

REPORT DOCUMENTATION PAGE			Form Approved OMB NO. 0704-0188		
<p>The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA, 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p> <p>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</p>					
1. REPORT DATE (DD-MM-YYYY) 14-09-2010		2. REPORT TYPE Final Report		3. DATES COVERED (From - To) 15-May-2004 - 14-Jan-2008	
4. TITLE AND SUBTITLE Human Signatures for Personnel Detection			5a. CONTRACT NUMBER W911NF-04-1-0190		
			5b. GRANT NUMBER		
			5c. PROGRAM ELEMENT NUMBER 611103		
6. AUTHORS J. Michael Cathcart, William T. Rhodes			5d. PROJECT NUMBER		
			5e. TASK NUMBER		
			5f. WORK UNIT NUMBER		
7. PERFORMING ORGANIZATION NAMES AND ADDRESSES Georgia Tech Research Corporation Office of Sponsored Programs Georgia Tech Research Corporation Atlanta, GA 30332 -0420			8. PERFORMING ORGANIZATION REPORT NUMBER		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Army Research Office P.O. Box 12211 Research Triangle Park, NC 27709-2211			10. SPONSOR/MONITOR'S ACRONYM(S) ARO		
			11. SPONSOR/MONITOR'S REPORT NUMBER(S) 46631-EL-MUR.1		
12. DISTRIBUTION AVAILABILITY STATEMENT Approved for Public Release; Distribution Unlimited					
13. SUPPLEMENTARY NOTES The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.					
14. ABSTRACT This MURI program continued into its final year; this report covers the eleven months from 1 August 2009 to 14 June 2010. The overall goal of this program is a comprehensive understanding of the phenomenology underlying the signatures generated by humans, the detection of those signatures using multiple sensor modalities, and the processing of those signatures to detect personnel. In particular, the objectives are to understand the physics of these signatures in traditionally difficult detection environments (buildings, caves, tunnels, camouflaged settings),					
15. SUBJECT TERMS human signatures, human signature physiology, signature physics, detection algorithms, multimodal signatures, sensor fusion, optical sensing, infrared sensing, radar detection, acoustical sensing, seismic sensing					
16. SECURITY CLASSIFICATION OF:		17. LIMITATION OF ABSTRACT		15. NUMBER OF PAGES	
a. REPORT UU	b. ABSTRACT UU	c. THIS PAGE UU	UU	19a. NAME OF RESPONSIBLE PERSON William Rhodes	
				19b. TELEPHONE NUMBER 404-894-2929	

Report Title

Human Signatures for Personnel Detection

ABSTRACT

This MURI program continued into its final year; this report covers the eleven months from 1 August 2009 to 14 June 2010. The overall goal of this program is a comprehensive understanding of the phenomenology underlying the signatures generated by humans, the detection of those signatures using multiple sensor modalities, and the processing of those signatures to detect personnel. In particular, the objectives are to understand the physics of these signatures in traditionally difficult detection environments (buildings, caves, tunnels, camouflaged settings), identify sensor network constructs to record these signatures, and develop processing techniques (e.g., distributed processing) to exploit these signatures for personnel detection.

During this reporting period, the technical efforts focused on three sensor domains: seismic/acoustics, electro-optics, and radar. Seismic/acoustics technical efforts investigated use of passive and active sensing modes. Electro-optics research focused on continued development of a multi-modal human signature & urban model, collecting motion data, development of motion-based detection algorithms, and developing techniques for detection in low pixel video sequences. Additional work focused on exploiting motion capture data for human motion characterization and discrimination. Radar-based research focused on investigating improving autoregressive modeling approaches for gait detection.

List of papers submitted or published that acknowledge ARO support during this reporting period. List the papers, including journal references, in the following categories:

(a) Papers published in peer-reviewed journals (N/A for none)

Number of Papers published in peer-reviewed journals: 0.00

(b) Papers published in non-peer-reviewed journals or in conference proceedings (N/A for none)

Number of Papers published in non peer-reviewed journals: 0.00

(c) Presentations

K. Prussing, J. Michael Cathcart, Sarah Lane, Shayla Otolorin, and Brian R. Kocher, "Urban Scene Simulation for Multimodal Sensing Applications," SPIE Defense and Security Symposium April 2010

Number of Presentations: 1.00

Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

J. Michael Cathcart, Brian R. Kocher, Keith Prussing, Sarah Lane, and Alan M. Thomas, "Multimodal Signature Modeling of Humans," Proc. SPIE, vol. 7687 (2010).

Alan M. Thomas, J. Michael Cathcart, and Brian R. Kocher, "Analysis of Human Motion in Video Imagery," Proc. SPIE, vol. 7697 (2010)

Number of Non Peer-Reviewed Conference Proceeding publications (other than abstracts): 2

Peer-Reviewed Conference Proceeding publications (other than abstracts):

Number of Peer-Reviewed Conference Proceeding publications (other than abstracts): 0

(d) Manuscripts

Alexander E. Ekimov and James M. Sabatier, "Rhythmic Analysis of Human Motion," Submitted Fall 2009, J. Acoust. Soc. Am.

Number of Manuscripts: 1.00

Patents Submitted

Patents Awarded

Graduate Students

<u>NAME</u>	<u>PERCENT SUPPORTED</u>
Keith Prussing	0.33
FTE Equivalent:	0.33
Total Number:	1

Names of Post Doctorates

<u>NAME</u>	<u>PERCENT SUPPORTED</u>
FTE Equivalent:	
Total Number:	

Names of Faculty Supported

<u>NAME</u>	<u>PERCENT SUPPORTED</u>	National Academy Member
Michael Cathcart	0.20	No
William Rhodes	0.10	No
FTE Equivalent:	0.30	
Total Number:	2	

Names of Under Graduate students supported

<u>NAME</u>	<u>PERCENT SUPPORTED</u>
Spencer Nichols	0.25
FTE Equivalent:	0.25
Total Number:	1

Student Metrics

This section only applies to graduating undergraduates supported by this agreement in this reporting period

- The number of undergraduates funded by this agreement who graduated during this period: 0.00
- The number of undergraduates funded by this agreement who graduated during this period with a degree in science, mathematics, engineering, or technology fields:..... 0.00
- The number of undergraduates funded by your agreement who graduated during this period and will continue to pursue a graduate or Ph.D. degree in science, mathematics, engineering, or technology fields:..... 0.00
- Number of graduating undergraduates who achieved a 3.5 GPA to 4.0 (4.0 max scale):..... 0.00
- Number of graduating undergraduates funded by a DoD funded Center of Excellence grant for Education, Research and Engineering:..... 0.00
- The number of undergraduates funded by your agreement who graduated during this period and intend to work for the Department of Defense 0.00
- The number of undergraduates funded by your agreement who graduated during this period and will receive scholarships or fellowships for further studies in science, mathematics, engineering or technology fields: 0.00

Names of Personnel receiving masters degrees

<u>NAME</u> Brian Kocher Ed Burdette Total Number:	 2
--	----------------------

Names of personnel receiving PHDs

<u>NAME</u> Total Number:	
---	--

Names of other research staff

<u>NAME</u>	<u>PERCENT SUPPORTED</u>	
Ryan Hersey	0.15	No
Bill Melvin	0.01	No
Alan Thomas	0.10	No
Sarah Lane	0.30	No
Brian Kocher	0.60	No
Ed Burdette	0.30	No
James Sabatier	0.04	No
Alexander Ekimov	0.34	No
Mark Weiderhold	0.28	No
FTE Equivalent:	2.12	
Total Number:	9	

Sub Contractors (DD882)

Inventions (DD882)

MURI: Human Signatures for Personnel Detection

Scientific Progress and Accomplishments

Georgia Institute of Technology

Conducted analysis of motion capture data to develop methods for characterizing human motion under different loads and gaits

Adapted lattice Boltzmann method to thermal signature computation

Continued analysis of phenomenology and signatures associated with personnel and background materials; emphasis was on urban environments

Continued development of a multi-modal signature model; with a focus on:

developing a common signature model across the multiple sensor domains;

developing an urban scene simulation to support signature analysis;

addressing complex radiative transfer issues

developing thermal and reflective band human signature models

adding dynamic clutter elements (e.g., vehicles) to modeled scene

Investigated human motion detection and tracking algorithms

Continued development of a kinematic human motion model

Continued investigations into advanced algorithms for RF dismount detection and characterization

Continued analysis of advanced post detection integration (PDI) techniques.

Continued investigation of autoregressive (AR) modeling for dismount detection.

Florida Atlantic University

Continued research on methods for detecting personnel in low pixel count images and video sequences; motion-based approaches investigated

Continued development of MatLab-based segmentation program for isolating candidate groups of pixels for human-likelihood testing.

Continued development of Bayesian-logic-based detection program using video-sequence features including (a) location in 3-D space as inferred from location in 2-D video projection, (b) velocity (speed and direction), and (c) up-down motion of center-of-mass characteristics

Tested detection and tracking methodology with collected video sequences

Massachusetts Institute of Technology

None

University of Mississippi

Continued work on extension of Draper Lab algorithms for seismic and cadence frequency

Continued adaptation of Boulic and Thalmann human motion model to conduct initial theoretical studies of human motion

Technology Transitions

Georgia Institute of Technology

Proposal to DARPA for Phase II of DYME program on human motion modeling and RF/EO-IR human signatures

Teamed with Computer Sciences Corporation on TASER proposal to NGA

Teamed with ATK on Ground Combat Vehicle Proposal

Briefing to DIA on MURI work

University of Mississippi

None

Massachusetts Institute of Technology

None

Report List

FAU

FAU Exploiting 3-D Models for Low Resolution Analysis – combined poster.ppt
FAU MURI Final report – FAU – 14Sep10.pdf
FAU Pava Diego MSEE dissertation.pdf
FAU Removing Ambiguity in 2-D Image Information by Means of 3-D Models.pdf

Georgia Tech

GT Hersey – MURI Final Presentation.ppt
GT LBM_summary_v5_no3d.ppt
GT MURI Final Report Biomechanical v4.pptx
GT MURI_Final_UrbanSceneModeling_v3.pptx
GT MURI_sensors_final_report_v2.pptx
GT Thomas Cathcart Human Motion Final 2010_v3a.ppt
GT Vehicle_modeling_v3.ppt

MIT

MIT 090827annualmurireport.doc

Univ of Mississippi

UM Final Report Presentation.ppt

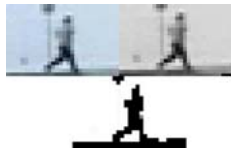
Exploiting 3-D Models for Low-Resolution Motion Analysis

Diego F. Pava, William T. Rhodes PhD.

Department of Computer and Electrical Engineering and Computer Science

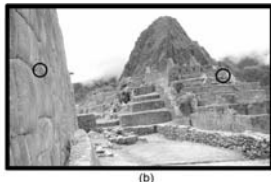
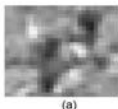
Motivation

- Given a video sequence of extremely low resolution objects moving in a fixed background (e.g., objects very far away from the camera), we want to extract the most information from the scene in order to determine whether objects may be human beings.



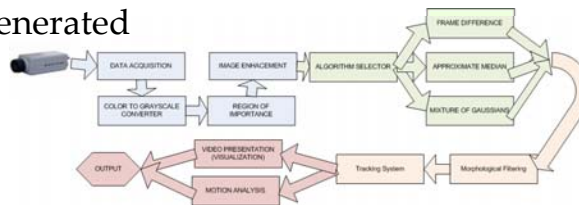
Related work

- To identify an object in the very low resolution regime, we need to extract data from the object itself (motion, size, etc) and to use the knowledge of the scene. The latter can be done by dividing the scene in regions of importance that minimizes decision errors.



Approach

- The system has four blocks. In the preprocessing block, regions of interest are selected and histogram equalization is performed. In the background subtraction block, foreground objects are extracted. In the information extraction block, the foreground video is morphologically filtered, the objects are tracked, and motion analyzed using a similarity matrix. Finally, in the post processing block, the data is stored in a structure and visual information is generated



Experiments

- Each block was tested separately using sample videos and images. Four different videos were taken on the FAU

campus depicting humans walking on natural scenes with occlusions. The system was tested with one and two objects at different speeds and occupying 12 to 30 pixels in the screen. All videos were tested for biological motion using similarities matrixes.

Results and Conclusions

- The system detected, tracked, and classified as objects with biological motion all subjects present in the videos under the background restrictions.

- The system compensated for wind movement on the scene and shades



- The system has limitations in the number of objects that can be present, the complexity of occlusions, and is not a real time application

Context based ROI



Background Subtraction



-Frame difference

$$\text{Foreground} = |\text{Frames}_i - \text{Frames}_{i-1}| > \text{Threshold}$$

-Approximate Median

$$\text{foreground} = |\text{frame}_i - \text{model}| > \text{Threshold}$$

$$\text{If } \text{frame}_i(x, y) > \text{model}(x, y) \Rightarrow \text{model}(x, y) = \text{model}(x, y) + 1$$

$$\text{If } \text{frame}_i(x, y) < \text{model}(x, y) \Rightarrow \text{model}(x, y) = \text{model}(x, y) - 1$$

-Mixture of Gaussians

$$F(i_t = \mu) = \sum_{i=1}^k \omega_{it} \cdot \eta(\mu, \sigma)$$

$$|i_t - \mu_{i,t-1}| \leq D \cdot \sigma$$

$$w = (1 - \alpha)w + \alpha$$

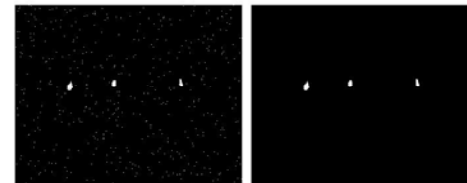
$$w = (1 - \alpha)w$$

$$p = \alpha / w$$

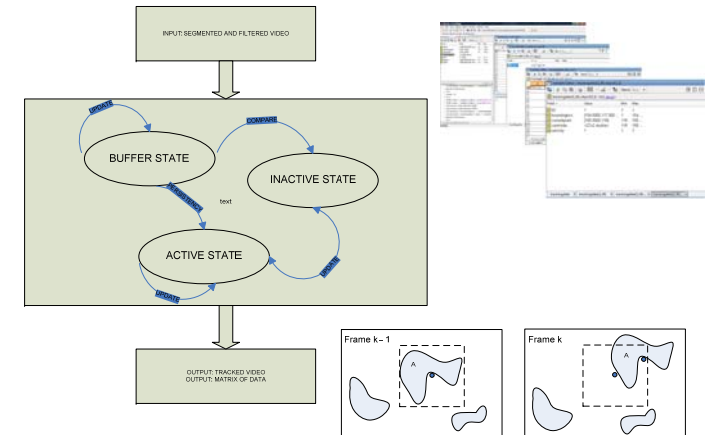
$$\mu = (1 - p)\mu + (p) \text{pixel}$$

$$\sigma = \sqrt{(1 - p)\sigma^2 + p(\text{pixel} - \mu)^2}$$

Morphological Filtering



Tracking System



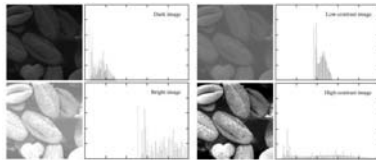
Histogram Equalization

$$s = T(r) = \int_0^r p(w)dw$$

$$p_o(s) = \begin{cases} 1 & \text{For } 0 \leq s \leq 1 \\ 0 & \text{Otherwise} \end{cases}$$

$$s = T(r) = \int_0^r p(w)dw$$

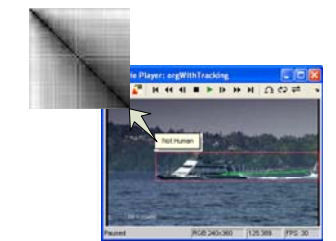
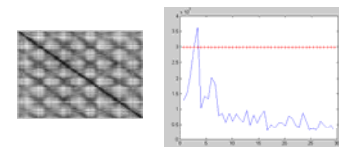
$$s_k = T(r_k) = \sum_{j=1}^k p_r(r_j) = \sum_{j=1}^k \frac{n_j}{n}$$



Motion Analysis

$$S_{t_1, t_2} = \sum_{(x,y) \in R_{t_1, t_2}} |O_{t_1}(x,y) - O_{t_2}(x,y)|$$

$$P > \mu_p + K\sigma_p$$



Extraction of human signature information from very-low-resolution video sequences

Researchers:

William T. Rhodes, Ph.D.

Professor of Computer and Electrical Engineering and Computer Science

Diego F. Pava

Masters and Ph.D. degree student in electrical engineering

Research at Florida Atlantic University, conducted by doctoral student Diego Pava under the direction of Prof. William Rhodes, centered on the development of methods for extraction of human signature information from very low resolution video sequences. Very low resolution is encountered when, for example, people are great distances from the camera and the field of view is large. In such cases, the humans may subtend only 10 to 15 pixels, a condition we might in fact take as defining very low resolution in this context.

The key to the method lies in the identification of motion characteristics of low-resolution objects in their specific context. Objects of interest observed in a particular location in the video sequence must move in ways consistent with physical constraints. For example, an object observed moving through a patch of sky cannot be a person, but is more likely a bird or an airplane—or a butterfly if the motion path is erratic. Motion of a small-pixel-count object is more likely to correspond to human activity if the pixels are located in a portion of the video imagery consistent with a large distance. If moving objects in low-resolution 2D video imagery are placed in their 3D context, object uncertainties can often be removed. It should be noted that very low resolution imagery presents special difficulties because the blurring of the background into the moving part of the scene makes the application of techniques such as centroid tracking unreliable.

The approach we took in addressing this problem ultimately requires that we have available to us a 3D model of the scene being viewed with our video camera. Exploiting our knowledge of the 3D world from which we have extracted the 2D video projections, we can then reduce, often by extremely large amounts, possible uncertainties concerning the nature of what we are viewing.

Figure 1 provides an illustration of the concept, albeit with 2D still images—i.e., no video—and without a true 3D representation of the scene available to us, only our own idea of what the 3D scene actually is. The left-hand part of the figure shows a small number of pixels extracted from somewhere in the larger image shown on the right (see caption).

In a video image, we would observe some motion within this small number of pixels. Does that motion represent human activity, or something else? The question is largely resolved if we know *where* in the scene the pixels in question are observed. If they are observed in the circled region at the left, they probably represent a moving leaf, a lizard, or some other small animal or insect; if in the circled region on the right, they almost certainly represent one or two humans climbing along an ancient pathway. With several frames of video, the probability that the changing pixels represent human(s) can be more accurately determined through the observation of the motion itself: Does it have an up-and-down component? Is the transverse motion consistent with people

struggling along a 2500 meter-high path? Most importantly, is the size of the moving pixel group consistent with people at that apparent distance?

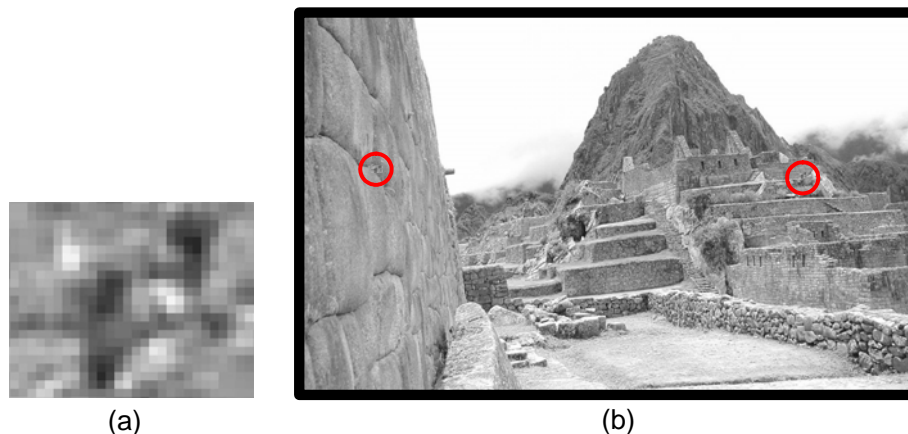


Figure 1. The small group of pixels on the left may or may not correspond to one or more humans. The location of these pixels within a 3D scene—two such locations being indicated on the right—makes the likelihood of their representing humans much easier to determine, even in the absence of motion cues. Motion cues would, in this case, make correct identification almost certainly correct.

The idea that even a subconscious understanding of a 3D setting can help disambiguate information contained in a 2D image is of course not new. What is relatively new is the greatly increased capability we have now to obtain and manage data on the 3D structure of settings of interest to us. Our ability to build a 3D model of buildings, trees, roadways, and the like from a stereo image pair has improved enormously over even the past decade, and today's computational power and huge computer memories make fine-scale 3D databases, along with the attachment of contextual information, comparatively easy. It is thus much easier for us to assess, probabilistically, and by checks against the 3D database, whether a moving object in a 2D image is likely to be a human or, instead, a goat or a butterfly. These key ideas were presented in two conference papers, listed as publications 1 and 2 at the end of this section and available as added separate uploaded documents.

The framework in which such disambiguation operates is of necessity probabilistic, and several methods, including traditional Bayesian, can be used. Of at least equal importance is the impact of very-low-resolution imagery on the image processing algorithms employed, and it was on this subject area that the work at FAU was concentrated. If an object of concern is so distant that it subtends only tens of pixels, then the normal approaches to motion tracking, such as optical flow methods, do not work well. Edges are fuzzy, and the interaction of the (usually) stationary background structure with the moving object structure creates additional problems. Indeed, most object detection studies employ video sequences where objects of interest are imaged at high resolution. The FAU research, by way of contrast, explored the very low resolution regime in order to assess how much information can be obtained in an early alarm system. The operation investigated had four stages—preprocessing, background modeling, information extraction, and post processing—and used context-based region-of-importance selection, histogram equalization, background subtraction, and morphological filtering techniques. The program was implemented in Matlab; output was presented in both data and visual form. The resulting system was capable of detecting and tracking low resolution objects (as low as 15 pixels in size) in a

controlled background scene. The system can serve as the basis for systems with much higher complexity. The work was documented in the master’s degree thesis—Ref. 3 at the end of this section—written by Mr. Pava, the FAU student working on this project.

Pava’s thesis is available as a separate document accompanying this one. His major accomplishments are presented in the following.

Regions of Importance: A program was written that allows the user to select one or more regions of importance (ROI). Motion outside of those areas is ignored by subsequent portions of the program. When objects occupy just a few pixels in a scene, there are usually important portions of the video sequence where the presence of objects of such characteristics is unlikely or unimportant. Security applications may require some regions to be attended while others can be ignored. Furthermore, through the establishment of regions of importance, inevitable noise coming from unimportant regions can be ignored with a resulting improvement in overall performance of the system and computing resources management. Because regions of importance depend on so many factors, user-creation of ROIs is preferred over automatic approaches. The system employed in our study requires that the user draw with the mouse the ROI. A binary mask of the ROI is then created and applied to the video after the image enhancement. Figure 2 illustrates the ROI algorithm.

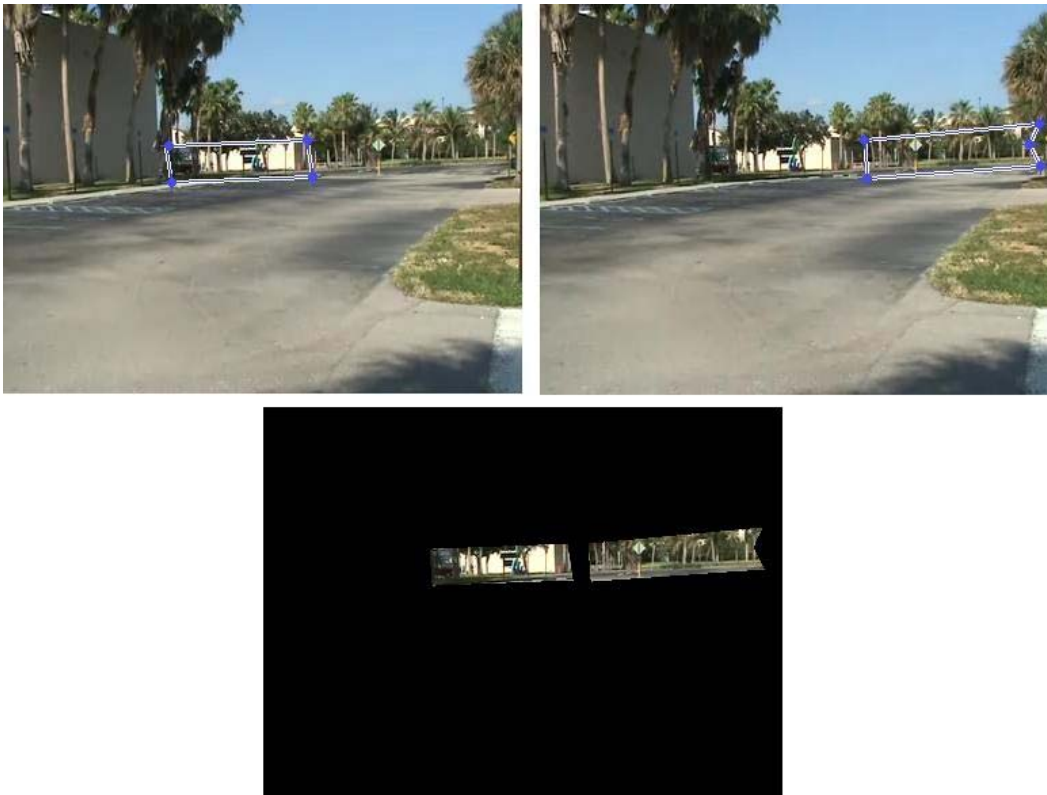


Fig. 2. The video sequence processing program allows manual selection of regions of importance. Motion/change outside of the specified regions is ignored. In the figures above, two different regions of importance were specified.

Histogram Equalization: Contrast enhancement, accomplished by means of histogram equalization, was used to improve the behavior of low-resolution object tracking algorithms. In some cases the objects of interest were binarized. Figure 3 illustrates. The regions of importance are usually small regions where the analysis is in detail and hence where the enhancement of the contrast is most desired: the unimportant regions add pixels of different intensities; to take the histogram equalization over the whole scene could have the contrary effect of contrast enhancement. Take for example the scene depicted in at the bottom of Fig. 3, where histogram equalization over the whole picture will have an undesired effect. Contrast enhancement is desired because it facilitates the differentiation between the object and the background. The shirt of the person in the figure has less contrast than the pants as can be appreciated in the color and grayscale versions of the image. Note how after the contrast enhancement, the object and the background tend to be mostly black and mostly white which makes the object easier to recognize.

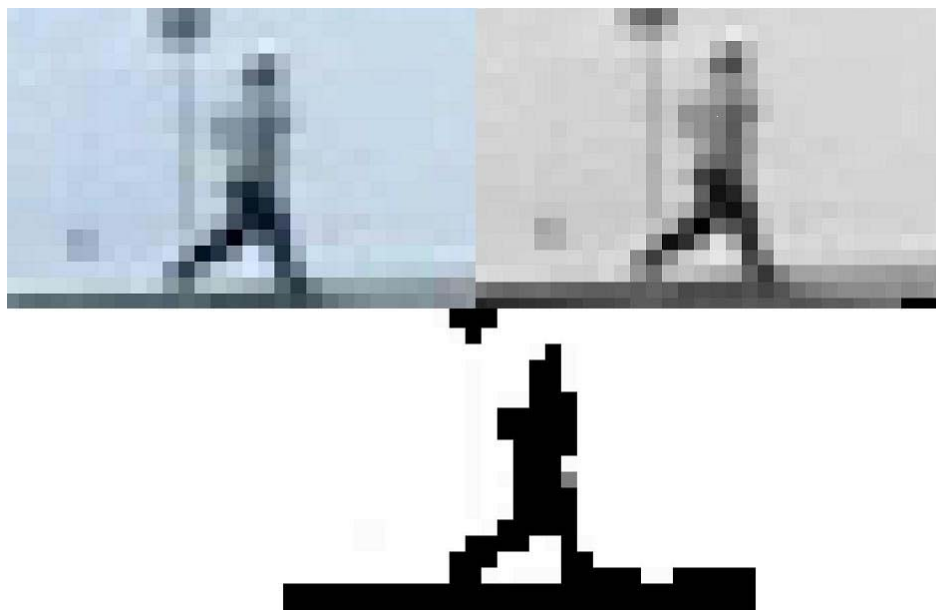


Fig. 3. Example of contrast enhancement by histogram equalization. Color information was first removed from the imagery. The bottom-left scene illustrates a case where global histogram equalization is not desired, since the scene brightness in the region of interest is more or less uniform. The benefits of histogram equalization are shown bottom right in the extraction of an object of interest: left with histogram equalization, right without.

Background Subtraction: Three different methods for background subtraction were investigated at the low-resolution extreme: frame difference analysis, approximate median analysis, and mixture of Gaussians analysis. The approximate median analysis, the results of which are illustrated in Fig. 4, worked best for low-resolution imagery.

