# ECE 445 Project Proposal

# A Parallelized Algorithm for Hyperspectral Biometrics

Christopher Baker

Timothée Bouhour

Akshay Malik

**Table of Contents** **Page No**

# I. Introduction

**Motivation:**

In the modern world unique person identification has become an increasing challenge, central to strategies in combating terrorism and crime to provide global security. Recent research has shown that hyperspectral imaging provides new and improved biometric data, which can be leveraged to meet this challenge by examining features in different spectral bands.

Despite its promise, this method of identification still has some challenges, which must be addressed before it can be applied in the real world. One of those challenges is dealing with the massive amount of data that a hyperspectral sensor generates. We will be developing an algorithm to solve this data processing problem based on GPU parallel processing. Our algorithm will be scalable to allow it to be expanded and used as the technology develops into real world application.

**Objectives:**

Our overarching goal in this project is to provide a high throughput processing back end for biometric identification based on hyperspectral images of a person's face. Our product will be an algorithm coded and running on a GPU using a language such as CUDA. Our specific objectives include the following:

1. Determine which facial features work best for hyperspectral biometrics
2. Develop, code, and test an optimized parallel identification algorithm

We aim to provide a solution such that any hyperspectral camera can be used to collect data. Once the data is formatted properly, we should then be able to identify the person in the picture given that we have previous data on that person. The functions of our final product include the following:

1. Identify a person against a database based on a hyperspectral image of their face
2. Match a hyperspectral image of a face against a database using a GPU
3. Efficiently store and access a database of hyperspectral images
4. Send a notification locking or unlocking a door upon recognition

Using our product will provide the customer with the following benefits:

1. Provide a better and more reliable way to recognize faces
2. Provide processing for massive facial data input in near real time
3. Identify a face despite changes (mask, aging, plastic surgery, etc.)
4. Automate area security (locking/unlocking doors, alarms, etc.)
5. Provide a key tool for creating a more secure and controlled environment

6. Possible to provide target identification for military applications

Our products features would include the following:

1. Accept hyperspectral image of a face as input
2. Lock/unlock doors wirelessly based on recognition
3. Ability to utilize GPU resources to achieve order of magnitude processing speed advantages over simple CPU based algorithms
4. Efficiently store a database of subjects
5. Efficiently access database for comparison purposes
6. Efficiently scale to different sizes of system processing resources
7. Output the closest match and a feature match percentage upon completion.

While our product has many benefits, we feel that it is important to not here that we are not going to be dealing with the sensor or automated preprocessing for input to our algorithm. We are simply handling the backend processing, data basing, and notification challenges.

## II. Design

**Block Diagram**

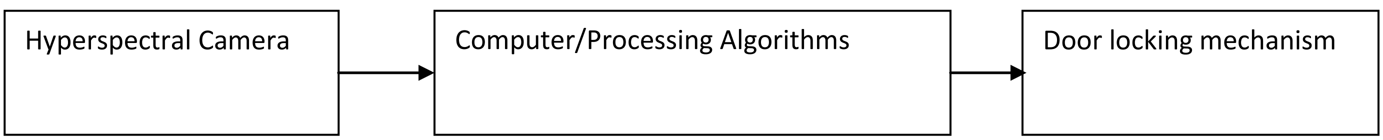


Figure 1. Block Diagram

**Block description**

Hyperspectral Camera:

This module is responsible for collecting the hyperspectral information to be processed. We will obtain preliminary data from a research group at Carnegie Mellon University[1] and a research group at Hong Kong Polytechnic University[2]. We also have a verbal commitment from Raytheon to get direct access to a hyperspectral camera in their facilities and obtain data from there. Although this module is an integral part of our design, we are not addressing its development in the scope of this project. We are only going to use data that has been collected with pre-existing systems.

1. Louis J. Denes, Peter Metes, and Yanxi Liu, "Hyperspectral Face Database," tech. report CMU-RI-TR-02-25, Robotics Institute, Carnegie Mellon University, October, 2002

2. Wei Di, Lei Zhang, David Zhang, and Quan Pan "Studies on Hyperspectral Face Recognition in Visible Spectrum with Feature Band Selection" *IEEE Trans. on System, Man and Cybernetics, Part A*, vol. 40, issue 6, pp. 1354 – 1361, Nov. 2010

Computer / Processing Algorithm

This module forms the bulk of our project. It utilizes a CPU, a GPU, and a database of hyperspectral information from multiple subjects. A description of the functionality of this module is given in the flow chart (figure 2). The data contains an image of a person's face and is obtained from a hyperspectral sensor. It is in the form of varying intensities for different optical bands varying from 400-1100nm (visible to near-infrared region). We will be identifying a set of features whose hyperspectral information can uniquely identify a person. These features will be automatically extracted from the data. The algorithm will then process this information and compare it to an existing database. The algorithm will be developed using CUDA and parallelized using a GPU. When a match is detected, it will generate a signal that will indicate whether the door locking mechanism should be engaged. The output will show the closest match and percentage accuracy.

