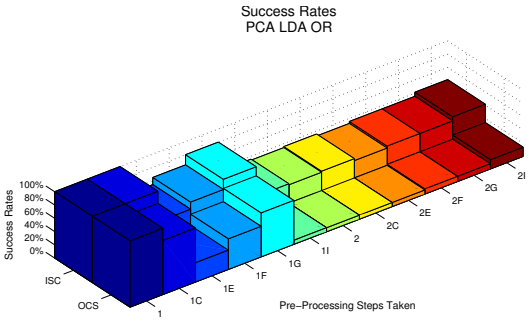

Fig. 8. The success rates of recognizing 5 hand gestures, just using 1 image for each class (gesture). Within class success rates (ICS) are calculations of correctly labeling images within their respective classes. Out-of-class (OCS) success rates are determined from correctly not labeling images into classes to which they do not belong. The success rates are averaged over the five images of each pre-processing scenario and over 13 different image resolutions, ranging from 4×4 images to 64×64 images. The alpha-numeric labels correspond to the pre-processing methods as follows: 1. All, 1C. All But Intensity Normalization, 1E. All But Cropping, 1F. All But Arm/Wrist Removal, 1G. All But Centering, 1I. All But Rotation, 2. None, 2C. Only Intensity Normalization, 2E. Only Cropping, 2F. Only Arm/Wrist Removal, 2G. Only Centering, and 2I. Only Rotating.

needed in order to for any feature selection method to properly identify the true center of the hand. Also, identification, localization, segmentation, and background removal will always be a crucial aspect of pre-processing, and is most likely the most time consuming part as well. Additionally, rotation may still be an issue for some background removal methods to work properly. Finally, sometimes the best features may not come from translation, scale and rotation invariant methods. Referring back to Table 2, it is clear that the PCA methods out-performed the invariant methods of circular generalized projects and invariant moments in both accuracy and speed.

7. CONCLUSIONS AND OUTLOOK

Various combinations of methods that perform the steps of the recognition process were implemented and studied. It was seen that, even at image resolutions as low as 8×8 pixels, accuracies of 99%, using PCA feature selection, and 95%, using generalized projection techniques, could be achieved given the right kind of pre-processing (CPP), and of course with a somewhat idealized dataset that starts with dark, uniform backgrounds. It is likely that such accurate results are achieved because the pre-processing is tailored to this particular application and experiment. Not all applications will be able to produce such performance, but the pre-processing schemes can be optimized around the given application.

The classification scheme implemented seemed to be least influential factor in producing high accuracy recognition algorithms. In fact, all of the method combinations employed in the case study performed reasonably well, and had accuracies that scaled with the image resolution, as would be expected. This means that there are already well-known feature selection methods and classification schemes that can perform excellently if they're executed on properly pre-processing images. Since pre-processing can't be avoided, and because it dominates the overall computational expense (processing time) and performance of the recognition process, it is a valid conclusion that more effort and focus ought to be committed to improving and optimizing the pre-processing stage of the recognition process, instead of constructing more elaborate and sophisticated feature selection and classification schemes. This strategy also depends on the progress and development of algorithms for identifying the hand in a potentially complex image background, a field of ongoing and intense research for any gesture recognition software.

Knowing that the speed, efficiency, and robustness of gesture recognition problems depend on the quality of the pre-processing of the images, further investigations will be made in order to determine the best features that can be selected, and what trade-offs between extra pre-processing and invariant features are warranted. Also, one ought to consider how to best select an appropriate set of hand gestures that are optimally well separated in a statistical sense, and yet are easy to articulate and are suitable to the desired application. Future work will include methods for determining the best gestures to be used in a small vocabulary, and the corresponding best features for the greatest statistical separation of these gestures.

Future work will incorporate modern hand identification routines and attempt to improve the pre-processing so that the best results can be achieved in even shorter times. Indeed, the differences in accuracy in Figs. 7 and 8 indicate that pre-processing has more of an overall effect on accuracy than does any other step or method in the recognition process. Eventually dynamic gestures will be considered, with real backgrounds that are not so ideal as the dark, uniform backgrounds used throughout this paper. Further, one may envision using the gestures to interface with a laptop or some other portable electronic device so that the functionalities of accessories, like an external mouse or a laser pointer, could be replaced by hand gestures captured through a built-in camera. Ultimately, algorithms will be developed that will track and recognize hand gestures in real-time for applications of this nature.

Acknowledgements

J. N. Kutz acknowledges support from the National Science Foundation (NSF) (DMS-1007621) and the US Air Force Office of Scientific Research (AFOSR) (FA9550-09-0174).

8. REFERENCES

[1] Y. Benezeth, P. Jodoin, B. Emile, H. Laurant, and C. Rosenberger. *Review and Evaluation of Commonly-Implemented Background Subtraction Algorithms.* In *IEEE International Conference on Pattern Recognition*, pages 1–4, 2008.

[2] Henrik Birk, Thomas B. Moeslund, and Claus B. Madsen. *Real-Time Recognition of Hand Alphabet Gestures Using Principal Component Analysis.* In *In 10th Scandinavian Conference on Image Analysis*, 1997.

[3] S. Cobos, M. Ferre, M. A. Sanchez-Uran, J. Ortego, and R. Aracil. *Human hand descriptions and gesture recognition for object manipulation. Comp. Meth. Biomechanics Biomedical Eng.*, pages 1–13, 2010.

