FINDING BLOOD IN CAPSULE ENDOSCOPY VIDEO

by

JAY FREDERICK COX

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT ARLINGTON

August 2005

ACKNOWLEDGEMENTS

There are many more people and friends I could thank here for helping me through these past three years, however, if I really gave due diligence to all who helped, this document wouldn't past the final mechanical check, and then I wouldn't graduate, and then I'd become a crazy lunatic with a finger up his nose in a yoga pose chanting "I'm a fireman; where's my hose?".

All silliness aside, much thanks goes out to my parents, without whom I would doubt this document would ever be created. I must thank Julie Evans, as before she helped with this research I had absolutely know concept of where to go with it. Mahmudul "Tazim" Khan did a good deal of work to our Capsule Endoscopy project, and I was glad to have such a friendly soul as him to work with. Sae Hwang, JeongKyu Lee and Emre Celebi proved to be good lab mates. The random conversations I had with Mr. Lee and Emre proved most rewarding. I also must thank Dr. Oh for having the patience to deal with me!

July 20, 2005

ABSTRACT

FINDING BLOOD IN CAPSULE ENDOSCOPY

VIDEO

Publication No. _____

Jay Cox, M.S.

The University of Texas at Arlington, 2005

Supervising Professor: JungHwan Oh

Capsule Endoscopy (CE) is a new procedure where endoscopists can visualize and discriminate anomalies of the human gastrointestinal track by allowing the patient to swallow a camera pill. The digital video obtained from the pill is used to safely pinpoint the location of these abnormalities in areas previously unavailable for view. One of the main functions of CE is determining the location of bleeding. Given Imaging, currently the only maker of a FDA approved CE device, has a proprietary and unspecified blood test, called Suspected Blood Index, which can be used to automatically detect where blood occurs in the video. Its precision and recall are fair, but can be improved upon. In this paper we discuss the journey taken in the attempt to

attempt to create a better blood detector. The final algorithm seems to deliver good results with precision and recall above 92% and 85% respectively.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

CHAPTER 1

INTRODUCTION

Capsule Endoscopy (CE) is a relatively new technology (FDA approved 2002) allowing doctors to visualize what previously could not be visualized through traditional endoscopy procedures. Previous endoscopy and colonoscopy procedures could be used to visualize up to the stomach or most of the large intestine, but there existed no method to view most of the small intestine without surgery. With the miniaturization of wireless and camera technologies came the ability to view the entire gestational track with little effort required by either the practitioner or the patient. Currently the only commercial maker of such an FDA approved capsule endoscopy device is Given Imagining (http://www.givenimaging.com). Their camera pill (26mm long by 11mm wide) is swallowed, transmitting two images a second to an array of sensors worn by the patient. During an approximately 8-hour course, the images are recorded to a worn device and then later downloaded to a computer where the doctor can see what might be troubling the patient. The pills are inexpensive to make and disposable. Furthermore, because the pill is moved only by normal processes and not forced anywhere, it is generally considered to be a safer procedure then regular endoscopy [12].

To diagnose, an endoscopist is required to stare at the video produced from the sequence of images created by the capsule endoscopy device. General

recommendations are to set the frame rate at 8 or so frames per second, get a good cup of coffee, some relaxing music, and enjoy! Obviously, this can be a very droll process for the doctor, and he would welcome any way of shortening the time span he needs to take in analyzing the video. Some conceivable methods would be to create an algorithm to highlight statistically aberrant images or to recognize areas where the pill has not moved for quite some time and consolidate the images to a single frame. Research has been attempted to automatically find abnormal images or to reduce the time needed to analyze the video by locating the boundaries separating the gastric organs [3]. But it appears little research is being actively pursued.

Of course, CE is used to diagnose all types of ailments, but one of the main reasons to have the procedure is to find the occurrence of bleeding. Here we focus on the application of finding blood.

Blood detection is not new. Given Imaging has created with their video viewer a proprietary, non-disclosed method called Suspected Blood Index. The published findings about the quality of the method show that it is fair [5,9], but certainly not good enough to be reliable. SBI also only works in the small bowel.

Our goal here is to come up with a better method of detecting blood in video than SBI that works in all areas of the gestational track.

CHAPTER 2

BACKGROUND

2.1 A BLOOD PIXEL?

Human beings see all light as combination of three distinct colors, Red, Green, and Blue. The pill cam, as well as most commercially sold digital cameras, is designed to see the world as we see it, using filters specially designed to respond to frequencies of light corresponding to the three colors. It might be conceivable that a camera sensitive to more frequencies in the power spectrum of light frequencies might be better able to detect blood then our visual measure of color alone. Current day blood-oxygen monitors that are applied to the finger work by measuring the differential response of two waveforms of light transmitted through the finger. This measures the amount of oxygenated hemoglobin in the blood. It might be possible that something akin to this technology would prove a better method to see if we have free flowing blood in front of the pill camera. Until then, we can only work with what we have got.

The colors Red, Green, and Blue form the RGB colorspace. The data extracted from the videos is in standard 8-bit per color form; with color values ranging from 0 to 255, and giving us a total of 16777216 potential colors.

The hypothesis behind my thesis is that there is a range of colors that is unique to blood or blood tainted fluid. This hypothesis does not seem unreasonable, however, it is possible that that the color that blood takes is highly dependent upon the context in

which it is viewed, say from within the small bowel or the esophagus. Colors that might in one context define blood may in a different context define poorly lit or ulcerated small bowel flesh. If there is no set of pixel values for which we can ultimately declare "This is blood," then our research is in vain.

However the concept of delineating a region of the RGB colorspace as blood is enticing because delineation is simple. On average, we have around 50,000 images to process. As each image has approximately 45,000 viewable pixels, that means we have 2,250,000,000 pixels to process in an entire video! With that amount of data, if we want any expediency in processing, we cannot consider any image processing methods except the most simple. With the blood pixel concept, after we have found the blood range, we have at worst a 2MB bit array defining for every combination of Red Green and Blue whether or not it likely belongs to a blood flow, and in reality, we could implement it with much smaller memory with range predicates and compression techniques. Even if we want to consider some sort of fuzzy membership concept, with the massive memory of today's machines, we could store grades for every pixel with no problem.

So then, if we are to pursue the concept further, how do we find the appropriate range for blood? That, in essence, is the topic of this research. The proposed method for identifying an image is as follows:

1. Count the number of blood pixels in a capsule Endoscopy image.
2. If the number of blood pixels is beyond a given threshold, classify image as an image containing blood.

