Yi-Ting Chou, chou1@cs.uiuc.edu
Peter Bajcsy, pbajcsy@ncsa.uiuc.edu

Automated Learning Group
National Center for Supercomputing Applications
605 East Springfield Avenue, Champaign, IL 61820

Toward Face Detection, Pose Estimation and Human Recognition from Hyperspectral Imagery

Abstract

We present our work on face detection and pose estimation from hyperspectral imagery. The long-term goal of our research is to explore the use of hyperspectral imagery for building an automated system for human recognition. We report our preliminary results obtained with the face detection algorithm proposed by Paul Viola and Michael Jones. The algorithm was extended to work not only with monochromatic images but also with any dimensional images by pre-processing images using principal component analysis (PCA) and applying the algorithm to the first principal component.

The second part of this report focuses on spectral-based and spatial-based pose estimation. Due to the complexity of human head pose estimation; we analyze the problem of human hand pose estimation in our initial study. We investigate the hand material properties by spectral and spatial analyses, and identify the primary limitations of each analysis from the pose estimation viewpoint. A summary of pros and cons for spectral and spatial analyses is provided at the end.

1 Introduction

Human identification (HID) have becomes one of the most prosperous fields in the past ten years. Among the wide range of HIDs, human face analysis is the one that has been devoted the most extensive research in computer vision. Human face analysis research can be divided into several subfields: *face detection*, *face tracking*, *face recognition*, *pose estimation*, and *facial expression recognition*. In this paper, we will focus on *face detection* and *pose estimation*.

In section 2, we discuss the face detection algorithm developed for our system. Face detection is usually the first stage for any automated face analysis system. After extracting the faces from the image, we can further perform recognition, pose estimation, and etc. Several face detection algorithms have been developed in the past decades, and

the following surveys provide a good overview of all existing algorithms [1] [2]. For example, according to [1], face detection methods can be classified into four categories: *knowledge-based methods*, *feature invariant approaches*, *template matching methods*, and *appearance-based methods*.

In this work, we focus on the algorithm proposed by Viola and Jones [7], which can be view as an *appearance-based method*. The *appearance-based method* is a supervised classification technique that is based on learning from a set of training images, and capturing the most representative variability of facial appearance. Section 2.1 describes invariant image features that represent facial characteristics. Each feature provides an input for building a supervised classifier (denoted as a weak classifier). Section 2.2 outlines how to select the most informative features and combine the weak classifiers into the final strong classifier for face detection. This step is executing using a Boosting technique, where the final strong classifier is a linear combination of the weak classifiers. Section 2.3 describes how to achieve the real-time detection by decomposing the strong classifier into several parts. Section 2.4 presents a principal component analysis method that is used for extending the grayscale face detection algorithm to process color and hyperspectral images as well. Section 2.5 shows the experiment results of the implemented face detection system.

In section 3, we discuss several possible approaches to pose estimation from hyperspectral imagery. The goal of pose estimation is to calculate 3D orientation information from a 2D image. In general, pose estimation still remains a hard problem in the computer vision domain. Previous pose estimation methods can be roughly divided into two categories: *template-matching methods* and appearance-based methods. *Template-matching methods* usually construct 3D head model as template and match it with 2D images [3] [4]. *Appearance-based methods* view pose estimation problem as a classification problem. It usually requires several training images for each pose [5] [6]. In this paper, we investigate if spectral information could be sufficient for pose estimation.

In section 3.1, we introduce hyperspectral imaging, and describe why it could provide more information for pose estimation. In section 3.2, we describe a hyperspectral camera calibration procedure. In section 3.3, we discuss segmentation of hyperspectral images with the goal of separating image background from hand regions. In section 3.4, we present our experiments for pose estimation based on spectral information and discuss the distance metrics for skin/nail classification. In section 3.5, we rely on the phone illumination model used by computer graphics community and compute spatial variance as a feature for detecting nail regions. In section 3.6, we summarize the limitations of spectral-based and spatial-based pose estimation approaches.

2 Face Detection

In order to achieve a real-time face detector, we implemented the algorithm proposed by Paul Viola and Michael Jones [7] [8]. The algorithm starts with a large number of classifiers and follows by using the *AdaBoost* technique [9] [10] to obtain a single classifier for face detection. To meet the requirements of a real-time system, feature detection is implemented as a cascade process that divides a single feature detector into

several large ones. The feature detectors applied early in the process eliminate most of the false image samples, so that the feature detectors applied later have to deal with very few image samples.

We also extended the original face detection algorithm operating on grayscale images to process color and hyperspectral images by performing a principal component analysis (PCA) and applying the algorithm to the first principal component. Faces detected in color and hyperspectral images can be used for further analyses, such as, face recognition or pose estimation.

2.1 Features

In general, image features should be very simple, easy to compute, and robust to background noise, so that face detection can take place in real time and perform well. We used seven types of such image features and they are shown in Figure 1.

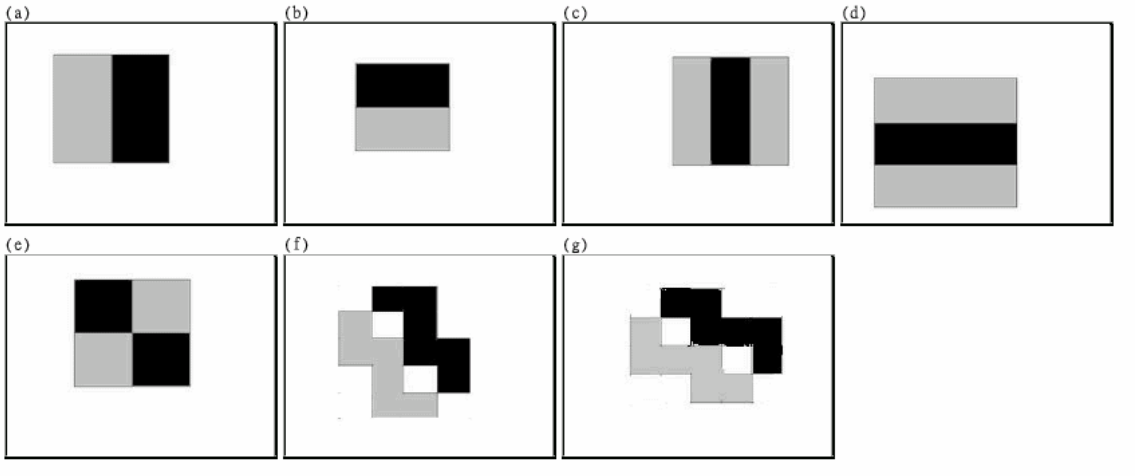


Figure 1: Filters defining seven face detection feature types. The feature value is the difference between the sums of the pixel values, which lie within the gray region subtracted by the sum of pixel values within the black region. Feature types (a-e) correspond to symmetric two-, three- and four-rectangle characteristics of human face. Feature types (f) and (g) model asymmetric characteristics of human face.

Each feature type could be of any size and located at any position in image. The feature value is the sum of the pixel values in gray area subtracted by the sum of pixel values in the black area. While building a model for supervised classifier during its training phase, we worked with 24 by 24 pixel training images, and chose this pixel size to be our base sub-window size. We enumerated exhaustively all possible spatial overlays of all seven filters in one 24 by 24 pixel image. The total number of possible spatial overlays is summarized for each feature type in Table 1.

Table 1: Number of features per feature type extracted from one 24x24 training image.

Class (a)	Class (b)	Class (c)	Class (d)	Class (e)	Class (f)	Class (g)	Total
43,200	43,200	27,600	27,600	20,736	3,300	3,300	168,936

We calculated that there are totally 168,936 features for a 24 by 24 image (576 pixels only). If we compute the sum of gray area and black area for each feature separately, then the complexity of extracting all features would be $O(h^3 w^3)$, where h and w are the height and width of a training image respectively.

One should be aware that although feature extraction is not a time critical operation during a training phase, it becomes a time critical computation during a testing phase. Thus, we try to decrease the time for extracting all features by introducing the concept of integral image and dynamic programming. We construct an *integral image* (*I*) for each training image. An *integral image* is formed as a table of locations (x, y) with the sums of the pixels above and left to it, inclusive. The mathematical expression for computing an integral image can be written as:

$$I(x, y) = \sum_{x' \le x, y' \le y} I^{ORIG}(x', y')$$

where $I^{ORIG}(x, y)$ is the original image intensity at location (x, y). The integral image table can be computed via dynamic programming:

$$I(x, -1) = I(-1, y) = 0$$
$$I(x, y) = I(x-1, y) + I(x, y-1) - I(x-1, y-1) + I^{ORIG}(x, y)$$

The integral image computation requires $O(hw)$ operations to construct the table, which is linearly increasing with the size of the original image. Each feature can be computed from the integral image table by a few operations. For example, a feature type (a) can be computed by performing only seven operations (see Figure 2).

