REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*: 22-02-2011

2. REPORT TYPE: Reprint

3. DATES COVERED *(From - To)*: Jan 2005 - Sept 2010

4. TITLE AND SUBTITLE

Feature extraction and object recognition in multi-modal forward looking imagery

5a. CONTRACT NUMBER: W911NF-05-1-0069

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Greenwood, G.
Blakely, S.
Schartman, D.
Calhoun, B.
Keller, James, M.
Ton, Tuan
Wong, David

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

University of Missouri
Office of Research
205 Jesse Hall
Columbia, MO 65211

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Department of the Army
US Army Research, Development and Engineering Command Acquisition Center
Research Triangle Park Contracting Division
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSOR/MONITOR'S ACRONYM(S): ARO Terrestrial Sciences

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; federal purpose rights.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The U. S. Army Night Vision and Electronic Sensors Directorate (NVESD) recently tested an explosive-hazards detection vehicle that combines a pulsed FLGPR with a visible spectrum color camera. Additionally, NVESD tested a human-in-the-loop multi-camera system with the same goal in mind. It contains wide field-of-view color and infrared cameras as well as zoomable narrow field-of-view versions of those modalities. Even though they are separate vehicles, having information from both systems offers great potential for information fusion. Based on previous work at the University of Missouri, we are not only able to register the UTM based positions of the FLGPR to the color image sequences on the first system, but we can register these locations to corresponding image frames of all sensors on the human-in-the-loop platform. This paper presents our approach to first generate libraries of multi-sensor information across these platforms. Subsequently, research is performed in feature extraction and recognition algorithms based on the multi-sensor signatures. Our goal is to tailor specific algorithms to recognize and eliminate different categories of clutter and to be able to identify particular explosive hazards. We demonstrate our library creation, feature

15. SUBJECT TERMS

Forward Looking Imagery, Image Processing

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
UU	UU	UU	UU	9	Jim Keller

19b. TELEPHONE NUMBER *(Include area code)*: (573) 882-7339

Reset

Feature Extraction and Object Recognition in Multi-Modal Forward Looking Imagery

G. Greenwood[a], S. Blakely[a], D. Schartman[b], B. Calhoun[b], J. M. Keller[b], T. Ton[c], D. Wong[c], M. Soumekh[d]

[a]Electrical and Computer Engineering Department
University of Missouri
Columbia, MO 65211

[b]Department of Electrical and Computer Engineering
Portland State University
Portland, OR 97201

[c]U. S. Army RDECOM CERDEC
Night Vision and Electronic Sensors Directorate
Fort Belvoir, VA 22060

[d]Department of Electrical Engineering
State University of New York at Buffalo
Amherst, NY 14260

The U. S. Army Night Vision and Electronic Sensors Directorate (NVESD) recently tested an explosive-hazards detection vehicle that combines a pulsed FLGPR with a visible-spectrum color camera. Additionally, NVESD tested a human-in-the-loop multi-camera system with the same goal in mind. It contains wide field-of-view color and infrared cameras as well as zoomable narrow field-of-view versions of those modalities. Even though they are separate vehicles, having information from both systems offers great potential for information fusion. Based on previous work at the University of Missouri, we are not only able to register the UTM-based positions of the FLGPR to the color image sequences on the first system, but we can register these locations to corresponding image frames of all sensors on the human-in-the-loop platform.

This paper presents our approach to first generate libraries of multi-sensor information across these platforms. Subsequently, research is performed in feature extraction and recognition algorithms based on the multi-sensor signatures. Our goal is to tailor specific algorithms to recognize and eliminate different categories of clutter and to be able to identify particular explosive hazards. We demonstrate our library creation, feature extraction and object recognition results on a large data collection at a US Army test site.

I. Introduction

Forward-looking ground penetrating radar (FLGPR) is a primary sensor used for landmine detection. The FLGPR can detect both above and below ground targets, but unfortunately it is can produce a large number of false detections. With proper registration FLGPR target hits can be mapped into a video image space where image-processing techniques can potentially filter out false positive (FP) target detections. These FPs can arise from various sources including bushes, cacti and even bare ground. Previous work used one monolithic model for all classes of FPs, but the results were less than desirable [Stone08]. In this paper we describe an effort to see if FP-specific models will improve FP filtering. Both model construction methods and results are presented.