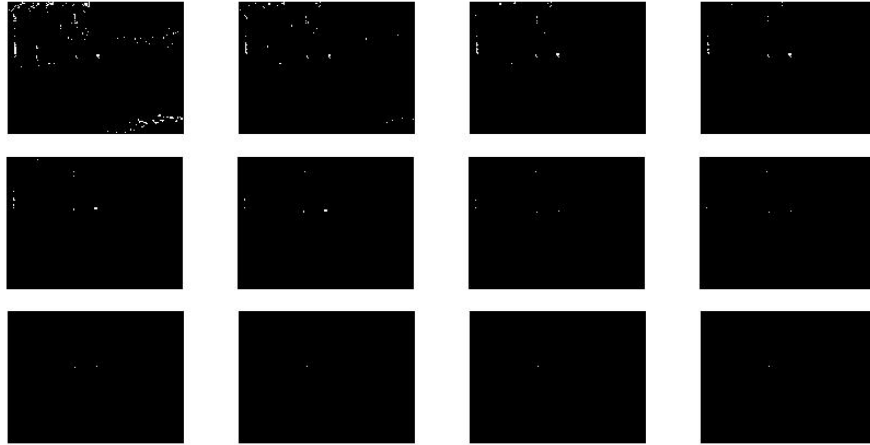


Fig. 4. Thresholding applied to imagery subjected to approximate median algorithm background subtraction operation.

Morphological Filtering: Morphological filtering was also employed to remove salt-and-pepper noise from the contrast-enhanced imagery and to make it easier to establish motion vectors for moving object structures. The results of such filtering is illustrated in Fig. 5.

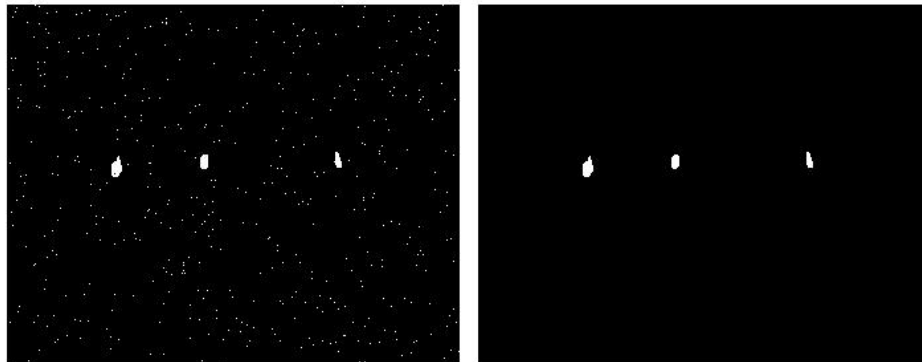


Fig. 5. Test of morphological filtering: Left, image with test objects and Poisson and salt and pepper noise; right, image after morphological filtering.

Tracking System: Objects moving behind occluding objects can present problems for motion analysis algorithms. A tracking system was developed that allowed temporarily occluded moving objects to be identified and successfully tracked. An example of the tracking algorithm operation is illustrated in Fig. 6.

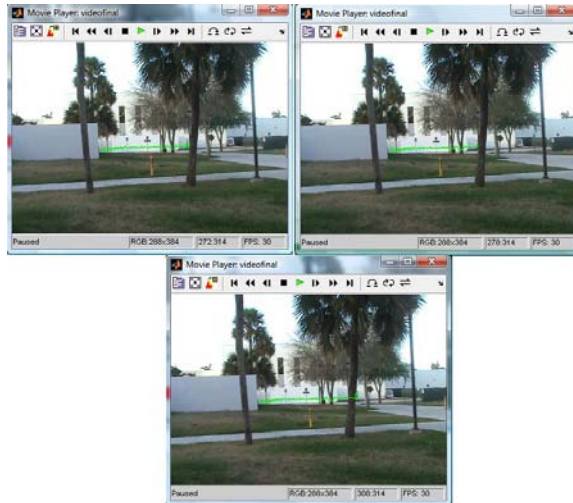


Fig. 6. Example of tracking algorithm operation illustrating handling of occlusions..

The tracking system operated with more than one moving object in the region of importance, as illustrated in Fig. 7.

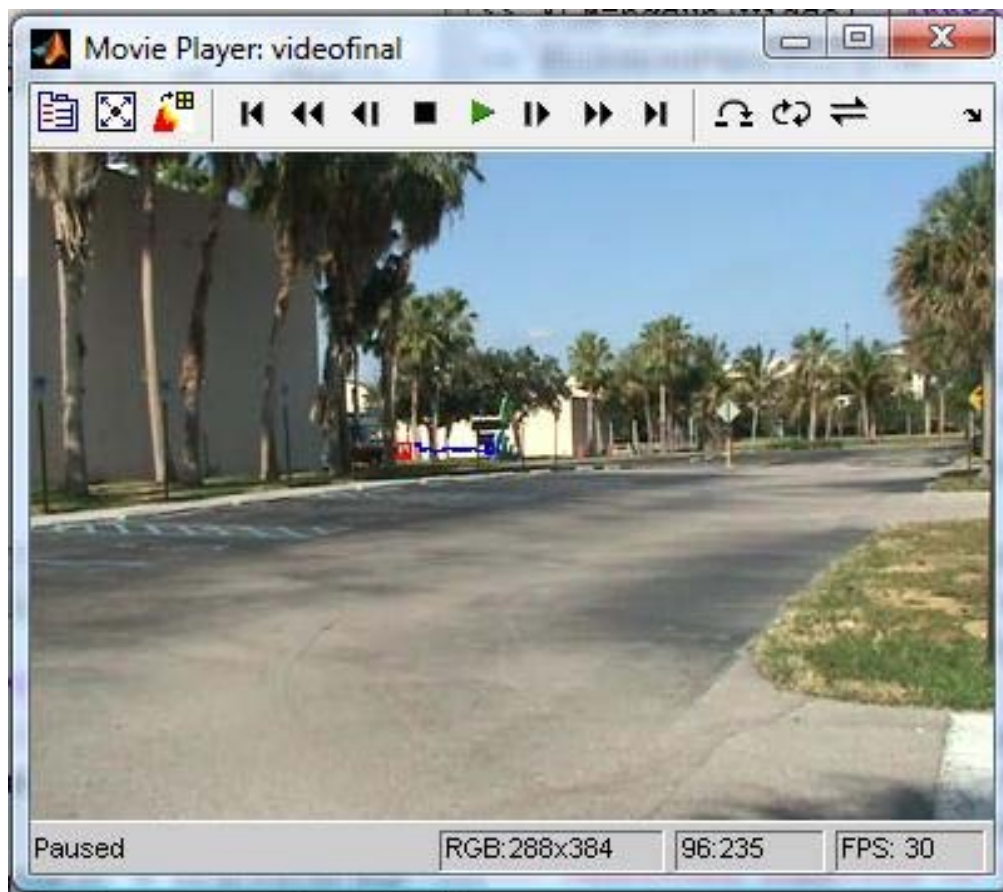


Fig. 7. Illustration of tracking operation. Note that two people entered the scene. One person is tracked in blue, the other (who returned to the left) is tracked in red. See video for entire sequence.

Time Analysis: The basic processing algorithms were subject to time analyses, which gave us an idea of how much time was required for different aspects of the processing. Comparative figures are shown in Fig. 8.

Block	Function	LAPTOP			WORKSTATION		
		Video 1 (311 Fr)	Video 2 (314 Fr)	Video 3 (235 Fr)	Video 1 (311 Fr)	Video 2 (314 Fr)	Video 3 (235 Fr)
Preprocessing	preproc.m						
	With His. Eq.	23.76	22.98	20.94	25.72	24.62	22.54
	Without	17.35	17.24	14.68	18.79	17.65	13.97
Background Mod.	bgsub.m						
	FrameDiff	8.86	8.00	6.78	8.02	7.54	5.67
	ApproxMed	242.86	255.02	177.45	11.19	10.98	8.25
	MoG	2461.08	2422.79	1807.44	1637.33	1681.16	1296.81
	morfil.m	51.16	50.50	35.35	79.13	78.62	56.71
Info Extraction and Postprocessing	tracksys.m	8.03	11.14	11.18	14.58	23.61	16.43

Fig. 8. Time analysis of the different processing operations.

In his thesis research Mr. Pava developed the following capabilities, all documented in his dissertation (Ref. 3):

- Detecting foreground objects (as small as 8 pixel of height and 15 pixels in total).
- Tracking objects and accumulate data along their trajectories.
- Handling occlusions.
- Implementing three different background subtraction algorithms.
- Choosing several regions of importance in a video sequence
- Handling noise due to weather conditions, video conditions, or random noise.

His system was subject to the following restrictions:

- A single, static camera setting.
- Implementation time and memory capacity limits affect the video size and the amount of information that can be extracted.
- Limited number of objects present on the video.
- The system needs contrast between in order to work.
- The system needs for the object to be moving.
- The solution for the occlusion problem is dependant in the condition of the scene.
- A real time solution is not feasible with the current implementation.

In addition, he reached the following conclusions:

- The introduction of region of interest selection to the overall system improves the response of the system to noise.
- The implementation of histogram equalization improves the contrast between the object and the background but also introduces more noise in the system.

- Of the background subtraction algorithms implemented, the approximate median method turned out to be the best option for most applications. Frame difference is fast and easy to implement but very susceptible to noise and very dependant on continuous movement of the object. Finally, mixture of Gaussians handles noise relatively well but is very slow and very difficult to tune.
- Morphological filtering proved to be a valuable method for removing noise that leaked from the background in the subtraction operation.
- The tracking system was able to detect and track objects occupying tens of pixels in the screen under controlled conditions.
- In low resolution objects, color contrast between the object and the background is the feature that provides more information about the object. Ultimately permits the detection of such objects.
- Information such as relative velocity, centroid, and position can be extracted from the system.
- MATLAB proved to be an important tool when developing prototypes due to its built-in video processing and mathematical tools. For real time implementation the use of lower level languages is required.
- The separation of the problem into blocks was designed to permit future improvements in each of the four blocks. This is a system that can be improved in each of its blocks separately allowing for future implementation to use all or part of the blocks and improve others.

Publications (uploaded with this report)

1. W. T. Rhodes and D. Pava, "Removing Ambiguity in 2-D Image Information by Means of 3-D Models," in *Digital Holography and Three-Dimensional Imaging*, OSA Technical Digest (CD) (Optical Society of America, 2008), paper DMA3.
2. D. Pava and W. T. Rhodes, "Low-Resolution Motion Analysis in a 3-D Model," in *Digital Holography and Three-Dimensional Imaging*, OSA Technical Digest (CD) (Optical Society of America, 2010), paper JMA39.
3. Diego F. Pava, Object Detection in Low Resolution Video Sequences, masters degree thesis, Department of Electrical Engineering, Florida Atlantic University, April 2009.

Additional attachment: Video sequence illustrating human tracking ability of program.

**OBJECT DETECTION IN LOW RESOLUTION VIDEO
SEQUENCES**

by

Diego F. Pava

A Thesis Submitted to the Faculty of
The College of Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degree of
Master of Science

Florida Atlantic University

Boca Raton, Florida

April 2009

OBJECT DETECTION IN LOW RESOLUTION VIDEO

by

Diego F. Pava

This thesis was prepared under the direction of the candidate's thesis advisor, Dr. William T. Rhodes, Department of Electrical Engineering, and has been approved by the members of his supervisory committee. It was submitted to the faculty of The College of Engineering and Computer Science and was accepted in partial fulfillment of the requirements for the degree of Master of Science.

SUPERVISORY COMMITTEE:



Thesis Advisor
William T. Rhodes, Ph.D.



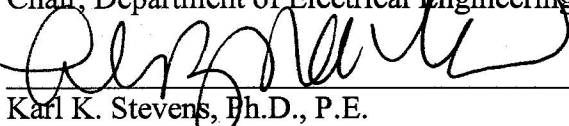
Hari Kalva, Ph.D.



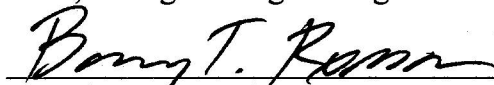
Nurgun Erdol, Ph.D.



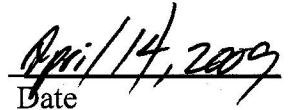
Salvatore Morgera, Ph.D.
Chair, Department of Electrical Engineering



Karl K. Stevens, Ph.D., P.E.
Dean, College of Engineering and Computer Science



Barry T. Rosson, Ph.D.
Dean, Graduate College


Date

ACKNOWLEDGMENTS

First of all, I would like to thank Dr. William T. Rhodes, more than a Thesis advisor he has been a mentor. I also want to thank to the members of the Thesis committee Dr. William Glenn and Dr. Hari Kalva. I also want to thank my colleagues of the MLAB especially Carlos Pertuz, Alvaro Fonseca, Paula Carrillo and Chris Holder who helped me a lot; Dr. Oge Marques who gave me a lot of advice; my colleagues of the Imaging Science and Technology Lab, among them Lt. Deissy Garces, John Healy and Jennifer Ward; Dr Maria Petrie and Maureen Manoly without whom I wouldn't be here and all my friends of FAU.

In addition, I would like to thank my family, my father that has been a model of hard work and honesty, my uncle Ricardo Arias who helped me in the quest of continuing my education, my eternal gratitude, my sister Isabel, Walter Hunter, but specially my mother whose incredible sacrifice is finally paying off, and Eileen who always was there lifting me up and supporting me when I needed it.

Last, but not least, I want to thank God, thanks to him I am here, thanks to him all this work have meaning.

ABSTRACT

Author: Diego Fernando Pava
Title: Object Detection in Low Resolution Video Sequences
Institution: Florida Atlantic University
Thesis Advisor: Dr. William T. Rhodes
Degree: Master of Science
Year: 2009

With augmenting security concerns and decreasing costs of surveillance and computing equipment, research on automated systems for object detection has been increasing, but the majority of the studies focus their attention on sequences where high resolution objects are present. The main objective of this work is the detection and extraction of information of low resolution objects (e.g. objects that are so far away from the camera that they occupy only tens of pixels) in order to provide a base for higher level information operations such as classification and behavioral analysis. The system proposed is composed of four stages (preprocessing, background modeling, information extraction, and post processing) and uses context based region of importance selection, histogram equalization, background subtraction and morphological filtering techniques. The result is a system capable of detecting and tracking low resolution objects in a controlled background scene which can be a base for systems with higher complexity.

OBJECT DETECTION IN LOW RESOLUTION VIDEO

SEQUENCES

TABLES	x
FIGURES	xi
Chapter 1 INTRODUCTION	1
1.1 Motivation	1
1.2 Problem Statement.....	2
1.3 Context and Scope	3
1.4 Main Contributions.....	4
1.5 Overview of the Thesis.....	5
Chapter 2 BACKGROUND AND RELATED WORK	6
2.1 General Framework	6
2.1.1 Background Modeling	7
2.1.2 Foreground Extraction.....	7
2.1.3 Filtering	8
2.1.4 Tracking.....	8
2.1.5 Actuator	9

2.2	Theoretical Background.....	9
2.2.1	Removing ambiguity in 2-D video by means of 3-D models [1].....	9
2.2.2	Histogram[[11].....	11
2.2.2.1	Histogram Equalization.....	12
2.2.3	Background Subtraction.....	14
2.2.4	Morphological filtering [11].....	19
2.2.4.1	Dilation and Erosion.....	20
2.2.4.2	Opening and Closing.....	23
2.2.5	Tracking.....	26
2.2.6	Finite State Machines []......	28
Chapter 3 PROPOSED SOLUTION		31
3.1	General Description.....	31
3.1.1	Block Diagram.....	31
3.2	Detailed Description.....	32
3.2.1	Preprocessing.....	33
3.2.1.1	Block Diagram.....	33
3.2.1.2	Data Acquisition.....	33
3.2.1.3	Color to Grayscale Converter.....	34
3.2.1.4	Region of Importance.....	35
3.2.1.5	Image Enhancement.....	35
3.2.2	Background Modeling.....	36
3.2.2.1	Block Diagram.....	36

3.2.2.2	Algorithm Selector	37
3.2.2.3	Algorithm Implementation	37
3.2.3	Information Extraction	38
3.2.3.1	Block Diagram.....	38
3.2.3.2	Morphological Filtering.....	39
3.2.3.3	Tracking System.....	39
3.2.4	Postprocessing	43
3.2.4.1	Block Diagram.....	43
3.2.4.2	Video Presentation.....	43
3.2.4.3	Data Cell.....	44
Chapter 4	IMPLEMENTATION.....	45
4.1	Preprocessing.....	46
4.1.1	Data Acquisition.....	46
4.1.2	Color to Grayscale Converter.....	47
4.1.3	Region of Importance	48
4.1.4	Image Enhancement	49
4.2	Background Subtraction	51
4.2.1	Algorithm Selection.....	51
4.2.2	Algorithm implementation	53
4.2.2.1	Frame Difference.....	53
4.2.2.2	Approximate Median.....	54
4.2.2.3	Mixture of Gaussians.....	55

4.3	Information Extraction	59
4.3.1	Morphological Filtering.....	59
4.3.2	Tracking System.....	60
4.4	Post processing	65
4.4.1	Video Presentation.....	65
4.4.2	Data Cell.....	67
Chapter 5 EXPERIMENTS AND RESULTS.....		68
5.1	Preprocessing.....	69
5.1.1	Data Acquisition and grayscale conversion	69
5.1.2	Region of Importance	69
5.1.3	Image Enhancement	71
5.2	Background Subtraction	76
5.2.1	Frame Difference.....	78
5.2.2	Approximate Median.....	80
5.2.3	Mixture of Gaussians.....	81
5.3	Morphological filtering	82
5.4	Information Extraction and post processing.....	86
5.5	Time Analysis.....	89
Chapter 6 CONCLUSIONS AND FUTURE WORK		91
6.1	Conclusions	91

6.2	Future Work.....	95
	Bibliography.....	97

TABLES

Table 1: Tracking System variables.	61
Table 2: Time analysis of the functions.	90

FIGURES

Figure 1: General framework of a visual surveillance system [2].....	6
Figure 2: Group of pixels (Left), probable position of the group (right).	10
Figure 3: Examples of Histogram diagram for different kinds of pictures. [18].....	12
Figure 4. Background subtraction process, background model (left), input frame (center) and foreground extraction (right).....	15
Figure 5: Basic Boolean operations.....	20
Figure 6: Example of dilation (green blocks are the ones added to the original).....	21
Figure 7: Example of erosion (the green blocks are the ones that will disappear from the image).	22
Figure 8: Example of Opening.	23
Figure 9: Example of Opening in a more complex binary image (structuring element a 20 pixel square).	24
Figure 10: Example of closing.	25
Figure 11: Example of closing with a more complex binary image (structuring element a 20 pixel square).....	26
Figure 12. Taxonomy of tracking parameters	28
Figure 13: Finite State Machine Logic.....	29
Figure 14: State diagram.	29
Figure 15: Block Diagram.	32

Figure 16: Block diagram of the Preprocessing.	33
Figure 17: Block diagram of the Background Subtraction algorithm.	37
Figure 18: Information Extraction Block diagram.	38
Figure 19: Tracking system state diagram.....	40
Figure 20: Object tracking based on bounding box and centroid position.....	42
Figure 21: Postprocessing Block Diagram.	43
Figure 22: Functional flowchart.	46
Figure 23: Scene where histogram equalization over the whole picture will have an undesired effect.	50
Figure 24: Example of selection of region of importance.....	70
Figure 25: Selection of the region of importance.....	71
Figure 26: Contrast enhancement test image.....	72
Figure 27: Contrast enhancement test. a) input image, b) output image, c) histogram of the input image, d) histogram of the output.	73
Figure 28: Moon surface with poor contrast (image and histogram).	74
Figure 29: Moon surface after the histogram equalization.....	74
Figure 30: Contrast enhancement applied to the videos. Video in color, grayscale, and contrast enhanced, respectively.	75
Figure 31: Object with the histogram equalization versus object without contrast enhancement.	76
Figure 32: Ground truth example (manually annotated segmentation).....	77
Figure 33: Very low resolution object.....	77
Figure 34: Threshold comparison for the Frame Difference algorithm.	79

Figure 35: Threshold test of the approximate median algorithm.	80
Figure 36: Parameter test of the mixture of Gaussians.....	81
Figure 37: Test image for the morphological filter operation.	82
Figure 38: Image with Gaussian random noise (left), same image after the morphological filter (right).....	83
Figure 39: Image with Salt and Pepper noise and output (top), image with Poisson and salt and pepper noise and output (middle), and image with artificially constructed noise and output.	84
Figure 40: Grayscale image (bottom), background subtraction algorithm (top-left), and morphological filter output (top-right).	85
Figure 41: Tracking system.	86
Figure 42: Occlusion handling of the tracking system.	87
Figure 43: Tracking of video No. 3.....	88
Figure 44: The trackingdata cell.....	89

Chapter 1 INTRODUCTION

1.1 Motivation

For years, automatic video recognition of moving objects has been one of the most rapidly developing topics in video image processing due to the great variety of fields that could potentially benefit from such advancement.. Robotic vision, medical imaging, space exploration, remote monitoring, and video surveillance are among the various fields that have attracted researchers to the problem of detecting and extracting information from moving objects.

Over the past decade, a numerous algorithms have been proposed for moving-object tracking, but a solution that clearly outperforms the human vision system is still missing, leaving room for new researchers to come up with new ideas on how to improve existing methods or develop new ones.

In this post 9/11 world, video surveillance has become a topic of great importance. Security is now a major concern not only to the government, but also to industries and the general public. With the prices of video surveillance systems dropping, each day it

becomes more common to find surveillance rooms where several screens receive video feeds from cameras distributed across a location under observation.

Even in the case of large objects, security personnel can sometimes be overwhelmed with the amount of information they must process, leading them to make costly mistakes by overlooking important information or losing time and resources on unimportant information [2].

Now imagine that the objects moving occupy only a few pixels in the screen, either because they are very small or because they are so far away from the camera. In such a case the work of security personnel without computerized help would be virtually impossible.

In this work effort is concentrated on the detection and extraction of information of such small objects in video sequences, especially in video sequences taken with conventional cameras.

1.2 Problem Statement

The task of detecting the presence of moving objects (human or vehicles for example) that are so far away as to only occupy a few pixels in the video sequence is not a simple

one. Blurring of background into the image of interest can degrade information exploited with conventional techniques such as shape and color.

This work investigates systems able to detect low-resolution moving objects in video sequences and extract as much information from the imagery with a program that is simple, fast, reliable, and robust.

1.3 Context and Scope

This Multi-University Research Initiative (MURI) supporting this research is a five-year, program that began in 2004. Participating universities are Georgia Tech, University of Mississippi, Florida Atlantic University, and MIT. The program PI is Prof. William T. Rhodes.

Research centers on the detection of humans in traditionally difficult circumstances (e.g., inside buildings or tunnels, under camouflage, etc.). The program is focused on two primary areas: human signature physics (i.e., what signals can be used to either uniquely register the presence of a human or provide a “hint” of human presence), and sensor networking., how can we combine and configure multi-modal sensors in a communication grid to enhance our ability to detect humans.

An important part of the MURI program is of an imaging nature and operate for large area coverage, at low resolution. Yet, the basic nature of low image resolution has not been extensively investigated. Most image-based human detection schemes presume a comparatively high number of pixels on target. The objective is then to improve the knowledge of the fundamental nature of objects in the low resolution regime placing emphasis first on a characterization of fundamental phenomena associated with the size, configuration, and motion of images of objects at low resolution, for ultimately, apply this knowledge to the human detection problem.

1.4 Main Contributions

The following are the main contributions of this work:

- Use of imaging processing techniques such as histogram equalization in order to extract as much information from the video as possible.
- Using the knowledge of the 3D-scene in order to improve the speed and robustness of the program. [1]
- Implementation and comparison of different background subtraction techniques with the goal of extracting relevant / moving objects from video frames.

- Implementation of morphological filtering techniques to complement the background subtraction techniques.
- Tracking of low resolution moving objects present in the sequence.
- Establishment of a framework for future FAU students and researchers who may concentrate efforts on related, more sophisticated topics of activity recognition.

1.5 Overview of the Thesis

This thesis is structured as follows: Chapter 2 provides background information on visual surveillance systems and algorithms. Chapter 3 describes in detail the proposed solution. Implementation aspects are contained in 4, experiments and results are included in Chapter 5, and Chapter 6 presents conclusions and possibilities for future work.

Chapter 2 BACKGROUND AND RELATED WORK

In this chapter, an introduction to general concepts and algorithms associated with this thesis is provided.

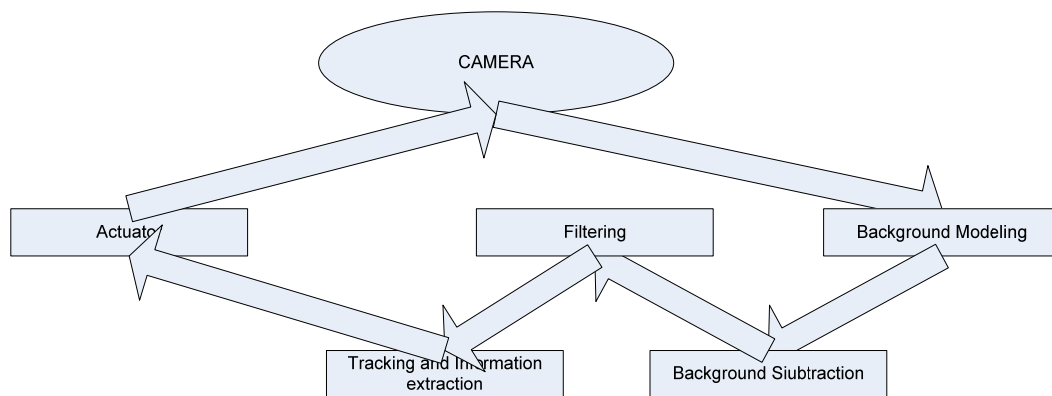


Figure 1: General framework of a visual surveillance system [2].

2.1 General Framework

The general framework of a visual surveillance system is composed of the blocks shown in Figure 1. Each block will be briefly explained in the next subsections (adapted from [2]).

2.1.1 Background Modeling

A *sequence* is a set of consecutive frames recorded at the same location. The group of common elements within the sequence is referred to as the *background*. In Figure 1, we can see the general surveillance process. After the information is recorded by the camera, the system creates a model of the background. Some of the background modeling techniques involve averaging the pixel values over a certain number of frames where foreground objects are not present [3], [4]. Other approaches are based on adaptive Gaussian estimations [5], parameter estimation based on pixel processes [6], and approximate median method [7]

2.1.2 Foreground Extraction

The objective is to separate foreground from background in the video sequence. This is usually accomplished by subtracting the output of the previous block from each frame [8]. There are, however, other techniques such as temporal differencing [9] and optical flow [10] that can be used.

The foreground can be represented in a binary image (the pixels corresponding to the foreground are labeled as one and those of the background labeled as zero) or a grayscale or color image where the foreground conserves its original characteristic.

2.1.3 Filtering

Mathematical morphology is a tool used for extracting image components that are useful in the representation of shapes such as boundaries and skeletons [11]. Of special importance for this work is morphological filtering.

Morphological filtering consists in a series of Boolean operations made to a binary image where objects occupy the region labeled as 1 and the background occupy the region labeled as 0. These techniques are used to remove unwanted elements in the sequence such as noise or non important moving objects.

2.1.4 Tracking

This block compares the group of characteristics that define each object in order to locate its position along the sequence ([12],[13],[14]). When an object is being tracked, important information such as position, velocity, centroid, distance from the camera, and periodicity becomes then available. For example, walking or running humans have characteristic periodic motion; this unique signature can be detected only after tracking a subject for a given period of time [15].

2.1.5 Actuator

After the system has extracted all the information from the sequence, the actuator stage takes on the problem of decision making based in the information obtained. The decision could be gathering information from different cameras to obtain advantages such as depth and overcome problems such as occlusion, or to raise an alarm so that the security personnel can take a closer look at an important event occurring. Camera installation [16] and calibration [17] are the kind of problems related to the functions associated with this block.

2.2 Theoretical Background

In this section, a theoretical description of algorithms relevant to this work is provided.

2.2.1 Removing ambiguity in 2-D video by means of 3-D models [1]

If moving objects in low-resolution 2D video imagery are placed in their 3D context, ambiguities concerning the identity of the objects can often be removed. In the identification of objects moving in a video sequence, the availability of a 3-D model of

the scene can reduce, often greatly, uncertainties in the nature of what is being observed.