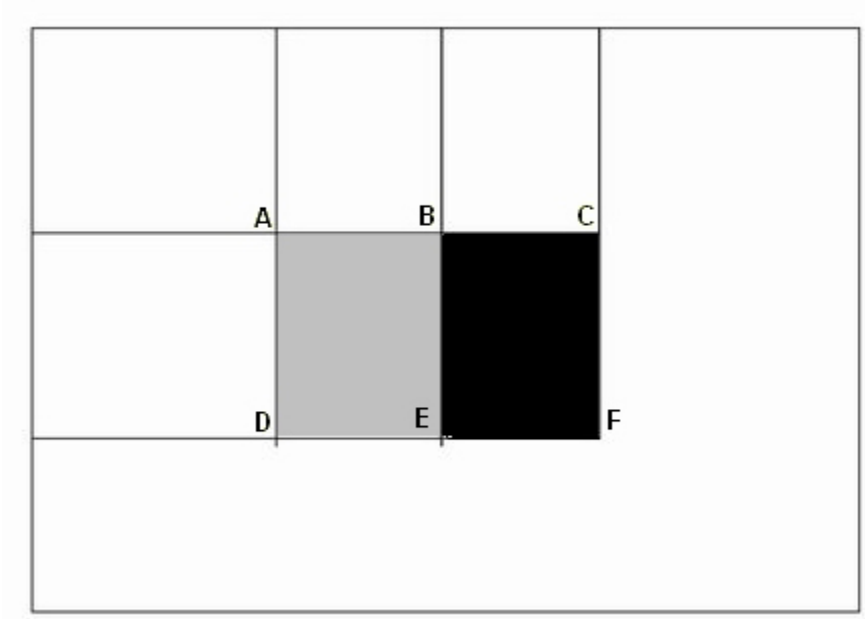


Figure 2: $A, B, C, D, E,$ **and** F **are the values of an integral image table. The feature of type (a) can be computed by subtracting black pixels** $(F + B - C - E)$ **from gray pixels** $(E + A - B - D)$**, which equals to** $(F + B - C - E) - (E + A - B - D) = (2 \times B + D + F) - (2 \times E + A + C)$**.**

The complexity of enumerating every feature is roughly $O(h^2 w^2)$. The total complexity of feature extraction is $O(hw) + O(h^2 w^2) = O(h^2 w^2)$ in comparison with the original complexity $O(h^3 w^3)$.

2.2 Classification Function

According to Table 1, there are N=168,936 features that should be extracted from each 24 x 24 image sub-window. However, if the number of training images M is less than N then the classification problem becomes over-determined and therefore over-fitting would likely occur. In our study, we obtained M=10,000 training images and had to deal with the case of M<N. Our approach is based on one of the *boosting* techniques in order to build the final strong classifier that does not over-fit the data.

According to [11], the goal of *boosting* is to improve the accuracy of any given learning algorithm. First, create a weak classifier with accuracy of more than 50%. Second, add new weak classifiers to learn new training sets that would increase accuracy of the new classifier. The process will iterate and choose the most informative features to improve the overall classifier.

AdaBoost, adaptive boosting, is one of the variations of boosting. It allows adding weak classifiers linearly to current strong classifier, and the overall training error rate will be a monotone non-increasing function with domain of the total number of chosen weak classifiers. The algorithmic steps are provided in Figure 3.

- Given: $(x_1, y_1), \cdots, (x_n, y_n)$, where $y_i = 0$ for negative (non-face) and $y_i = 1$ positive (face) training images.
- Compute the weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0,1$ respectively, where m and l are the number of negatives and positives training images.
- For $t = 1, \cdots, T$:
  - Normalize the weight:

$$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,i}}$$

  - For each feature, j, train a classifier h_j and evaluate its error with respect to w_t as $\varepsilon_j = \sum w_i \left| h_j(x_i) - y_i \right|$.
  - Choose the classifier, h_t with the lowest error ε_t.
  - Update the weights:

$$w_{t+1,i} = w_{t,i}\beta_t^{1-e_i}$$

where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise,

and $\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$.

- The final strong classifier is:

$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2}\sum_{t=1}^{T} \alpha_t \\ 0 & otherwise \end{cases}$$

where $\alpha_t = \log\frac{1}{\beta_t}$

Figure 3: The algorithmic steps of AdaBoost.

Here, $h_i, i = 1, \cdots, t$ corresponds to the weak classifiers. Each weak classifier is formed based on a feature described in Section 2.1. Using a threshold value between the feature value of face data and that of non-face data, we can obtain a simple weak classifier with error rate less than 50%. The final strong classifier is a linear combination of these weak classifiers. Furthermore, based on computational learning theory, the final strong classifier will have a monotonously non-increasing error rate for an increasing number of weak classifiers [9]. Knowing that the error rate is lower bounded, the overall error rate will converge to zero as long as we provide enough features with classification rate more than 50%.

2.3 Achieve the real-time detector

Viola and Jones also proposed an algorithm for constructing a cascade of classifiers that decreases the computation time dramatically during face detection (testing phase). The cascade of classifier algorithms decomposes a single *AdaBoost* procedure into several stages, and each stage executes its own AdaBoost procedure. In each stage, a window sub-detector will eliminate roughly 50% non-face images and preserve most of face images (e.g. 99.9%). Therefore, if there are 32 stages, then the final false positive would be only $0.5^{32} = 2.328 \times 10^{-10}$, and the detection rate would be $0.999^{32} = 96.85\%$. It is important to mention that the product of detection rates will decrease exponentially, and one should maintain a high detection rate at each stage. The complete algorithm can be found in [7].

By introducing a cascade of classifiers, most regions will be discarded in the early steps of face detection since usually large portions of images are typically without faces. Early steps need just small number of weak classifiers to achieve desired prediction rate because of the nature of the AdaBoost algorithm. Therefore, most non-face region can be discarded by computing relatively small number of features.

2.4 Extend to color and hyperspectral images

The feature detector discussed in Sections 2.1 to 2.3 is limited to processing monochromatic images. In order to detect faces in color or hyperspectral images, we pre-process images by *principal component analysis* (PCA) and perform face detection on the most informative principal component (the first eigenvector). We choose PCA because face features detected by the proposed feature detector rely on spatial variations in a small neighborhood, and the first principal component provides the majority of spatial variations. The face detection algorithm described before can be applied without any modifications to the first principal component of the PCA transformed image.

2.5 Experiments

Due to the cost of acquiring a large number of face images, we gathered 3,000 face images and 5,000 non-face images as training data. Examples of these images are shown in Figure 4. Most of the face data were obtained from AT&T face database **[13]** and Sinica face database **[14]**, in order to include face images from several races. The non-face images are randomly collected from the Internet, in order to discriminate all kinds of background in face images. Each training image is of size 24 by 24 pixels with only one band (grayscale), and eight bits per pixel intensity (unsigned byte value).

The training process is very time consuming and needs a lot of computation power. This is due to the computational complexity of learning from all features enumerated exhaustively for all training images. On the other side, the detection process can be done very efficiently, since it only relies on small number of features.

In order to detect faces of varying size in images, most face detection systems usually scan every level of a multi-resolution image pyramid. Instead of resizing the size of an analyzed image, we resize the sub-windows for face feature detection, and adjusted parameters of a classifier based on pre-computed look-up table values. For example, if we choose a scale factor s then the size of first detecting sub-window is 24 by 24, the second one is $24 \cdot s$ by $24 \cdot s$, and the size of nth sub-window is $24 \cdot s^{n-1}$ by $24 \cdot s^{n-1}$. The threshold for the strong classifier is just multiplied by a factor of s^2.

Another advantage of using a single image instead of a multi-resolution image pyramid is the fact that the integral image tables for large testing images can be computed in advance.

(a) (b)