If we cannot define a pixel set based upon color alone, which we can likely say, "this defines the color of blood", it may be possible to reduce false positives by further analyzing the pattern of pixels. Are they spread out? Or do they clump together? It only depends on the selection of pixels to see if this idea is worth consideration.

2.2 A HISTORY OF PIXEL METHODS

The ability to track objects of specific color has always been a topic in the computer science field. Often the goal of some research is facial and or expression tracking. Many examples exist, but a few are given here.

Stochasticks: Augmenting the Billiards Experience with Probabilistic Vision and Wearable Computers [8] describes a system intended to become a wearable computer and help the wearer win a billiards game. They describe using the Expectation Maximization algorithm to train probability distributions defining the color of the pool table. If a pixel has a calculated probability beyond a certain threshold, it is regarded as a pool table pixel. The EM algorithm is also used to find appropriate probability distributions for balls and pockets. Because some pool balls are at least two colors, the previous method to identify the pool table does not apply. Multiple Gaussians are chosen and trained to model the objects in question. At runtime, samples of pixels are taken from candidate objects and a single Gaussian distribution is modeled. To determine the type of the candidate object, the Kullback-Leibler divergence is calculated between these two models and the closest match determines the object label.

In *A Robust Skin Color Based Face Detection Algorithm* [14] three algorithms based on three different colorspaces are used to detect facial skin regions in an image.

Their explanation is rather difficult to understand, but it seems to get the combined results, if any pixel is classified as skin by one of the algorithms, the final result will have that pixel classified as skin. Then further processing is done to obtain the facial area.

Much of what is in this thesis is based on the ideas mentioned in *Comparison of Five Color Models in Skin Pixel Classification* [16]. They implement two classes of skin classifiers using five colorspaces. One method takes advantage of only the color of skin whereas the other is Bayesian. The first method, which they name "Lookup Table" involves a histogram normalized its maximum value and various thresholds are applied to the histogram bins to deem if the pixels assigned to that bin should or should not be classified as skin. The second method is really based on two related methods, maximum likelihood and maximum a priori. They conclude the maximum likelihood Bayesian model is the best of them all.

"Does Colorspace Transformation Make Any Difference on Skin Detection?" [13] attempts to show the performance of up to eight non-RGB colorspaces and compares them to standard RGB as a baseline classifier. Two models for each colorspace, skin and non-skin, are created. Classifiers are evaluated by a scatter matrix based metric and histogram comparison metrics. They conclude transforming to another colorspace does not help.

Skin Segmentation Using Color Pixel Classification: Analysis and Comparison [11] is a recent paper comparing skin detection algorithms. They give good justification for using a large (256^3) three-dimensional histogram as a Bayesian Classifier to classify

skin pixels. However, as they point out, large datasets are needed to get a well performing classifier.

Finally, if you need more papers to understand the current state of various face detection methods, read *Face Detection in Color Images* [7]. The method used in this paper is not superior to any of the method mentioned previously, but it contains more references than any of the previous papers.

CHAPTER 3

THE EVOLUTION OF A BLOOD DETECTION ALGORITHM

If we are going to use a range of colors to signify "blood" then we must have some way to delineate this range. This chapter is dedicated to past methods attempted in finding an appropriate range

3.1 A CONE METHOD

My first idea was to select a cone shaped area in the RGB colorspace and declare that to be blood. The intuition behind the idea is as follows: Consider an RGB triple (r1, g1, b1). To find brighter pixels with the same chrominance, just multiply each value by some *l* greater than 1. To find darker pixels, multiply each value by *m* such that 0< *m* <1. Another way to look at this is in terms of spherical coordinates. We define

1. $Brightness = \sqrt{R^2 + G^2 + B^2}$

2. $\theta = \tan^{-1}\left(\frac{y}{x}\right)$

3. $\phi = \cos^{-1}\left(\frac{B}{Brightness}\right)$

We say any pixel having the same chrominance would have the same θ and ϕ. Because we expect there to be some variance in the color of blood, we can define a distance metric between two colors to be the distance between two colors on the unit sphere, where these colors are projected to the unit sphere by assuming *Brightness* = 1.

Progress on this stopped when it was realized that there does not seem to be an easy method to obtain an accurate average or ideal color using this color scheme. Also, this colorspace is not in wide use, leading me to wonder about the true utility of it. As it is fairly apparent, we need to exclude true black (0,0,0) since that pixel has no chrominance value. But it is not exactly clear what range of values ought to be excluded. Also, it was assumed that only a circle drawn on the sphere would give a sufficient range, but what if a better area was an ellipse? The only concept that seemed to be applicable in this situation was the idea of modeling Gaussians on a sphere, however the mathematics were difficult to understand and papers hard to locate. These difficulties forced consideration of other options.

3.2 A CLUSTERING/BAYESIAN METHOD

As mentioned in Chapter 2, there are papers that attempt to model the color of skin. Likelihoods that pixels were skin were estimated and used to find skin regions in new images. Unfortunately, blood in Capsule Endoscopy does not always have a clear border. See Figure 3.1. The same methods that worked in those papers would not necessarily work here. Borders subjectively drawn could affect the results. Blood also often contains other matter and bubbles, which skew results, or require borders to be much more complicated than need be. Clearly, we needed a way to solve these issues.

Once I discovered the Expectation Maximization Algorithm, I thought I had found the perfect solution.

Figure 3.1 A picture of blood tainted fluid in capsule endoscopy. As can be seen, its hard to define where the blood stops and the bile or intestinal wall begins.

3.2.1 THE EXPECTATION MAXIMIZATION ALGORITHM

The Expectation Maximization (EM) algorithm is a method to cluster data based upon a mixture of probability distribution functions (PDFs). That is, given a family of probability distributions and a dataset, the EM algorithm attempts to find the linear combination of PDFs that most optimally match the dataset. Most implementations of the EM algorithm assume a mixture of Gaussians, but that need not be the case. We will assume a Gaussian mixture model. See Figure 3.2. Given that data follows the distribution given by the top red curve, the EM algorithm is able to constitute the underlying distributions signified by green and blue curves. I only give details about our implementation in the following pages, but more info on Expectation Maximization can be found in *The Expectation Maximization Algorithm* [10].