II. Model Description

A. Model Design

This project is actually part of a larger research project in which computer algorithms attempt to locate instances of specific objects within a large data set of color images; or, given any point on a color image, to return the probability that a specific object is local within some radius. The process involves characterizing images of certain types of objects. More specifically, for multiple sets of color images (frames) in which a consistent time interval separates every consecutive image in a set, the objective of this project is to develop a means for collecting, storing, and accessing images of specific objects (sub-image) extracted from the frames (super-images); to collect, calculate, and store information describing each sub-image; and to associate sets of temporally linked sub-images, which move through super-image space with respect to time (with respect to frame index).

The process involves characterizing images of certain types of objects. To aid in analysis, the sub-images and associated information are pre-extracted and sorted in a database. This allows specific sets of data to be analyzed at once while excluding other sets of data. It also reduces the computer processing time necessary to locate and analyze the data.

The database stores structured information pertaining to multiple sets of temporally linked sub-images. It is a collection of two object-oriented classes: *Sequence* and *Datanode*. Each instance of the Sequence class contains data regarding exactly one specific object over some range of frames. It holds information about the object's type and the data set in which it can be found, as well as an array of Datanodes. Each instance of the Datanode class contains data regarding exactly one frame of the specific object. It holds information about the file in which the sub-image can be found and its coordinates on the super-image.

The database does not directly store any image data. Image data is stored in a different directory within an umbrella directory. This approach allows loading the database without the overhead of loading hundreds of megabytes of images. This also greatly improves the efficiency of analyzing a partial data set and for developing new image features with which to characterize specific object types.

We developed the MATLAB application *Sequence Extraction Graphic User Interface* (SEG), which is shown in Figure 1, to conveniently collect data to populate the database. This application can display the sequential set of super-images from which to extract the data. The user can label the object as a certain type and select the positions and number of instances of extracted sub images. The SEG application organizes the multiple sub-images of a single object and stores them in a Sequence, which is then added to the database.