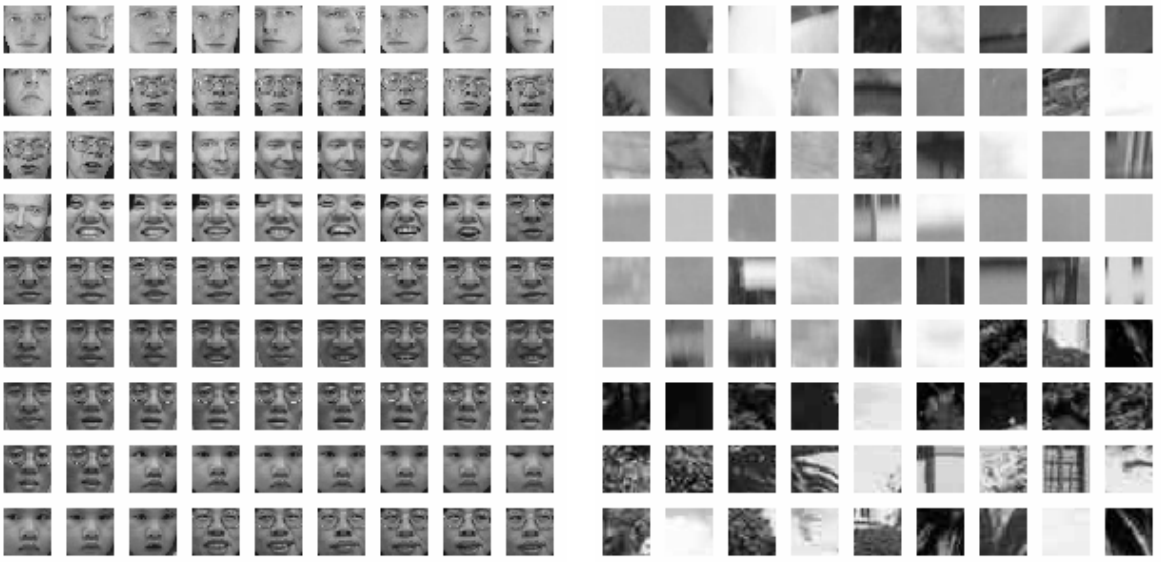
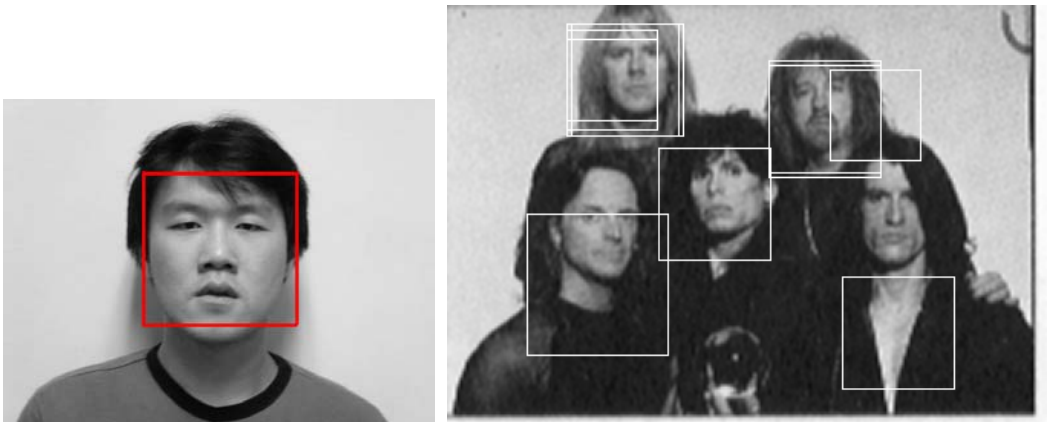


Figure 4: Examples of face images that vary in human race, gender and facial expression (a). Examples of non-face images collected randomly from the Internet (b).

A few experimental results of detected faces are in Figure 5. It is apparent from these examples that detecting faces in pictures with complex background is more error prone than in images with simple background. One way to improve the results is to collect false detections, and use them to train a new classifier with improved performance. Another way is to pre-process both training and testing images with histogram equalization to remove noise from light variation. We have also noticed that some detected face sub-windows mutually overlap. This type of false detection can be removed by analyzing overlapping face sub-windows.



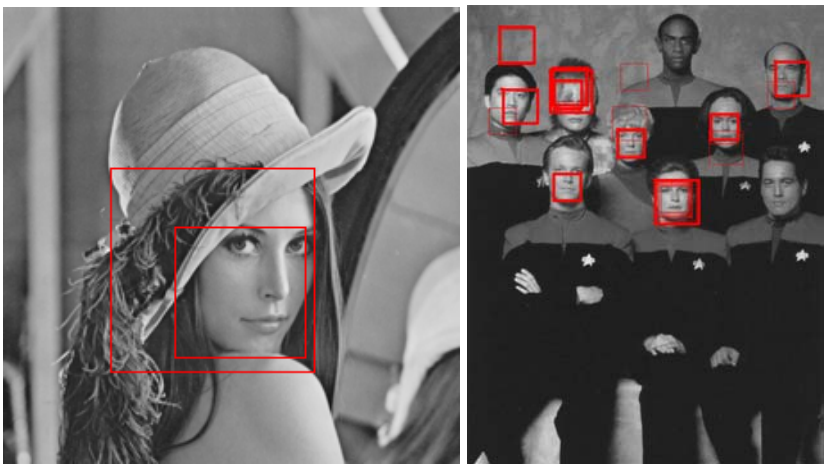
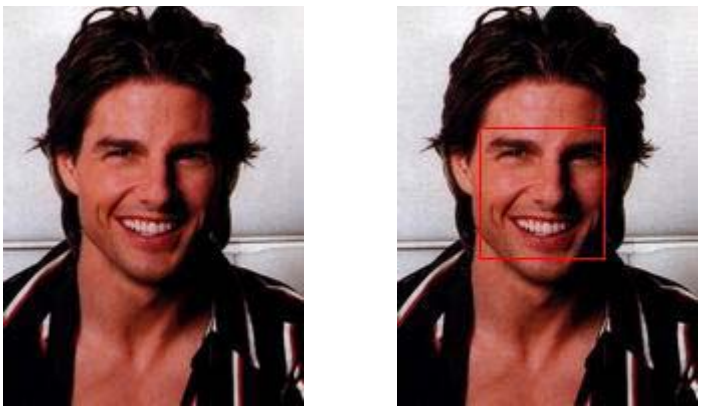
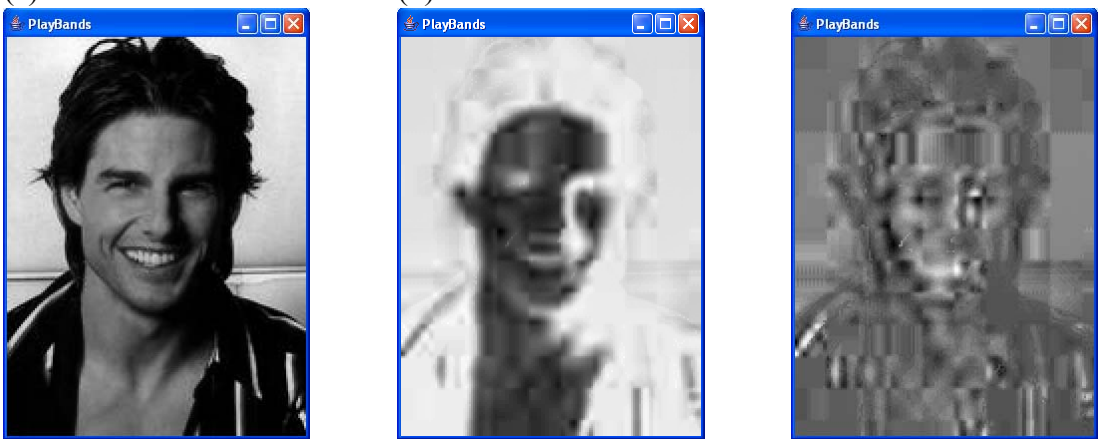


Figure 5: A few examples of face detections. The left-top image contains a sample white background and only one person in the scene. The right-top image has simple background and multiple persons, and the image quality is not very good. The left-down image contains a single person with the head orientation slightly away from the camera and with complex background. The right-down image contains multiple persons and complex cluttered background.

For color and hyperspectral images, we use the first principal component of PCA for face detection. Figure 7 shows an example of face detection from a color image. It shows the original image, its principal components, and the result of face detection. We can notice that most of the face spatial details are preserved in the first principal component.



(a) (b)



(c) (d) (e)

Figure 6: Original color image (a); detected face region (b); the first principal component (c); the second principal component (d); the third principal component (e).

3 Hand Pose Estimation

Given an image bounding box that encloses a human face, we would like to further estimate the 3D head pose of the detected face. Pose estimation algorithms developed previously relied usually on spatial information or on motion cues **[15]**. In this section, we will investigate the use of spectral information for pose estimation. Due to the high complexity of human face appearance, we first reduced the head pose estimation problem to a human hand pose estimation problem.

First, the problem of human hand pose estimation is approached by exploring the discrimination power of spectral information for human surfaces that are significantly different, for example, fingernails, skin on back of a hand and palm skin. Next, we explore the possibility of pose estimation using the directional change of spectral signatures per each human surface type.

3.1 Hyperspectral Imaging

Hyperspectral images provide ample spectral information to identify and distinguish spectrally unique materials. The word "hyper" refers to the large number of measured wavelength bands. In general, light interacts with an object and we are measuring the reflected radiation (electro-magnetic waves) at multiple wavelengths with a hyperspectral camera. A spectral wavelength of each wave is directly related to the energy level of photons as shown in Figure 7 [16]. Humans can visually perceive spectrum with wavelengths between $300nm$ and $700nm$.