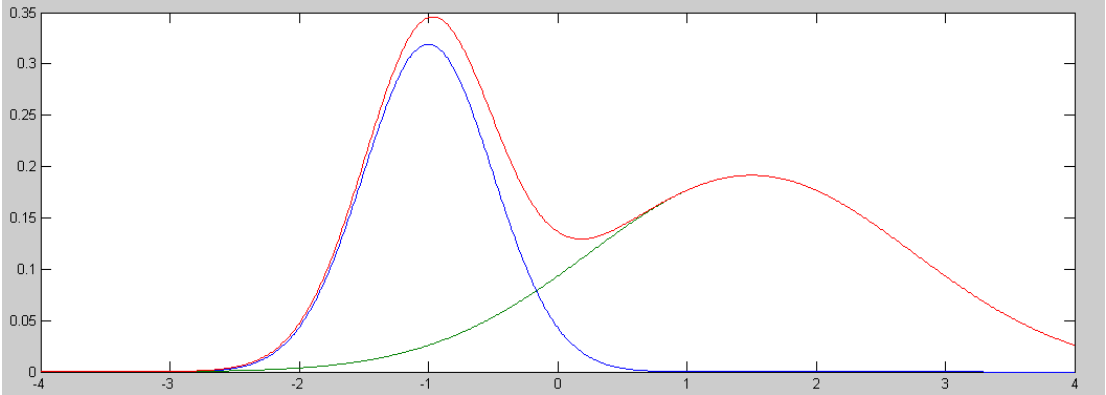


Figure 3.2 A linear combination of Gaussian Distributions. Matlab code to plot this is f = 0.4*normpdf(x,-1,0.5); g = 0.6*normpdf(x,1.5,1.25); plot(x,f,x,g,x,f+g);

3.2.2 INTUITION BEHIND USING THE EM TO FIND "IDEAL CLUSTERS"

The idea behind using the results of a clustering algorithm was simple. The clustering algorithm evaluates all data given and finds a matching cluster for every datum. Although it is true that many clustering methods cannot guarantee that the resulting clusters are the "best" clusters, in my experience with this dataset, the number of different resulting clusters tended to be few. Instead of the large variability of error that could have been introduced by having a human draw borders around blood regions, we only get the small error that the clusters might not be optimal. A two dimensional histogram of the pixels in HSV space (ignoring the "luminance" value V) potentially indicated the presence of two separate-able distributions. Furthermore, as a benefit of using the EM algorithm, we already have the key parameters needed to define PDFs.

3.2.3 THE EM IMPLEMENTATION

To implement the algorithm, formula were taken from Berkeley's Blobworld paper [4]. To ensure that I had the proper idea, other implementations were taken from the web and our implementation was tested against them. The Blobworld implementation required the use of Gaussians in N-Dimensional space parameterized by a mean and a covariance matrix. Two of the three implementations [1, 2, 15] found only worked with a simple variance. One [1] did indeed have covariance as a factor, but the algorithm was only applicable to two dimensions.

The covariance *covar(x, y)* between two random variables *x* and *y* is defined as follows where $E\{x\}$ designates the expected value (or mean) of random variable *x*.

$$covar(x, y) = E\{(x - E\{x\})(y - E\{y\})\}$$

Through algebraic reasoning and the use of identities on expected values, this expression becomes:

$$E\{xy\} - E\{x\}E\{y\} = \sum_{i=1}^{n} x_i y_i p(x_i, y_i) - \left(\sum_{i=1}^{n} x_i p(x_i)\right)\left(\sum_{i=1}^{n} y_i p(y_i)\right)$$

which is similar to how we chose to compute the co-variances for the covariance matrix.

The covariance matrix for random variables x_1, x_2, x_3, … , x_n is a *n* by *n* matrix *C* in which element *C*(*i*,*j*) is the covariance between x_i and x_j. Note this matrix is symmetric about the main diagonal since *covar*(*x*,*y*) = *covar*(*y*,*x*) and the diagonal is composed of the variances for every random variable x_i for $1 \le i \le n$. Thus a spherical

Gaussian with uniform variance could be represented with a covariance matrix that is the product of some scalar and the multiplicative identity matrix.

To start finding the optimal solution, the EM algorithm must be told how many kernels *K* there are and what values to "seed" the means and covariances. Then the EM algorithm is iterated until sufficient convergence of all the parameters we seek. For our implementation, this involves calculation of values based on the actual Gaussians we intend to model. The definition of the Gaussian is given below:

$$f_i(x \mid \theta_i) = \frac{1}{(2\pi)^{d/2}(\det C_i)^{1/2}} e^{-\frac{1}{2}\left[(x-\mu_i)^T * C_i^{-1} * (x-\mu_i)\right]}$$

where θ_i represents both the parameters mean μ_i and covariance matrix C_i, d is the dimension of x (in this case 3 for the three color values), $\det C_i$ represents the determinant of C_i, and "*" represents matrix multiplication.

To find the optimal linear combination, we must find the mean and covariance matrix for each PDF, but the also the weight α_i for which it contributes to the dataset. The update equations to find α_i^{new} and θ_i^{new} (represented by μ_i^{new} and C_i^{new}) follow:

$$p(i \mid x_j, \theta_i) = \frac{\alpha_i f_i(x_j \mid \theta_i)}{\sum_{k=1}^{K} \alpha_k f_k(x_j \mid \theta_k)}$$

$$\mu_i^{new} = \frac{\sum_{j=1}^{N} x_j p(i \mid x_j, \theta_i^{old})}{\sum_{j=1}^{N} p(i \mid x_j, \theta_i^{old})}$$

$$C_i^{new} = \frac{\sum_{j=1}^{N} p(i \mid x_j, \theta_i^{old}) x_j x_j^{T}}{\sum_{j=1}^{N} p(i \mid x_j, \theta_i^{old})} - \mu_i^{new} (\mu_i^{new})^T$$

$$\alpha_i^{new} = \frac{1}{N} \sum_{j=1}^{N} p(i \mid x_j, \theta_i^{old})$$

The algorithm itself was formulated such that in one iteration, $f_i(x \mid \theta_i)$ need only be calculated K times per data item, and the new values α_i^{new} and θ_i^{new} could be incrementally summed per each pixel. The value $C_i^{new} + \mu_i^{new} (\mu_i^{new})^T$ can be calculated as an intermediate value only to have $\mu_i^{new} (\mu_i^{new})^T$ be subtracted out once μ_i^{new} was known. As written, the algorithm is highly parallelizable. We have an 8-processor SMP machine to do our bidding, and it was tempting to employ threading to cluster the millions of pixels. However, there was found an even simpler method.