[4] H. Cooper and R. Bowden. *Large Lexicon Detection of Sign Language. Human-Computer Interaction, Lecture Notes in Computer Science*, 4796:88–97, 2007.

[5] Marcel datasets. (http://www.idiap.ch/resource/gestures/).

[6] P. Dreuw. *RWTH German Fingerspelling Database*, 2005.

[7] P. Dreuw, T. Deselaers, D. Keysers, and H. Ney. *Modeling Image Variability in Appearance-Based Gesture Recognition.* In *ECCV Workshop on Statistical Methods in Multi-Image and Video Processing*, pages 7–18, 2006.

[8] A. Erol, G. Bebis, M. Nicolescu, R. Boyle, and X. Twombly. *Vision-based hand pose estimation: A review. Computer Vision and Image Understanding*, 108:52–73, 2007.

[9] G. C. Feng and P. C. Yuen. *Variance projection function and its application to eye detection for human face recognition. Pattern Recogn. Lett.*, 19:899–906, 1998.

[10] J. Flusser. *On the Independence of Rotation Moment Invariants. Patt. Recog.*, 33:1405–1410, 2000.

[11] J. Flusser. *Moment Invariants in Image Analysis. World Academy of Science, Eng. & Tech.*, 11:376–381, 2005.

[12] Y. Freund and R. Schapire. *A Desicion-Theoretic Generalization of On-Line Learning and an Application to Boosting. Journal of Computer and System Sciences*, 55:119–139, 1997.

[13] Y. Freund and R. Schapire. *A Short Introduction to Boosting. J. of Jap. Soc. for Artificial Intelligence*, 14:771–780, 1999.

[14] Cambridge hand gesture datasets. (http://www.iis.ee.ic.ac.uk/tkkim/ges_db.htm).

[15] S. Haykin. *Neural Networks: a comprehensive foundation.* Macmillan, New York, 1994.

[16] M. Hu. *Visual Pattern Recognition by Moment Invariants. IRE Transactions Information Theory*, IT-8:179–187, 1962.

[17] B. Ionescu, D. Coquin, P. Lambert, and V. Buzuloiu. *Dynamic Hand Gesture Recognition Using the Skeleton of the Hand. EURASIP J. on App. Signal Processing*, 13:2101–2109, 2005.

[18] T. Khoshgoftaar, E. Allen, L. Bullard, R. Halstead, and G. Trio. *A Tree-Based Classification Model for Analysis of a Military Software System.* In *IEEE Comp. Soc. Proc.of the IEEE High-Assurance Systems Engineering Workshop*, pages 244–251, 1997.

[19] J. Lafferty, A. McCallum, and F. Pereira. *Conditional Random Fields: Probabilistic Models for Segmentng and Labeling Sequence Data. Proceedings of the International Conference on Machine Learning*, pages 282–289, 2001.

[20] J. Lin, Y. Wu, and T. Huang. *Modeling the Constraints of Human Hand Motion.* In *In IEEE Human Motion Workshop*, pages 121–126, 2000.

[21] R. Lockton and A. W. Fitzgibbon. *Real-time gesture recognition using deterministic boosting.* In *Proceedings, British Machine Vision Conference*, 2002.

[22] S. Menard. *Logistic regression: From Introductory to Advanced Concepts and Applications.* SAGE Pub., Inc., 2010.

[23] S. Mika, G. Rätsch, J. Weston, B. Schölkopf., and K. Müller. *Fisher Discriminant Analysis with Kernels.* In *Proc. IEEE Workshop Neural Networks for Sig. Proc.*, pages 41–48, 1999.

[24] A. Nefian and M. Hayes III. *Hidden Markov Models for face identification.* In *Proceedings of the Int'l Conf. on Acoustics, Speech, and Signal Processing*, pages 2721–2724, 1998.

[25] M. Russell and R. Moore. *Explicit Modelling of State Occupancy in Hidden Markov Models for Automatic Speech Recognition.* In *Proceedings of the Int'l Conf. on Acoustics, Speech, and Signal Processing*, pages 5–8, 1985.

[26] F. Samaria and S. Young. *HMM-based architecture for face identification. Image and Vision Comp.*, 12:537–543, 1994.

[27] P. Soille. *Morphological Image Analysis: Principles and Applications.* Springer-Verlag, New York, second edition, 2010.

[28] M. Turk and A. Pentland. *Eigenfaces for Recognition. Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[29] M. Turk and A. Pentland. *Face Recognition Using Eigenfaces.* In *Proceedings of IEEE Computer Society Conference on Comp. Vision and Patt. Recog.*, pages 586–591, 1991.

[30] V. Vapkin. *Statistical Learning Theory.* Wiley & Sons, 1998.

[31] P. Viola and M. Jones. *Robust Real-time Object Detection.* In *IEEE ICCV Second Int'l Workshop on Stat. and Comp. Theories of Vision*, volume 20, pages 1254–1259, 2001.

[32] H. Wu and A. Sutherland. *Dynamic Gesture Recognition Using PCA with Multi-scale Theory and HMM. Image Extr. Segm. Recogn.*, 4550:132–139, 2001.

[33] X. Zabulis, H. Baltzakis, and A. Argyros. *Vision-based hand gesture recognition for human-computer interaction*, 2009. The Universal Access Handbook. LEA.

[34] D. Zhang and Z.-H. Zhou. $(2D)^2$*PCA: Two-directional two-dimensional PCA for efficient face representation and recognition. Nanocomputing*, 69:224–231, 2005.