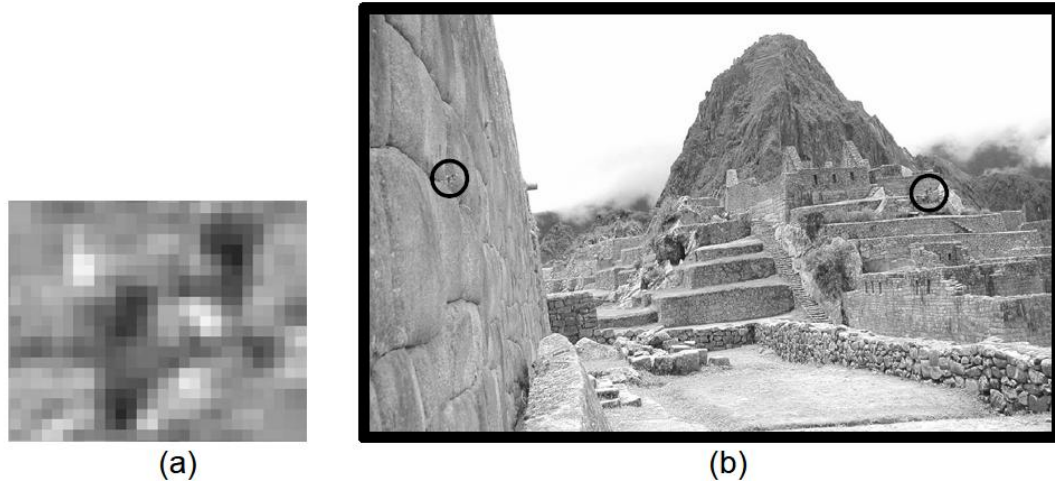


Figure 2: Group of pixels (Left), probable position of the group (right).

Figure 2 shows a group of pixels that could represent humans far away from the camera. Just by watching the group of pixels there is no way of telling what they represent. If the group of pixels corresponds to the area encircled in the left of image b), then the probability of the group of pixels being humans is zero. However, if the pixels correspond to the area encircled to the right of the same image, a path in the Machu Picchu ruins, then the probability of the group of pixels being humans walking along a path far away from the camera increases dramatically.

If the system receives several frames of information and track for motion in the group of pixels, then the probability of assessing correctly whether the pixels correspond to a human walking increases. If the motion makes sense in the 3D context (the pixel group

moves along a path) and have certain characteristics such as periodicity [15], velocity (moving at human speed) and consistency (the objects does not suddenly disappear from the scene), then we can be relatively certain regarding the nature of the group of pixels and what they represent. The same applies for other objects of interest different from humans.

In synthesis, in a complete solution, we can get better results if we can exploit our knowledge on the 3D models by creating regions of interest. These regions are those where the presence of meaningful moving objects are more probable and therefore need more computational resources to analyze.

2.2.2 Histogram[18][11]

Histograms are the basis for numerous spatial domain processing techniques. Histogram manipulation, in addition to providing useful image statistics, can also be very useful when using image enhancement techniques. Histograms are simple to implement in software and usually cheaper for hardware implementations, thus making them a popular tool.

An image with low contrast has a narrow histogram, usually centered toward the middle of the grayscale, while a high-contrast image has the characteristics of covering a broad range of the grayscale and, in addition, having the pixels almost uniformly distributed.

High contrast images present very few vertical lines in the histogram that are much higher than the others. A high contrast image will exhibit a large variety of gray tones and great detail and also will have high dynamic range.

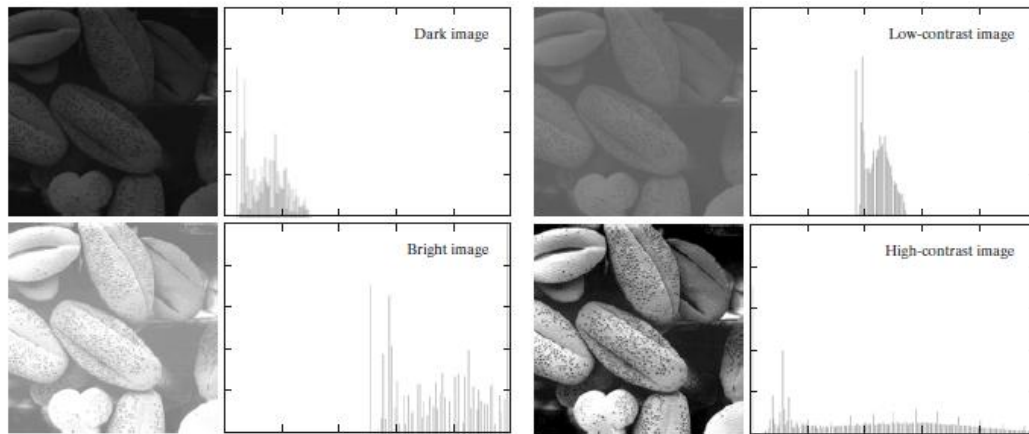


Figure 3: Examples of Histogram diagram for different kinds of pictures. [18]

2.2.2.1 Histogram Equalization

Let's assume that a given image has a continuous range of intensity levels from 0 to 1 and let $p(r)$ be the probability density function (PDF) of the intensity levels. We proceed to perform the transformation:

$$s = T(r) = \int_0^r p(w)dw \quad (1)$$

Gonzalez and Woods [18] show that the output of such a system will have a uniform PDF at the output $p_o(s)$, thus:

$$p_o(s) = \begin{cases} 1. & \text{For } 0 \leq s \leq 1 \\ 0. & \text{Otherwise} \end{cases} \quad (2)$$

The result, then, will consist on an image whose intensity is distributed equally throughout the range, or, in other words, a high contrast image. From the equation, it is clear that $T(r)$ is simply, the cumulative distribution function (CDF) of the system.

If instead of continuous signals, we are working with discrete intensity levels, then $p(r_j)$ is really the normalized histogram of the input image with $j=0,1,2,\dots,L$ being the discrete intensity levels and the transformation $T(r_k)$ is then known as the Histogram Equalization. Since we are working with discrete values, integration becomes summation and the transform function is then

$$s_k = T(r_k) = \sum_{j=1}^k p_r(r_j) = \sum_{j=1}^k \frac{n_j}{n} \quad (3)$$

where n_j is the number of pixels with the intensity level j and n the total number of pixels.

Due to the discrete nature in the system, the output will not be completely uniform, although its dynamic range will increase dramatically. In Figure 3: Examples of Histogram diagram for different kinds of pictures. [18], the image in the lower right is the histogram-equalized version of the image in the upper left.

It is important to note that by using histogram equalization the image is going to have its contrast enhanced, but that does not necessarily means that is going to be better

visually, For this study the contrast enhancement is performed as a middle step and therefore is not important if the image is better visually or not since this step is transparent to the user.

2.2.3 Background Subtraction

The *Background Modeling* and *Foreground Extraction* blocks mentioned in section 2.1 can be merged into a single block denoted *Background Subtraction*. This block represents one of the most common approaches to the problem of detecting moving objects in video sequences. In the background subtraction scheme, each frame is individually compared to a reference background model pixel by pixel. the current pixel deviates significantly from the background model, it is considered to be a foreground object and labeled as foreground. Background subtraction is thus usually the first step prior to other implementations such as position tracking, velocity of the objects and the alarm system.

Several background subtraction algorithms have been proposed, each of them with its own advantages and limitations. There are, however, certain requirements. The algorithm must be robust; it must adapt to changes in the environment such as wind, rain or illumination changes; it must be fast enough so that the information being analyzed is still meaningful; and lastly it must consume as little computing resources as possible.

The algorithm used to separate the foreground from the background impacts the quantity of noise present in the output. Figure 4 shows the background subtraction process.



Figure 4. Background subtraction process, background model (left), input frame (center) and foreground extraction (right).

Some of the most frequently used background subtraction techniques are: [19]:

- *Frame differencing*: This method is perhaps the simplest background subtraction method available. In this method, each frame is subtracted from the previous frame and the difference is then compared with a threshold. If the difference is bigger than the threshold then the pixel is foreground, otherwise it is background. The equation of the algorithm is as follows:

$$Foreground = |Frames_i - Frames_{i-1}| > Threshold \quad (4)$$

This approach has two important advantages. First, the fact that the background is constantly changing makes this algorithm a fast adapting one. It adapts quickly to changes in illumination and shadows as well as to changes in the

weather conditions of the video. Besides, is simple to implement and therefore it is fast and consumes less resources than other approaches.

But it has also serious flaws. All the objects must be moving constantly because the moment they stop they will be recognized as background in subsequent frames. Furthermore, the inside of the objects would be recognized as background if the objects are big enough with little internal structure.

- *Temporal median filter:* In this algorithm, the information from previous frames is accumulated in order to get the average value for each pixel [20]. The median method creates a buffer of the last N frames and models the background as the median of those frames. This approach has proven to be very robust and have good performance in most applications. It is also very adaptative as the frame difference approach (although not as fast to adapt). However, this approach consumes a lot of memory resources as it is necessary to store several frames.

- *Approximate Median Method:* A good approximation to the median approach was created by McFarlane and Schofield in 1995 and is currently known as the approximate median method. In this method each pixel in the current frame is compared with the one in the background, if the pixel in the current frame is larger then the background intensity is incremented by one, if on the other hand the background pixel is larger then it is decreased by one. The

background will then tend to be a good approximation of the median being the time of stabilization a function of the number, the size and velocity of the objects moving. This method will have less memory usage at the expense of some stabilization time

- *Mixture of Gaussians (MoG)*: This technique takes into account changing elements in the background such as moving trees or falling snow. In order to create the model of the background, a combination of different Gaussian pdf's is required to model each pixel [21].

In MoG, the background is not modeled as a frame of values. Instead, the model is purely parametric with each pixel location represented by a number (mixture) of Gaussian functions that sum together to form a probability distribution function of the form:

$$F(i_t = \mu) = \sum_{i=1}^k \omega_{i,t} \cdot \eta(\mu, \sigma) \quad (5)$$

The μ corresponds to the mean of each Gaussian component that can be thought of as an educated guess of the pixel value in the next frame assuming that pixels are usually background. The ω , which is the weight, and the σ which is the standard deviation of each component, can be thought as measures of our confidence in that guess (higher weight and lower standard deviation equals higher confidence). There are usually 3 to 5 Gaussian components per pixel, depending on memory limitations.

To determine if a given pixel is part of the background, a comparison is made between the current pixel to the Gaussian components tracking it. If the pixel value is within a scaling factor of one of component's standard deviation σ , then it is considered part of the background. Otherwise, it's foreground.

- *Running Gaussian average:* In this approach, a Gaussian probability density function (pdf) is compared to the last n frames of a video sequence, and the average for each pixel is updated according to its previous values [22].it is a mixture between the Gaussian and the median approximation.
- *Kernel density estimation (KDE):* The KDE calculates the pdf of a random variable, instead of assuming a Gaussian distribution for each pixel, the “true” pdf is extracted according to the values of previous frames [23].
- *Co-occurrence of image variations:* Blocks are used over individual pixels. It is based on the fact that neighboring blocks of pixels belonging to the background should experience similar variations over time [24].
- *Eigenbackgrounds:* In this technique eigenvectors¹ of the background are obtained by averaging a specific number of frames. The new frames are then projected to the eigenspace and back again to the image space; in this process

¹ “Let \mathbf{A} be a p by p matrix and \mathbf{w} a p -element vector. If it is true that $\mathbf{A}\mathbf{w} = \lambda \mathbf{w}$ for some scalar λ , then \mathbf{w} is an *eigenvector* of \mathbf{A} and λ is the corresponding *eigenvalue*. That is, an *eigenvector* of a matrix is a vector such that when we multiply the matrix by the vector we get the vector back again except that it has been multiplied by a particular constant, called the *eigenvalue*” [25]

the static portions of the sequence are revealed. The frame is then subtracted from the model therefore obtaining the foreground and background components ([25],[26]).

2.2.4 Morphological filtering [11]

Morphology is technique for the analysis and processing of geometrical structures based on set and lattice theory. Let Z be the set of integers. The sampling process used to transform the continuous environment in a digital image may be organized as a grid on the XY -plane, if each center of the grid is associated with an integer pair of numbers (x,y) and is assigned a intensity value (which could be a real number) then the image is said to be a digital image.

Groups of neighboring grid centers (referred to in imaging processing as pixels) can be thought as sets, and Boolean algebra can be applied to them. Morphological operations are then Boolean algebraic operations applied to the mapping of selected regions in a digital image. They can perform tasks such as finding the skeleton of a figure, filtering and restoration. To apply such methods we first transform the digital image into a binary one where the meaningful sectors are labeled one and the rest labeled zero.

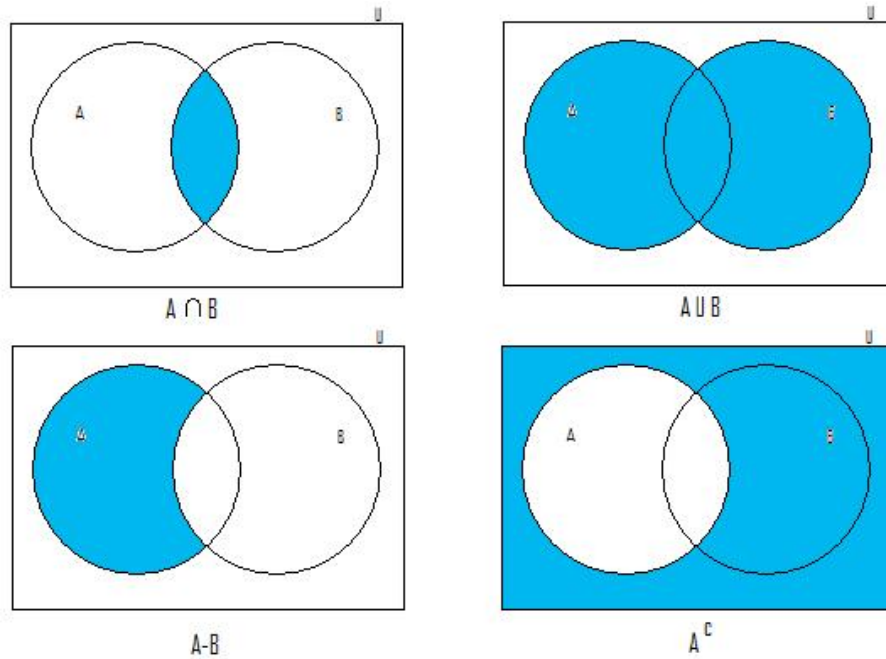


Figure 5: Basic Boolean operations.

2.2.4.1 Dilation and Erosion

Dilation and erosion are the building blocks of many operations in morphological image processing.

- *Dilation* is an operation that enlarges the objects present in a binary image. The extent to which the objects grow depends on a controlling object referred to as the *structuring element*.

E being the entire grid system, dilation is defined mathematically as:

$$A \oplus B = \{z \in E | (B^s)_z \cap A \neq \emptyset\} \quad (6)$$

In the equation, B^s is the symmetric of B, which means B^s is the group of elements b such that $-b$ belongs to B. In other words, the dilation of image A by the structuring element B is the set consisting of all structuring element locations when the symmetric of B overlaps with at least a portion of A. In that sense, dilation behaves in a similar manner to the 2-D convolution, and like convolution, dilation is commutative .

$$A \oplus B = B \oplus A = \bigcup_{a \in A} B_a \quad (7)$$

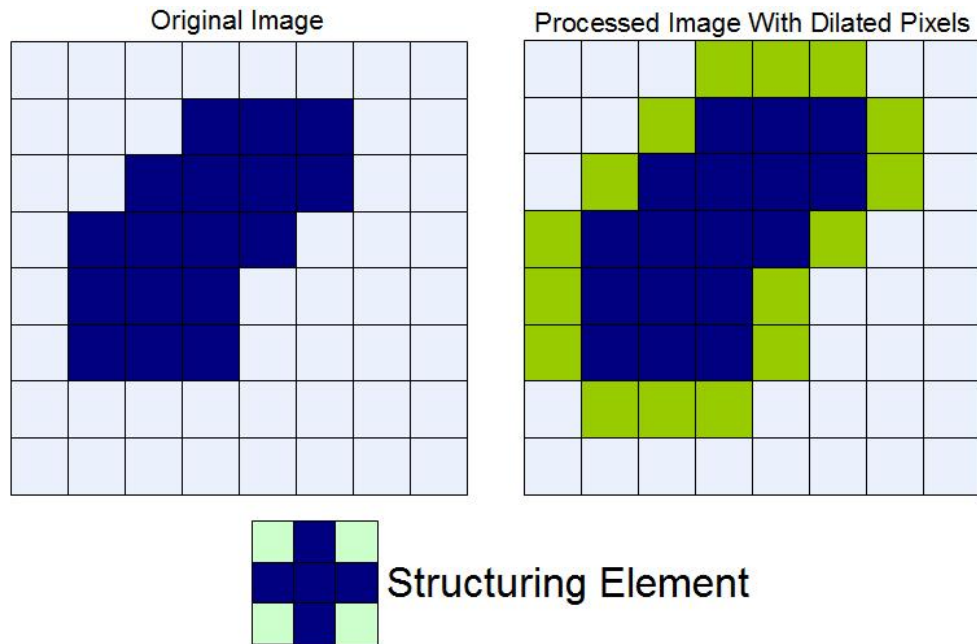


Figure 6: Example of dilation (green blocks are the ones added to the original).

- *Erosion* is the opposite of dilation; it shrinks the objects present in a binary image, the extent to which the objects thin depending again on a controlling object referred to as the structuring element.

E being the entire grid system, mathematically, dilation is defined as:

$$A \ominus B = \{z \in E | B_z \subseteq A\} \quad (8)$$

Which means that the erosion of A by B is the set of all structuring element locations where the structuring element does not overlap with the background of A. Note that erosion is not commutative.

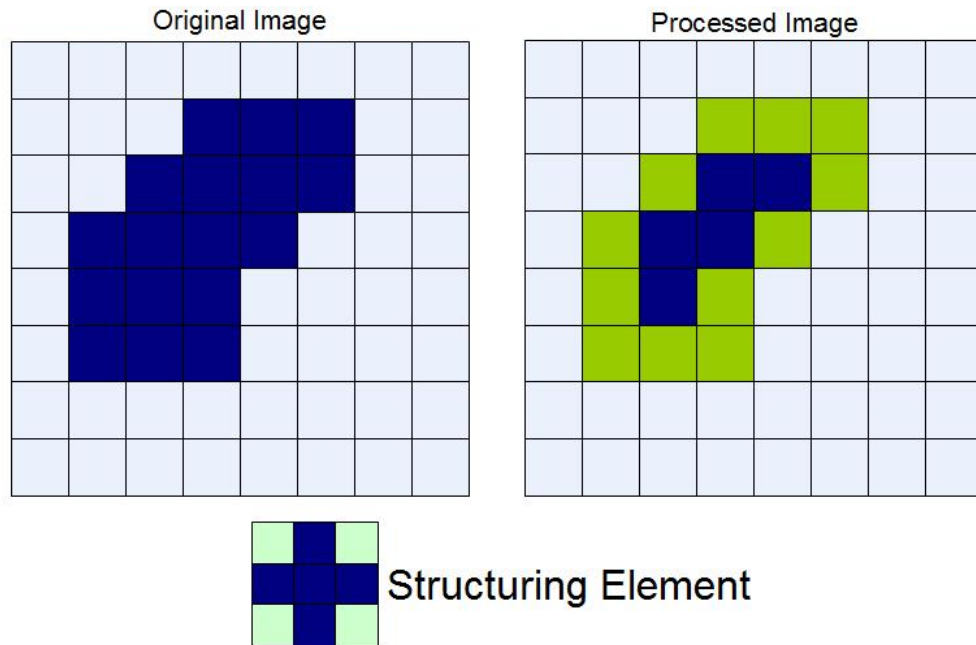


Figure 7: Example of erosion (the green blocks are the ones that will disappear from the image).

2.2.4.2 Opening and Closing.

- *Opening* is simply the erosion of A by a structuring element B followed by a dilation of the output by the same structuring element. In synthesis:

$$f \circ s = (f \ominus s) \oplus s \quad (9)$$

Opening is then the Union of all possible locations of structuring element B where B fits entirely inside A.

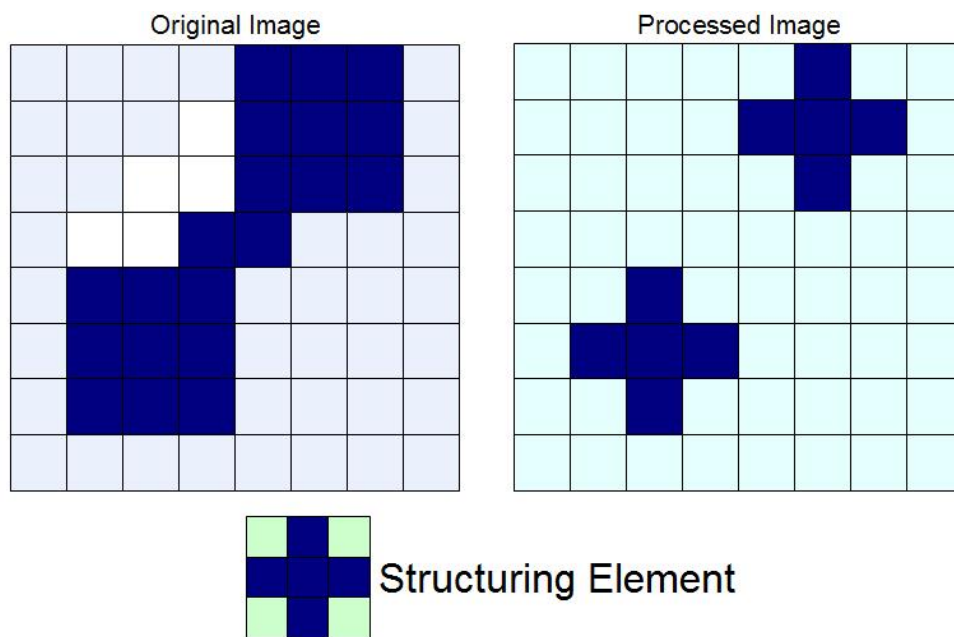


Figure 8: Example of Opening.

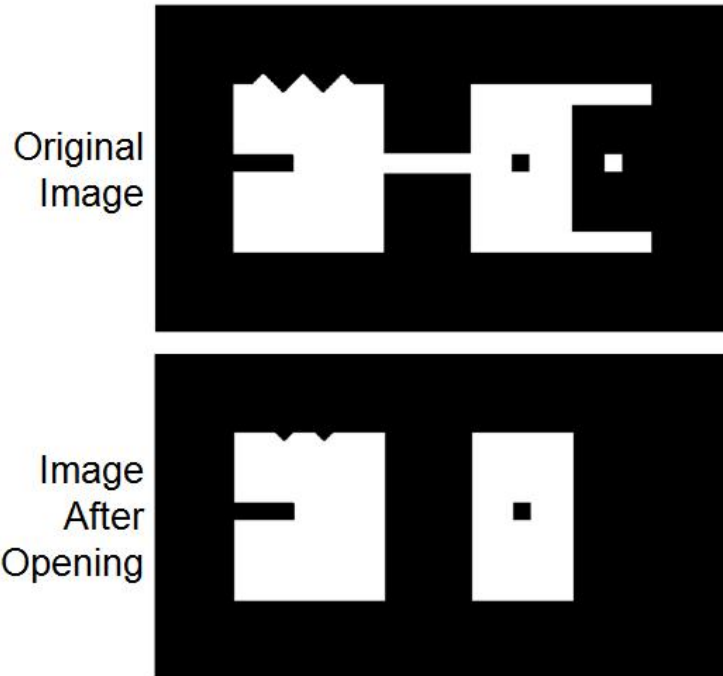


Figure 9: Example of Opening in a more complex binary image (structuring element a 20 pixel square).

- *Closing* is simply the dilation of A by a structuring element B followed by an erosion of the output by the same structuring element. In synthesis:

$$f \bullet s = (f \oplus s) \ominus s \quad (10)$$

Closing is then the complement of the union of all possible locations of structuring element B where B fits entirely outside A.

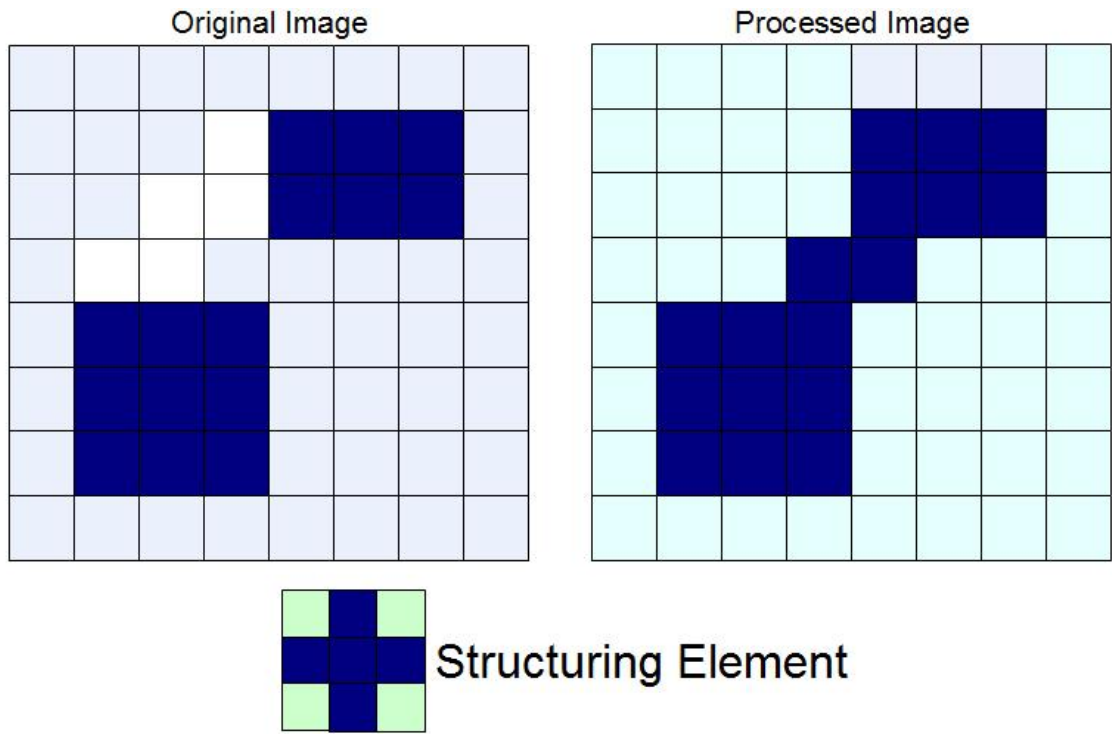


Figure 10: Example of closing.

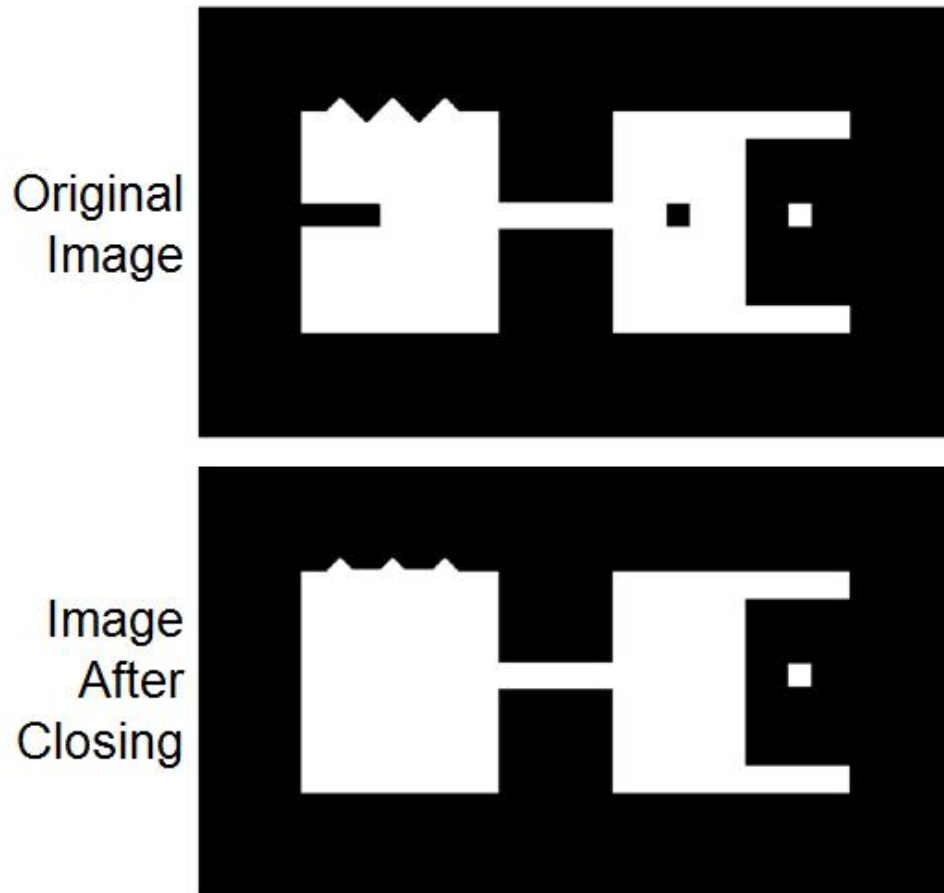


Figure 11: Example of closing with a more complex binary image (structuring element a 20 pixel square)

2.2.5 Tracking

Several solutions have been explored to the problem of tracking objects across multiple frames. A classification can be made according to the way they combine the following set of parameters [27]:

- *Object representation:* Objects can be represented according to their shapes (primitive shapes, silhouettes, contour, etc) and appearances (templates, active and multiview models, probability densities, etc). The representations depend greatly on the application. For small objects, point representation is appropriate in video sequences.
- *Feature selection for tracking:* The object should be distinguishable from any other object present in the scene. For that purpose, tracking systems could look for a particular feature that is unique to the object. Among the features we find: color, edges, optical flow, and texture. The most common feature is color; however, combinations of different features usually improve tracking performance.
- *Object detection:* Every tracking method requires mechanisms for finding new objects. Classification can be made between techniques that achieve this goal by using information from a single frame and techniques that use accumulated information from a sequence of frames, such as background subtraction techniques.
- *Object Tracking:* Correlates the different instances of an object along the frames that compose a video sequence. The output of this block is the object's trajectory.