3.2.4 CLUSTERING OVER HISTOGRAMS

Although the standard RGB colorspace is huge, around 16 million potential values, I found the number of actual unique colors per image set to be a very small fraction of that number. So it was hypothesized that we could cluster the data faster if we didn't cluster directly on the data but first create a histogram of the data and then cluster on the histogram. With very slight adjustments to the EM algorithm, it was found we could implement this optimization. So, to cluster on the data, we first generate a histogram based on the whole colorspace (256*256*256). Every bin has a count w_i of how many times a certain pixel value occurs in the dataset. We then take the non-zero bins along with their associated counts and fed them into the modified algorithm. The modifications to the EM algorithm are shown below:

$$p(i \mid x_j, \theta_i) = \frac{\alpha_i f_i(x_j \mid \theta_i)}{\sum_{k=1}^{K} \alpha_k f_k(x_j \mid \theta_k)} \qquad \rightarrow p(i \mid x_j, \theta_i) = w_j \frac{\alpha_i f_i(x_j \mid \theta_i)}{\sum_{k=1}^{K} \alpha_k f_k(x_j \mid \theta_k)}$$

$$\alpha_i^{new} = \frac{1}{N} \sum_{j=1}^{N} p(i \mid x_j, \theta_i^{old}) \qquad \rightarrow \quad \alpha_i^{new} = \frac{\sum_{j=1}^{M} p(i \mid x_j, \theta_i^{old})}{\sum_{j=1}^{M} w_j}$$

where N is the number of data points, and M is the number of unique data points. Thus $N = \sum_{j=i}^{M} w_j$. For some of the datasets that we clustered, the total number of pixels to cluster was around 8 million. The total number of unique colors was under 150 thousand. Thus, for that data set, the clustering speed increased by at least a 53 times by implementing this minor change.

3.2.5 THE EXPERIMENT AND RESULTS

To provide for the clustering, around 200 "blood" images were selected and their pixels extracted. Clustering into two groups was attempted in multiple colorspaces: HSV (ignoring V) YUV (ignoring Y), CIE LAB, and RGB. The two dimensional colorspaces were only considered because since HSV and YUV involve a linear orthonormal transformation, which means distances between pixels in the 3 dimensional colorspaces would not change (but they would in the 2 dimensional variants!), and thus clustering in them would prove absolutely no better than in RGB. To identify the blood cluster, the cluster with mean closest to the Red axis was selected. To find out which was closest to red, sometimes we had to transform the mean values back into RGB.

Classification using this method was based on the parameters for PDFs created as a function of clustering. In the two-cluster method, for every incoming pixel, the

pixel was evaluated for each PDF. For whichever the evaluation was higher, the pixel would be assigned to that class. The process was extended for the three-cluster method, with two non-blood PDFs and one blood PDF.

The results of this particular method proved too poor to even consider for further evaluation or explanation. The two dimensional colorspaces performed the worst. Surprisingly, the two-cluster classification results for RGB and CIE LAB proved quite similar visually in the pixel classification results of images, yet also similarly abysmal.

3.3 A BAYESIAN METHOD BASED ON SAMPLING

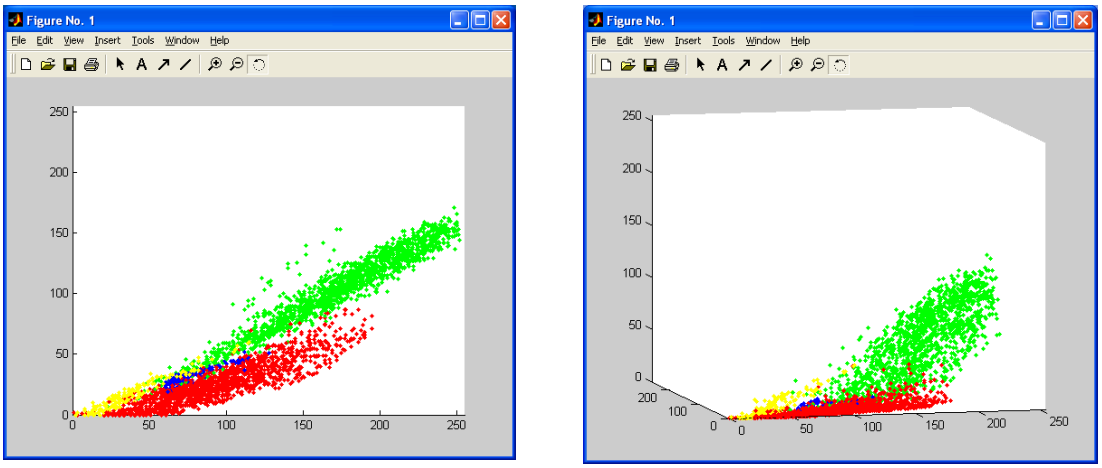
With the last method proving disappointing, I decided that in fact there had to be some human intervention necessary in order to accurately tune a blood collector. There had to be more than just a human selecting images, someone had to approximately select the regions that identified blood. I decided on creating a model based on two Gaussian distributions, one blood and one non-blood. Sixteen hundred blood pixels (25 pixels each from a set of 64 images) and sixteen hundred non-blood pixels were selected. The blood pixels consisted of pixels from blood and blood-tainted fluid. The non-blood pixels consisted mainly of gastrointestinal wall. To select the pixels, a human "randomly" selected pixels from all over the blood or non-blood regions. To select from the blood regions, the human only had to avoid bubbles and such. Means and covariances were collected from the data and used in the modeling of Gaussians. Again, the idea behind this method was to use the probability generated by the Gaussians to determine whether or not to classify a pixel was likely blood or not.

To classify a pixel x as blood, the pixel must satisfy two rules. First, the Euclidean distance of the pixel from the origin in RGB colorspace had to be greater than 60. This threshold is a little arbitrary; there exist a number of images of esophageal tissue where dark regions may be considered bloody if only looking at those specific regions. Furthermore, in some experiments a threshold proved efficacious in removing false positives from consideration. Unfortunately, I know of no method of selecting an optimal "darkness threshold" except by manual experimentation. Second, it must be the case that $p(x \mid \text{blood}) > p(x \mid \text{nonblood})$ as defined by the two Gaussians in the previous paragraph.

Images were determined to be blood by a blood pixel count threshold of 3000. If the blood pixel count for an image was more than or equal to this threshold, it was deemed to be a blood image.

To test this method, 400 clearly blood and 400 clearly non-blood images were selected. The accuracy and precision of this method for this test proved to be 94% and 99% respectively. However, when extending the test further

To further prove the efficacy of the method the datasets were extended. It seemed reasonable that the non-blood dataset needed to be much larger to test the huge variability of potential images captured by CE. It was in this extension that problems were discovered with this method.



a) b)

Figure 3.3 a) A view of the Red-Green (red is X, green is Y) Distribution of misclassified esophagus pixels (Yellow) misclassified small bowel pixels (Blue), normal bowel training dataset (Green), and the blood pixel training set (Red). b) a three dimensional view of them(Red axis is pointing to the right, Green axis points slightly to the left, and Blue is up.