Door locking mechanism

This module will contain a wireless receiver and a microcontroller. The microcontroller will control a small motor to operate a deadlock. When a positive match is detected, the motor will be controlled to lock the deadbolt, and unlock when the match is negative.

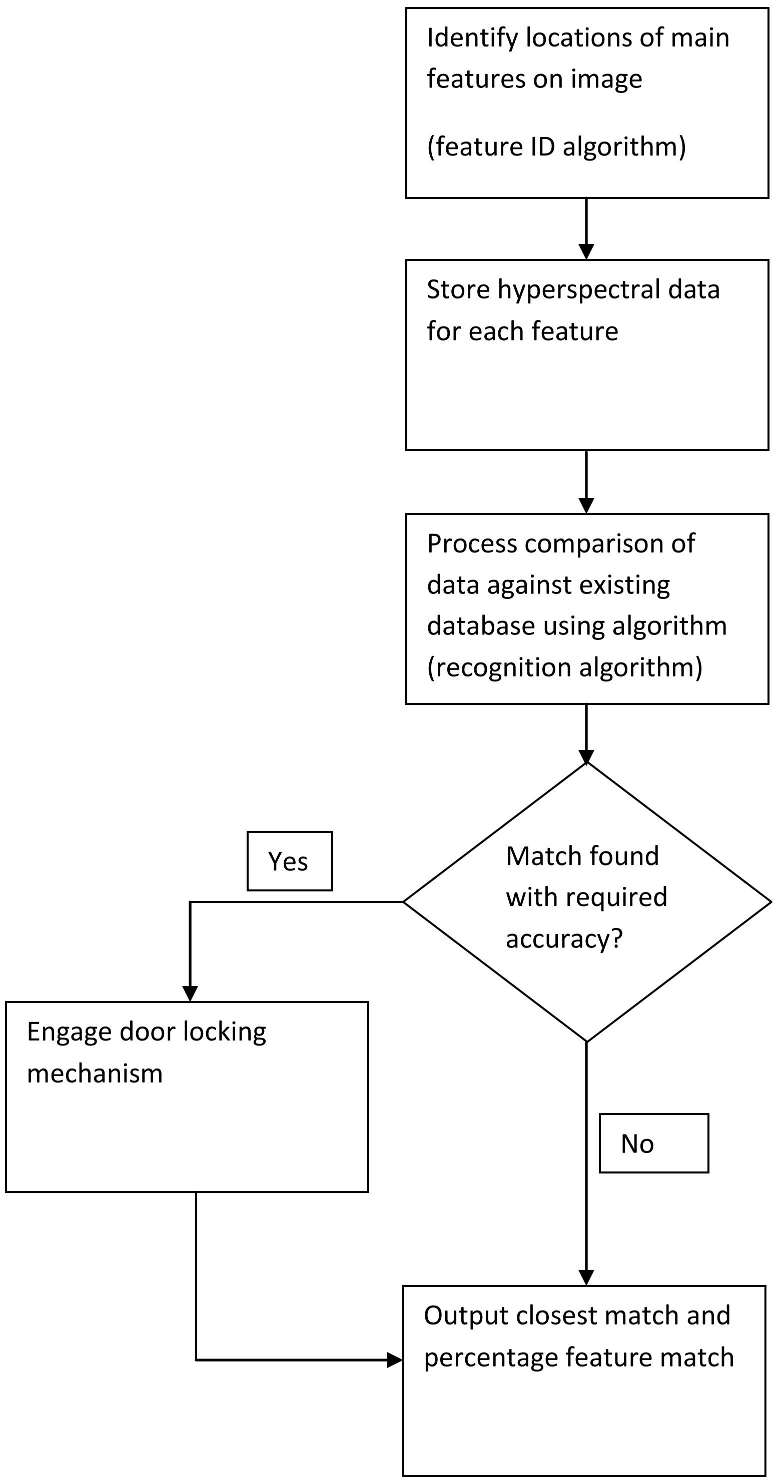


Figure 2. Algorithm Flow chart

# III. Requirements and Verification

| Block | Requirement | Verification |
|---|---|---|
| Hyperspectral camera data | Data is hyperspectral | Wavelength of data between 400 and 1200 nanometers and intensity information in different bands |
| Hyperspectral camera data | Data can be recognized by computer | Data is in a file format readable by Linux operating system. |
| Computer / Algorithm | Database is sufficiently and correctly populated | At least 20 unique entries in the database, with complete hyperspectral data for each feature. At least 200 total entries including automated generated data for testing. |
| Computer / Algorithm | Algorithm can correctly read data from new pictures and compare it against the database to output best match and accuracy number | Running the algorithm against a picture should lead to an identification and an accuracy rating. |
| Computer / Algorithm | Comparison of a new picture of an individual with an old, existing picture in the database of the same individual should yield that individual as best match | Use at least two different hyperspectral images for each individual, one in the database and another to compare. All images should yield the correct match. |
| Computer / Algorithm | If a match within 70% feature correlation is found, must output a positive signal to the door locking mechanism | Use a database photo for good match, pass through the photo and ensure proper signal output. |
| Door Locking Mechanism | The door locking mechanism must properly receive positive signals from the computer block. | Output a positive signal from the computer block; ensure proper reception through a LED. |
| Door Locking Mechanism | If a positive signal is properly received from the Computer Block, the door locking mechanism must engage the lock | Pass a positive signal to the door locking mechanism; ensure that it properly engages the lock. |