A synthesis of the parameter taxonomy of the tracking system is shown in Figure 12:
 Taxonomy of tracking parameters [28].

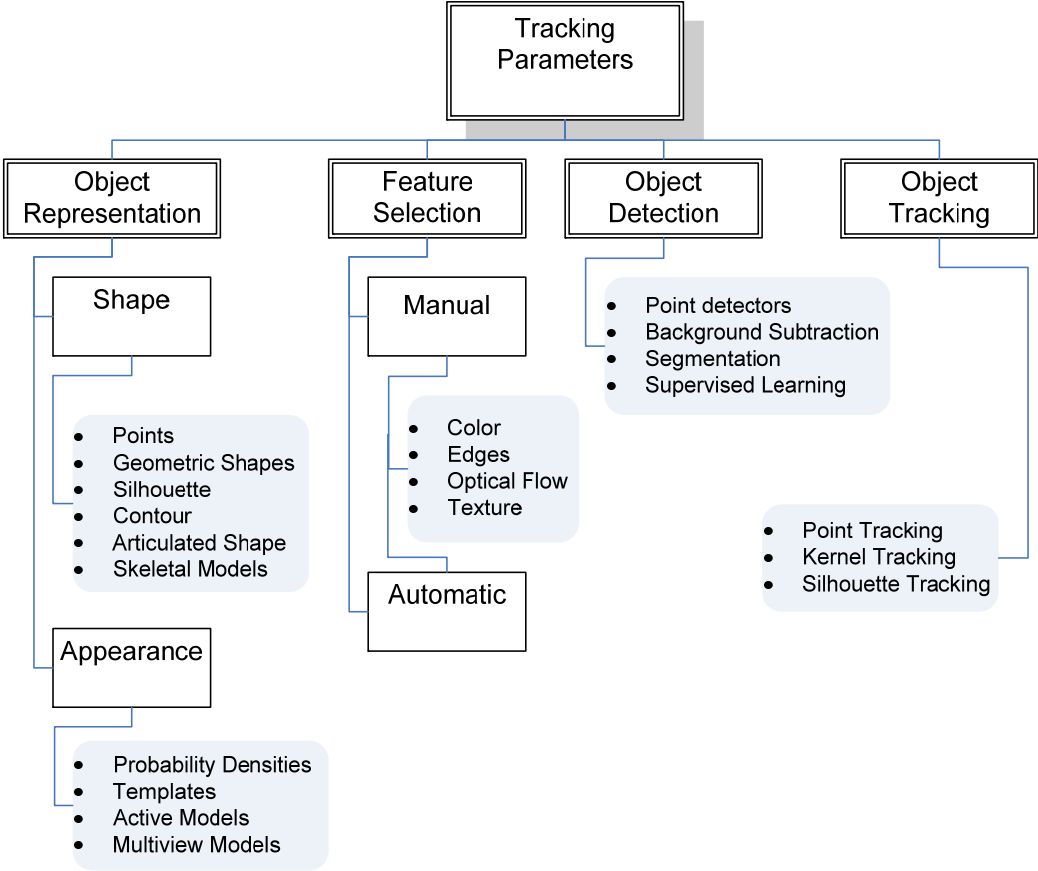


Figure 12: Taxonomy of tracking parameters

2.2.6 Finite State Machines [29]

A Finite State Machine (FSM) is an abstract machine that is used to describe behavior. It consists in a set of states, a set of inputs events, a set of outputs, and a set of transition functions which completely describe the behavior of a system. The current state of a

FSM is determined by the past events of the system and by the events occurring at the moment. If the event occurring fulfills certain conditions, then a transition between states occurs. The general logic of a FSM can be seen in Figure 13.

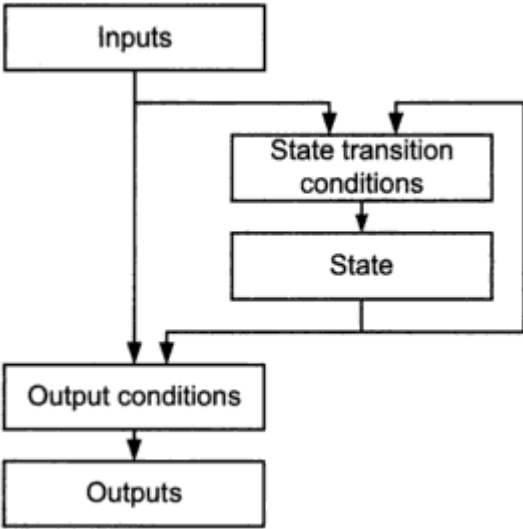


Figure 13: Finite State Machine Logic.

A FSM can be represented graphically with a state diagram similar to the one depicted in Figure 14: State diagram..

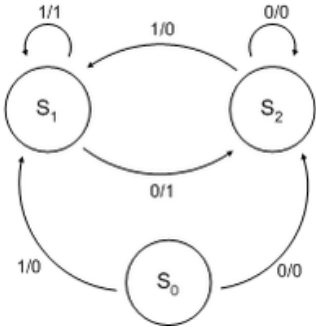


Figure 14: State diagram.

When the output of the FSM depends on the current state as well as on the input, the FSM is known as a Mealy machine.

Chapter 3 PROPOSED SOLUTION

3.1 General Description

The proposed solution follows the guidelines of surveillance systems highlighted in Chapter 2. This version of the system is not intended to work in real time due to restrictions in memory usage and computational resources. Instead, a sample video is collected and processed in order to obtain the needed information; the system then analyzes this information and displays it so that the user can see the results in different forms.

3.1.1 Block Diagram

Figure 15: Block Diagram. shows the general block diagram for the proposed solution:

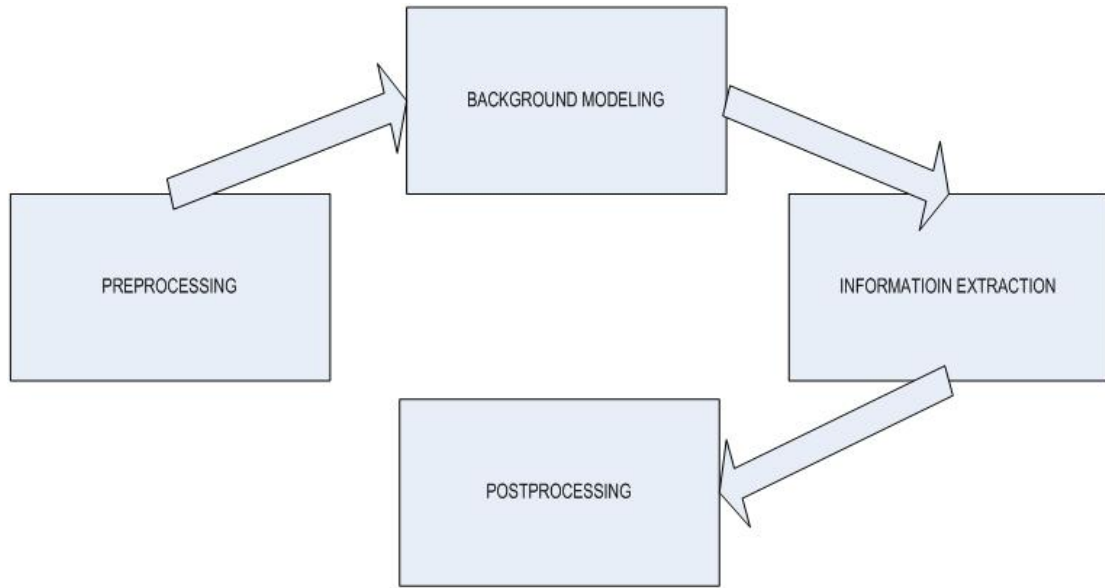


Figure 15: Block Diagram.

The Preprocessing block receives the information from the source and transforms it so that the essential information from the video is analyzed while unimportant information is ignored. The Background Modeling block receives the preprocessed information and separates the background from the foreground. Once modeled, the Information Extraction block obtains important data from the background and foreground. Lastly, the Postprocessing block receives and organizes this data so that the user can visualize the results obtained.

3.2 Detailed Description

3.2.1 Preprocessing

3.2.1.1 Block Diagram

The Block diagram for the preprocessing block is composed of four stages: a Data Acquisition block, where the information from the camera is received; a Color-to-grayscale converter block, where the data is transferred to grayscale matrix form; a prompt requesting the user to choose regions of importance that take advantage of the knowledge of the scene in the process; and, finally, an Image Enhancement block, where there is further data manipulation to optimize the information extracted.

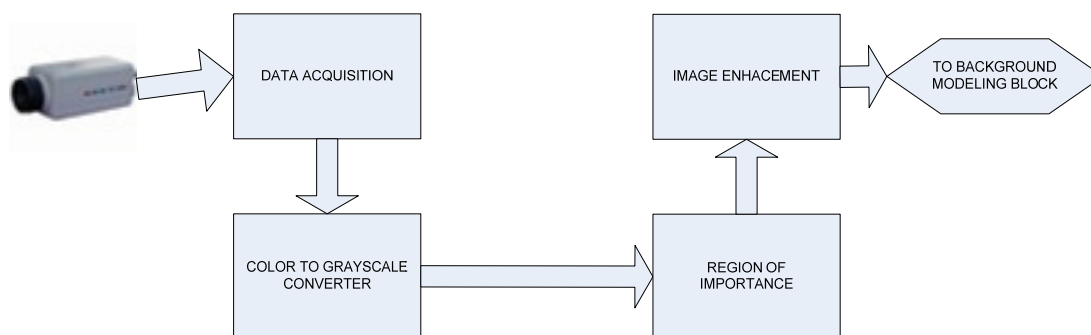


Figure 16: Block diagram of the Preprocessing.

3.2.1.2 Data Acquisition

The Data Acquisition block receives the video stream from the surveillance camera, decompress the data if compressed, and transform it from color-indexed to truecolor form if necessary.

The output of this block is a 4-D matrix containing the color video information. This matrix will also be used in the post processing block in order to present the results to the user.

3.2.1.3 Color to Grayscale Converter

In regular high definition object recognition systems, color may give important information about the nature of the object. For example, in one study, color is used for human recognition in high-definition images by exploiting the fact that skin color in human beings has a distinctive distribution in the chromaticity diagram [30].

However, as the objects become smaller or are located farther away from the camera, color information is less important except with respect to contrast. Since contrast can be achieved also in grayscale, and since grayscale video leads to systems more computationally efficient (use less memory and are usually faster) for background modeling analysis, the conversion is preferred.

3.2.1.4 Region of Importance

When objects occupy just a few pixels in a scene, there are usually important portions of the video sequence where the presence of objects of such characteristics is unlikely or unimportant. Security applications may require some regions to be attended while others can be ignored. Furthermore, through the setting regions of importance (ROI), inevitable noise coming from unimportant regions can be ignored with a resulting improvement in overall response of the system and computing resources management.

Because regions of importance depend on so many factors, user-created ROIs are preferred over automatic approaches. The system employed in this study requires that the user draw with the mouse the ROI. A binary mask of the ROI is then created and applied to the video after the image enhancement.

3.2.1.5 Image Enhancement

A small object can be sensed if its contrast is large enough for our visual system (or the computer vision system) to detect. Contrast depends on multiple factors such as color difference (not only hue difference but saturation and brightness as well), level of illumination of the scene, quality of the camera, etc.

Although most of these features are out of the control of the object detection system, some improvement can be achieved through the application of image processing techniques. For our system we implement histogram equalization to the image in order to enhance the contrast between foreground objects and the background.

3.2.2 Background Modeling

3.2.2.1 Block Diagram

The block diagram for the Background Modeling Block is composed of two stages. The first stage selects which of the three types of background subtraction algorithm is to be used, while the second stage actually implements the algorithm on the video, separating the foreground from the background. The input of this stage is the video after the histogram enhancement and with the unimportant regions extracted. The output is a binary video with zero representing the background and 1 representing the foreground pixels.

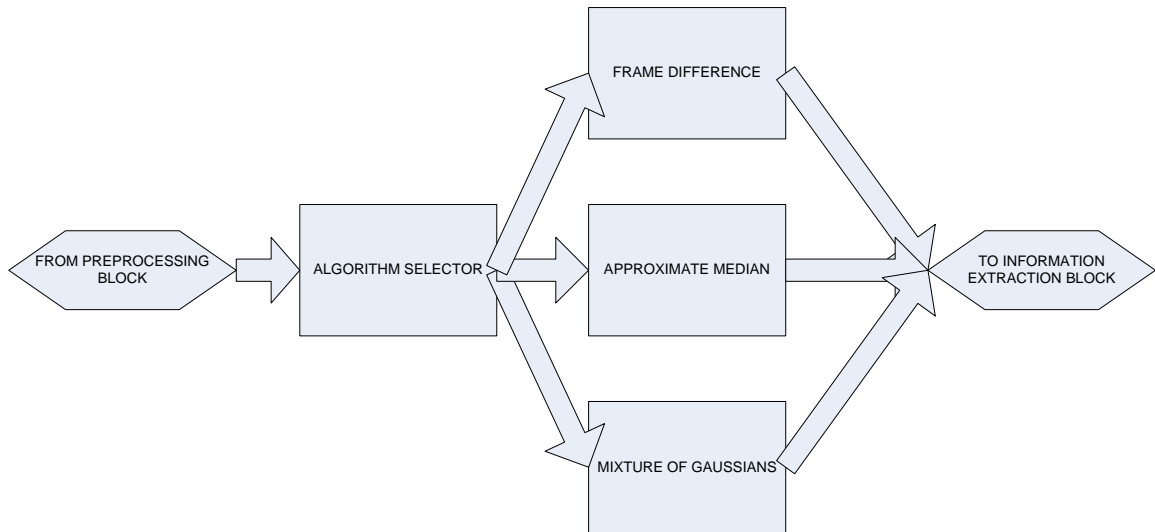


Figure 17: Block diagram of the Background Subtraction algorithm.

3.2.2.2 Algorithm Selector

The Algorithm selector prompts the user to choose between three background subtraction algorithms: Frame difference, Approximate Median, and Mixture of Gaussians. These three algorithms were explained in section 2.2.3.

3.2.2.3 Algorithm Implementation

According to the Algorithm Selector, the program implements one of the three algorithms available. The three algorithms were selected because they are quite different in their approach. Frame difference is very fast and easy to implement but is susceptible to noise and requires continuous movement, as explained in section 2.2.3.

Mixture of Gaussians is complex and elegant but takes a significant amount of time and computer resources, and its optimization is more difficult due to it having many variables (the other two implementations only have one variable). Finally, Approximate Median is of middle complexity, being as easy to optimize as the frame difference method but with added robustness and being less susceptible to noise.

The three algorithms have as output a binary image for each frame of the video, with zero representing the background and one representing the foreground. The images still contain some noise due to the different conditions of the video sequence.

3.2.3 Information Extraction

3.2.3.1 Block Diagram

The block diagram for the Information Extraction Block is composed of two stages. The first stage is a morphological filtering that handles the noise present after the background subtraction. The second stage is a tracking and information extraction system, which analyses the images and provides information about objects present in the video and their properties (position, velocity, etc).

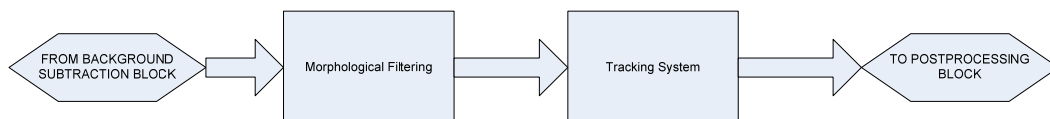


Figure 18: Information Extraction Block diagram.

3.2.3.2 Morphological Filtering

The result of the background subtraction operator contains some unwanted noise. The morphological filtering operation is intended to reduce the noise as much as possible. Morphological filtering in background subtraction systems but in the case of low resolution objects special care has to be taken.

Due to the nature of the object (objects occupying just a few pixels), a morphological operation could easily either remove important information (even remove the object entirely) or allow noise to pass. The morphological filters were chosen to reduce spatially small noise components. Noise comparable to or bigger in size to the object is handled partially in the selection of the Region of Interest (section 3.2.1.4) and partially by the buffering system in the tracking algorithm (section 3.2.3.3).

3.2.3.3 Tracking System

The tracking system implemented is a Mealy finite state machine with three definite states: a buffer state, an active state, and an inactive state. The diagram is shown in Figure 19: Tracking system state diagram.

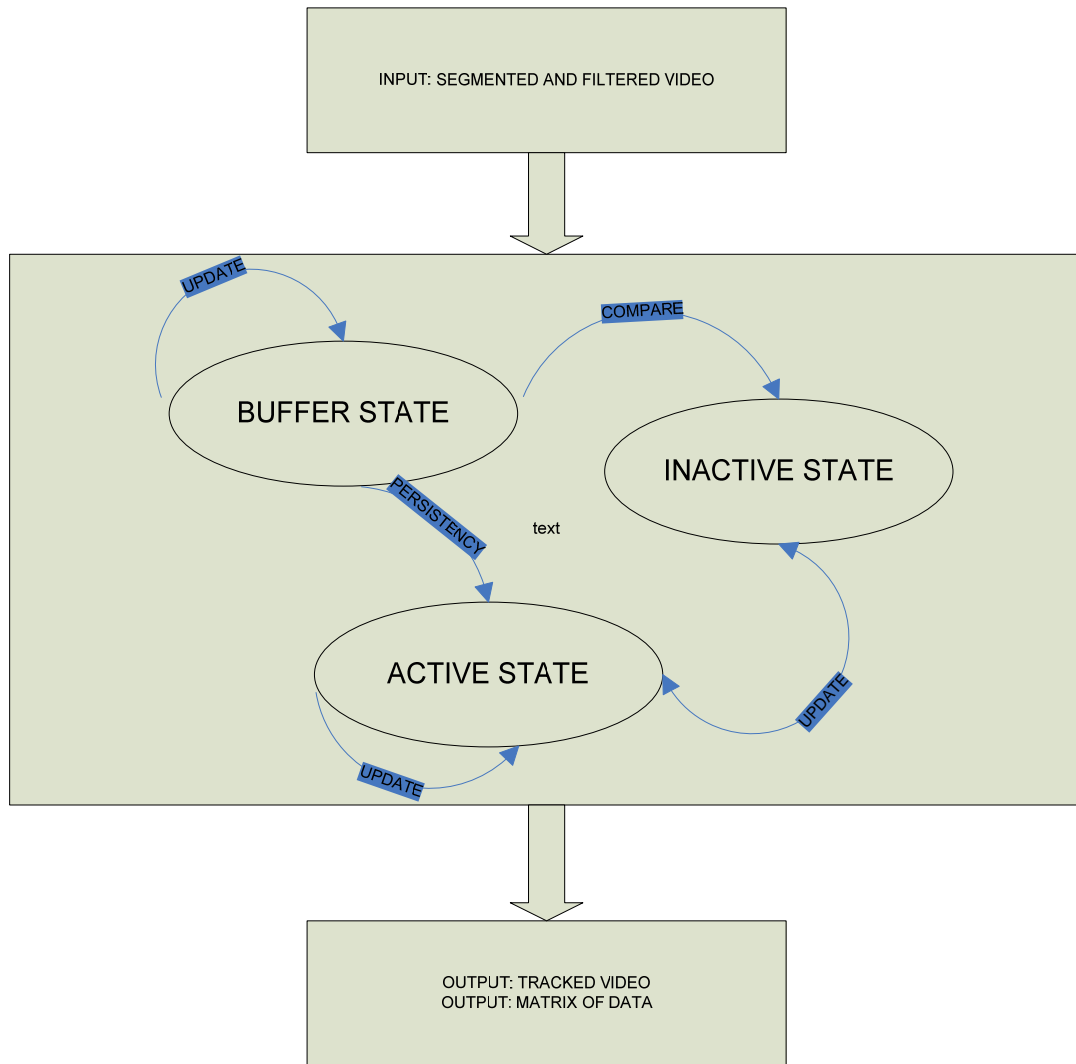


Figure 19: Tracking system state diagram.

- *Buffer State:* When a new object is detected, the buffer state keeps track of the object in the first three frames; this is done to avoid the appearance of ghost objects

The buffer state saves system resources by allowing the FSM to keep track only of persistent objects in the video. When the object has been in the buffer state

for three consecutive frames, its information is compared with that of the Inactive State to check if the new object is in fact an old object that previously disappeared due to an occlusion. If no object in the inactive list is comparable to the new object, then the object is labeled as a new object and its information is transferred to the Active State.

- *Active State:* The active state keeps track of the objects while they are present in the video and after they have passed the buffer state. The active state keeps track of the centroid position, past centroid positions, and the index for each of the pixels that compose the object.

If an active object disappears in the middle of the video, the ID of the object is stored in the Inactive State. and the Active State stops tracking it until the buffer state finds a match between a new object that appeared in the middle of the video and the stored inactive object. When that happens, the buffer state transfers the information to the Active State and the tracking is resumed.

- *Inactive State:* The inactive state is the only state of the system where information about the physical properties of the object is not stored or generated. Instead, it keeps a list of IDs or pointers of the objects that were being tracked by the Active State and that disappeared in the middle of the video, probably because of an occlusion.

When an object is ready to go out of the buffer state, the inactive state sends the ID to the buffer state where a comparison is made to check whether or not the new object is in fact an inactive object reappearing.

To determine if an object in the current frame is the updated version of an object being tracked, the first step is to create an extended bounding box around the object being tracked and check for centroids of objects inside this region in the current frame. If there is only one object in that region, then it is considered a match and the information for that object is updated accordingly. If, on the other hand, there are more than one object inside the region, the system compares the object sizes of the candidates with that of the previous frame to decide which one is a match. Lastly, if there are no matches, the object is either discarded or transferred to the inactive state if it has been a persistent object. An object is considered persistent when it has been in the active list for some minimum number of frames. The process can be seen in Figure 20: Object tracking based on bounding box and centroid position..

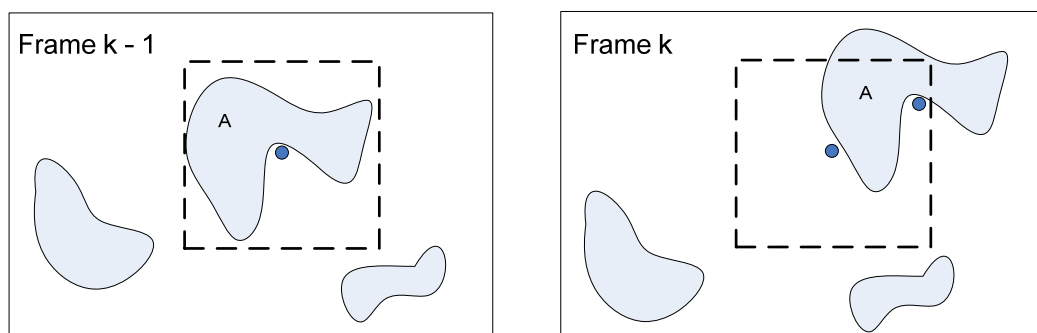


Figure 20: Object tracking based on bounding box and centroid position.

3.2.4 Postprocessing

3.2.4.1 Block Diagram

The post processing stage organizes the data obtained from the data extraction stage and presents it to the user. The data generated is a video presentation, which is a visualization of the results, and a cell containing all the information (centroid, bounding box, pixels coordinates and instantaneous velocity) from each object tracked by the system along the frames where the object was present. The block diagram is as follows:

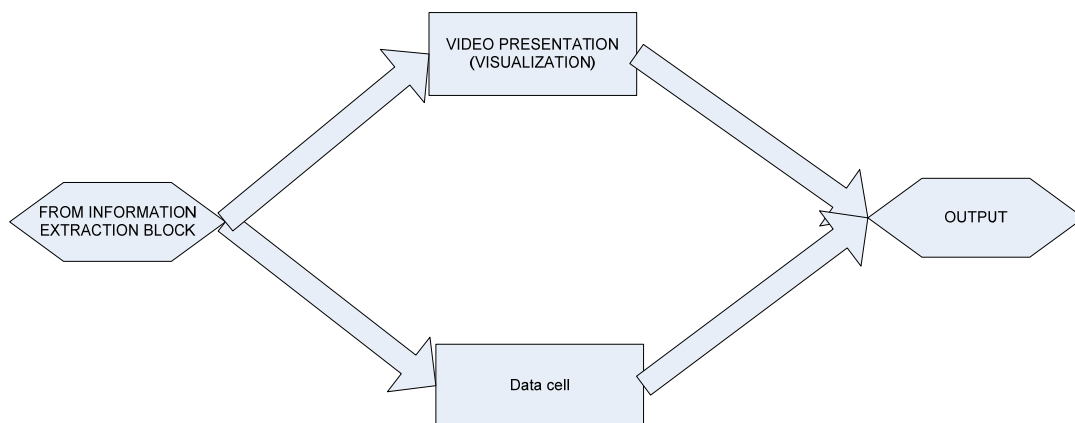


Figure 21: Postprocessing Block Diagram.

3.2.4.2 Video Presentation

The video presentation generates an output video that is like the original video but with the objects detected being circled and the trajectory highlighted, The video presentation does not present exact data but it gives a good idea of how the system is behaving. It is also an ideal early alarm system telling the user where the activity is in the video so that the user can understand the data from the cell.

3.2.4.3 Data Cell

A cell is a matrix where each of its elements is of different nature (e.g., one of the elements is a vector, another one is a matrix, another is a string of characters, etc). The cell generated by the program stores information from new objects such as the frame in which it appeared, the history of the position of the centroid and the list of pixels of the object, the bounding box information, and the instantaneous velocity.

Chapter 4 IMPLEMENTATION

MATLAB Version 7.6 and its Image Processing Toolbox (IPT) Version 6.1 were the main tools used to implement the algorithms. Four MATLAB functions were created for the system:

- Function `preproc.m`: Implements all the preprocessing operations from data acquisition up until region of importance analysis.
- Function `bgsub.m`: Implements the background subtraction selecting between three different types of algorithms. It is a part of the background modeling block.
- Function `morfil.m`: Implements the morphological filtering. It is .part of the background modeling block.
- Function `Tracksys.m`: Implements the tracking system, data analysis, and the post processing operations.

The following is the flowchart of the functions with their respective inputs, outputs, and the block to which each of them belongs.

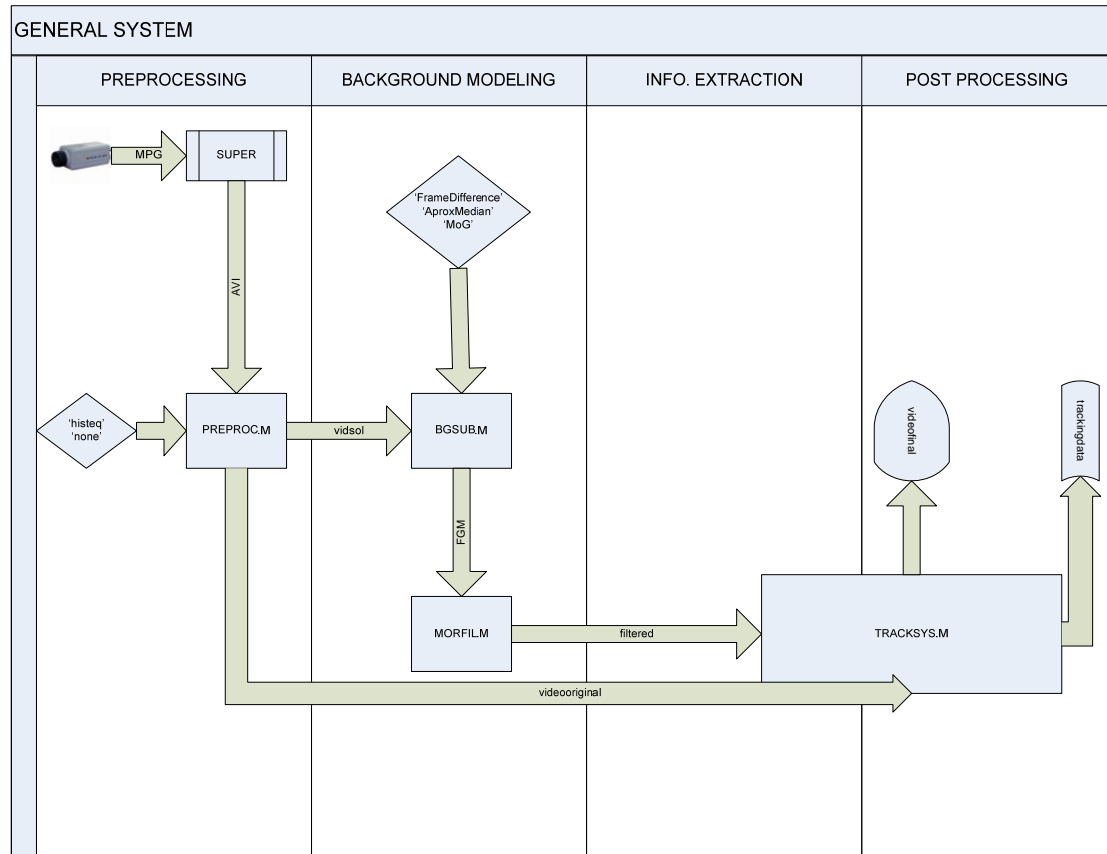


Figure 22: Functional flowchart.