Firstly, the blood area in RGB spanned a greater volume than it ought. We used only two distributions to model the blood and non-blood respectively. But what if there were pixels which fit neither model well, but for which the PDF value for blood was still greater than the PDF value for non-blood? Since the non-blood distribution contained no pixels from bile regions, the model was not sufficiently trained to handle that substance and yet the greenish yellow substance appeared to be closer to the blood distribution.

Secondly, as more images were added, an inconsistency with the way pixels were classified (by human eyes) was revealed. See Figure 3.3. It appeared for certain pixels the modeled likelihood that these pixels were blood was calculated to be far greater than the true value. Several investigations showed that these misclassified

pixels were indeed quite within the cluster of blood pixels, but very localized in the RGB colorspace, consisting of only a small region of the blood pixel region.

Finally, doubts arose upon whether a single Gaussian would be good enough to model the entire variety of pixel colors in an image. Although the pixels of various blood regions tended to be clumped together in a large ellipsoid, the 1600 non-blood pixels viewed in 3d space showed the non-blood distribution to be very odd shaped, almost forming a fuzzy triangle. The three endpoints generally consisted of points around the origin, light points from the esophagus or gastric regions, and light pixels from the small bowel.

CHAPTER 4

FINAL PROPOSED METHOD: DISCUSSION AND RESULTS

In the previous sections, I have given details of failed algorithms. Now I will give you details of the algorithm that seems to work the best.

4.1 IMPLEMENTATION DETAILS

In implementation of the last method, I discovered that the Gaussian modeled after the blood data was possibly giving blood likelihood probabilities higher than they ought to be. The first fix was to sample more data. Pixels from 137 blood-free images from 6 videos were clustered into one (G1), two (G2), and three clusters (G3), and then I used the Gaussian mixtures obtained from the clustering to model the non-blood data. Video segments were taken from video of the esophagus and the small bowel. Still, there were issues of pixel misclassification, and they tended to be in the same region of RGB space as before. So, it seemed there ought to be some way to filter out the bad pixels from the good. But how? The simplest method seemed to be to use one of the Gaussian mixtures to filter out the bad pixels. If the probability as measured from the Gaussian mixture was greater than the uniform probability, that is, for pixel x, $p(x \mid \text{nonblood}) = \sum_i^n w_i f_i(x \mid \theta) > 1/256^3$, the pixel was regarded as a non-blood pixel, and thus thrown away. To perform the filtering the three-cluster model (G3) was utilized as it was thought to model the non-blood distribution the best. Eight hundred eighteen pixels of the sixteen hundred survived the selection process, and then were used to model a new Gaussian.

4.1.1 ALGORITHM DEFINITION

We only consider a pixel to be blood when the following three conditions are met:

First, The distance between the pixel and the origin is greater than 40. This is lower than the "darkness threshold" mentioned in Section 3.3, and we keep this thresholding technique for the same reasons mentioned in 3.3. Because the accuracy of the classifier has improved, the threshold was lowered to account for pixels in the 40 to 60-distance range. However, whenever pixels are at a distance somewhere below 40, it is very hard for the human eye to separate blood from non-blood. In the current implementation, this clause may never actually get tested, however, we keep it in because we know that it would be considerably hard to verify if pixels in this RGB colorspace region do indeed identify blood.

Second, the probability that a pixel is blood is greater than the uniform probability that the pixel might be something else, or in mathematical terms, $p(x|\text{blood}) > 1/265^3$. In essence, this is an outlier detector and throws out pixels for which the likelihood that the pixel is actual blood is very small but still may pass the third test.

Third, The probability that a pixel is blood is significantly larger than the measured probability that the pixel is non-blood. We say $p(x|\text{nonblood})*\tau < p(x|\text{blood})$ or $\tau * \sum_i^n w_i f_i(x \mid \theta_i) < p(x \mid \text{blood})$. In Bayesian classification literature, it is common to give a threshold τ to determine if a something belongs to a class or not, however, we

experiment with various thresholds empirically to find a good τ. We found τ=3 to be reasonably good.

4.2 FINAL EXPERIMENT

To run these tests, we had to extract images from CE videos. Because Given Imaging's videos are in a proprietary format, we could not directly read the video. The Given Imaging reader provides a functionality to export videos from the main video, so this was utilized. To further simplify matters, images were extracted as JPEGS from the exported videos using Ulead 5.0 software.

To perform the final experiment, clips were selected using the Given Imaging 3.0 reader. Our group has access to 27 capsule endoscopy videos. Of those, only 6 were found to contain any blood. The blood images were whittled down to a selection of 1731 images that the eye could ascertain definitely contained blood. To create a final non-blood training set, 5 clips of about 100 images each were taken from each of the videos giving a non-blood dataset of 13491 images. At least one of the clips was taken from the esophagus or gastric regions whenever available.

The images are 256 by 256, however the center circle contains around 45000 pixels. A radius of 120 pixels was used to select pixels for classification.

To classify pictures, we needed a blood pixel threshold. Thresholds of 1000 and 3000 were utilized. Training of the classifiers was described in section 4.1.

4.3 CONCLUSIONS

When applied to images, each of the classifiers gave similar results in regions where pixels were classified. Appendix A gives a slew of images along with their classification results by G3 with $\tau = 3$. The borders of the blood regions generally tended to be located close to where one might subjectively define the border. As can be seen, it would be hard to give pixel misclassification results.

Appendix B gives confusion matrices for each of the classifiers with thresholds of 1000 and 3000, respectively. Note the true positive rate, while high, may not be as high as we would like. Generally speaking, the blood cases that were missed went into three classes: Dark red images (where most of the pixels fell below the threshold), weakly tainted blood images, or low blood content images. To be fair to test the algorithm, images were selected which did not have as much blood as those images in appendix A. We know such images may be of import, because sometimes the only case of bleeding in the bowel may be small.

To test the quality of the classifiers, accuracy, precision, and recall are calculated for the various classifiers. These three measures of performance are calculated in the following way. In testing any classifier, you know the ground truth, and the classifier is tested over two sets of data: one positive for something that is sought and one negative. So you will get four values: The true positive count (TPC) for those values correctly classified as positive, the false positive count (FPC) for those items incorrectly tested as positive, the false negative count (FNC) for those incorrectly tested as negative, and the true negative count for those correctly tested as negative.