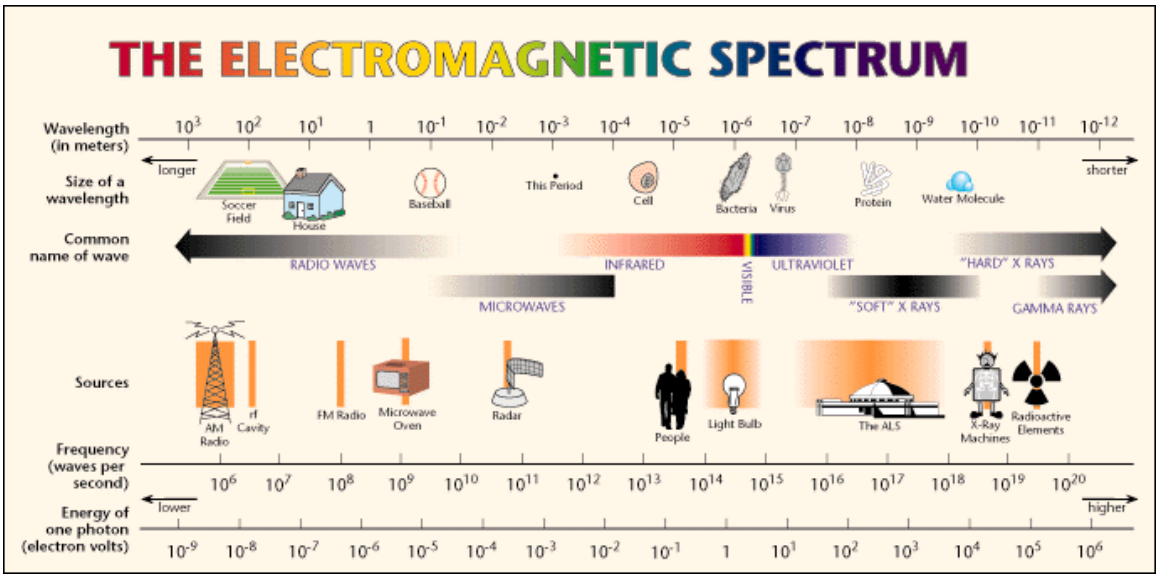


Figure 7: The electromagnetic spectrum and its energy.

The hyperspectral camera used in our experiments operates in a spectral wavelength range [400, 1100] nanometers with two distinct spectral regimes, such as, visible [400nm, 720nm] and near infrared [650nm, 1100nm]. Figure 8 shows an example of a hyperspectral image with wavelengths from the visible wavelength range.

We considered near infrared (NIR) hyperspectral images for hand pose estimation since waves at this range have larger depth penetration than the waves from the visible wavelength range. We are looking for unique features that could be correlated with 3D pose and would be hard to conceal or modify. Thus, near infrared information could provide some subsurface skin characteristics that would be hard to conceal and modify by humans [12]. Figure 9 shows an example of a hyperspectral (HS) image with 46 spectral bands from the infrared wavelength range.

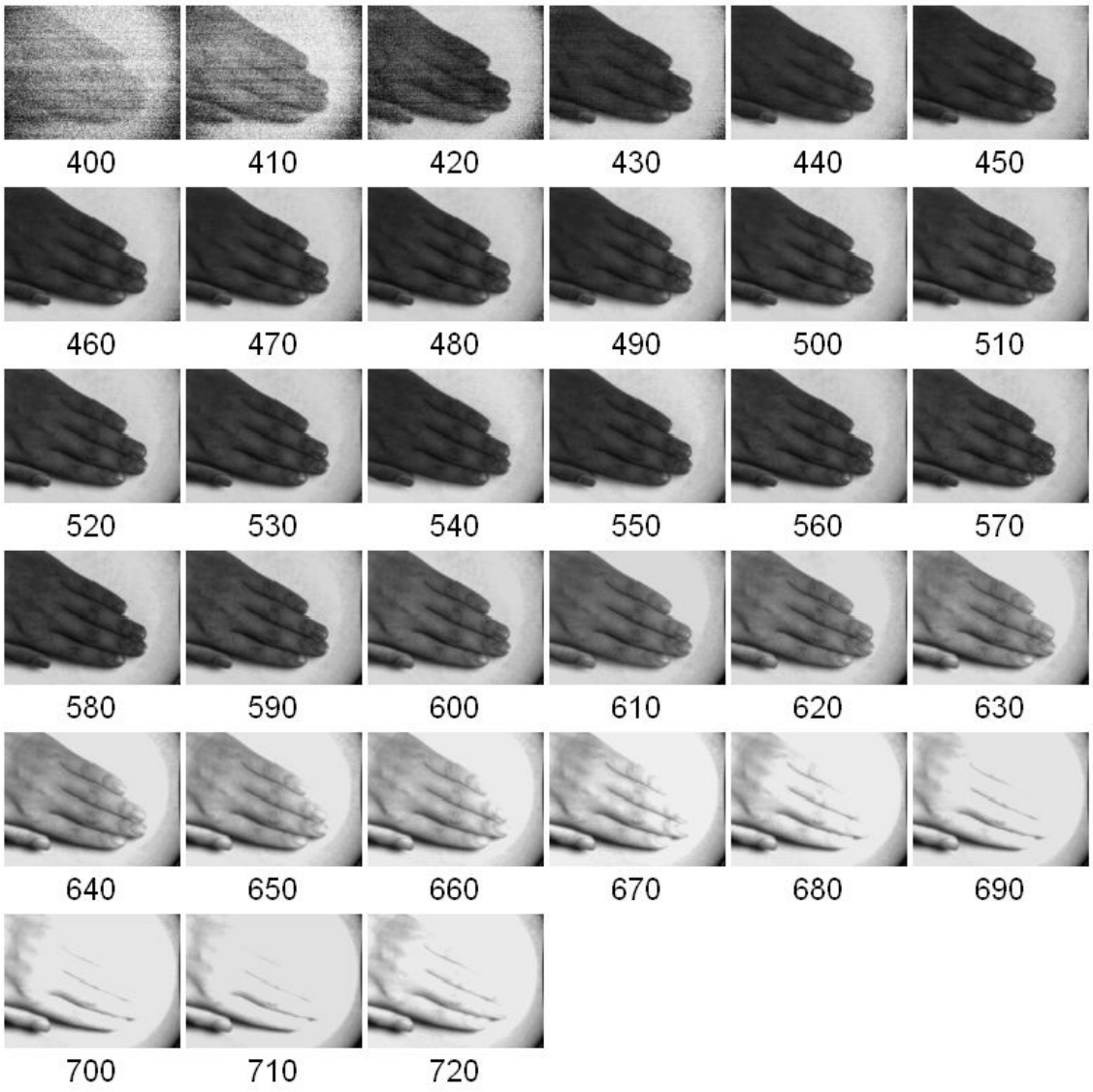


Figure 8: A series of visible spectrum wavelength images from a hyperspectral image (or an image cube) after spectral calibration. The bands are displayed from the smallest to the largest wavelengths (shown under each image in [*nm*]).

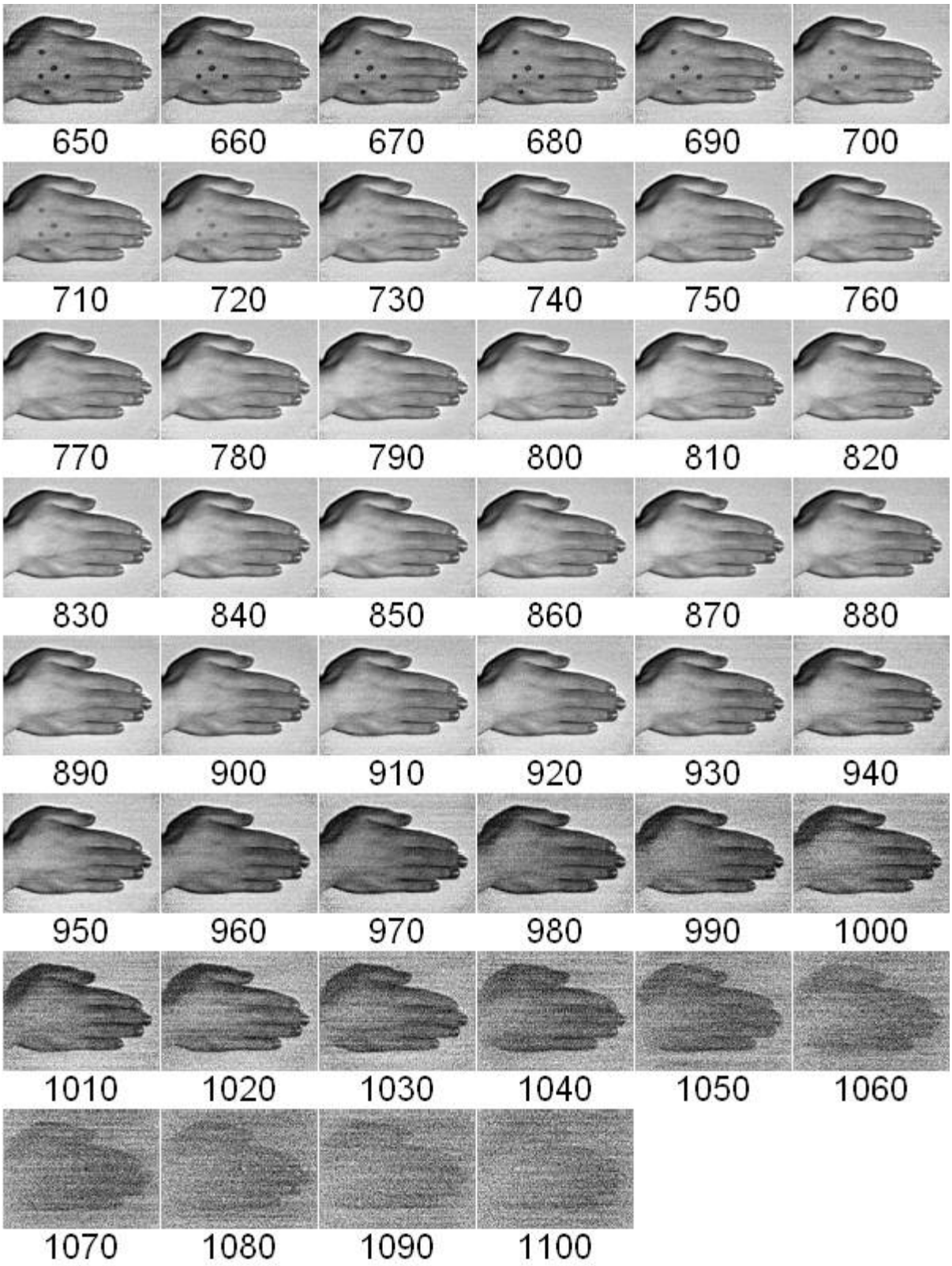


Figure 9: A series of near infrared spectrum wavelength images from a hyperspectral image (or an image cube) after spectral calibration. The bands are displayed from the smallest to the largest wavelengths (shown under each image in [*nm*]).