4.1 Preprocessing

4.1.1 Data Acquisition

Before running the program the video should be in AVI format, so if the original video is in another format such as MPG, WMV or MOV a conversion is needed prior to the operation. The freeware Simplified Universal Player Encoder and Renderer (SUPER©) from eRightSoft [31] were used when it was necessary to convert formats.

From the AVI file, information is extracted using the aviinfo function from the IPT. If the video is in indexed color format it is first converted to RGB format using IPT function ind2rgb. With the function aviread, the RGB AVI video is then stored in a 4-D matrix structure named videooriginal with dimensions height, width, number of frames, and a fourth dimension of magnitude three for storing separately the R, G and B component of the video.

```
videoinfo=aviinfo('C:\MATLAB\R2006a\work\TESIS\prueba.avi');  
switch videoinfo.ImageType  
    case 'truecolor'  
video=aviread('C:\MATLAB\R2006a\work\TESIS\prueba.avi');  
    case 'indexed'  
video=aviread('C:\MATLAB\R2006a\work\TESIS\prueba.avi');  
video=ind2rgb(video);  
end
```

4.1.2 Color to Grayscale Converter

The conversion from color to grayscale is performed by the IPT function rgb2gray, and the grayscale version of the video is then stored in a 3-D matrix named videogray.

```
videogray=zeros(videoinfo.Height,videoinfo.Width,videoinfo.NumFrames,'  
uint8');
```

```

for n=1:videoinfo.NumFrames
    videogray(:, :, n)=rgb2gray(video(n).cdata);
end

```

4.1.3 Region of Importance

The mask of the Region of Importance is generated using function `roipoly` from the IPT.

As a sample to be displayed to the user, the program shows the second frame.

```
baseimage=videogray(:, :, 2);
```

The user is then prompted to draw a polygon with the mouse around the ROI. After the enter key is pressed the user is asked whether another ROI is needed for the image or not. This is because some images may have different ROIs that are not connected.

```

baseimage=videogray(:, :, 2);
[MASK R C]=roipoly(baseimage);
question='y';
while question=='y'
    clc
    question=input('Do you want to declare another region of interest?
(Y/N)\n', 's');
    if question=='y'
        [MASKTEM RTEM CTEM]=roipoly(I);
        MASK=MASK|MASKTEM;
    elseif question=='n'
    else
        error('wrong input')
    end
end
end

```

When the user tells the program that there are no more ROIs in that particular video, the MASK is then created and used to remove the unimportant regions. The result is stored in a 3-D Matrix named vidsol, which is the output of the preprocessing block.

```
MASKuint8=uint8(MASK);
vidsol=zeros(videoinfo.Height,videoinfo.Width,videoinfo.NumFrames,'uint8');
for z=2:videoinfo.NumFrames
    vidsol(:,:,z-1)=videogray(:,:,z).*MASKuint8;
end
```

4.1.4 Image Enhancement

Function histeq performs histogram equalization in the video (see Section 2.2.2.1), In order to take real advantage of the function, there are some previous considerations to take into account.

As seen in Section 4.1.3, the regions of importance are usually small regions where the analysis is in detail and hence where the enhancement of the contrast is most desired: the unimportant regions add pixels of different intensities; to take the histogram equalization over the whole scene could have the contrary effect of contrast enhancement. Take for example the scene depicted in Figure 23: Scene where histogram equalization over the whole picture will have an undesired effect.:



Figure 23: Scene where histogram equalization over the whole picture will have an undesired effect.

The region of importance is inside the red box. Since the surrounding is visibly darker than the region of importance, a histogram equalization over the whole scene would enhance contrast of the whole picture leading to less contrast in the region of importance due to the histogram compensating for the dark region.

To avoid this, `preproc.m` takes the values inside the region of importance and extracts the mean over those values; it then assigns the mean value to the unimportant region. This is done by creating an `ANTIMASK` (the negative of the `MASK` obtained in 4.1.3) and multiplying it by the mean and storing it in variable `meanmask`. The system then adds the `meanmask` to the `vidsol` obtained from the previous section and stores the new video in variable `vidhis`. Finally, `histeq` is performed over `vidhis`, and the video is once again masked and stored in variable `vidsol`.

```

ANTIMASK=~MASK;
[I,J,V] = find(vidsol(:,:,2));
meanbck=floor(mean(V));
ANTIMASKuint8=uint8(ANTIMASK);
meanmask=ANTIMASKuint8*meanbck;

for z=1:videoinfo.NumFrames
    vidhis(:,:,z)=vidsol(:,:,z)+meanmask;
end

for n=1:videoinfo.NumFrames
    vidhiseq(:,:,n)=histeq(vidhis(:,:,n));
end

for n=1:videoinfo.NumFrames
    vidsol(:,:,n)=vidhiseq(:,:,n).*MASKuint8;
end

```

4.2 Background Subtraction

The Background subtraction block (selector and implementation) are all executed simultaneously.

4.2.1 Algorithm Selection

The background subtraction algorithm is implemented in the MATLAB function `bgsub`.

The selection of the algorithm is made in the calling of the function.

```
function [FGM time] = bgsub(vidsol,varargin),
```

where FGM is the output video with the background pixels at zero value and the foreground pixels at one, and time stores how many seconds the program took to analyze the video. The different options to call `bgsub` are:

- `bgsub(vidsol, 'FrameDifference', threshold)`: Uses the frame difference algorithm to analyze the video `vidsol`. The variable `threshold` set the comparison parameter of the frame difference algorithm (see Sections 2.2.3 and 3.2.2.3)
- `bgsub(vidsol, 'ApproxMedian', threshold)`: Uses the approximate median algorithm to analyze the video `vidsol`. The variable `Threshold` sets the comparison parameter similar to the frame difference method (see Sections 2.2.3 and 3.2.2.3)
- `bgsub(vidsol, 'MoG', threshold, components, sdthreshold, alpha, initials)`: Uses the Mixture of Gaussians algorithm. `Threshold` is the comparison parameter, `components` are the number of Gaussian components (typically 3 to 5), `sdthreshold` is the positive deviation threshold, `alpha` is the learning rate, and `initials` is the initial standard deviation value (see the set of equations 8 in Section 4.2.2.3)

If the parameters are not specified so that only `vidsol` and `method` are given, the system takes the default values specified in Section 4.2.2.3.

4.2.2 Algorithm implementation

4.2.2.1 Frame Difference

The first frame is used as the base background and stored in the `bg` variable. The threshold is set to 70 and temporary processing variables are declared.

```
thr=70;
bg=vidsol(:,:,1);
[Height Width Numframes]=size(vidsol);
fg = zeros(Height, Width);
FGMfd=zeros(Height, Width ,Numframes-1);
```

The frame is then compared with `bg` pixel by pixel. If the absolute difference between the pixels is above the threshold the pixel is set to 255, otherwise the pixel is set to 0. After the image for that particular frame is created, `bg` is set to take the value of the current frame and the process starts again.

```
for n = 2:Numframes
    diframes = abs(double(vidsol(:,:,n)) - double(bg));
    for m=1:Width
        for l=1:Height
            if ((diframes(l,m) > thr))
                fg(l,m)=255;
            else
                fg(l,m) = 0;
            end
        end
    end
    bg=vidsol(:,:,n);
    FGMfd(:,:,n-1) = uint8(fg);
end
```


At the end the foreground is stored in the 3-D Matrix FGMfd.

4.2.2.2 Approximate Median

The approximate median follows the same steps as the frame difference method. The difference is that bg is not replaced for the current frame at the end. Threshold is set in 50 in this case.

```
thr=50;
bg=vidsol(:,:,1);
[Height Width Numframes]=size(vidsol);
fg = zeros(Height, Width);
BGM=zeros(Height, Width ,Numframes-1);
FGMam=zeros(Height, Width ,Numframes-1)
```

Instead of merely replacing the entire bg with the current frame, the first frame is used as a model and after the frame difference comparison, if the value for a particular pixel in the current frame is greater than the stored value in bg, the pixel in bg is updated by increasing its model value by 1. If, on the other hand, the value of the intensity of a particular pixel in the current frame is less than its bg model counterpart, then the bg model pixel intensity is decreased by 1. At the end the foreground is stored in the 3-D Matrix FGMam and background in the 3-D Matrix BGM.

```
for n = 2:Numframes
    diframes = abs(double(vidsol(:,:,n)) - double(bg));
    diframes = uint8(diframes);
    for m=1:Width
        for l=1:Height
            if ((diframes(l,m) > thr))
```

```

        fg(1,m) = 255;
    else
        fg(1,m) = 0;
    end
    if (vidsol(1,m,n) > bg(1,m))
        bg(1,m) = bg(1,m) + 1;
    elseif (vidsol(1,m,n) < bg(1,m))
        bg(1,m) = bg(1,m) - 1;
    end
end
end
FGMam(:, :, n-1) = fg;
BGM(:, :, n-1) = bg;
end

```

4.2.2.3 Mixture of Gaussians

As stated in section 2.2.3, for a Mixture of Gaussians algorithm to succeed, three to five Gaussian components are needed. In the system implemented for this work, three components were implemented mainly because of computational reasons.

The initialization of variables is as follows; the values were chosen according to [21] with slight trial and error adjustments.

```

Components = 3;           % number of gaussian
components
M = 3;                   % number of components
Dev = 2.5;               % positive deviation thr.
alpha = 0.01;           % learning rate
threshold = 0.25;       % foreground threshold
initialsd = 6;          % initial standard deviation
w = zeros(Height,Width,Components); % initialize weights array
mean = zeros(Height,Width,Components); % pixel means
sd = zeros(Height,Width,Components); % pixel standard deviations
diffframes = zeros(Height,Width,Components); % difference of each pixel
p = alpha/(1/Components); % initial p variable
rank = zeros(1,Components); % rank of components

```

The mean 3-D matrix is initialized with random numbers between 1 and 255, the 3-D weight matrix is initialized with 1/3 for each component, and the standard deviation 3-D matrix is initialized with the initial value of 6.

```

for i=1:Height
    for j=1:Width
        for k=1:Components
            mean(i,j,k) = rand*range;    % means random (0-255)
            w(i,j,k) = 1/Components;    % weights uniformly dist
            sd(i,j,k) = initialsds;    % initialize to
        initialsds
        end
    end
end

```

The frame difference operation is similar to the other algorithms implemented in the system, but this time, diffframes is a 3-D matrix of three components instead of the 2-D matrix of the other methods.

```

for m=1:Components
    diffframes(:,:,m) = abs(vidsoltemp - double(mean(:,:,m)));
end

```

For each pixel, if the absolute value of the difference is less than the positive deviation threshold multiplied by the standard deviation, there is a component match and the weights and standard deviation matrixes are updated as follows:

$$\begin{aligned}
 w &= (1 - \alpha)w + \alpha \\
 p &= \alpha / w \\
 \mu &= (1 - p)\mu + (p)pixel \\
 \sigma &= \sqrt{(1 - p)\sigma^2 + p(pixel - \mu)^2}
 \end{aligned}
 \tag{11}$$

with α the learning rate, μ the mean, σ the standard deviation, pixel the current pixel value, and w the weight.

If, on the other hand, the absolute value of the difference is more than the positive deviation threshold multiplied by the standard deviation, there is a no match and only the weight matrix is decreased, as follows:

$$w = (1 - \alpha)w \quad (11)$$

```

ComponentMatch = 0;
for k=1:Components
    if (abs(diffframes(i,j,k)) <= Dev*sd(i,j,k))
        ComponentMatch = 1;
        w(i,j,k) = (1-alpha)*w(i,j,k) + alpha;
        p = alpha/w(i,j,k);
        mean(i,j,k) = (1-p)*mean(i,j,k) + p*vidsoltemp(i,j);
        sd(i,j,k) = sqrt((1-p)*(sd(i,j,k)^2) +
p*(vidsoltemp(i,j) ... - mean(i,j,k)).^2);
    else
        w(i,j,k) = (1-alpha)*w(i,j,k);
    end
end

```

The weights are normalized over the three components and the weight matrix is updated. The background is then updated by the mean multiplied by the weight.

```

w(i,j,:) = w(i,j,:)./sum(w(i,j,:));
bg_bw(i,j)=0;
for k=1:Components
    bg_bw(i,j) = bg_bw(i,j)+ mean(i,j,k)*w(i,j,k);
end

```

If there is no match, a new Gaussian component is created with the mean just obtained and the initial standard deviation.

```

if (match == 0)
    [min_w, min_w_index] = min(w(i,j,:));
    mean(i,j,min_w_index) = double(fr_bw(i,j));
    sd(i,j,min_w_index) = initialsd;
end

```

The confidence that the algorithm has that a given pixel is in the background is reflected by larger associated weights and smaller associated standard deviations. In this case w/σ will give a good measure of how confident the guess is. In order to model the new background we organize the component from the most confident to the least confident and take the first M components (M could vary and depends on computational resources).

```

rank = w(i,j,:)./sd(i,j,:);
rank_ind = [1:1:Components];
for k=2:Components
    for m=1:(k-1)
        if (rank(:, :,k) > rank(:, :,m))
            rank_temp = rank(:, :,m);
            rank(:, :,m) = rank(:, :,k);
            rank(:, :,k) = rank_temp;
            rank_ind_temp = rank_ind(m);
            rank_ind(m) = rank_ind(k);
            rank_ind(k) = rank_ind_temp;
        end
    end
end

```

The foreground is then extracted by comparing it with the background model.

```

fg(i,j) = 0;
while ((match == 0)&&(k<=M))
    if (w(i,j,rank_ind(k)) >= threshold)
        if (abs(diffframes(i,j,rank_ind(k))) <=
Dev*sd(i,j,rank_ind(k)))
            fg(i,j) = 0;
            match = 1;
        end
    end
end

```

```

        else
            fg(i,j) = fr_bw(i,j);
        end
    end
    k = k+1;
end
end

```

The result is stored in 3-D Matrix FGM.

4.3 Information Extraction

4.3.1 Morphological Filtering

For the morphological filtering, using IPT function `bwmorph`, we perform several tasks.

First we remove isolated foreground pixels with ‘clean,’ isolated background pixels with ‘holes’ and ‘majority,’ and connect adjacent pixels with ‘bridge’:

```

filtered(:,:,n)= bwmorph(FGMBW(:,:,n), 'clean');
filtered(:,:,n) = bwmorph(filtered(:,:,n), 'bridge');
filtered(:,:,n) = bwmorph(filtered(:,:,n), 'close');
filtered(:,:,n) = bwmorph(filtered(:,:,n), 'majority');
filtered(:,:,n) = imfill(filtered(:,:,n), 'holes');

```

Finally we perform an opening operation using the 3-pixel shaped disk as the structuring element.

```

se=strel('diamond',3);
filtered(:,:,n)=imopen(FGMBW(:,:,n),se);

```

4.3.2 Tracking System

The tracking system and the post processing block are implemented in the MATLAB function `tracksys.m`. The call for such a function is as follows:

```
[videofinal trackingdata] = tracksysCHANGES(videooriginal,filtered)
```

with `videooriginal` being the original color video, `filtered` being the output from the morphological filtering block, and `videofinal` and `trackingdata` being the video presentation and the cell as described in Section 3.2.4.3.

The global variables and initial states of the most important variables of the function are as follows:

```
buffer=3;
height=tempsize(1);
width=tempsize(2);
persistence=9;
externalbox = [ hormargin width-hormargin vermargin height-
vermargin ];
labeledframe=zeros(height,width);
activeIDlist=[];
inactiveIDlist=[];
bufferlist=[];
trackedlist=[];
```

These variables are important for the rest of the algorithm; Table 1 shows the purpose of each of the variables.

NAME	FUNCTION
buffer	Determines the size of the buffer (default in 3).
height	Height of the video sequence.
width	Width of the video sequence.
persistency	The minimum number of frames an object should be being tracked before it can go to the inactive state.
externalbox	Is a margin created across the video in order to distinguish when an object appeared “in the middle” of the video (e.g. because it was being occluded or when it appeared because it just entered the sight of the camera.
labeledframe	Contains the information about the label of the objects for the current frame
currentframestats	Contains the statistical information about each object from the current frame.
activeIDlist	Keeps the list of pointers or IDs to each object being tracked
inactiveIDlist	Keeps the list of pointers or IDs to each object that disappeared inside the externalbox
bufferlist	Keeps all the information from the objects while they are still in the buffer state.
trackedlist	Keeps all the information from the objects that had being or are being tracked.

Table 1: Tracking System variables.

The system extracts the information from the current frame and stores it in the cell labeledframe. The information is obtained by first extracting the number of objects present in the frame with the IPT function bwlabel and then extracting the centroid, the list of pixels, and the bounding box with the IPT function regionprops.

```
labeledframe=bwlabel(filtered(:,:,currentframe));
currentframestats =
regionprops(labeledframe, 'Centroid', 'PixelIdxList', 'BoundingBox');
```

The extended bounding box is created from the bounding box extracted from the regionprops with a 10 pixel extension in all directions. This is done because we are dealing with very small objects and hence with very small bounding boxes. If the

bounding box is small then objects moving fast would have their centroids outside the region.

```
currentboundingbox = trackedlist(currentID).boundingbox;
centroidrow = currentframestats(newobjectnumber).Centroid(2);
centroidcolumn = currentframestats(newobjectnumber).Centroid(1);
largerbox=zeros(1,4);
    if currentboundingbox(1)>10
        largerbox(1)=currentboundingbox(1)-10;
    else
        largerbox(1)=0;
    end
    if currentboundingbox(2)>10
        largerbox(2)=currentboundingbox(2)-10;
    else
        largerbox(2)=0;
    end
    if largerbox(1)+currentboundingbox(3)+20<width
        largerbox(3)=largerbox(1)+currentboundingbox(3)+20;
    else
        largerbox(3)=width;
    end
    if largerbox(2)+currentboundingbox(4)+20<height
        largerbox(4)=largerbox(2)+currentboundingbox(4)+20;
    else
        largerbox(4)=height;
    end
```

All the candidates are obtained by comparing the centroid of the objects with the extendedbox. Each candidate ID is stored in a temporal vector called possiblematch.

```
    if centroidrow >= largerbox(2) && ...
        centroidrow <= (largerbox(2) + largerbox(4)) && ...
        centroidcolumn >= largerbox(1) && ...
        centroidcolumn <= (largerbox(1) + largerbox(3))
        possiblematch = [possiblematch newobjectnumber];
    end
```

The decision-making process of finding the match between the current object and the objects being tracked as explained in Section 3.2.3.3 is implemented as follows.

```

case 0
if size(trackedlist(currentID).Centroids,1)>persistency && ...
    centroidrow >= externalbox(1) && ...
    centroidrow <= externalbox(2) && ...
    centroidcolumn >= externalbox(3) && ...
    centroidcolumn <= externalbox(4)
    inactiveIDlist=[inactiveIDlist currentID];
else
    removefromactive = [removefromactive existentobjectnumber];
end
case 1
    matchID = possiblematch;
    trackedlist(currentID).Centroids =
[trackedlist(currentID).Centroids;
currentframestats(matchID).Centroid];
    trackedlist(currentID).prevCentroid =
currentframestats(matchID).Centroid;
    trackedlist(currentID).boundingbox =
currentframestats(matchID).BoundingBox;
    trackedlist(currentID).PixelIdxList =
currentframestats(matchID).PixelIdxList;
    currentframestats(matchID) = [];
otherwise
    trackedobjectsize = length(trackedlist(currentID).PixelIdxList);
    sizedifference = [];
for candidatenum = 1:length(possiblematch)
    currentobjectsize =
length(currentframestats(possiblematch(candidatenum)).PixelIdxList)
;
    sizedifference(candidatenum) = abs(trackedobjectsize -
currentobjectsize);
end
matchID = possiblematch(find(sizedifference == min(sizedifference)));
matchID = matchID(1);
%...Repeat steps from case 1
End

```

Note that when there are no candidates for a match, implying that the object disappeared from the scene, the system checks to determine whether the object was inside the externalbox when it disappeared. If so, it sets the pointer information to the inactive state.

After all the objects from the active state have been attended, the next step is to track the objects from the buffer state. The way the system tracks objects in the buffer state is

similar to the way just showed for the active state with the exception that no pointer is ever sent to the inactive state; instead when an object disappears it is erased from the buffer.

When an object has been in the buffer for three frames and its initial centroid was inside the externalbox, then the object is compared with the inactive list in size and distance from the last known location, and then a decision is made. If there is no match with the inactive list then the object is updated in the active list as a new one.

```

if centroidcolumnb >= externalbox(1) && ...
centroidcolumnb <= externalbox(2) && ...
centroidrowb >= externalbox(3) && ...
centroidrowb <= externalbox(4) && ...
length(inactiveIDlist)~=0
distance=[];
  for currentinactive = 1:length(inactiveIDlist)
      currentdistance=sqrt((centroidrowb-
          trackedlist(inactiveIDlist(currentinactive)).prevCentroid(
              2))^2+ ...
          (centroidcolumnb-
              trackedlist(inactiveIDlist(currentinactive)).prevCentroid(1))^2);
      distance=[distance currentdistance];
  end
  [minvalue index]=min(distance);
  activeIDlist=[activeIDlist inactiveIDlist(index)];
  trackedlist(inactiveIDlist(index)).Centroids =
  [trackedlist(currentID).Centroids;
  bufferlist(bufferobjectnumber).Centroids];
      trackedlist(inactiveIDlist(index)).prevCentroid =
  bufferlist(bufferobjectnumber).prevCentroid;
      trackedlist(inactiveIDlist(index)).boundingbox =
  bufferlist(bufferobjectnumber).boundingbox;
      trackedlist(inactiveIDlist(index)).PixelIdxList =
  bufferlist(bufferobjectnumber).PixelIdxList;
      removefrominactive = [removefrominactive index];
      removefrombufferlist = [removefrombufferlist
  bufferobjectnumber];
else
  nextID = length(trackedlist) + 1;
  %...Fill information the same way as in the other part of the if
  statement.
end

```

The remaining objects from labeledframe not yet identified are then new objects and handled in the following statement.

```
    for remaininobjectnumber = 1:length(currentframestats)
        bufferlist(length(bufferlist)+1).Centroids =
currentframestats(remaininobjectnumber).Centroid;
        bufferlist(length(bufferlist)).prevCentroid =
currentframestats(remaininobjectnumber).Centroid;
        bufferlist(length(bufferlist)).boundingbox =
currentframestats(remaininobjectnumber).BoundingBox;
        bufferlist(length(bufferlist)).PixelIdxList =
currentframestats(remaininobjectnumber).PixelIdxList;
    end
```

4.4 Post processing

4.4.1 Video Presentation

At the beginning of the tracksys function videofinal is equated to videooriginal. After each frame is analyzed, videofinal is separated in its R,G, and B components.

```
R = videofinal(:,:,1,currentframe);
G = videofinal(:,:,2,currentframe);
B = videofinal(:,:,3,currentframe);
```

The trail is drawn in videofinal by connecting the different centroids using the function func_drawline [32]. The function func_drawline connects two points with a gray line. By manipulating each color separately we can choose a color for each object detected.

```

currentnr = round(trackedlist(currentID).Centroids(centroidnumber,2));
currentnc = round(trackedlist(currentID).Centroids(centroidnumber,1));
nextnr = round(trackedlist(currentID).Centroids(centroidnumber + 1,2));
nextnc = round(trackedlist(currentID).Centroids(centroidnumber + 1,1));
colorselector = mod(currentID,3);
switch colorselector
    case 0
        R = func_Drawline(R,currentnr,currentnc,nextnr,nextnc,255);
        G = func_Drawline(G,currentnr,currentnc,nextnr,nextnc,0);
        B = func_Drawline(B,currentnr,currentnc,nextnr,nextnc,0);
    case 1
        R = func_Drawline(R,currentnr,currentnc,nextnr,nextnc,0);
        G = func_Drawline(G,currentnr,currentnc,nextnr,nextnc,255);
        B = func_Drawline(B,currentnr,currentnc,nextnr,nextnc,0);
    otherwise
        R = func_Drawline(R,currentnr,currentnc,nextnr,nextnc,0);
        G = func_Drawline(G,currentnr,currentnc,nextnr,nextnc,0);
        B = func_Drawline(B,currentnr,currentnc,nextnr,nextnc,255);
end

```

The system also draws the bounding box. The corners of the boundingbox are found as follows:

```

r1 = floor(trackedlist(currentID).boundingbox(2));
c1 = floor(trackedlist(currentID).boundingbox(1));
r2 = floor(trackedlist(currentID).boundingbox(2) +
trackedlist(currentID).boundingbox(4));
c2 = floor(trackedlist(currentID).boundingbox(1) +
trackedlist(currentID).boundingbox(3));

```

Then the bounding box is drawn by connecting (c1,r1) with (c1,r2), (c1,r2) with (c2,r2), (c2,r2) with (c2,r1), and (c2,r1) with (c1,r1).

```

R = func_Drawline(R,r1,c1,r1,c2,255);
R = func_Drawline(R,r1,c2,r2,c2,255);
R = func_Drawline(R,r2,c2,r2,c1,255);
R = func_Drawline(R,r2,c1,r1,c1,255);

```

4.4.2 Data Cell

The Data cell with all the information of the objects is called trackingdata. The information is obtained from the trackedobject and activeIDlist cells.

```
currentID = activeIDlist(existentobjectnumber);
trackingdata(currentframe).object(existentobjectnumber).ID =
currentID;
trackingdata(currentframe).object(existentobjectnumber).boundingbox =
trackedlist(currentID).boundingbox;
trackingdata(currentframe).object(existentobjectnumber).currentpoint =
trackedlist(currentID).prevCentroid;
```

The instantaneous velocity is found by finding the distance between two contiguous centroids.

```
numberofcentroids = size(trackedlist(currentID).Centroids,1);
currentcentroid = trackedlist(currentID).prevCentroid;
previouscentroid = trackedlist(currentID).Centroids(numberofcentroids-
1,:);
trackingdata(currentframe).object(existentobjectnumber).velocity =
round(norm(currentcentroid - previouscentroid));
```

Chapter 5 EXPERIMENTS AND RESULTS

In this chapter, a summary of the results that were obtained is provided. Three videos were fully analyzed, the first one with a person walking through multiple occlusions, the same scene with the person running instead of walking, and a different scenario with two persons present and one occlusion. The scenes present manmade features such as walls and sidewalks and natural features such as sunlight and trees.

All the programs were tested on two computers. The first computer is an HP-Pavilion dv5-1004nr Laptop PC with 2.1GHz AMD Turion X2 Mobile ZM-80 processor, 4,096MB DDR2 SDRAM 667MHz, ATI Mobility Radeon HD 3200 video card, and Windows Vista Home Premium Edition. The version of MATLAB in that computer is Version 7.6.0.324 (R2008a), and the Image Processing Toolbox (IPT) installed is Version 6.1. The second computer is a Dell Precision Workstation 650 with Intel Xeon Dual Processor (3.06 GHz and 3.2 GHz respectively), 4GB of SDRAM, 128MB nVidia QuadroFX 1000 video card and Windows XP SP2 Professional. The version of MATLAB in that computer is Version 7.2.0.232 (R2006a) and the IPT installed is Version 5.2.

5.1 Preprocessing

5.1.1 Data Acquisition and grayscale conversion

When using the `aviread` function of the IPT, it is important to have in mind that AVI files can be created from a variety of compression codecs. In order for `aviread` to work properly, it is necessary to make sure that the proper codec is installed on the PC. The codec can be found with the `aviinfo` function.

MATLAB has a serious limitation in memory usage, so the maximum size of the video that can be handled depends on the machine running the program. The Laptop could handle videos of up to 800 frames, while the workstation could handle videos of up to 600 frames. The test videos had 314, 311 and 235 frames respectively.

5.1.2 Region of Importance

Figure 24: Example of selection of region of importance. shows a typical surveillance scene, with multiple paths, occlusions, and shades. The objective of the system is to detect objects that are near the building at the end of the scene in the orange and red regions.



Figure 24: Example of selection of region of importance.