Figure 4.1 A very reddish and yet apparently blood-free image.

Formulas for the various quality measures including are as follows [6]:

Accuracy = (TPC +TNC)/(TPC+TNC +FPC+FNC)

Precision = TPC/(TPC+FPC)

True Positive Rate =Recall = TPC/(TPC+FNC)

False Positive Rate = 1 - Recall

While the False Positive Rate is low for all three classifiers, we must remember that the expected number of non-blood images is quite high. However there is a strong tendency for the pixels that are misclassified to be in a localized region in the RGB colorspace. Most of the images that were misclassified were of a bright orange-red hue. See figure 4.1 for an example image. This leads me to believe that the selection of images for training the non-blood dataset was not broad enough to encompass the full range of values that non-bloody pixels can take, especially those rather close to the blood distribution.

To get an impression of how well the best classifier G3 is working, a Precision/Recall graph [6] constructed from the test is included in Appendix C considering only the G3 classifier. Also included is a graph measuring the overall quality of the classifier at various thresholds using the distance the precision/recall pair from that of the perfect classifier, which has both a precision and recall of 100%. Or, in formulaic terms, $E(P,R) = \sqrt{(1-P)^2 + (1-R)^2}$. While it appears a blood pixel count threshold of 1000 is close to optimal using this metric, the actual optimal threshold is around 1500 with a precision of 88.5% and recall of 89.3%. However, the applicability of this optimality metric to the precision and recall measurements of a blood classifier is not really known.

In one study [9], SBI, when only considering bleeding lesions, has a Sensitivity (recall), Positive Predictive Value (precision) and overall diagnostic accuracy of 81.2%, 81.3%, and 83.3%, respectively. On this study, the new algorithm apparently beats SBI. However, it should be noted that while this test was over a significant number of images, the SBI study was over several videos. It is not terribly clear how they are calculating these metrics. More tests are necessary to prove that this method can in fact beat SBI. I also suggest that one needs much more than 137 images to train the non-blood classifier.

Furthermore, sometimes it is incredibly hard for the human eye to ascertain that an image indeed contains blood. And still, some images were included in the blood dataset that might not actually contain a relevant amount of blood. So there are a few

questions about the accuracy of this method, however, so the same can be said about SBI. In one of the videos our group has in its collection, SBI marks a massive number of images of apparently ulcerated small bowel positive. While this has clinical value, the positive predicative value and accuracy (with regard to bleeding lesions) would certainly drop.

Much can be only resolved with more study. However, I hope this paper proves a good stepping-stone to others who may want to attempt the creation of their own suspected blood index. This method is a start. I hope you can learn from it.

APPENDIX A

A SAMPLE OF IMAGES AND THEIR BLOOD CLASSIFICATIONS

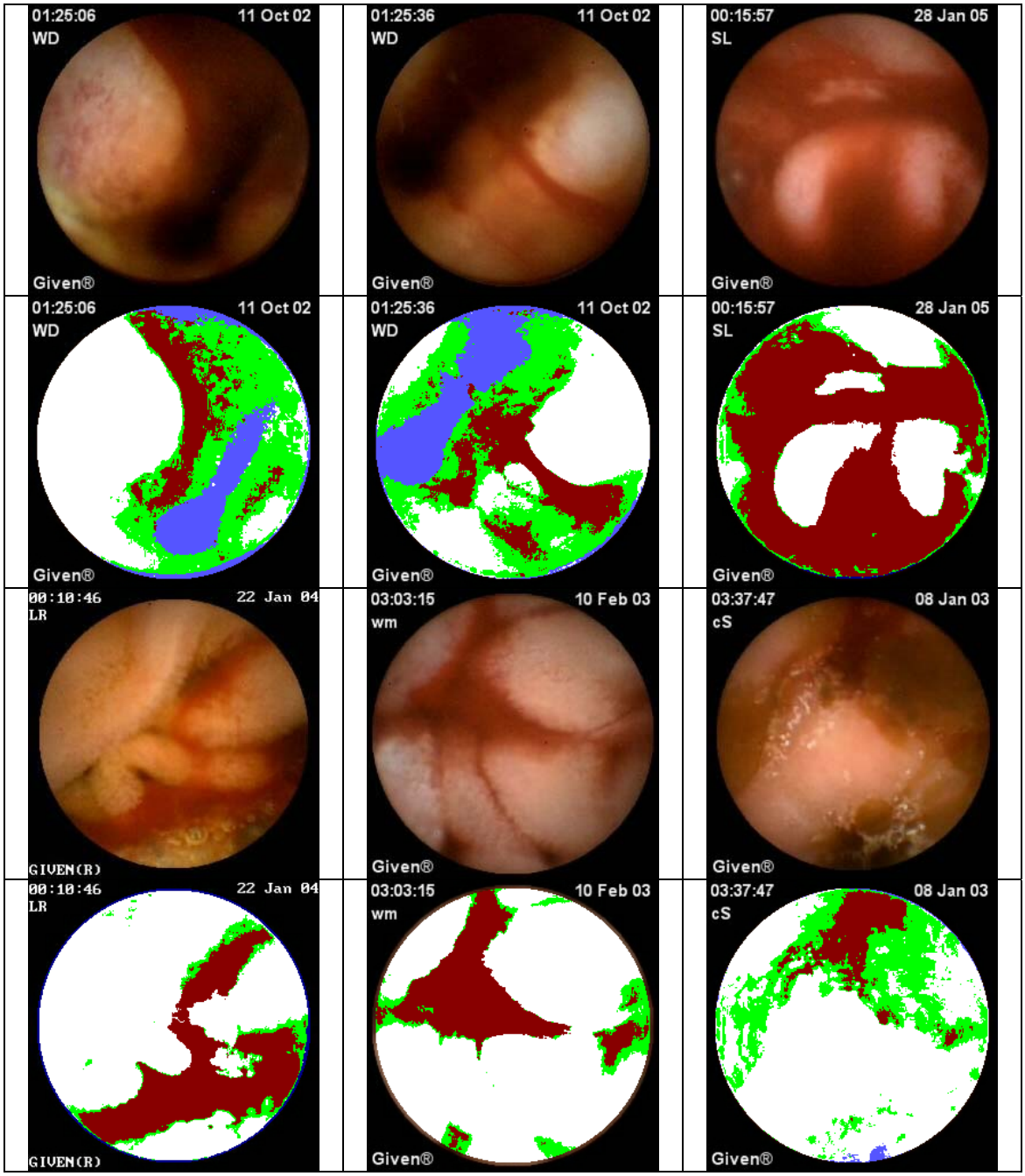


Figure A.1 A sample of images showing the areas classified as blood by the G3-classifer. White shows regions with low probability as being blood. Green shows areas with a higher probability of being blood, but the probability is less than 3 times the probability of being non-blood. Blue shows areas where the pixels are below the "darkness" threshold. Dark red shows blood pixel areas.