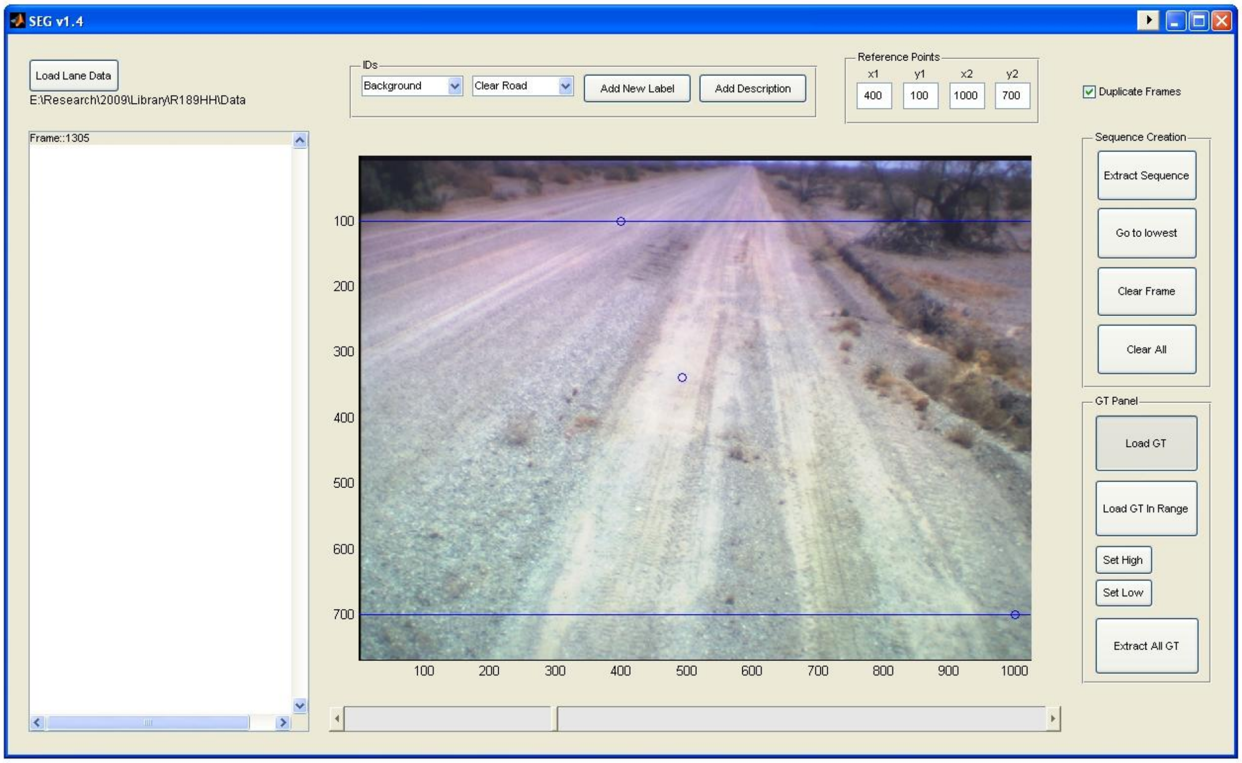

Figure 1: Example rendering of the SEG MATLAB application

The SEG was developed to help create a database of image sequences. In its current version, SEG requires only two files to run. SEG needs the GPS locations of the cart for a particular image and a lane info file. The lane info file contains identification information that is stored in the database along with any information that is extracted from the images. Once the necessary files are loaded, data for a particular object can be extracted based on mouse clicks from the user or a ground truth file with northing and easting coordinates.

By default, the ground truth file only shows object locations, but this file can also be used to extract a single object or the entire lane of objects. Extracting information based on the ground truth is an automated process and allows the user to quickly enter hundreds of sequences into the database with minimal trouble. SEG's ability to label sequences and add descriptions before the sequence is added in the database makes it easy to sort through the database to find what you are looking for.

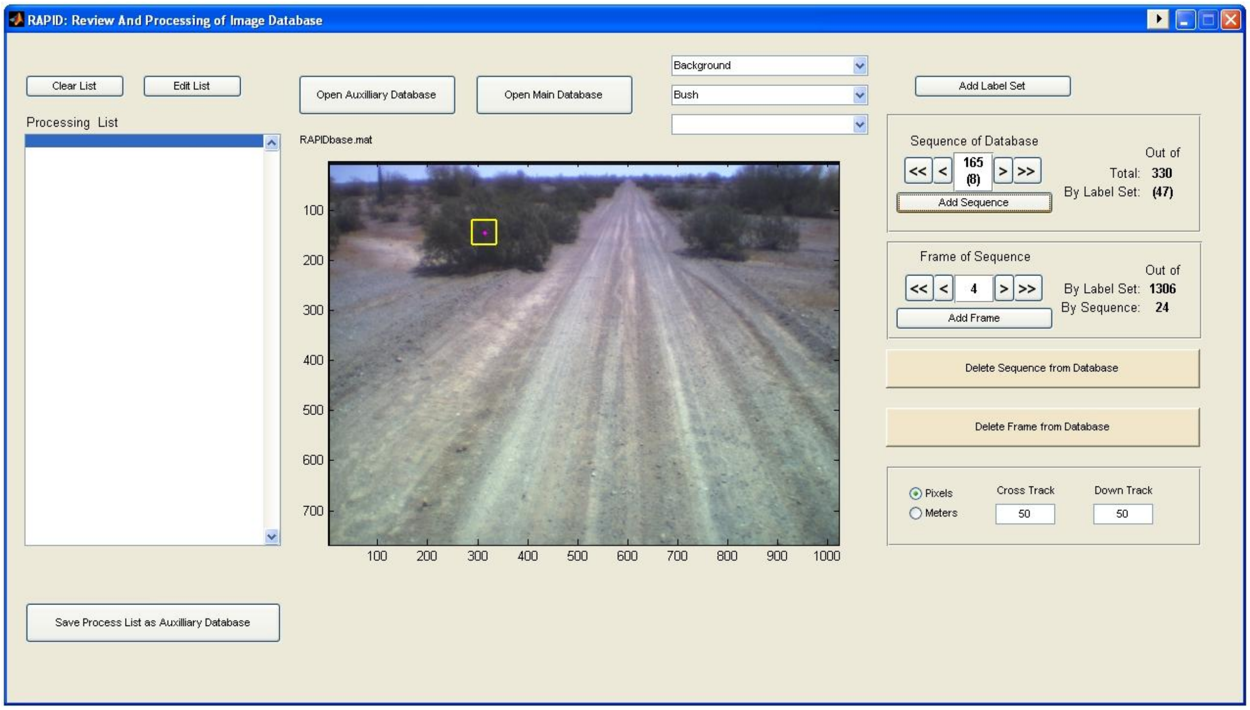

Figure 2: Example rendering of the RAPID MATLAB application.