From the scene, it is clear that the majority of the video provides no information to the task. Instead, it can be an important source of noise. As can be seen, outside the region of importance there are shadows and vegetation that could affect negatively the background subtraction algorithm.

The preproc function prompts the user to select the regions of importance with the mouse; after their selection a mask of the regions selected is applied. Note that the system has the capability of allowing multiple regions even though they are not joined. Also note that the regions do not have to be rectangular but can be polygonal.

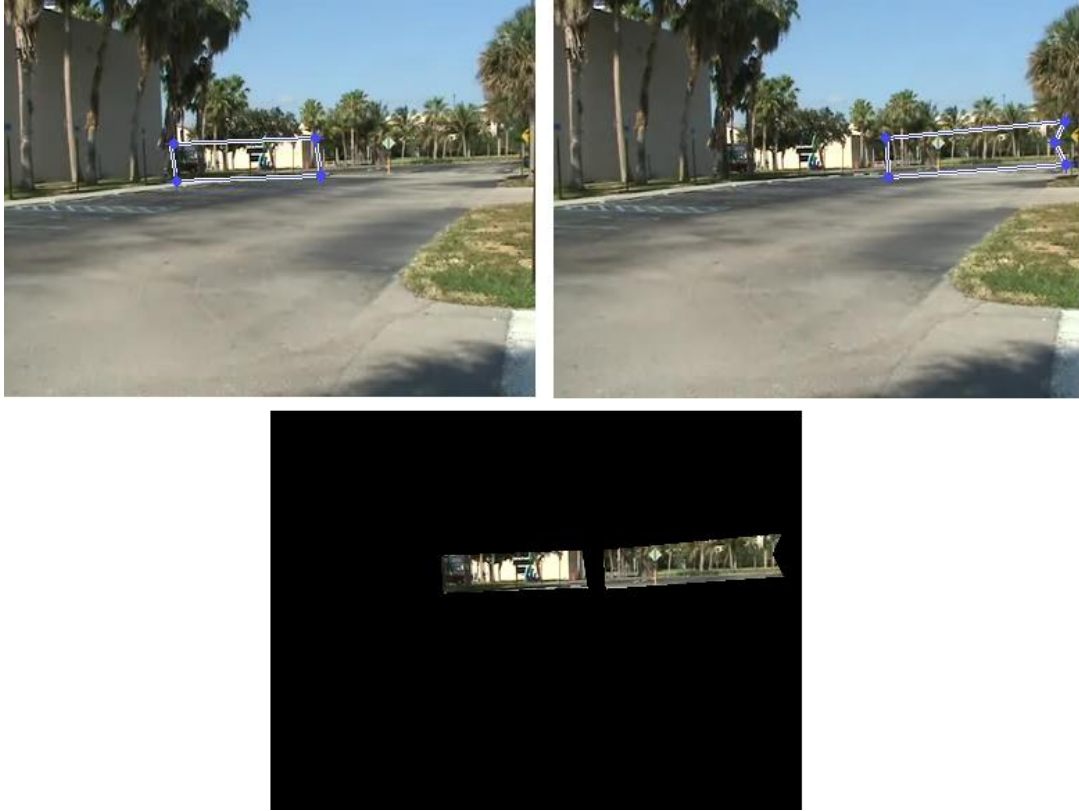


Figure 25: Selection of the region of importance

5.1.3 Image Enhancement

The following test image was used to prove the contrast enhancing capability of the histogram equalization.



Figure 26: Contrast enhancement test image

The image is divided in four quadrants each of them is painted with a shade of green in the value from 100 to 104. The MATLAB script `testcontrast` reads the image and then performs the histogram equalization. The output shows us the input test image and its histogram and the output contrast-enhanced image and its histogram. The result can be seen in the following Figure.

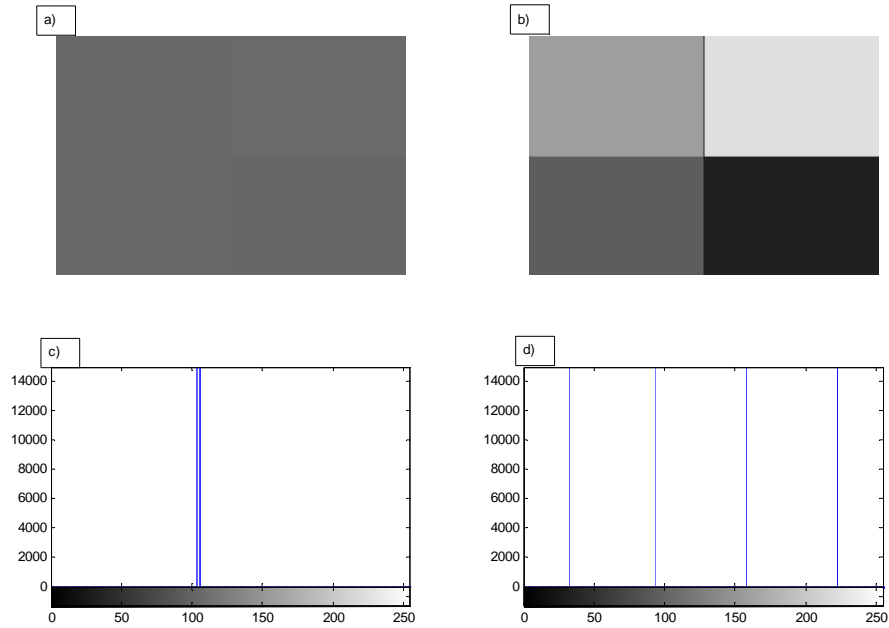


Figure 27: Contrast enhancement test. a) Input image, b) output image, c) histogram of the input image, d) histogram of the output.

Consider the following figure that shows a picture of the surface of the moon with poor contrast and its respective histogram:

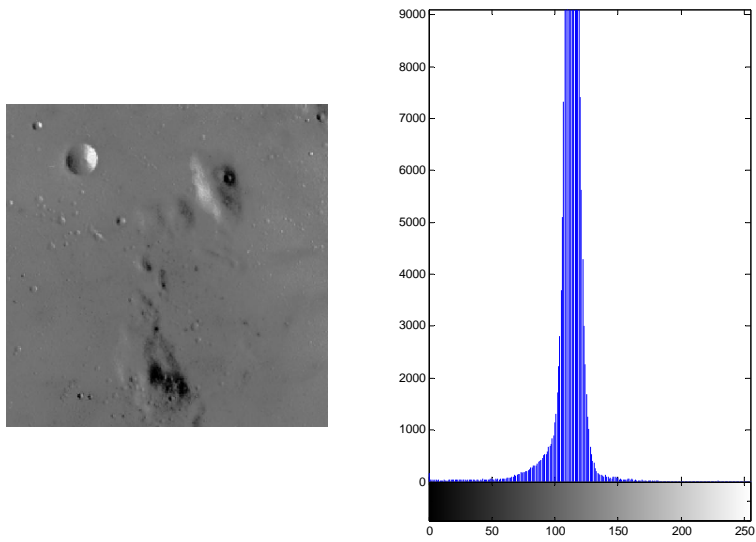


Figure 28: Moon surface with poor contrast (image and histogram).

After the histogram equalization, the change in the level of detail is evident.

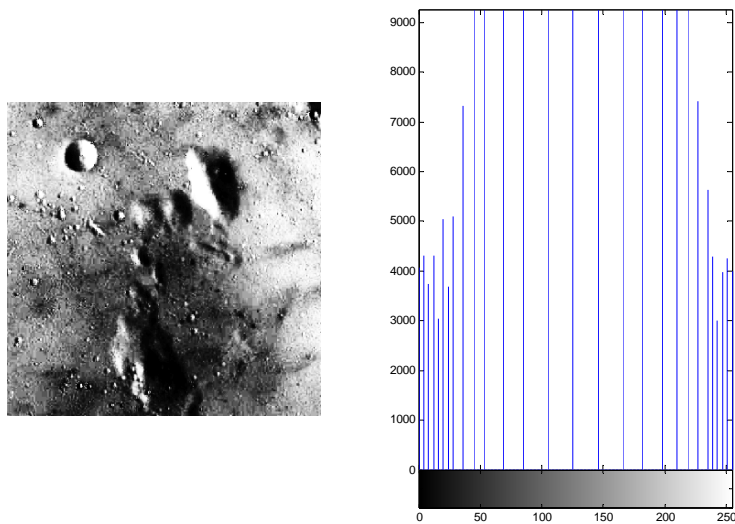


Figure 29: Moon surface after the histogram equalization.

For the system, contrast enhancement is desired because it facilitates the differentiation between the object and the background. Consider the following figure that depicts a person walking in the distance and occupying just a few pixels. The shirt of the person has less contrast than the pants as can be appreciated in the color and grayscale versions of the image. Note how after the contrast enhancement, the object and the background tend to be mostly black and mostly white which makes the object easier to recognize..

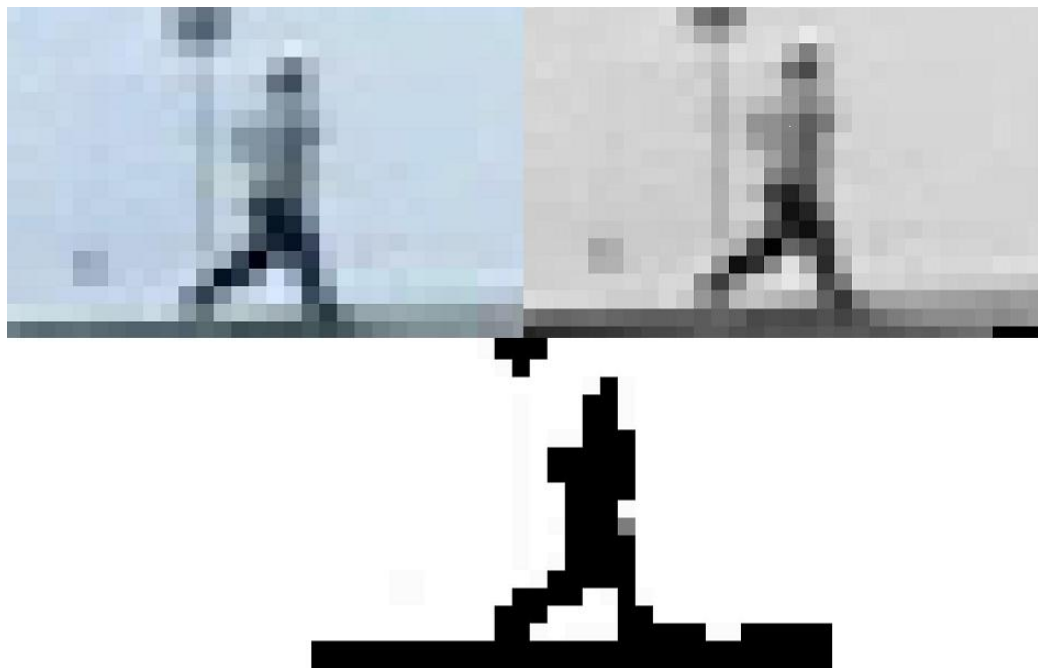


Figure 30: Contrast enhancement applied to the videos. Video in color, grayscale, and contrast enhanced, respectively.

The result is that more information can be extracted as more pixels from the objects are detected by the foreground. At the same time, there is more noise in the system due to

the contrast enhancement that will enhance changes in the scene as well as the object-background contrast.

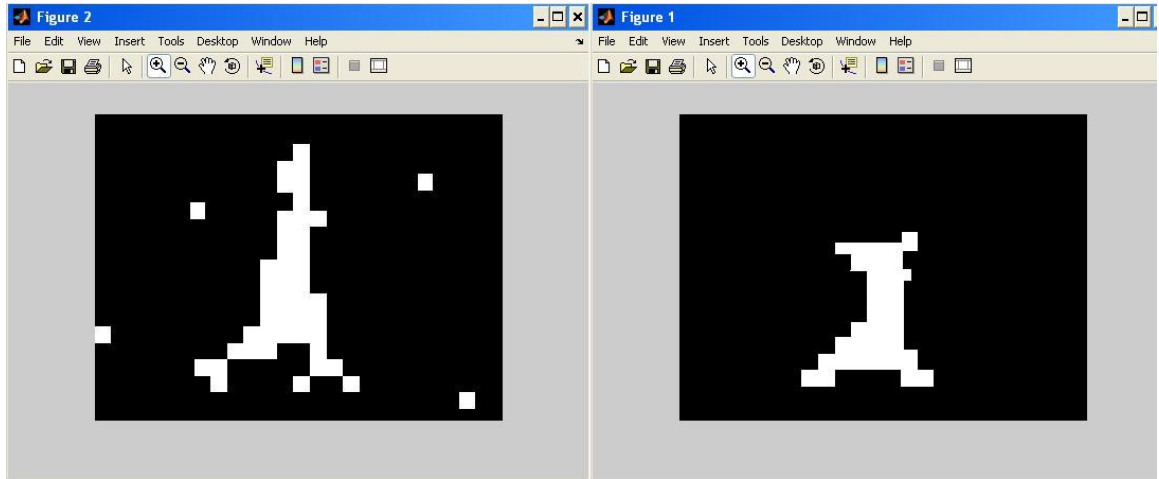


Figure 31: Object with the histogram equalization versus object without contrast enhancement.

The image on the left has more information but the system has more overall noise, while the image on the right has better noise handling but some information is lost in the process due to poor contrast in some regions.

5.2 Background Subtraction

One of the traditional methods for comparing background subtraction algorithms is the use of the ground-truth comparison [33]. In such a scheme, background subtraction algorithms are compared with images annotated by hand and the result is analyzed using detection theory techniques.



Figure 32: Ground truth example (manually annotated segmentation).

In high-resolution images, ground Truth analysis is useful because the limits between objects and background are well defined. In low-resolution images, however, the object is blended with the background in such a way that for some regions it is not clear if a given pixel is a part of the object or the background.

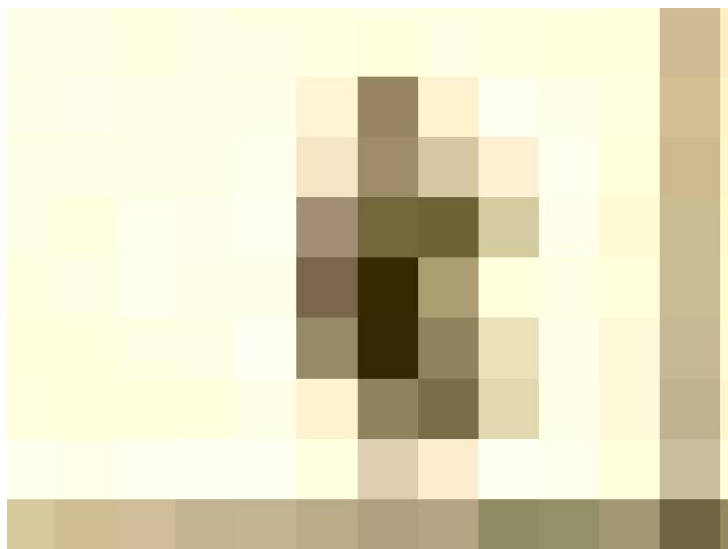


Figure 33: Very low resolution object.

Rather than through the ground-truth analysis, the algorithms were compared when applied to the same set of videos. Prior to the comparison, a tuning process was applied to the Frame Difference and the Approximate Median methods. The Mixture of Gaussian Method is a multivariate parametrical method and thus its tuning is rather complex. In this case, values were moved around the ideal set (see Section 4.2.2.3).

5.2.1 Frame Difference

With the same set of videos, the frame difference algorithm was applied with a varying value for the threshold. For comparison, one particular frame of the video was chosen. The chosen frame was the one where the objects and a fair amount of noise were present. When the threshold is low enough, considerable noise from the background is leaked into the foreground. On the other hand, when the threshold is too high, information from the foreground is lost, since the system understands it as background. The objective is then to find a point where the most information from the foreground pixels remains while the level noise is reduced. An example of the threshold comparisons follows.

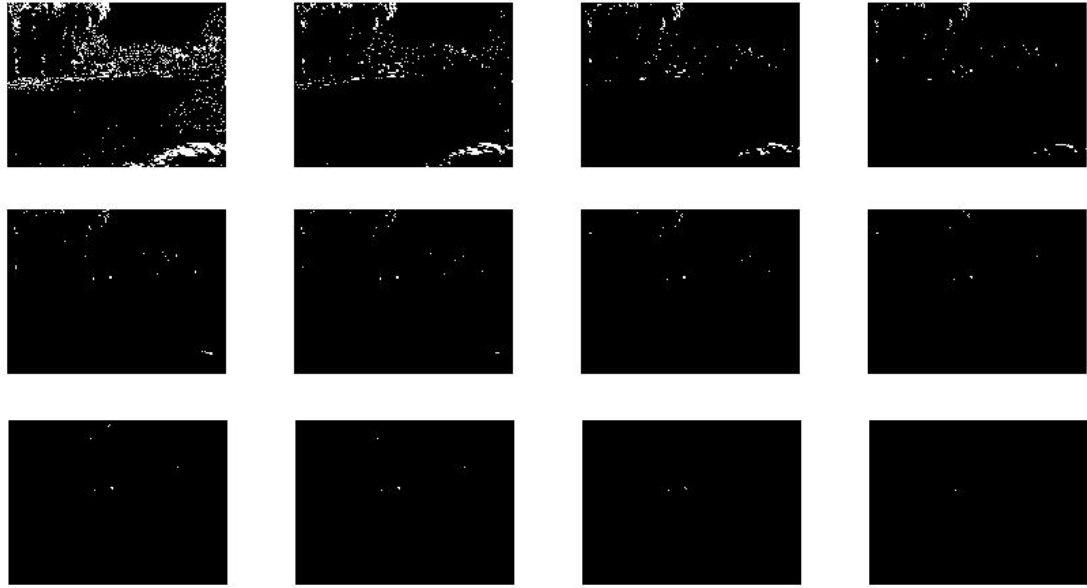


Figure 34: Threshold comparison for the Frame Difference algorithm.

The value of the threshold was varied from 10 to 120; the two objects present are clearly visible around the middle of the frame from a threshold value of 30. For threshold values over 90 there is almost no noise, but the objects are disappearing as well. At 120, one of the objects is completely lost.

The analysis of three videos showed that a threshold value of 70 exhibited a good balance between preventing noise from leaking and keeping the most information about the objects present. For that reason, 70 is the default value for the frame difference algorithm, although the MATLAB function `bgsub` allows the user to change the threshold parameter from the call.

5.2.2 Approximate Median

A technique similar to frame difference was applied to the Approximate Median. An example of the results is as follows:

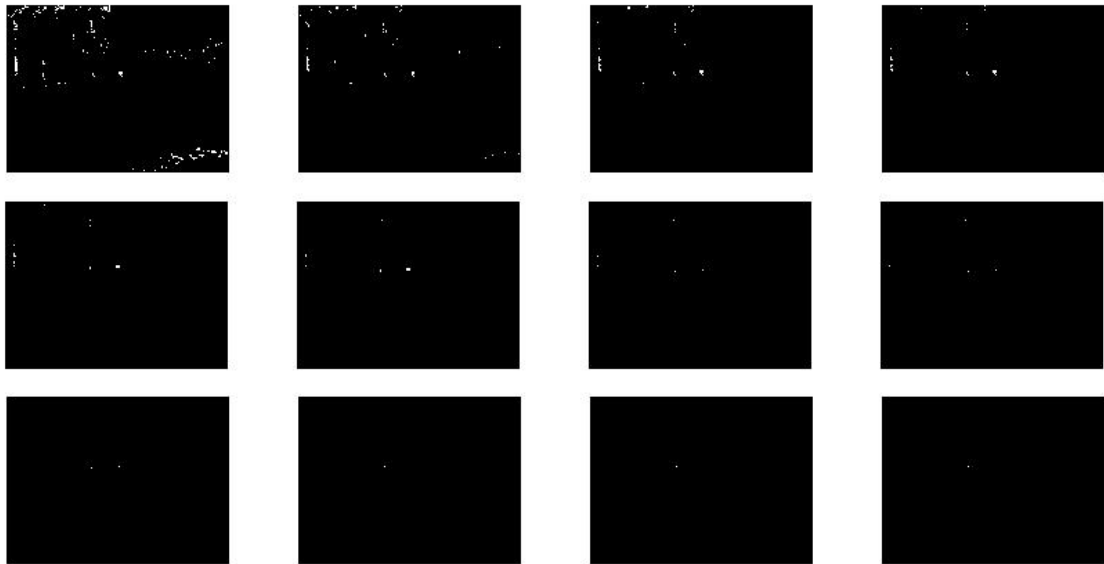


Figure 35: Threshold test of the approximate median algorithm.

Again the threshold is varied from 10 to 120. Objects are visible even when the threshold value is 10, and information about the objects is retained until the threshold has a value of 60. Beyond that point information is clearly lost.

From the analysis of three videos, a threshold value of 40 exhibits good balance and stability. For that reason, 40 is the default value for the Approximate Median algorithm.

The MATLAB function `bgsub` allows the user to change the threshold parameter from the call.

5.2.3 Mixture of Gaussians

Because the Mixture of Gaussians algorithm has many parameters, the tuning problem was approached one parameter at a time, sweeping the algorithm for a particular parameter and then sweeping another parameter with the previous one fixed at the best response. This approach is far from ideal, since it does not take into account possible interaction between parameters.

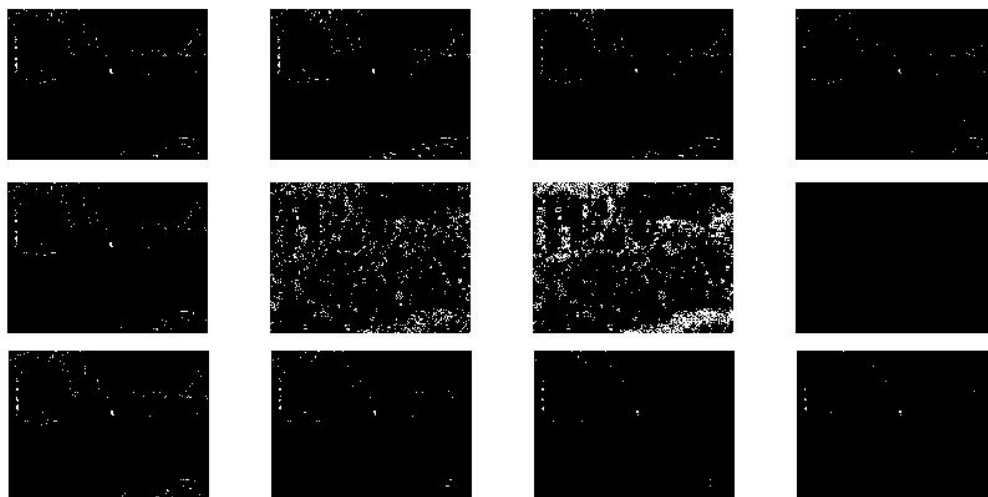


Figure 36: Parameter test of the mixture of Gaussians.

The figure shows a change in the threshold parameter between 0.25 and 0.75 in the top four images, a change in the learning rate between 0.01 and 1 in the middle four images,

and a change in the positive deviation parameter in the bottom four images. The best results were stored as default for the bgsub function, although the user can change the parameters from the call.

5.3 Morphological filtering

To test the morphological filter, a random image was created using the Microsoft program Paint. The image consists of three random object-like groups of pixels that can be seen in the following image:

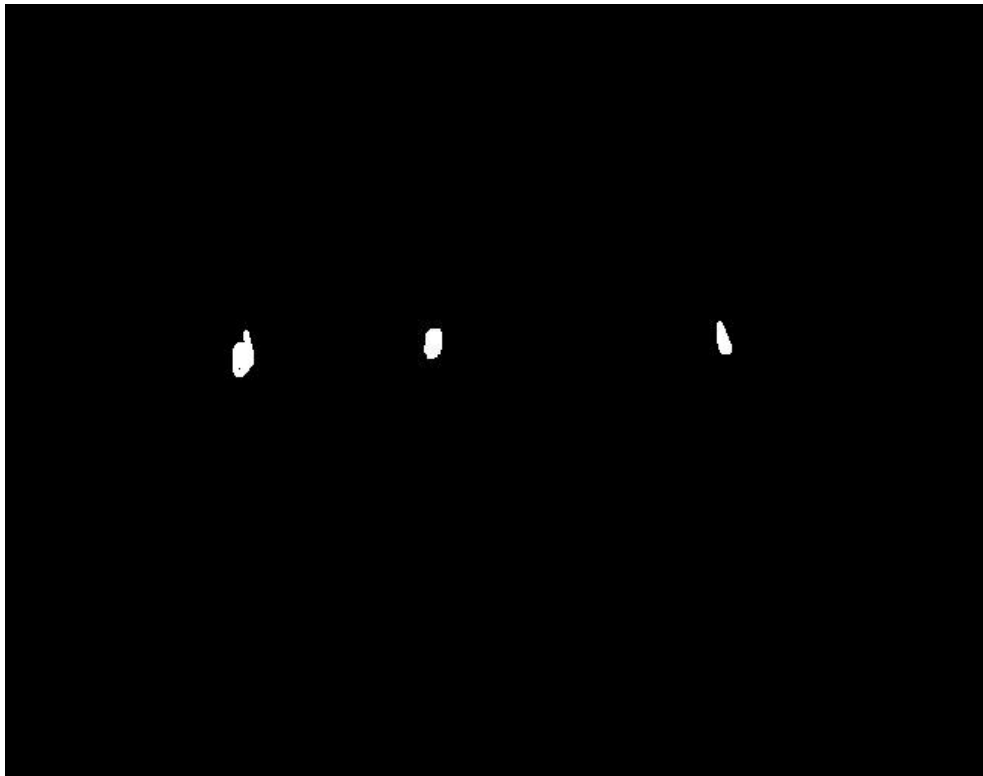


Figure 37: Test image for the morphological filter operation.

The test consisted then in adding artificial noise to this image and using the morphological filter to eliminate the noise. First random Gaussian noise was added with a normalized mean of 0.1 and a normalized variance of 0.007.

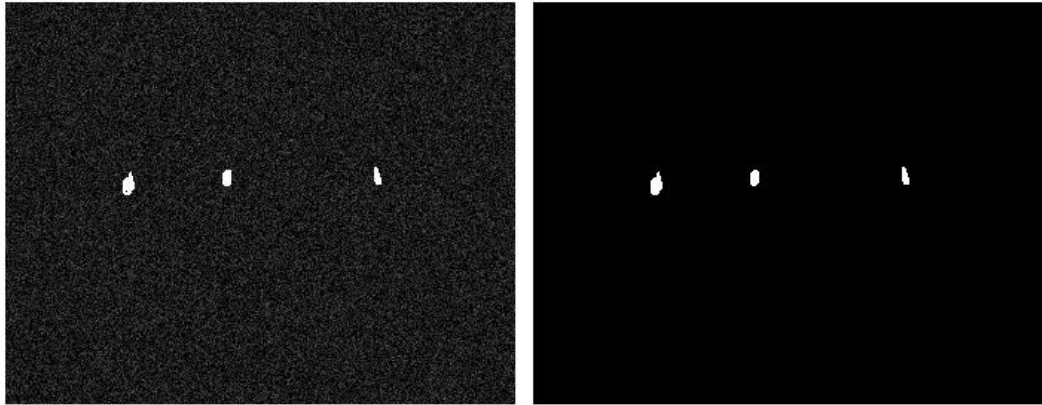


Figure 38: Image with Gaussian random noise (left), same image after the morphological filter (right).