APPENDIX B

CONFUSION MATRICES FROM FINAL EXPERIMENT

Table B.1 Confusion matrices from the final experiment.

Threshold of Blood Pixel Count =1000	Threshold of Blood Pixel Count = 3000

		G1 Classified	
		Positive	Negative
Actual	Positive	1595	136
	Negative	274	13217

Accuracy	97.3%	Recall	92.1%
Precision	85.3%	FPR	7.9%

		G2 Classified	
		Positive	Negative
Actual	Positive	1606	125
	Negative	283	13210

Accuracy	97.3%	Recall	92.7%
Precision	85.1%	FPR	7.3%

		G3 Classified	
		Positive	Negative
Actual	Positive	1602	129
	Negative	265	13226

Accuracy	97.4%	Recall	92.5%
Precision	85.8%	FPR	7.5%

		G1 Classified	
		Positive	Negative
Actual	Positive	1418	313
	Negative	156	13335

Accuracy	96.9%	Recall	81.7%
Precision	90.1%	FPR	18.3%

		G2 Classified	
		Positive	Negative
Actual	Positive	1435	296
	Negative	161	13330

Accuracy	97.0%	Recall	82.7%
Precision	89.9%	FPR	17.3%

		G3 Classified	
		Positive	Negative
Actual	Positive	1421	310
	Negative	138	13353

Accuracy	97.1%	Recall	81.9%
Precision	91.2%	FPR	18.9%

APPENDIX C

PRECISION/RECALL GRAPH OF G3 CLASSIFIER

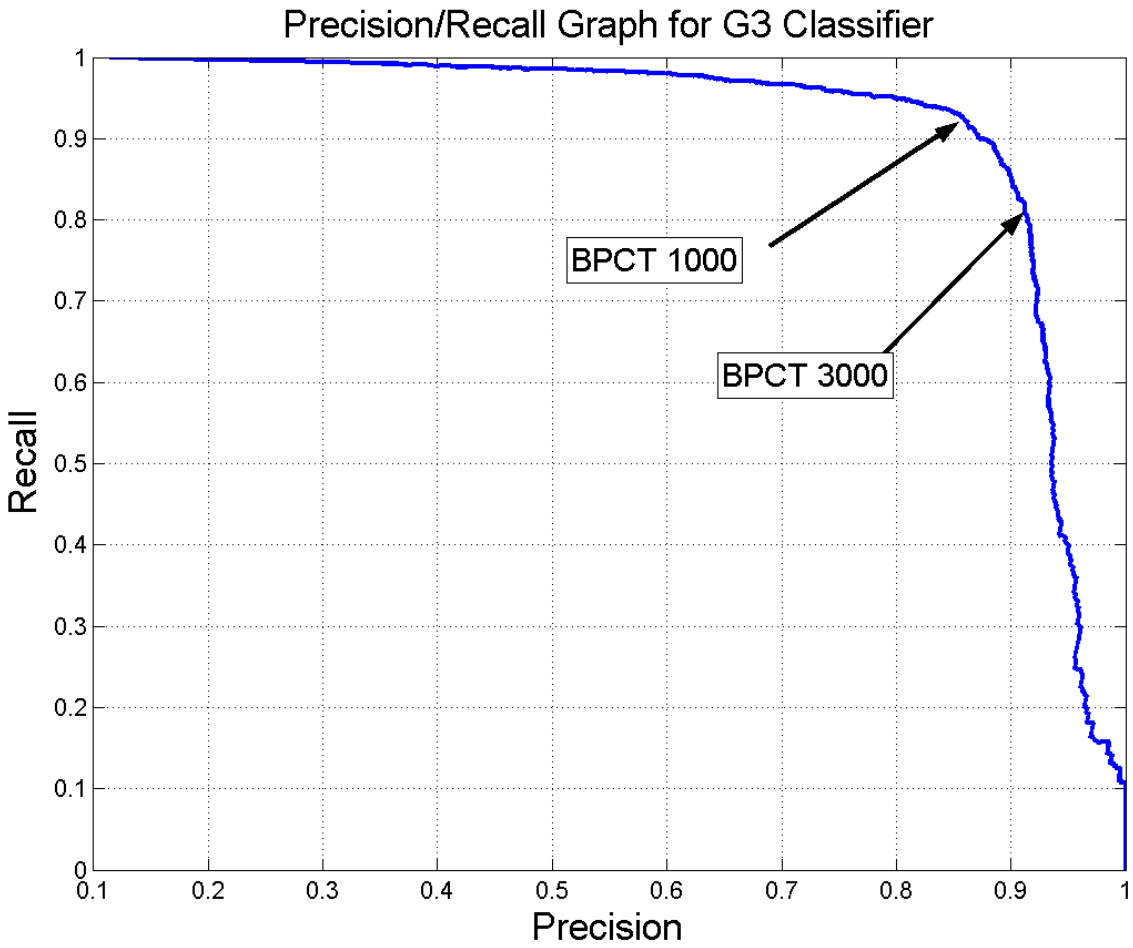


Figure C.1 A Precision Recall Graph for the G3 classifier. The arrows point to the precision/recall values for blood pixel count thresholds of 1000 and 3000 respectively.