**Tolerance Analysis**

We will run our algorithm with the data we have from actual hyperspectral images as well as automatically generated data for speed testing purposes (it is not possible for us to get a very large data set on the order of hundreds or thousands of subjects). Two major components affect the performance of our system the most, both related to the algorithm.

The first component is accuracy. The testing procedure for this would be to run hyperspectral images against the database. The requirement is a minimum of 20 subjects, each with an existing image in the database and a different image for testing, and a minimum of 180 other database entries (automatically generated). In these conditions, the algorithm should correctly identify the subject with his database entry at least 70% of the time.

The second component is speed. Because we are developing a basic parallel algorithm, the speed of the algorithm using a GPU should be greater than when using the CPU. Thus, we will run the algorithm on the computer without the utilization of the GPU and compare it to the speed with a GPU. To obtain a good benchmark, we will use automatically generated data in addition to our obtained data to produce a large enough data set.

## IV. Cost and Schedule

**Cost Estimates:**

We expect our main costs aside from labor to come from additional data collection in this project and building a computer dedicated for testing our algorithm. We expect our total project cost not including labor to be significant, but we will be receiving sponsorship from Raytheon Space and Airborne Systems. Our current plan is to travel to California and use one of Raytheon’s labs and sensor setups to collect our data.

***Labor***:

| Name | Rate/Hour | Overhead(x2.5) | Hours* | Total |
|---|---|---|---|---|
| Chris | $45 | $112.5 | 240 | $27,000 |
| Timothee | $45 | $112.5 | 240 | $27,000 |
| Akshay | $45 | $112.5 | 240 | $27,000 |
| | | | Total Labor Cost: | $81,000 |

*Assuming a 12 hr. work week for 12 weeks

***Parts:***

** = Financial support expected from Raytheon SAS via the university*

*** = Support expected in kind directly from Raytheon SAS*

*Door Locking Mechanism*

| Part/Service | Quantity | Cost Estimate | Potential Supplier | Total |
| --- | --- | --- | --- | --- |
| 9v DC electric motor* | 3 | $7 | Amazon | $21 |
| Arduino Microcontroller* | 1 | $30 | ECE Shop | $30 |
| Bluetooth or wifi shield for Arduino* | 1 | $70 | Mouser | $70 |
| Dead bolt * | 1 | $15 | Menards | $15 |
| Various wiring, soldering supplies, etc. * | 1 | $50 | | $50 |
| | | | **TOTAL:** | $186 |