To conveniently review the content of the database, we developed the MATLAB application *Review and Processing of Image Database* (RAPID). This application can display any super-image or sub-image of a target. The user can browse the database by data set and/or object type. Sequences or individual Datanodes can be permanently removed from the database. The user can also make a list of interesting data and save it as a separate, auxiliary database (e.g. all sub-images of green bushes). The end result of the RAPID application is a refined database set with a common format, which can be efficiently analyzed using additional MATLAB tools.

Our objective here is use the above MATLAB applications to construct an FP model for various specific classes of FP hits rather than one monolithic model for all FP hits. These FP class-specific models are called *eigenmodels*. After scanning several video image files, it was determined that FPs associated with bushes and clear ground were common FP classes. Therefore, eigenmodels for bush FPs and clear ground FPs were constructed for this investigation.

Each hit instance appears in a sequence of typically 20 to 30 consecutive video frames. SEG constructs a set of statistical feature vectors for each video sequence corresponding to a hit instance. Each vector contains statistical information relating to a 100 x 100 set of pixels centered on each hit (approximately 2m down range and 1m cross-range). See Figure 3. Seven statistics are computed for each hit instance: (1) image intensity, (2) Laplacian of intensity, (3) Sobel edge feature of intensity, (4) Local standard deviation of intensity, (5) red channel, (6) green channel, and (7) blue channel. The following attributes are computed for each statistic: (1) max, (2) min, (3) mean, (4) median, (5) standard deviation, (6) skewness and (7) kurtosis. Thus, each vector associated with a hit instance has 49 components.

Figure 3: One frame of a typical video image sequence. The faint white circles indicate potential target hits in this video frame.

A *principal component analysis* (PCA) was conducted to reduce the dimensionality of the model. The first decision was whether to use a PCA based on data covariance or one based on data correlation since either one could be used. Scaling effects principal components. This means if one variable has a greater variance than the others, then this variable will tend to dominate the first principal component of the covariance matrix. However, if the variables are scaled to unit variance then this problem is mitigated. Using a correlation matrix is therefore preferred especially if all variables are considered equally important [Chat80].

A covariance PCA was conducted on a bush-class FP eigenmodel and Table 1 shows the variance contribution of the eight largest eigenvalues (λ). Clearly the first eigenvalue contributes the most to the overall data variance. Looking at the first variable (image intensity) in the statistics vector it was apparent it did indeed have the largest variance. Nevertheless, we had no reason to believe this variable was any less important than any other variable. We therefore decided to use a covariance PCA and picked the p = 6 largest eigenvalues to construct the eigenmodels.

Eigenvalues	Variation (%)
λ_1	94.75
λ_2	3.17
λ_3	1.15
λ_4	0.42
λ_5	0.32
λ_6	0.16
λ_7	0.04
λ_8	0.02

Table 1: Variance contribution per eigenvalue of the covariance matrix. Eigenvalues decrease in magnitude as the index number increases.

The database had 1631 total hit instances although not all of them corresponded to actual targets. Only 48 actual targets were present and some of the other hits were for fiducials. Nevertheless, the vast majority of hits were FPs. We physically scanned the video files and chose hits with no discernable targets present. (Ground truth coordinates were available for all targets.) We randomly selected M = 10 FP hits that had bushes and another 9 hits that had clear ground. As mentioned above, SEG constructs a set of statistical feature vectors for the video frames associated with a hit instance. A median vector for each set was extracted and the set of M such vectors$\{x_1, x_2, \ldots, x_M\}$ are used to construct an FP class-specific eigenmodel. The following steps created the bush eigenmodel:

STEP 1: Compute the mean vector

$$\bar{x}_B = \frac{1}{M}\sum_{j=1}^{M} x_j$$

This mean vector must be saved since it will be needed during classification.

STEP 2: Subtract the mean

$$\phi_j = x_j - \bar{x}_B \qquad j = 1, 2, \ldots, M$$

STEP 3: Form the matrix

$$A = \begin{bmatrix} \phi_1 & \phi_2 & \cdots & \phi_M \end{bmatrix}$$

and then compute the *covariance matrix* $C = AA^{T}$.

STEP 4: Perform a principal component analysis on C and keep the $p < M$ largest eigenvalues (λ) where $\lambda_1 > \lambda_2 > \cdots > \lambda_p$ along with their corresponding eigenvectors $e_1, e_2, \ldots, e_p$.

STEP 5: Normalize the eigenvectors so that $\| e_j \| = 1$.