The same procedure was repeated with Salt and Pepper noise, Poisson and Salt and Pepper noise, and with a randomly constructed noise created in paint. The results are as follows:

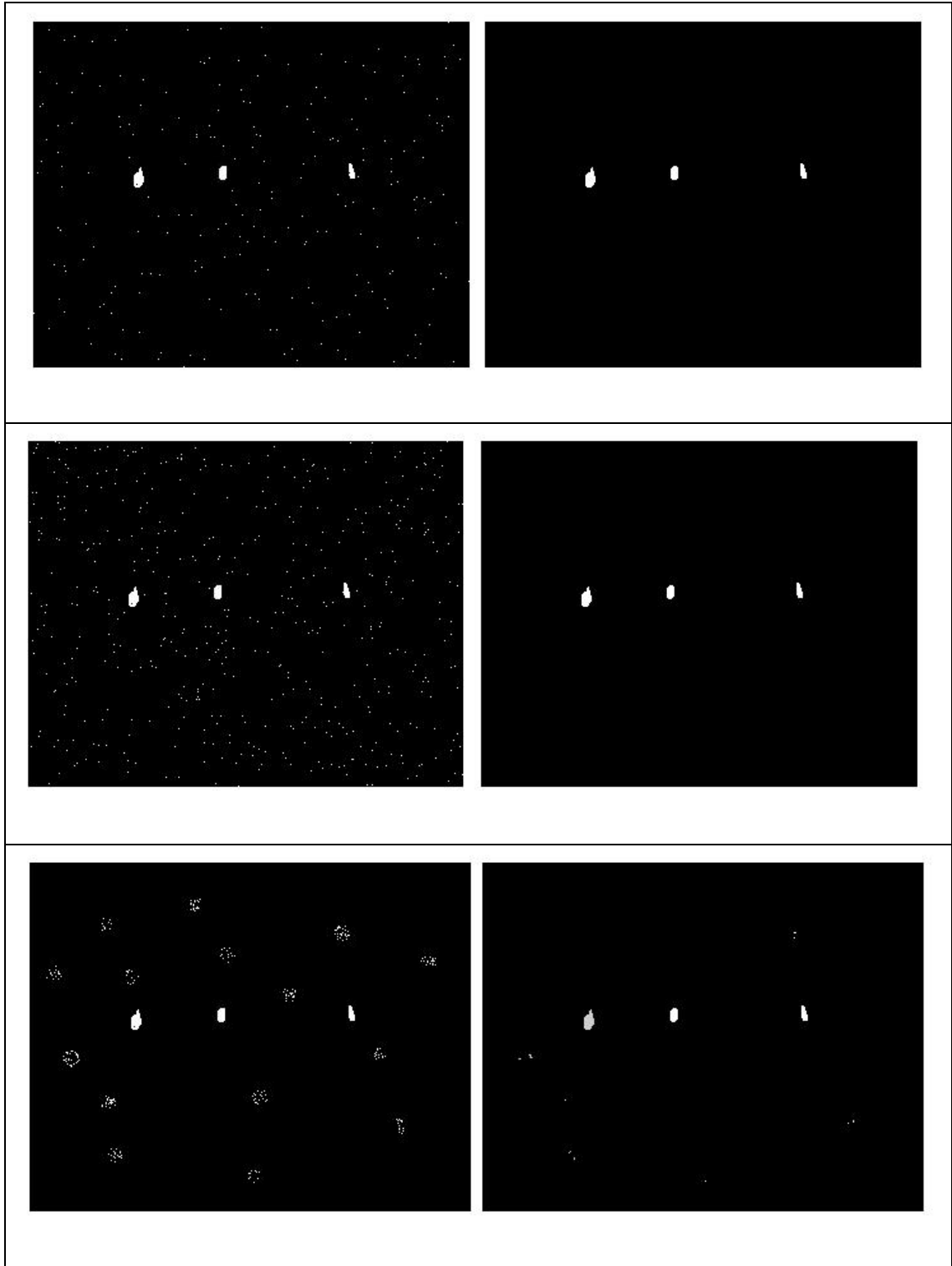


Figure 39: Image with Salt and Pepper noise and output (top), image with Poisson and salt and pepper noise and output (middle), and image with artificially constructed noise and output.

The morphological system was able to clean most of the noise with little loss of information of the images. Some of the noise remained as is expected but the reduced noise can be handled either by the region of importance or by the buffer state in the tracking system (see Sections 3.2.1.4 and 3.2.3.3).

When the morphological operations were implemented in the test video, a reduction of the noise was appreciable:



Figure 40: Grayscale image (bottom), background subtraction algorithm (top-left), and morphological filter output (top-right).

The noise that remained after the morphological filtering operation is due to vegetation moving with the wind. Note that the noise is in a region outside the usual region of importance, while the information loss of the foreground was acceptable.

5.4 Information Extraction and post processing

Each of the videos that were tested featured different scenarios. First a person is walking through multiple occlusions caused by nearby trees. The person occupies a maximum height of 18 pixels. The variable videofinal shows the tracking of the centroid through the screen.

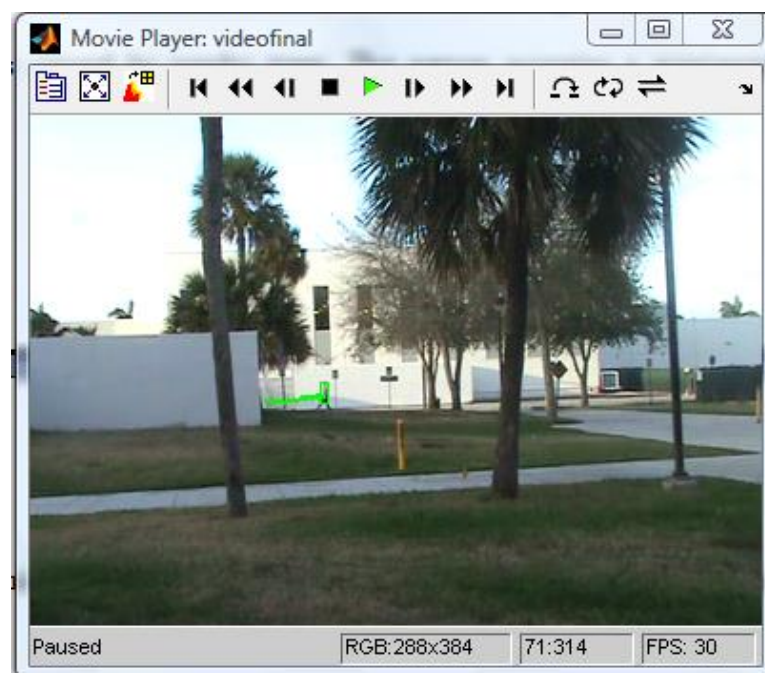


Figure 41: Tracking system.

The handling of occlusions can be seen in the following figure:

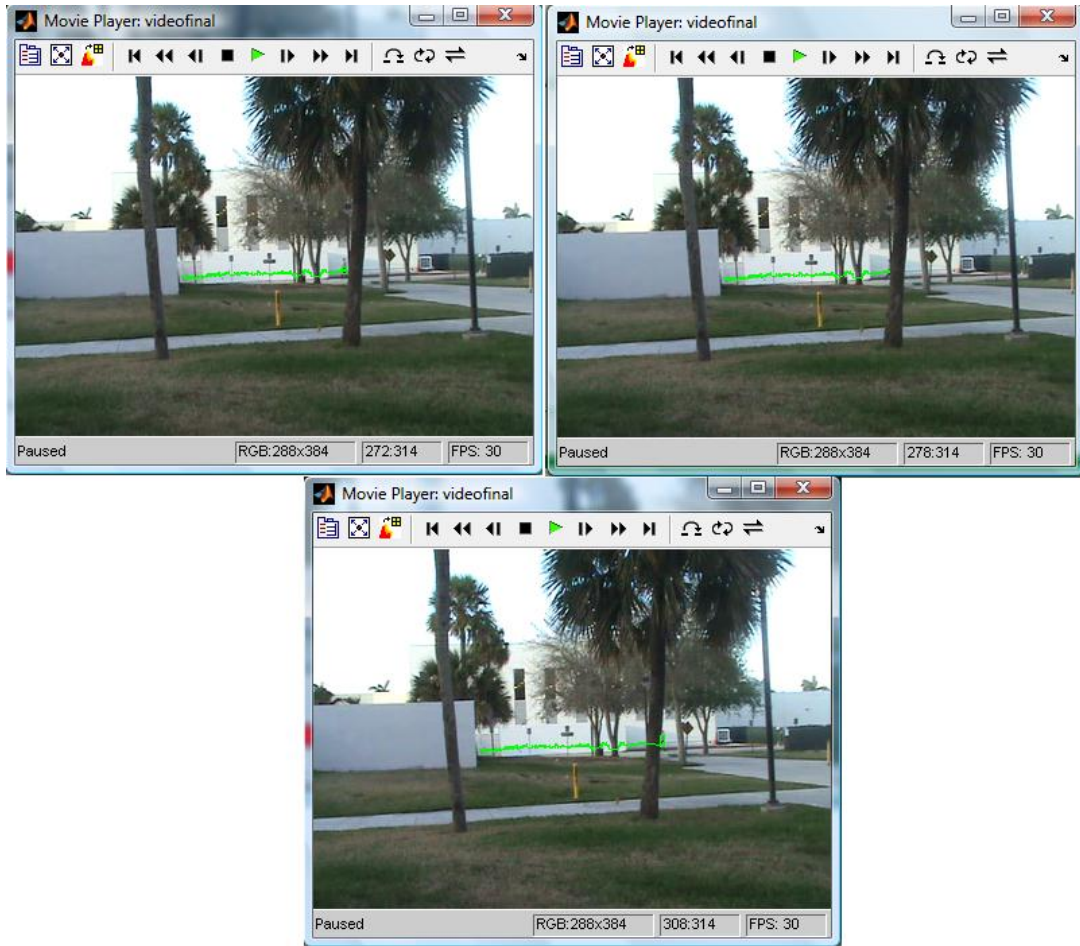


Figure 42: Occlusion handling of the tracking system.

From the figure, the system stops tracking the object once it disappears due to the occlusion. As soon as the object reappears at the other side, the previous information is retrieved and updated with the new.

The second video is identical to the first but with the subject running instead of walking. This is intended to show that the system does not lose track of an object when it is moving faster. The final results are similar to those of Figure 42: Occlusion handling of the tracking system.

The third video consisted of two persons, one walking while the other was almost still, as in hiding. The maximum height of either person in the video was 8 pixels.

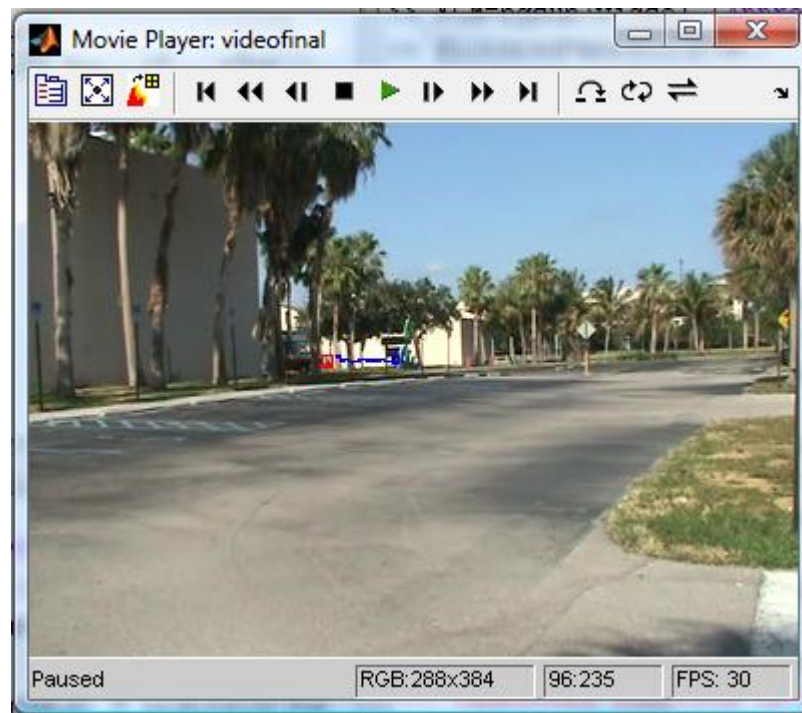


Figure 43: Tracking of video No. 3.

The first object's trajectory is being tracked as well as the second object (in red), which is barely moving.

The trackingdata cell stores the data obtained from the analysis., videofinal providing the visual aid but with the real information lying in the cell. The information from object 1 in frame 30 can be seen as follows.

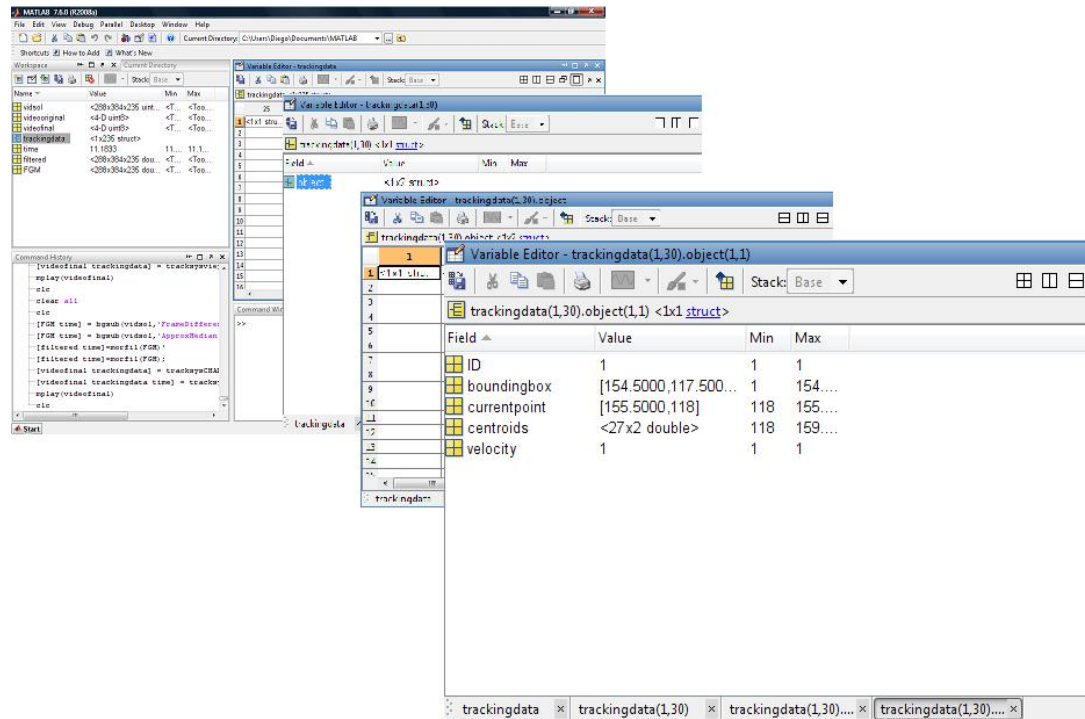


Figure 44: The trackingdata cell.

The cell is constructed so that information is easily retrieved for future developments.

5.5 Time Analysis

The following table shows the time in seconds that each algorithm spends on both computers.

Block	Function	LAPTOP			WORKSTATION		
		Video 1 (311 Fr)	Video 2 (314 Fr)	Video 3 (235 Fr)	Video 1 (311 Fr)	Video 2 (314 Fr)	Video 3 (235 Fr)
Preprocessing	preproc.m						
	With His. Eq.	23.76	22.98	20.94	25.72	24.62	22.54
	Without	17.35	17.24	14.68	18.79	17.65	13.97
Background Mod.	bgsub.m						
	FrameDiff	8.86	8.00	6.78	8.02	7.54	5.67
	ApproxMed	242.86	255.02	177.45	11.19	10.98	8.25
	MoG	2461.08	2422.79	1807.44	1637.33	1681.16	1296.81
	morfil.m	51.16	50.50	35.35	79.13	78.62	56.71
Info Extraction and Postprocessing	tracksys.m	8.03	11.14	11.18	14.58	23.61	16.43

Table 2: Time analysis of the functions.

As can be seen the preprocessing algorithm time depends mostly on whether or not histogram equalization is used, and post processing algorithm depends mostly on how many objects are present. Also, morfil depends mostly on the video length.

Certain results of the background subtraction algorithm are worth mentioning. The frame difference algorithm is fast, but very susceptible to noise, as seen in section 5.2.1. Approximate median is slower than the frame difference (although in the workstation it is almost as fast, perhaps because the workstation uses Windows XP while the laptop uses Windows Vista) but it provides more stability and less noise. Finally, Mixture of Gaussians, even though it is a high complexity algorithm, has a response and stability similar to or worse than the approximate median, but is 300 times slower.

Chapter 6 CONCLUSIONS AND FUTURE WORK

This chapter presents conclusions and directions for future work.

6.1 Conclusions

This thesis presented a working system for detection of low resolution objects in video sequences.

The proposed system is capable of:

- Detecting foreground objects in video sequences, proven to work with objects as small as 8 pixel of height and 15 pixels in total.
- Tracking objects and accumulate data along their trajectories.
- Handling occlusions.
- Implementing three different background subtraction algorithms.

- Choosing several regions of importance in a video sequence.
- Handling noise due to weather conditions, video conditions, or random noise by using a conjunction of four mechanisms: A selection of a region of importance, a selection of the background subtraction algorithm, the morphological filters operation, and the buffer state of the tracking system.

On the other hand, the system is subject to the following restrictions:

- A single, static camera setting.
- Implementation time and memory capacity limits affect the video size and the amount of information that can be extracted.
- Limited number of objects present on the video. The system was designed to handle objects as relevant events so limitation both in memory and time of analysis would limit the number of objects that can be present at a given time.
- The system needs a minimum of contrast between the object and the background in order to work.

- The system needs for the object to be moving across the scene: a standing object or an object that is moving but stops doing so for a long time will eventually blend with the background and will therefore not be detectable.
- The solution for the occlusion problem behaves relatively well with a small number of objects present, but it is very dependant in the condition of the scene.
- A real time solution is not feasible with the current implementation unless more sophisticated equipment is used such as dedicated computing systems.

Through study and experimentation, this work has reached the following conclusions:

- The introduction of region of interest selection to the overall system improves the response of the system to noise such as climatic conditions, wind, and movement of shades.
- The implementation of histogram equalization improves the contrast between the object and the background but also introduces more noise in the system. Depending on the application and the condition of the scene, the histogram equalization can be a useful technique.
- Of the background subtraction algorithms implemented, the approximate median method turned out to be the best option for most applications because it is fast,

easy to implement, and handles noise relatively well. Frame difference is fast and easy to implement but very susceptible to noise and very dependant on continuous movement of the object. Finally, mixture of Gaussians handles noise relatively well but is very slow and very difficult to tune.

- Morphological filtering proved to be a valuable method for removing noise that leaked from the background in the background subtraction operation.
- The tracking system was able to detect and track objects occupying tens of pixels in the screen under controlled conditions such as relatively simple background, stable weather and lightning conditions.
- In low resolution objects, color contrast between the object and the background is the feature that provides more information about the object. Ultimately it is this feature that permits the detection of such objects.
- Information such as relative velocity, centroid, and position can be extracted from the system.
- MATLAB proved to be an important tool when developing prototypes due to its built-in video processing and mathematical tools. For real time implementation the use of lower level languages is required.

- The separation of the problem into blocks was designed to permit future improvements in each of the four blocks. This is a system that can be improved in each of its blocks separately allowing for future implementation to use all or part of the blocks and improve others.

6.2 Future Work

Possible areas for future work related to this thesis include:

- The data extracted in the cell tracking data can be used for finding periodicity in the movement in order to help determine whether the object is human or not.
- The information from cell tracking data can also be used to find probable distance, velocity, and size of the object using the context of the scene as an aid. That would give the system more information about the nature of the object.
- If an implementation in real time based on this system could be developed, video final will constitute a good early alarm device for security applications.
- If a system with several cameras is used, the system could constitute a trigger event that will direct cameras that are closer to the object to track and focus on the object in order to extract more valuable information..

- The region of importance can be improved if, instead of a binary mask, a mask with several levels of importance can be implemented. The system then would not rule out any information but instead will organize the information according to its importance.
- A system that automatically evaluates if histogram equalization is worthy could be developed.
- More background subtraction algorithms have been implemented and researched; implementing the system with more complex background subtraction algorithms could improve the response of the system.
- A more complex occlusion handling could involve probability and predictors; furthermore, more data can be incorporated in the tracking and the occlusion handling algorithm than merely position and size.
- A wide variety of videos with varying conditions and objects could improve the overall evaluation of the system.
- Optimization of the code, migration to other programming language, or hardware implementation must be taken into account for a real time implementation.

Bibliography

- [1] W. T. Rhodes and D. Pava, "Removing Ambiguity in 2-D Image Information by Means of 3-D Models," in *Digital Holography and Three-Dimensional Imaging*, OSA Technical Digest (CD) (Optical Society of America, 2008), paper DMA3.
- [2] Weiming Hu; Tieniu Tan; Liang Wang; S. Maybank, "A Survey on Visual Surveillance of Object Motion and Behaviors," *IEEE Trans. Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 34, no. 3, 2004, pp. 334–352.
- [3] N. Friedman and S. Russell, "Image segmentation in video sequences: a probabilistic approach," in *Proc. 13th Conf. Uncertainty in Artificial Intelligence*, 1997, pp. 1–3.
- [4] D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, and S. Russel, "Toward robust automatic traffic scene analysis in real-time," in *Proc. Int. Conf. Pattern Recognition*, Israel, 1994, pp. 126–131.

- [5] M. Köhle, D. Merkl, and J. Kastner, "Clinical gait analysis by neural networks: Issues and experiences," in Proc. IEEE Symp. Computer-Based Medical Systems, 1997, pp. 138–143.
- [6] H. Z. Sun, T. Feng, and T. N. Tan, "Robust extraction of moving objects from image sequences," in Proc. Asian Conf. Computer Vision, Taiwan, R.O.C., 2000, pp. 961–964.
- [7] N. J. B. McFarlane¹ and C. P. Schofield, "Segmentation and tracking of piglets in images", Machine Vision and Applications, Vol 8-3 pp. 187-193.
- [8] I. Haritaoglu, D. Harwood, and L. S. Davis, "W⁴: Real-time surveillance of people and their activities," IEEE Trans. Pattern Anal. Machine Intell., vol. 22, pp. 809–830, Aug. 2000, pp.809-830.
- [9] A. J. Lipton, H. Fujiyoshi, and R. S. Patil, "Moving target classification and tracking from real-time video," in Proc. IEEE Workshop Applications of Computer Vision, 1998, pp. 8–14.

- [10] D. Meyer, J. Denzler, and H. Niemann, "Model based extraction of articulated objects in image sequences for gait analysis," in Proc. IEEE Int. Conf. Image Processing, 1998, pp. 78–81.
- [11] Gonzales, Woods, Eddins, (2002) "Digital Image Processing using MatLab", Prentice Hall, ISBN 0130085197.
- [12] JPL, "Traffic surveillance and detection technology development," Sensor Development Final Rep., Jet Propulsion Laboratory Publication no. 97-10, 1997.
- [13] D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, and S. Russel, "Toward robust automatic traffic scene analysis in real-time," in Proc. Int. Conf. Pattern Recognition, Israel, 1994, pp. 126–131.
- [14] R. Polana and R. Nelson, "Low level recognition of human motion," in Proc. IEEE Workshop Motion of Non-Rigid and Articulated Objects, Austin, TX, 1994, pp. 77–82.
- [15] R. Cutler and L. S. Davis, "Robust real-time periodic motion detection, analysis, and applications," IEEE Trans. Pattern Anal. Machine Intell., vol. 22, Aug. 2000, pp. 781–796.

- [16] I. Pavlidis, V. Morellas, P. Tsiamyrtzis, and S. Harp, "Urban surveillance system: from the laboratory to the commercial world," Proc. IEEE, vol. 89, Oct. 2001, pp. 1478–1497.
- [17] G. P. Stein, "Tracking from multiple view points: self-calibration of space and time," in Proc. IEEE Conf. Computer Vision and Pattern Recognition, vol. I, 1999, pp. 521–527
- [18] R. C. Gonzalez and R. E. Woods: "Digital Image Processing", Prentice Hall, Second edition (2002).
- [19] Piccardi, M., "Background subtraction techniques: a review," Systems, Man and Cybernetics, 2004 IEEE International Conference on , vol.4, Oct. 2004, pp. 3099-3104.
- [20] B.P.L. Lo and S.A. Velastin, "Automatic congestion detection system for underground platforms," Proc. ISIMP2001, May2001, pp. 158-161.
- [21] C. Stauffer and W.E.L. Grimson, "Adaptive background mixture models for real-time tracking," Proc. IEEE CVPR 1999, June 1999, pp. 24&252.

- [22] C. Wren, A. Azarhayejani, T. Darrell, and A.P. Pentland, "Pfinder: real-time tracking of the human body," *IEEE Trans. on Pattern Analysis. and Machine Intelligence.*, vol. 19, no. 7, 1997, pp. 780 - 785.
- [23] A. Elgammal, D. Hanwood, and L.S. Davis, "Nonparametric model for background subtraction," *Proc. ECCV 2000*, June 2000, pp. 751-767.
- [24] M. Seki, T. Wada, H. Fujiwara, and K. Sumi, "Background subtraction based on cooccurrence of image variations," *Proc. CVPR 2003*, Vol. 2, pp. 65-72,2003.
- [25] Cliff, N, "Analyzing Multivariate Data", Harcourt Brace Jovanovich, San Diego, 1987.
- [26] N.M. Oliver, B. Rosario, and A.P. Pentland, "A Bayesian computer vision system for modeling human interactions," *IEEE Trans. on Pattern Anal. and Machine Zntell.*, vol. 22, no. 8, 2000, pp. 831-843.
- [27] A. Yilmaz, O. Javed, M. Shah, "Object Tracking: A Survey," *ACM Computing Survey*, vol. 38, 2006.
- [28] C.Pertuz,, "Methods and Algorithms for Human Detection in Video Sequences ", Master thesis defense, Florida Atlentic University, 2007.

[29] Wagner, F., "Modeling Software with Finite State Machines: A Practical Approach", Auerbach Publications, 2006, ISBN 0-8493-8086-3.

[30] Chai, D.; Ngan, K.N., "Face segmentation using skin-color map in videophone applications," Circuits and Systems for Video Technology, IEEE Transactions on , vol.9, no.4, Jun 1999, pp.551-564.\

[31] <http://www.erightsoft.net/SUPER.html>

[32] func_drawline version 1.0 Jing Tian Oct. 31 2000.

[33] Abdou and W. Pratt, Quantitative design and evaluation of enhancement/thresholding edge detectors, Proceedings of the IEEE 67 (5) (1979), pp. 753–763.

Removing Ambiguity in 2D Video Imagery by Means of 3D Models

William T. Rhodes^{1,2} and Diego Pava²

¹ Georgia Institute of Technology, School of Electrical Engineering, Atlanta, Georgia 30332, USA

² Florida Atlantic University, Imaging Technology Center, Boca Raton, Florida 33431 USA
wrhodes@fau.edu

Abstract: If moving objects in low-resolution 2D video imagery are placed in their 3D context, ambiguities concerning the identity of the objects can often be removed. We consider the case of detecting distant humans that subtend only tens of pixels in digital video. We also make observations on what appears to be a potential paradigm shift regarding what constitutes an image. ©2007 Optical Society of America.

OCIS codes: (100.0100) Image Processing; (110.2960) Image Analysis; (110.6880) Three-dimensional image acquisition

1. Introduction: A problem to be addressed

Assume that you are given the task of detecting the presence of humans through the aid of a wide-field-of-view video camera. Assume further that humans of interest could be quite near to your camera, subtending many pixels of the video imagery, or quite far, subtending very few. In the latter case, the problem is much more difficult, because the number of pixels associated with the moving object—human or not human?—can be so much smaller, perhaps numbering only in the tens. Low-resolution imagery, as would be encountered with sufficiently distant objects, presents additional difficulties, because the blurring of the background into the moving part of the scene makes the application of such techniques as tracking the center of mass of the object unreliable.

2. A possible solution: Exploit knowledge of the 3D scene

The approach we are taking in addressing this problem requires that we have available to us a 3D model of the scene being viewed with our video camera. Exploiting our knowledge of the 3D world from which we have extracted the 2D video projections, we can then reduce, often by extremely large amounts, possible uncertainties concerning the nature of what we are viewing.

Figure 1 provides an illustration of the concept with 2D still images—i.e., no video—and without a true 3D representation of the scene available to us, only our own idea of what the 3D scene actually is. The left-hand part

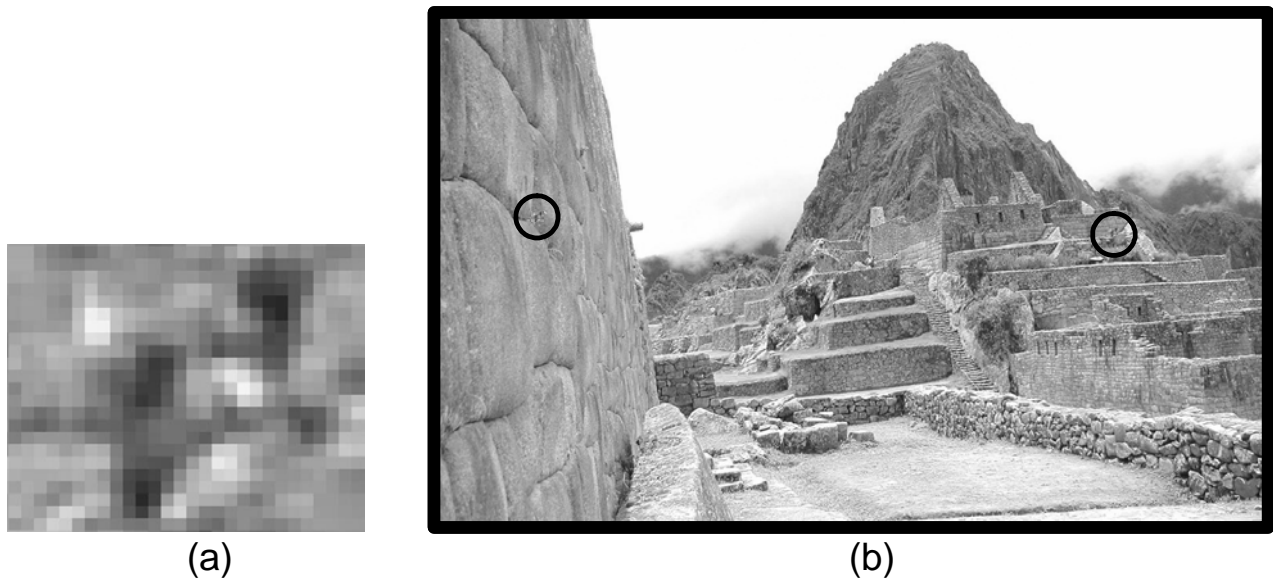


Figure 1. The small group of pixels on the left may or may not correspond to one or more humans. The location of these pixels within a 3D scene, two such locations being indicated on the right, makes the likelihood of their representing humans much easier to determine, even in the absence of motion cues.

of the figure shows a small number of pixels extracted from somewhere in the larger image shown on the right. In a video image, we would observe some motion within this small number of pixels. Does that motion represent human activity, or something else? The question is largely resolved if we know *where* in the scene the pixels in question are observed. If they are observed in the circled region at the left, they probably represent a moving leaf, a lizard, or some other small animal or insect; if in the circled region on the right, they almost certainly represent one or two humans climbing along an ancient pathway. With several frames of video, the probability that the changing pixels represent human(s) can be more accurately determined through the observation of the motion itself: Does it have an up-and-down component? Is the transverse motion consistent with people struggling along a 2500 meter-high path? Most importantly, is the size of the moving pixel group consistent with people at that apparent distance?