*Computer Platform for Running Algorithm*

| Part/Service | Quantity | Cost Estimate | Potential Supplier | Total |
| --- | --- | --- | --- | --- |
| Mother Board* | 1 | $110 | Newegg.com | $110 |
| 8G Ram* | 1 | $50 | Newegg.com | $50 |
| i5 CPU* | 1 | $250 | Newegg.com | $250 |
| Nvidia GPU* | 1 | $1100 | Newegg.com | $1000 |
| Case* | 1 | $100 | Newegg.com | $100 |
| Hard Drive* | 1 | $200 | Newegg.com | $200 |
| | | | **TOTAL:** | $1710 |

*Additional Data Collection* (The current plan is for the team to travel to California where Raytheon has sensors in place that we can use cheaply rather than deal with shipping costs and risks etc. We will attempt to estimate the value of contributions here.)

| Part/Service | Quantity | Cost Estimate | Potential Supplier | Total |
| --- | --- | --- | --- | --- |
| Lab Usage** | 1 | $2000 | Raytheon SAS | $2000 |
| Technician labor time** | 1 | $1000 | Raytheon SAS | $1000 |
| Plane travel to and from site * | 3 | $500 | Raytheon SAS | $1500 |
| Hotel* | 3 | $150 | Raytheon SAS | $450 |
| Food during travel* | 1 | $600 | Raytheon SAS | $600 |
| | | | **TOTAL:** | $5550 |

*TOTAL PARTS COST = $5550+$186+$1710 = $7446*

**TOTAL COST = $81000 + $7446 = $88446**

Raytheon will directly fund the computer hardware, door locking mechanism hardware and costs of trip for data collection. This amounts to a total direct funding of $4646. Raytheon will indirectly fund the costs of lab and technician during the data collection trip.

In case of unforeseen circumstances, our total necessary parts cost is only for the door locking mechanism. The processing requirements can be met by utilizing personal and university computing resources. We can use data from the Hong Kong Polytechnic University and Carnegie Mellon University to compensate for collection of data, and we will still be able to have a functional product.

**<u>Schedule</u>**

| **Week** | **Chris Baker** | **Tim Bouhour** | **Akshay Malik** |
|---|---|---|---|
| 9/16 | Work on proposal. | | |
| 9/23 | Design Review: Recognition algorithm design lead. Aid with lock mechanism, feature ID algorithm, and system level development. Obtain data sets. Purchase parts. | Design Review: Lock mechanism design lead. Aid with recognition algorithm, feature algorithm, and system level development. Purchase parts. | Design Review: Feature ID algorithm design lead. Aid with recognition algorithm, system level design, and lock mechanism development. Purchase parts. |
| 9/30 | Develop standard data format for database storage and algorithm input. Follow up on obtaining data sets. | Assemble computer. Install Linux operating system and programs needed for development in C++, CUDA C, and MATLAB. | Build and test lock mechanism with independent trigger. |
| 10/7 | Research best features for recognition. Begin work on parallel recognition algorithm implementation focusing on feature reference points. | Process data into a uniform format and create database with generic input filler and data sets. Research into feature ID algorithm | Set up wireless communication between the lock and computer. Set the lock successfully through a program. Begin research into feature ID algorithm |
| 10/14 | Work on parallel recognition algorithm development. Research into accounting for data | Work on feature ID algorithm implementation. Research into | Work on feature ID algorithm implementation. Debug any problems with lock |

| | | | |
|---|---|---|---|
| | source variation based on light and other factors. | accounting for data source variation based on light and other factors. | trigger from computer. |
| 10/21 | Individual Project Report. Continue coding of parallel recognition algorithm. Support debugging and finish feature ID algorithm. | Individual Project Report. Finish feature ID algorithm and debug. | Individual Project Report. Finish feature ID algorithm and debug. |
| 10/28 | Finish parallel recognition algorithm implementation. | Finish parallel recognition algorithm implementation. Focus on integration of database. | Finish parallel recognition algorithm implementation. Focus on integration of feature ID algorithm. |
| 11/4 | Write serial version on parallel recognition algorithm and run speed comparison test against parallel algorithm. | | |
| 11/11 | Work on presentation/ compare performance further between serial and parallel/ make optimization tweaks. | | |
| 11/18 | Tentative on site data collection in Raytheon SAS labs/ format data to be accepted by feature ID algorithm /testing and debugging of algorithms | | |
| 11/25 | Further systems integration and test including lock hardware; debugging. Prepare for demo and presentation | | |
| 12/2 | Final Systems tweaks/ demo/ presentation/ final report. | | |
| 12/9 | Schedule Buffer/ finish report. | | |