In our experiments, we acquired hyperspectral images of the size 676 columns by 516 rows. We analyzed only the spectral range between $700nm$ and $1{,}000nm$ due to the presence of a large amount of camera noise. In order to reduce the camera noise, we further down-sample the near infrared images by two.

3.2 Hyperspectral Camera Calibration

Raw hyperspectral images are calibrated first to obtain reflectance values. For calibration purposes, we acquired two additional images, such as, black and white images. The black image refers to near zero-reflectance image and is acquired with a lens cap on. The white image refers to near 100%-reflectance image and is acquired with a white calibration reflectance board in front of the camera at the distance of an imaged hand.

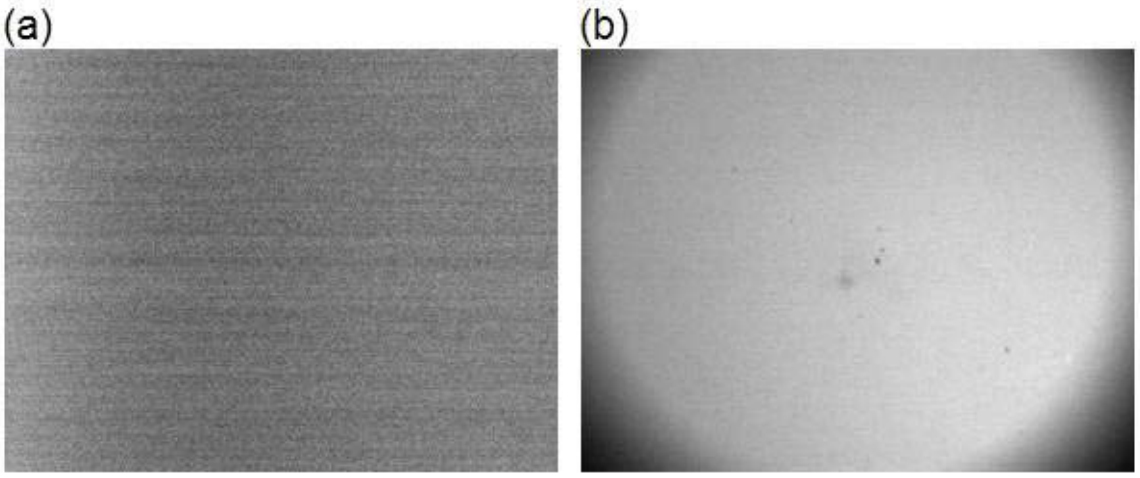


Figure 10: Black (a) and white (b) calibration images acquired at the wavelength $740nm$.

According to [17], [18], image intensity at spatial coordinate (x, y) and wavelength λ_i can be modeled as

$$I(x, y, \lambda_i) = L(x, y, \lambda_i)S(x, y, \lambda_i)R(x, y, \lambda_i) + O(x, y, \lambda_i),$$

where $L(x, y, \lambda_i)$ refers to the illumination, $S(x, y, \lambda_i)$ refers to the system spectral response, $R(x, y, \lambda_i)$ refers to the reflectance of the viewed surface, and $O(x, y, \lambda_i)$ refers to the offset which includes dark current and stray light. For a white background image, the intensity of coordinate (x, y) for wavelength λ_i can be modeled as

$$I_W(x, y, \lambda_i) = L(x, y, \lambda_i)S(x, y, \lambda_i)R_W(\lambda_i) + O(x, y, \lambda_i),$$

and for black background image, the intensity of coordinate (x, y) for wavelength λ_i can be modeled as

$$I_B(x, y, \lambda_i) = L(x, y, \lambda_i)S(x, y, \lambda_i)R_B(\lambda_i) + O(x, y, \lambda_i),$$

where $R_W(\lambda_k)$ and $R_B(\lambda_B)$ are reflectance functions for these two images. $R_W(\lambda_k)$ and $R_B(\lambda_B)$ are theoretically independent of (x, y), because the viewed surface has the same reflectance property for all image pixels. Raw black and white images are shown in Figure 10. We can notice that the intensity of white background image is not uniformly distributed within the image plane, and our goal is to compensate this non-uniformity.

By using the equations for $I_W(x, y, \lambda_i)$ and $I_B(x, y, \lambda_i)$, we can derive

$$L(x, y, \lambda_i)S(x, y, \lambda_i) = \frac{I_W(x, y, \lambda_i) - I_B(x, y, \lambda_i)}{R_W(\lambda_i) - R_B(\lambda_i)},$$

and plug it into the equation for $I(x, y, \lambda_i)$. We then obtain a calibration equation provide below.

$$R(x, y, \lambda_k) = \frac{(I(x, y, \lambda_i) - I_B(x, y, \lambda_i))R_W(\lambda_i)}{I_W(x, y, \lambda_i) - I_B(x, y, \lambda_i)} + \frac{(I_W(x, y, \lambda_i) - I(x, y, \lambda_i))R_B(\lambda_i)}{I_W(x, y, \lambda_i) - I_B(x, y, \lambda_i)}.$$

Figure 11 compares images before and after calibration. We can observe that the raw images have low intensity values near the image border and have high intensity values at the image center, just like the white image in Figure 10.

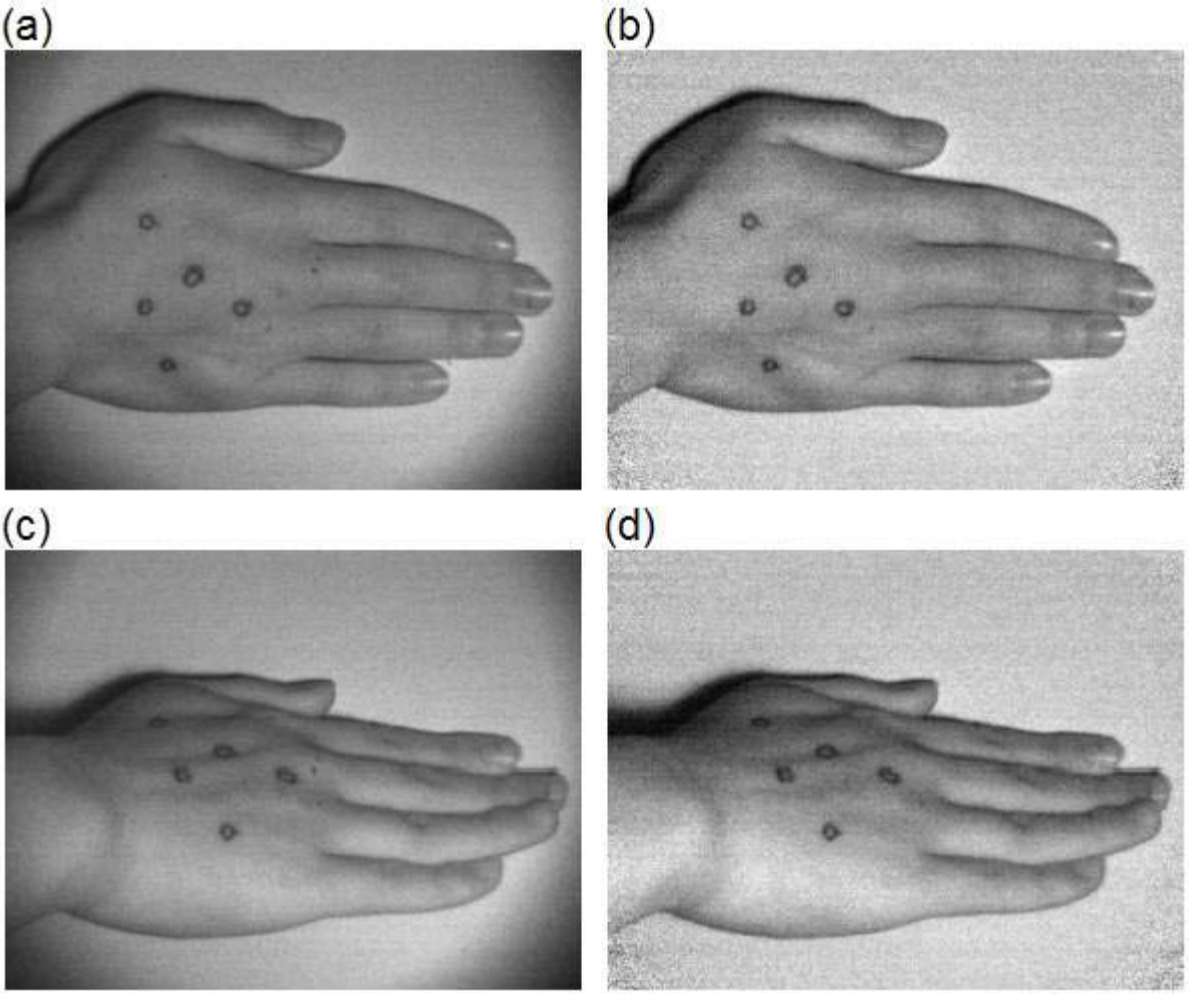


Figure 11: Illustration of hyperspectral image calibration. Two images before calibration (left column or (a) and (c)) and after calibration (right column or (b) and (d)) that were acquired at the wavelength $700nm$ **.**