STEP 6: Create the matrix

$$V_B = \begin{bmatrix} e_1 & e_2 & \cdots & e_p \end{bmatrix}$$

V_B is a matrix with normalized eigenvectors as columns ordered from left-to-right by decreasing magnitude of their corresponding eigenvalues. This matrix must be saved as it used during classification. The above process creates an bush FP eigenmodel

$$\Omega_{\mathrm{BUSH}} = \left\{ V_B, \bar{x}_B \right\}$$

This algorithm can be repeated as necessary to create other class-specific FP models.

B. *Model Testing*

Testing was conducted with eigenmodels for two FP classes: a "bush" eigenmodel (Ω_{BUSH}) and a "clear ground" eigenmodel (Ω_{GND}). SEG will construct a set of statistical feature vectors for the 1631 hit instances, but the fiducials and the hit instances used to construct the eigenmodels were excluded. The median vector from each of these sets was computed. Let Ω be this set of median vectors and let θ be a user-selected threshold.

STEP 1: Randomly choose a median vector $y \in \Omega$.

STEP 2: Compute

$$\omega = y - \bar{x}_B$$

STEP 3: Find the mapping of ω into the bush eigenmodel space

$$z = V_B^T \omega$$

STEP 4: Compute the Mahalanobis distance D between z and the origin of the bush eigenmodel space origin. If $D < \theta$, then classify y as a FP. Otherwise classify y as a target.

STEP 5: If all $y \in \Omega$ not checked, go to **STEP 1**. Otherwise, exit.

Repeat the above steps to compute the Mahalanobis distance for the clear ground eigenmodel Ω_{GND}. If the Mahalanobis distance $D < \theta$, then classify y as a FP. Otherwise classify y as a target.

III. Results

Table 2 shows the percent correct classifications of the 1631 total hit instances in the database. (There are 1420 FPs after excluding the fiducials, the targets and the hit instances used for eigenmodel construction).

	Bush Eigenmodel		Clear Ground Eigenmodel		Combined Eigenmodel	
	$\theta = 6$	$\theta = 8$	$\theta = 6$	$\theta = 8$	$\theta = 6$	$\theta = 8$
Targets	77.1%	47.9%	79.2%	77.1%		
FPs	21.9%	35.9%	34.8%	44.1%		

Table 2: Detection accuracy for targets and FPs at two different threshold levels.

The thresholds value (θ) was chosen to be above the average Mahalanobis distance (from the origin) of the hit instances used to construct the eigenmodels. Targets are correctly detected if their Mahalanobis distance from the eigenmodel origin is larger the threshold whereas FPs are correctly detected if their distance is less than the threshold. This explains why the target (FP) detection accuracy decreases (increases) as the threshold increases. It is worth noting the FP accuracy reflects the ability to detect *any* FP and not just class-specific FPs.

The moderately low FP detection accuracy is not surprising since we constructed eigenmodels for only two FP classes. No analysis was done to see what percent of the 1410 FPs were bush class or clear ground class. Additional FP classes do exist. For example, we noticed some hits were near the road berm where the ground is cluttered. Other FPs were mixed, containing a variety of diverse objects. However, there were an insufficient number of these FPs to reliably construct an eigenmodel. One reason why all of the bush FPs might not be correctly detected is we noticed sometimes a large bush would not necessarily generate a hit throughout the sequence of video frames while in other cases it would.

The majority of the targets were detected. One reason why all of them could not be detected is some targets were apparently buried in the road. (Ground truth coordinates were available for all of the targets so their locations were precisely known.) The road surface was smoothly graded and therefore lacked any visible features; some target hits appeared no different than some clear ground hits.

IV. Future Work

The FP detection accuracy shown in Table 2 is quite reasonable since it reflects the ability of one particular FP class-specific eigenmodel to detect *any* FP class. We did not investigate how well a conjunction of FP class-specific models would behave, but we conjecture the detection accuracy would be relatively high. Another area of investigation is whether a judicious choice of hit instances for specific type of FPs would lead to higher quality eigenmodels.

BIBLIOGRAHY

[Stone08] K. Stone, J. Keller, K.C. Ho, M. Busch and P.D. Gader, "On the registration of FLGPR and IR data for a forward-looking landmine detection system an its use in eliminating FLGPR false alarms", *SPIE Proceedings*, 2008

[Chat80] C. Chatfield and A. Collin, *Introduction to Multivariate Analysis*, Chapman and Hall in Association with Methuen, Inc., NY 1980