APPENDIX D

GRAPH OF PERFORMANCE BY THRESHOLD

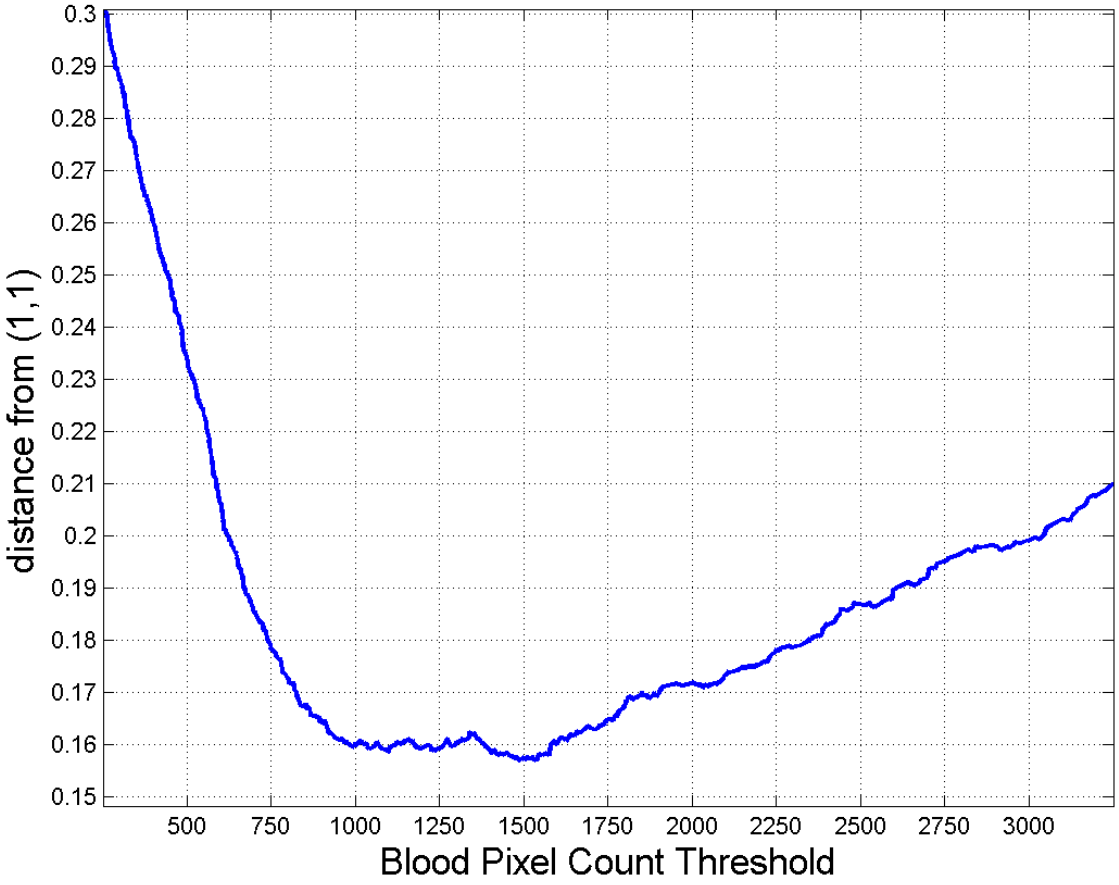


Figure D.1 Classifier Performance as measured by the metric

$$E(\text{Precision}, \text{Recall}) = \sqrt{(1-\text{Precision})^2 + (1-\text{Recall})^2}$$

REFERENCES

1. Shotako Akaho. EM algorithm for Mixture models (test version). http://www.neurosci.aist.go.jp/~akaho/MixtureEM.html

2. Ismet Bayraktaroglu. Expectation Maximization Algorithm, by Ismet Bayraktaroglu. http://www.cs.ucsd.edu/users/ibayrakt/java/em/

3. Jeff Berens, Micheal Mackiewicz, Duncan Bell. Stomach, Intestine and Colon tissue discriminators for Wireless Capsule Endoscopy images. In *Proceedings of the SPIE*, Volume 5747, Bellingham, WA, April 2005.

4. Chad Carson, Serge Belongie, Hayit Greenspan, Jitendra Malik. Blobworld: Image Segmentation Using Expectation-Maximization and Its Application to Image Querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence archive*. Volume 24, Issue 8, Pages: 1026 – 1038, August 2002.

5. Pierre-Nicolas D'Halluin, Michel Delvaux, Marie-Georges Lapalus, Sylvie Sacher-Huvelin, Emmanuel Ben Soussan, Laurent Heyries, Bernard Filoche, Jean-Christophe Saurin, Gerard Gay, Denis Heresbach. Does the ''Suspected Blood Indicator'' improve the detection of bleeding lesions by capsule endoscopy? *Gastrointestinal Endoscopy.* Volume 61, No. 2, Pages: 243-249, 2005.

6. Howard Hamilton, Ergun Gurak, Leah Findlater, Wayne Olive. Computer Science 831: Knowledge Discovery in Databases. http://www2.cs.uregina.ca/~hamilton/courses/831/index.html.

7. Rein-Lein Hsu, Mohamed Abdel-Mottaleb, Anil K. Jain. Face Detection in Color Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 5, May 2002.

8. T. Jebara, C. Eyster, J. Weaver, T. Starner, and A. Pentland. Stochasticks: Augmenting the Billiards Experience with Probabilistic Vision and Wearable Computers. In *Proc. of the Intl. Symposium on Wearable Computers*, Cambridge MA, October 1997.

9. Suthat Liangpunsakul, Lori Mays, Douglas K. Rex. Performance of Given Suspected Blood Indicator. *American Journal of Gastroenterology*. Vol 98, No. 12, 2003.

10. T.K. Moon. The expectation-maximization algorithm. *Signal Processing Magazine*, IEEE. Volume 13, Issue 6, Pages: 47 – 60, November 1996.

11. Son Lam Phung, Abdesselam Bouzerdoum, Douglas Chai. Skin Segmentation Using Color Pixel Classification: Analysis and Comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No 1 , January 2005.

12. M. L. Remedios, M. Appleyard. Capsule Endoscopy, Current Indications and Future Prospects. *Internal Medicine Journal*, Vol. 35 Issue 4, Pages 234-239, April 2005.

13. Min C. Shin, Kyong I. Chang, Leonid V. Tsap. Does Colorspace Transformation Make Any Difference on Skin Detection? In *Proceedings. Sixth IEEE*

Workshop on Applications of Computer Vision (WACV 2002). Pages: 275 – 279, December 2002.

14. S.K. Singh, D.S. Chauhan, M. Vasta, and R. Singh. A Robust Skin Color Based Face Detection Algorithm. *Tamkang Journal of Science and Engineering*, Vol. 6, No. 4, Pages: 227-234, 2003.

15. Ian H. Witten, Eibe Frank. Data Mining: Practical machine learning tools and techniques, 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

16. B.D. Zarit, B.J. Super, and F.K.H. Quek. Comparison of Five Color Models in Skin Pixel Classification. In *Proc International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, Pages: 58-63, 1999.

BIOGRAPHICAL INFORMATION

Jay Cox was born in Fort Worth, Texas on July 22, 1978. He graduated from Angelo State University in 2000 with a Bachelor of Science in Computer Science and Mathematics. In 2002, Jay enrolled at the University of Texas at Arlington. He will graduate having earned membership in various honor fraternities such as Tau Beta Pi and Upsilon Pi Epsilon. Jay's plans are to research life in general, and maybe to do an occasional stint as a technical researcher.