3.3 Hand Detection

In order to perform 3D hand pose estimation, we need to detect hand regions. In our simple experimental setup, this task was equivalent to identifying the background and foreground regions. To accomplish this task, we used a supervised classification technique known as the k-nearest neighborhood method. The need for a non-linear supervised classification method came from the fact that the hand and background spectral values were not linearly separable. However, this step could be simplified by the use of a simple thresholding method if one would use a highly absorbent background material.

Given the high dimensionality of hyperspectral images, we investigated several metrics for comparing pixel similarity that are part of any classification algorithm. In our experimental study, we tried *Euclidean Distance* (*ED*) and *Spectral Angle Mapper* (*SAM*)

similarity metrics. The metric *ED* is also known as l_2 distance. Suppose there are two spectra $S_1 = [a_1, a_2, \cdots, a_n]$ and $S_2 = [b_1, b_2, \cdots, b_n]$, then their *ED* is defined as

$$ED(S_1, S_2) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2} .$$

The metric *SAM* is computed as a spectral angle between two spectra (or vectors) according to the equation below.

$$SAM(S_1, S_2) = \cos^{-1}\left(\frac{S_1 \cdot S_2}{\|S_1\| \cdot \|S_2\|}\right),$$

The smaller the spectral angle, the more similar the two vectors are. Spectral angles will be relatively insensitive to illumination changes. This property is due to the fact that increasing or decreasing pixel values in all bands (e.g., multiplying all coordinates of vectors) would change only vector magnitudes but would not change the directions of the compared vectors.

We detect hand pixels and background pixels by using the k-nearest neighbor algorithm in two stages. In the first stage, we manually select a few pixels from a hand image region. Given a metric, image pixels are then labeled as follows. If a metric computed from a pair of pixels (one selected pixel and one pixel already labeled as hand) is less than a threshold then the pixel is marked as hand, otherwise the pixel is marked as background. In the second stage, manually select a few pixels that belong to the background or shadow regions of the hand object. For all pixels labeled as hand, modify their label to the background label if the metric evaluation computed from a hand labeled pixel and one of the manually selected pixels is less than a threshold. Finally, we blur the label image to obtain spatially contiguous foreground (hand) region.

We compared ED and SAM similarity metrics visually. The results based on the *ED* metric appeared better than the results based on the *SAM* metric. This observation could be explained by the insensitivity of the SAM metric to illumination changes. Figure 12 shows several masks computed by *ED*, where the white part represents hand areas, and the black areas correspond to background.

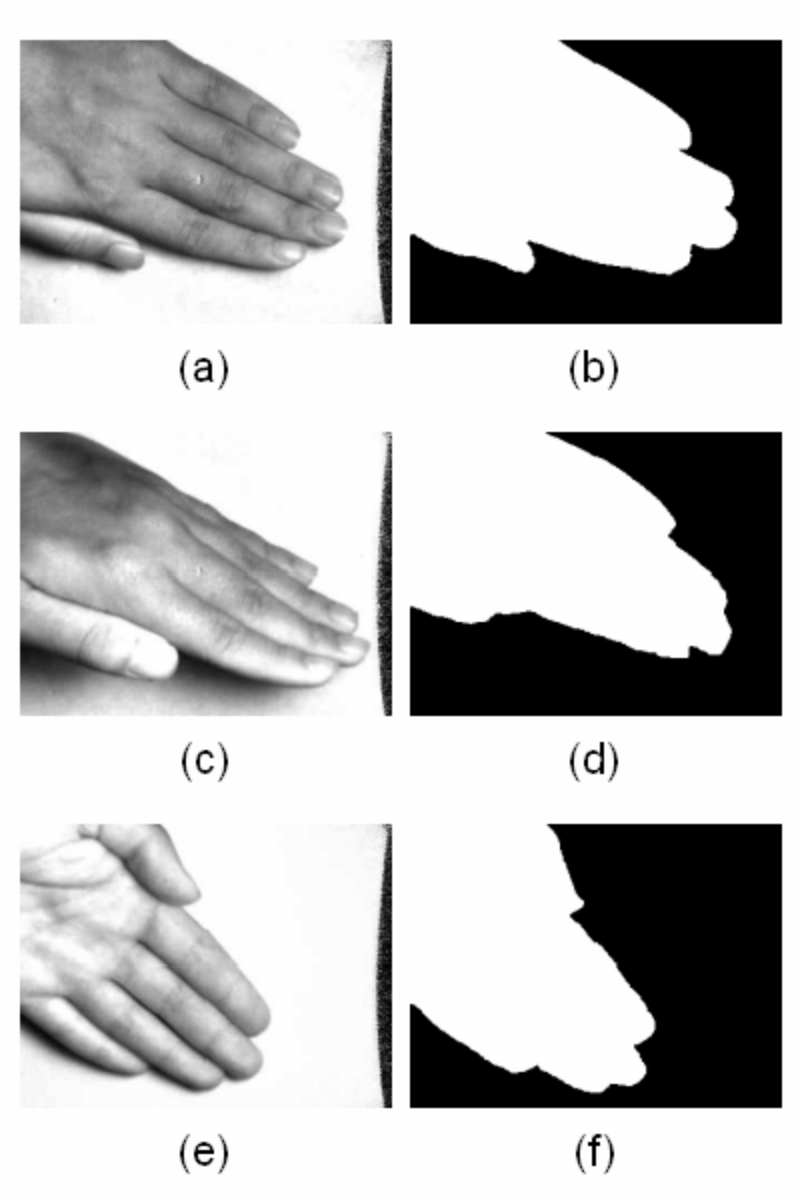


Figure 12: Original hand images (left column) and their label images (right column) identifying background (black) and foreground/hand (white).

3.4 Spectral Analysis

In this section, we attempt to separate hand skin and hand nail regions based on their hyperspectral information. First, we focused on choosing a metric that discriminates hand classes (nail and skin). Second, we investigated the use of unsupervised and supervised methods for nail and skin classification.

3.4.1 Metric Selection

In order to classify pixels into the skin and nail classes, we must define an appropriate distance metric as a measure of feature similarity. We investigated two such distance metrics, and compared their discriminating power for nail and skin spectral features.

We compared the discrimination power of *Euclidean Distance* and *Spectral Angle Mapper* as defined in Section 3.3 using a statistical hypothesis testing approach and four representative (randomly selected) patches from each class (nail and skin). We computed similarity measures according to these two metrics and reported the results in Table 2 (a and b parts).

The statistical hypothesis testing follows the description in [19] and can be outlined as follows. Let us suppose that the similarity measure (or the distance) between two nail patches is a random variable following the distribution A; the distance between two skin patches is a random variable following the distribution B; and the distance between a nail patch and a skin patch is a random variable following the distribution C. For the Euclidean Distance metric ED shown in Table 2 (a), we obtained samples $x_1, x_2, \cdots, x_6$ from the distribution A, $y_1, y_2, \cdots, y_6$ from the distribution B, and $z_1, z_2, \cdots, z_{16}$ from the distribution C. Let us suppose that the distributions A, B, and C are all normal distributions, and share a common unknown variance σ^2. In order to test whether one nail region is statistically different from another nail region, we examine the following two hypotheses:

$$H_0 : A = C \text{ and } H_A : A \neq C$$

Since we do not know the real variance σ^2 of A and C, we rely on the pooled sample variance s_p:

$$s_p = \frac{(n-1)\cdot {s_A}^2 + (m-1)\cdot {s_C}^2}{m+n-2} = 80{,}208{,}818$$

where $n = 6$ and $m = 16$ are the number of samples from the A and C distributions. Next, we used the t-statistic value with $(m+n-2) = 20$ degrees of freedom to test if two distributions are identical. The t-statistic value is 4.460246. The p-value is 99.976%. Based on these two values, we can conclude with a high confidence that the distribution A and C are different, and the hypothesis H_A is very likely to be true. Similarly, we computed the t-statistic value for the B and C distributions that is equal to 861.6607, and the p-value is almost equal to 1. We concluded that the distributions B and C are different. The main conclusion of this hypothesis testing is that the separation of nail and skin classes should be possible with a high confidence by using the ED metric.

Table 2: Randomly picked representative regions of nail and skin classes. The values in the table are the similarity measures (distances) between sample means of a pair of regions. The table (a) contains values computed with the *Euclidean Distance (ED)*, and the table (b) contains the *Spectral Angle Mapper (SAM)* values. The intra-class variance is greater then the inter-class variance for both *ED* and *SAM*. Nail 1 and Nail 2 sample regions contain more "shining" (highly reflective) parts of nail

than the Nail 3 and Nail 4 sample regions. Skin 2 and Skin 3 sample regions contain more "shadowed" (less reflective) parts of skin than the Skin 1 and Skin 4 sample regions.

(a)

ED	nail 1	nail 2	nail 3	nail 4	skin 1	skin 2	skin 3	skin 4
nail 1	0	25,779	25,663	31,012	27,808	37,702	33,990	21,553
nail 2		0	21,200	20,700	45,012	56,297	50,675	38,878
nail 3			0	19,160	41,703	50,785	43,485	38,030
nail 4				0	48,166	57,664	51,009	42,961
skin 1					0	20,802	18,602	17,937
skin 2						0	15,930	25,821
skin 3							0	23,101
skin 4								0

(b)

SAM	nail 1	nail 2	nail 3	nail 4	skin 1	skin 2	skin 3	skin 4
nail 1	0.0000	0.2633	0.2591	0.3083	0.2103	0.2351	0.2645	0.1763
nail 2		0.0000	0.3164	0.3109	0.3196	0.3645	0.3746	0.2891
nail 3			0.0000	0.2875	0.2415	0.2283	0.2063	0.2678
nail 4				0.0000	0.3078	0.3077	0.3027	0.2990
skin 1					0.0000	0.1671	0.1740	0.1698
skin 2						0.0000	0.1302	0.1955
skin 3							0.0000	0.2094
skin 4								0.0000

By using the same statistical hypothesis testing approach, we computed the t-statistic and p-values for the Spectral Angle Distance metric. The p-value for the B and C distributions is 99.9506%. However, the p-value for the A and C distributions is 49.2445%, and hence H_0 : A=C is more likely to be true, and the two distributions appear to be identical. We concluded that Spectral Angle Distance is not appropriate for separating nail from skin based on our statistical hypothesis testing.

Overall, Euclidean Distance should provide better separation of skin and nail classes than *Spectral Angle Mapper*. Therefore, we use the ED metric with supervised and unsupervised classification algorithms.

3.4.2 Unsupervised Classification

First, we use the *Isodata Clustering* to separate nail and skin materials. Isodata clustering is an advanced K-means unsupervised clustering algorithm [21]. Figure 13 shows the result of Isodata Clustering. Due to the fact that the intra-class noise is much greater than the inter-class distance, it is hard to classify skin and nail pixels using an unsupervised algorithm. We attempted to perform classification with a supervised algorithm.

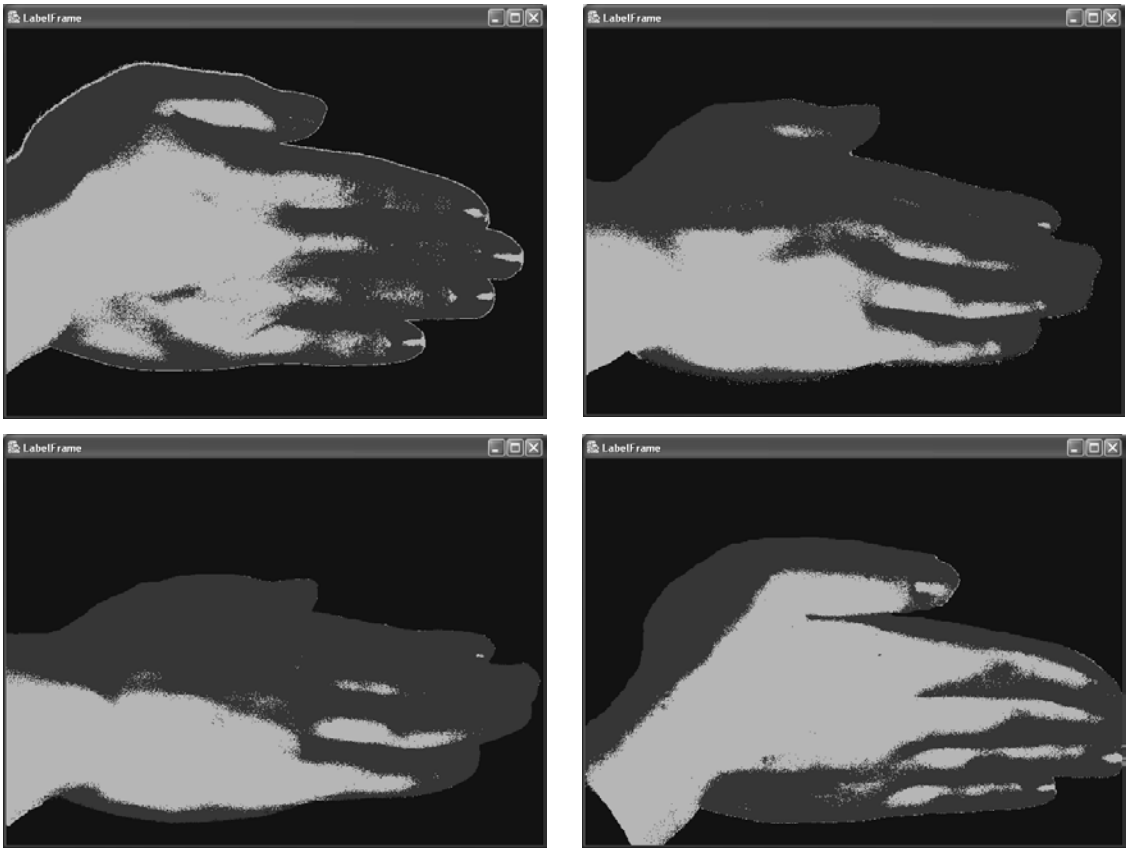


Figure 13: Isodata clustering cannot separate the nail from the skin. Instead, the two clusters are decided by the orientation of the hand to the light.

3.4.3 Supervised Classification

We choose the k-Nearest Neighbor Algorithm to classify nail and skin pixels. However, we obtained similar results as with the unsupervised Isodata clustering method. The results of supervised classification with the k-Nearest Neighbor Algorithm are shown in Figure 14.

Figure 14: The results of supervised classification with the k-Nearest Neighbor Algorithm. The black region denotes skin and the white region is nail.

The reason for failing to classify nail and skin with the supervised method can be attributed to the fact that all data in Table 2 are averaged (intra-class variance is reduced) and the supervised algorithm is classifying pixels with large intra-class variance. Without eliminating the inner-class variance, satisfactory classification is still not feasible. This leads to considering spatial information as it will be discussed in section 3.5.

3.5 Spatial Analysis

According to the Phong Illumination Model [20], the intensity of each wavelength is formed by three sources: (1) ambient light, (2) diffuse light, and (3) specular light, and can be modeled by the equation:

$$I_\lambda = I_{a\lambda} \cdot k_a \cdot O_{d\lambda} + f_{att} \cdot I_{p\lambda} \cdot [k_d \cdot O_{d\lambda} \cdot (\vec{N} \bullet \vec{L}) + k_s \cdot O_{s\lambda} \cdot (\vec{R} \bullet \vec{V})^n]$$

$\vec{N}$ is the unit normal vector; $\vec{L}$ is the unit light source vector; $\vec{R}$ is the unit light reflection vector; and $\vec{V}$ is the unit-viewing vector. I_a and I_p are the intensities of ambient light and point light sources respectively. k_a , k_d and k_s are the coefficient of ambient, diffuse and specular reflection. f_{att} is the light-source attenuation factor. O_d and O_s are the diffuse color and spectral color of the object.

We can notice that specular light is powered by a factor *n*, and hence it demonstrates larger intensity variations than diffuse light. It is known that nail material reflects more light that than skin material for most visible wavelengths. This motivates our effort to separate nail image regions from skin regions by compute a sample variance over a small spatial neighborhood of pixels.

Spatial variances over a small set of image pixels are calculated by introducing a round filter. The *round filter* is a filter that computes a sample spatial variance for every wavelength within a circular area. We obtain the output image, so called *variance map*, by convolving an original image with the *round filter* of a fixed radius. We also use the principal component analysis (PCA) technique described in Section 2.4 for finding regions that correspond to the overall hyperspectral image variance.

The results after filtering seem promising in terms of nail versus skin separation for small rotational angles. We conducted experiments with both visible and near infrared images, and visible images lead to more robust nail and skin separation for varying angular hand poses than near infrared images.

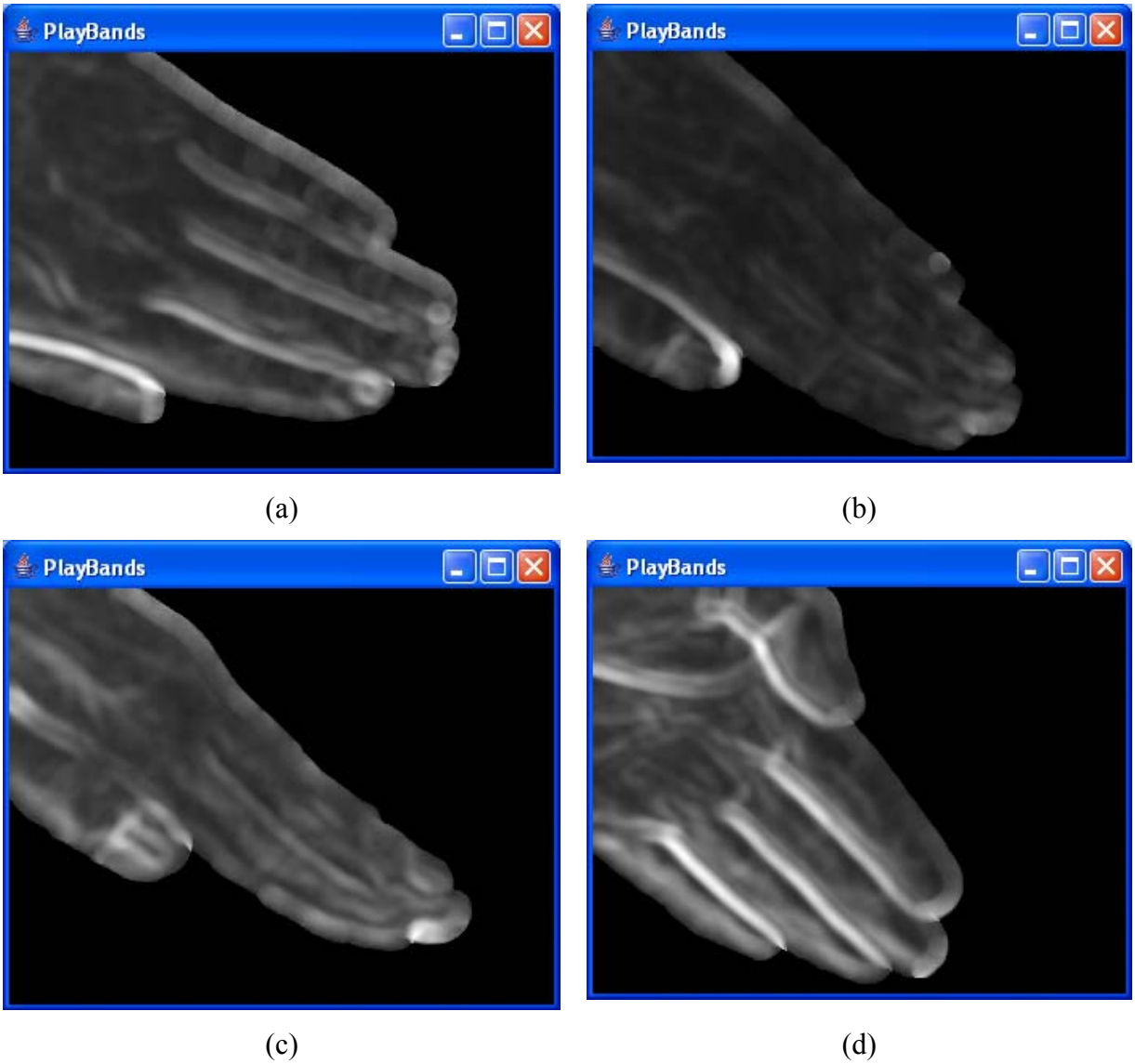


Figure 15 : The first principal component of variance maps for four different hand pose angles. The images (a), (b), and (c) are views of the backside of a human hand and include nail regions. The image (d) is the view of the palm of a human hand and it does not include nail regions. The four hand pose angles correspond to 0 degree (a), 60 degrees (b), 85 degrees (c), and 180 degrees (d). We can notice that nail region can be distinguished easier at smaller angles.

Figure 15 shows the first principal component of variance maps for four different hand pose angles. Images acquired at hand pose angles less than 30 degree usually have circular regions of high variance, for instance, nail regions in Figure 15 (a). These results appeared to be promising assuming that we would collect more hyperspectral hand pose images, and train an efficient classifier for separating nail and skin regions from the variance maps.

3.6 Discussion

Based on our preliminary experiments of hand pose estimation approaches, we can conclude the following. *The Euclidean Distance* similarity metric is more appropriate for classification than the *Spectral Angle Mapper* similarity metric. Pixel-based spectral classification with unsupervised or supervised methods is not reliable for separating nail and skin classes because of large intra-class variations. Finally, nail and skin separation seems promising from variance maps that are based on a spatial filtering approach. By collecting more hyperspectral images of human hands, we might be able to compute their variance maps and train an efficient classifier for finding nail positions. Given the presence of nail regions and their relative 3D position with respect to a human hand (and 2D projected shape computed by background segmentation introduced in Section 3.3), we may further estimate hand pose angles by exploiting the 3D spatial relationships among hand, fingers, and nails in the future.

4 Summary

We implemented a fast face detection system that can process multi-spectral images. We researched several approaches to hand pose estimation based on spectral and spatial analyses. We detected hand (foreground) in a background first, and then attempted to separate nail and skin regions using unsupervised and supervised approaches, as well as, spatial filtering. The preliminary outcomes of multiple approaches were summarized in Section 3.6.

There are three primary future directions. First, we could improve the skin/nail classification by acquiring more hyperspectral images of human hands. We would then compute variance maps from these images and use their first principal component of PCA analysis to train a classifier for nail/skin separation. Second, we would model the 3D spatial relationships among hand shapes, fingers, and nail positions. We would use these relationships as inputs into our model for estimating 3D poses of human hands. Third, we would extend the hand pose estimation to human face pose estimation. A human face is much more complex than a human hand, since we would need to separate several materials including skin, lips, eyes, and hair.

5 References

[1] Ming-Hsuan Yang, David Kriegman, and Narendra Ahuja, "Detecting Faces in Images: A Survey", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no.1, 2002.

[2] Erik Hjelmas and Boon Kee Low, "Face Detection: A Survey". Computer Vision and Image Understanding, vol. 83, 2001.

[3] Michael Chan and Dimitri Metaxas, "Physics-Based Object Pose and Shape Estimation from Multiple Views", Image Processing and Pattern Recognition, 1994.

[4] Qian Chen, Haiyuan Wu, Tadayosi Shioyama, and Tetsuo Shimada, "3D Head Pose Estimation using Color Information", IEEE International Conference on Multimedia Computing and Systems, 1999.

[5] Stan Z. Li, XianHuan Peng, XinWen Hou, HongJiang Zhang, and QianSheng Cheng, "Multi-View Face Pose Estimation Based on Supervised Learning". Int. Conf. on Automatic Face and Gesture Recognition, 2002.

[6] Volker Kruger, Sven Bruns, and Gerald Sommer, "Efficient Head Pose Estimation with Gabor Wavelet Networks". Proc. British Machine Vision Conference, 2000.

[7] Paul Viola and Michael Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features", Computer Vision and Pattern Recognition, 2001.

[8] Paul Viola and Michael Jones, “Fast and robust classification using asymmetric AdaBoost and a detector cascade.” Neural Information Processing Systems, 2001.

[9] Yoav Freund and Robert E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting.” Computational Leaning Theory: Eurocolt, 1995.

[10] Robert E. Schapire, and Yoav Freund, “Improving Boosting Algorithms Using Confidence-rated Predictions,” 1999.

[11] Richard Duda, Peter Hart, and David Stork, “Pattern classification.” Wiley-Interscience, 2000.

[12] M. Hiraoka, M. Firbank, M. Essenpreis, M. Cope, S. Arridge, P. van der Zee, and D. Delpy, “A Monte Carlo Investigation of Optical Pathlength in Inhomogeneous Tissue and Its Application to Near-Infrared Spectroscopy,” Physics in Medicine and Biology, vol. 38, pp. 1859-1876, 1993.

[13] AT&T (Olivettti) Database, http://www.uk.research.att.com/

[14] Sinica Face Database, http://smart.iis.sinaca.edu.tw/html/face.html

[15] K Seo, I. Cohen, S. You, and U. Neumann, “Face pose estimation system by combining hybrid ICA-SVM learning and re-registration,” Asian Conference on Computer Vision, 2004

[16] Electromagnetic spectrum description, URL: http://www.lbl.gov/MicroWorlds/ALSTool/EMSpec/EMSpec2.html

[17] Zhihong Pan, Glenn E. Healey, Manish Prasad, and Bruce J Tromberg, "Face Recognition in Hyperspectral Images," IEEE Transactions on Pattern Analysis and Macjone Intelligence, Vol. 25, No. 12, December 2003.

[18] Louis J. Denes, Peter Metes, and Yanxi Liu, "Hyperspectral Face Database," CMU-RI-TR-02-25, Oct 2002.

[19] John Rics, “Mathematical Statistics and Data Analysis”, Duxbury Press, 1994.

[20] Edward Angel, “Interactive Computer Graphics: A Top-Down Approach using OpenGL.” Addison-Wesley, 2002.

[21]J.T. Tou and R.C. Gonzales. “Pattern Recognition Principles.” Addison-Wesley Publishing Company, 1974, (pp. 97-1050)