The idea that even a subconscious understanding of a 3D setting can help disambiguate information contained in a 2D image is of course not new. What is relatively new is the greatly increased capability we have now to obtain and manage data on the 3D structure of settings of interest to us. Our ability to build a 3D model of buildings, trees, roadways, and the like from a stereo image pair or other forms of 3D scene observation has improved enormously over even the past decade, and today's computational power and huge computer memories make fine-scale 3D database representations, along with the attachment of contextual information, comparatively easy.

The human detection problem can rely on many forms of information, all conditioned on being consistent with location in the scene. Thus, the probability that a moving object is a human in the scene of Fig. 1(b) depends on such things as the speed and up-and-down amplitude of the object being consistent with human locomotion at the assumed feet-on-the-ground distance, the vertical position y vs. the horizontal position x of the object, and so forth. Similar principles can be applied to other possible moving objects, such as vehicles, airplanes, boats, etc.

The framework in which such disambiguation operates is of necessity probabilistic, and several methods to be discussed in the presentation, including traditional Bayesian, can be used. Of at least equal importance is the impact of very-low-resolution imagery on the image processing algorithms employed. If an object of concern is so distant that it subtends only tens of pixels, then the normal approaches to motion tracking, such as optical flow methods, do not work well. Edges are fuzzy, and the interaction of the (presumably) stationary background structure with the moving object structure presents problems. These issues will also be addressed in the talk and examples given.

Acknowledgment

This work was supported by a Multidisciplinary University Research Initiative grant from the U.S. Army Research Office.

Dismount Detection and Characterization Using Radar

Ryan K. Hersey, Ph.D.

Georgia Tech Research Institute (GTRI)

Sensors & Electromagnetic Applications Laboratory

ryan.hersey@gtri.gatech.edu, 404.407.7524

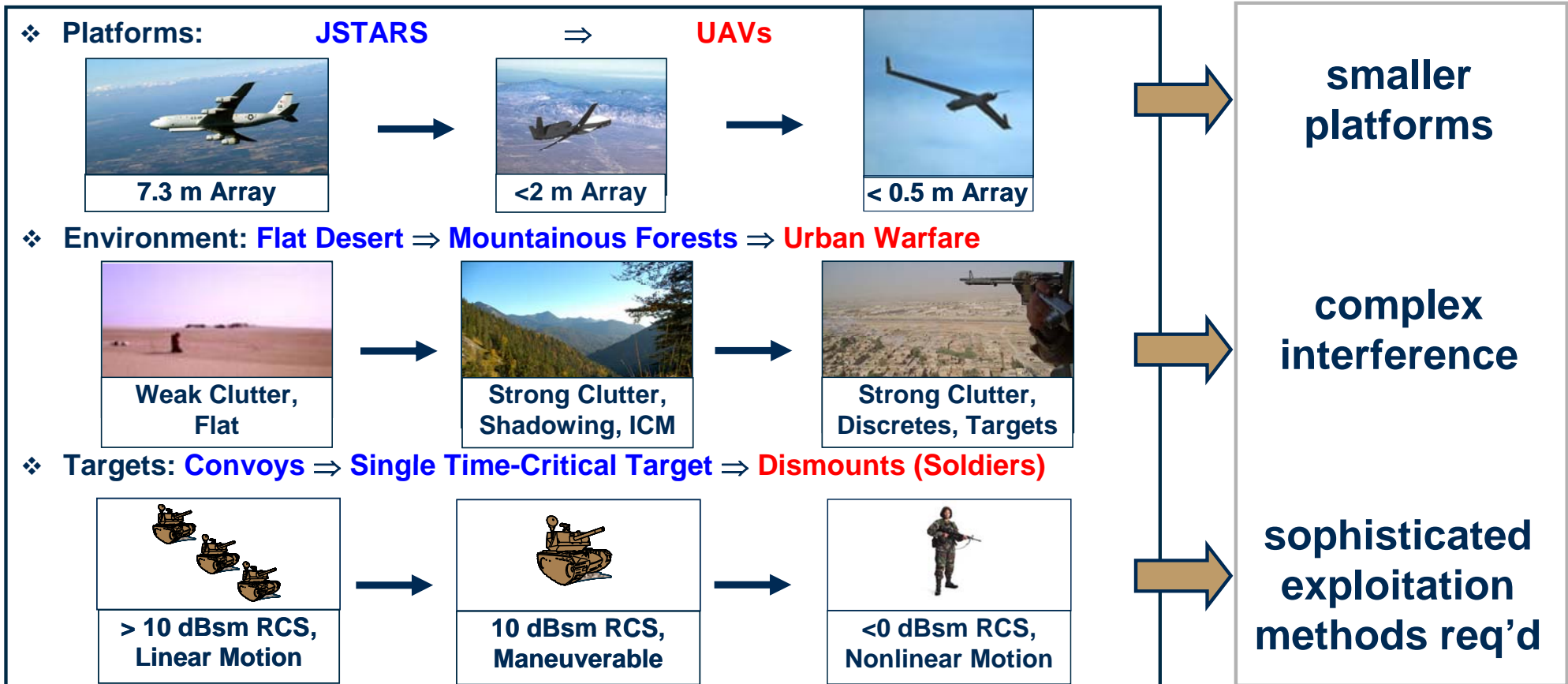


Outline

- **Introduction**
- **Dismount modeling and measurements**
- **Dismount algorithm development**
- **Summary**

Challenging GMTI Problem

- Smaller platforms → smaller antennas and radar subsystems → more clutter, lower SNR
- Challenging environments → clutter and RFI adversely affect detection performance, dense target environments
- Threat “systems” are different → dismounts, individual vehicles



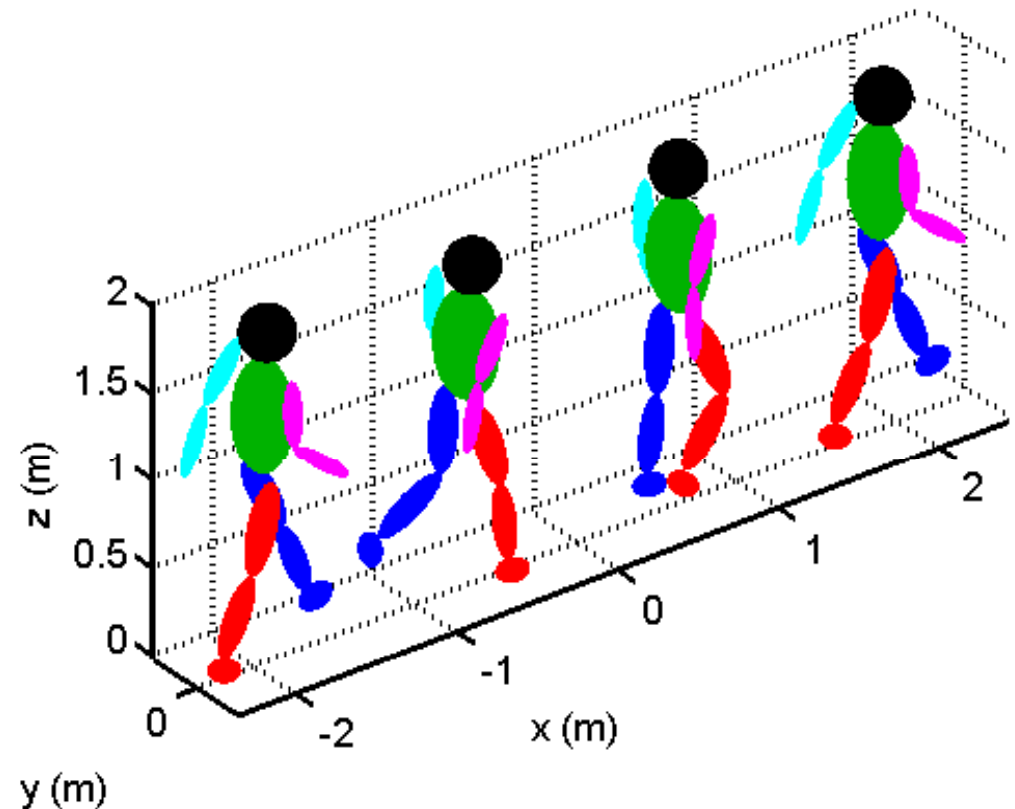
Outline

- Introduction
- **Dismount modeling and measurements**
- Dismount algorithm development
- Summary

Dismount Modeling

- Human body modeled using 12 body parts
 - Spherical head
 - Cylindrical or ellipsoidal torso, arms, and legs
- Kinematic model applied to move each body part
 - Body part centers give phase histories
 - Body part orientations give radar cross section (RCS)

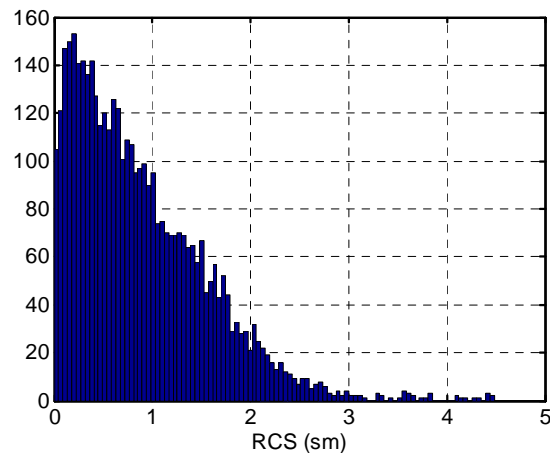
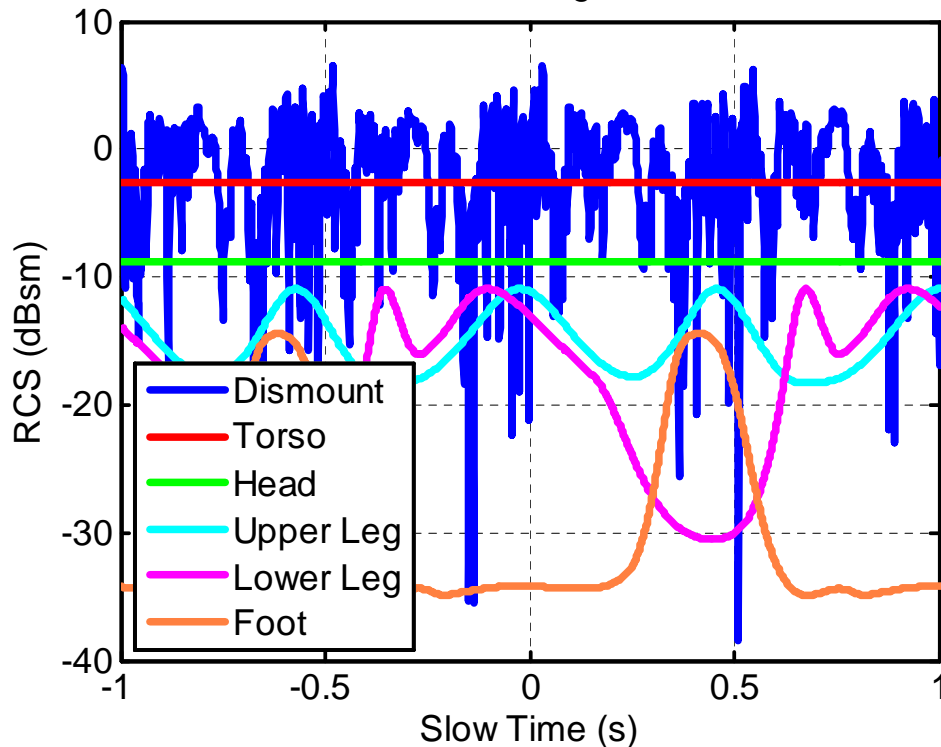
Dismount Walking at 1.85 m/s



- P. van Dorp and F.C.A Groen, "Human walking estimation with radar."
- R. Boulic, N. Magnenat Thalmann, and D. Thalmann, "A global human walking model with real-time kinematic personification."

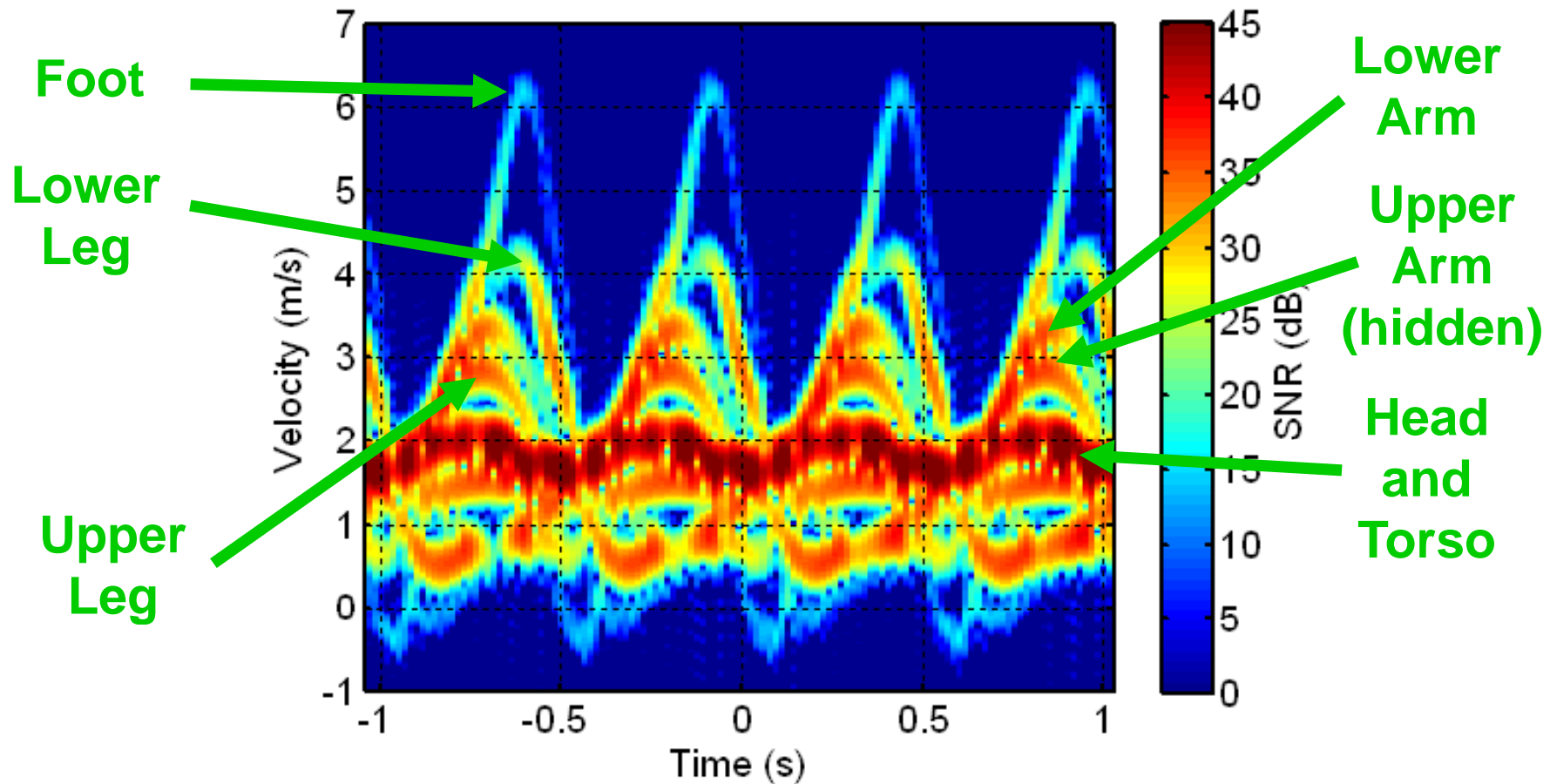
Dismount Radar Cross Section

RCS of a Dismount, Average = -0.53686 dBsm



- Variations in pose angle causes the RCS of each body part to vary over time
- The total dismount RCS varies significantly in time due to constructive and destructive interference
 - Average Dismount RCS: -3 to 0 dBsm
 - Torso and Head account for approximately 50% to 75% of the dismount RCS, depending on the subject
 - Dismount RCS gives a distribution similar to that of a Swerling 3 target

Simulated Dismount Spectrogram

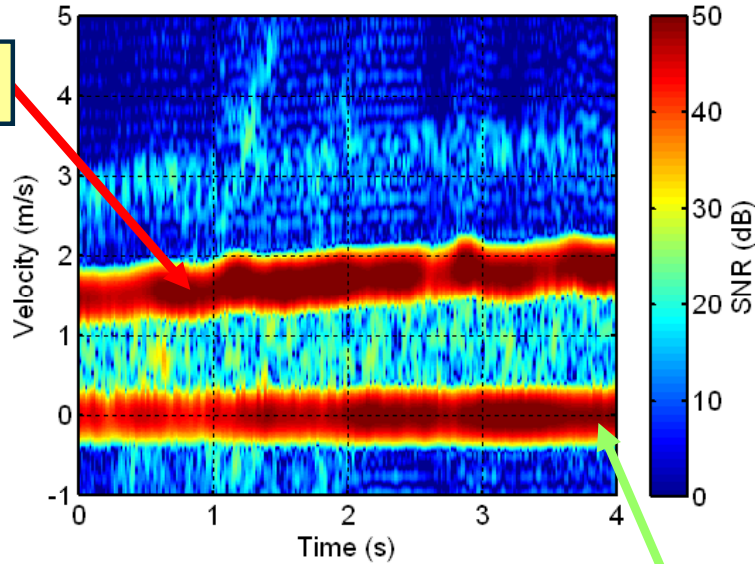


Spectrogram generated with a Hamming Window with 256 integrations (65.5 ms)

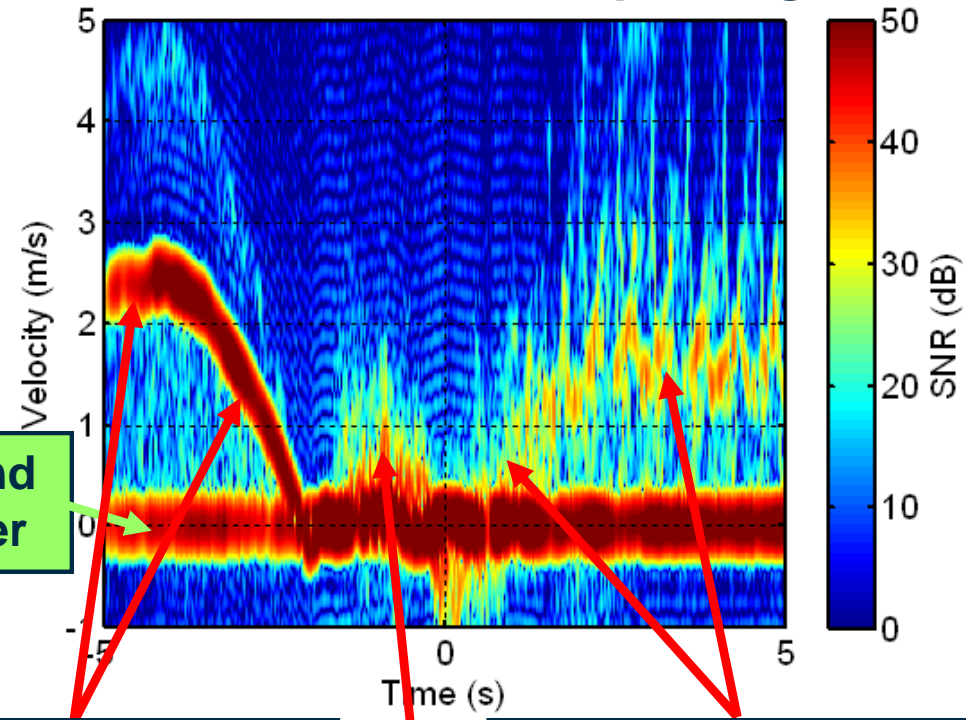
Dismount and Vehicle Radar Measurements

GTRI has acquired data from a mixture of vehicles and dismounted combatants

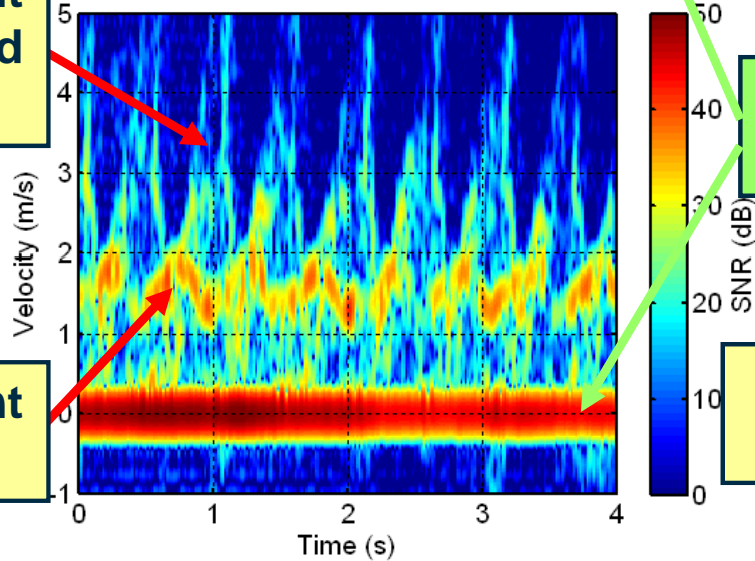
Vehicle Spectrogram



Vehicle with Dismount Spectrogram



Dismount Spectrogram



Ground Clutter

Vehicle Approaches and Slows to a Stop

Door Opens

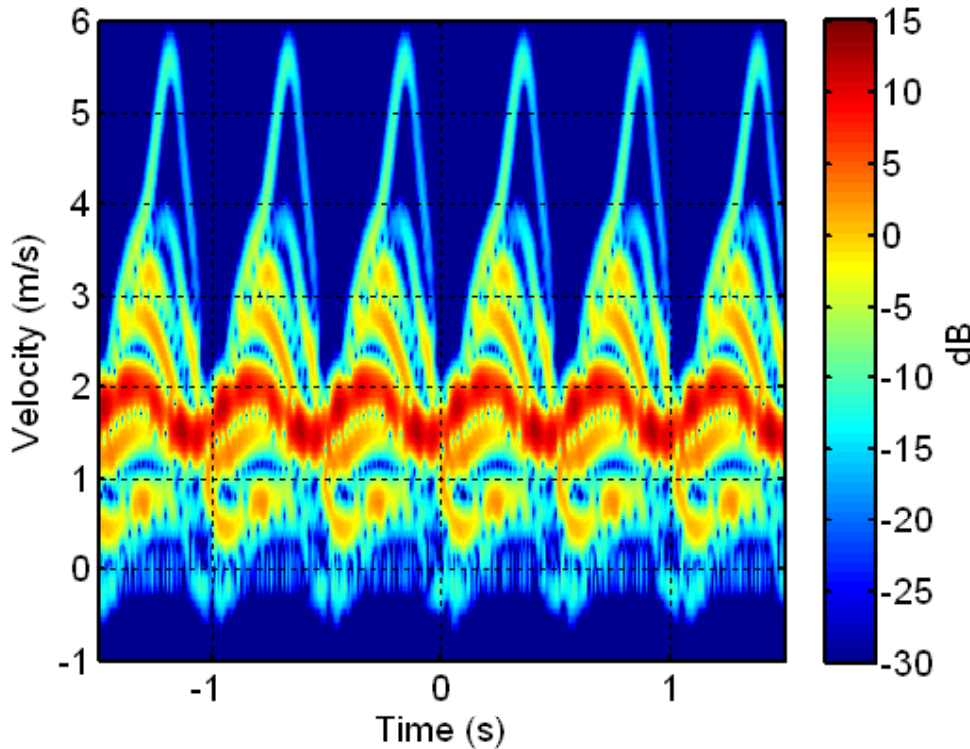
Person Dismounts and Walks Away

Dismount Arms and Legs

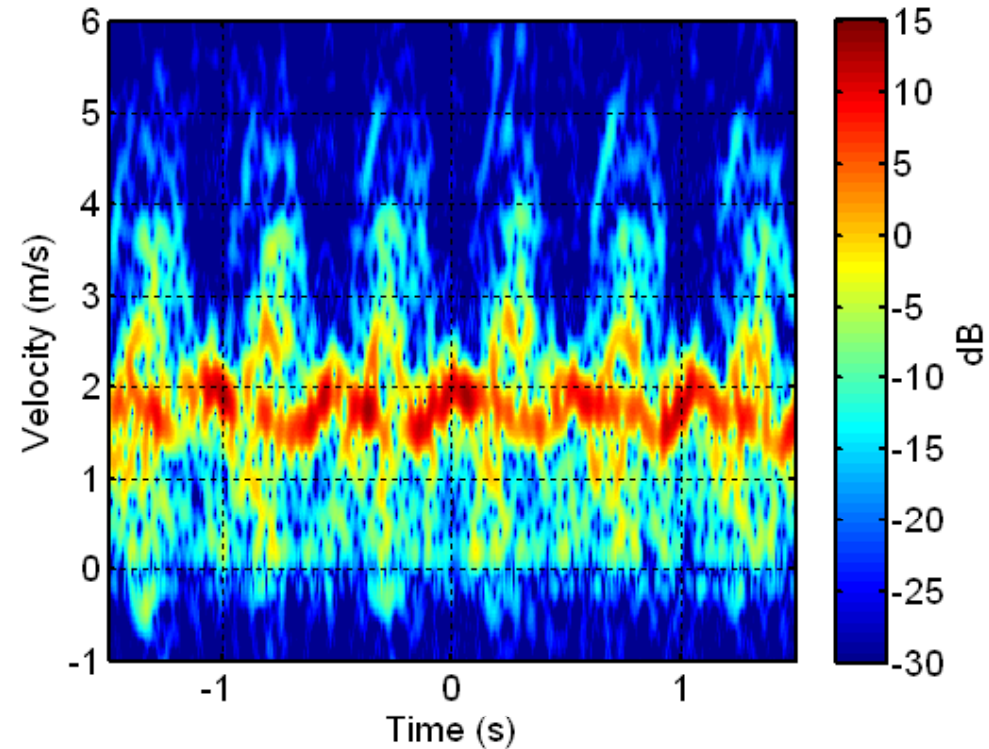
Dismount Torso

Comparison of Measurements and Simulation

Simulation



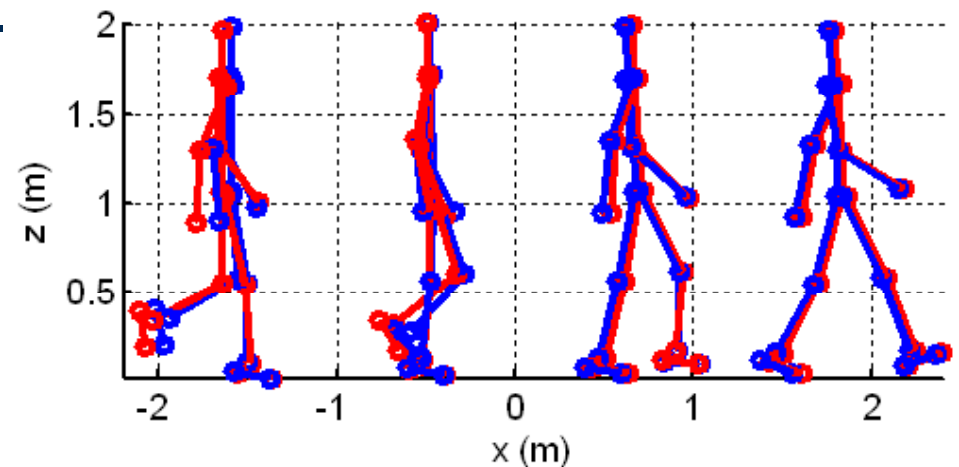
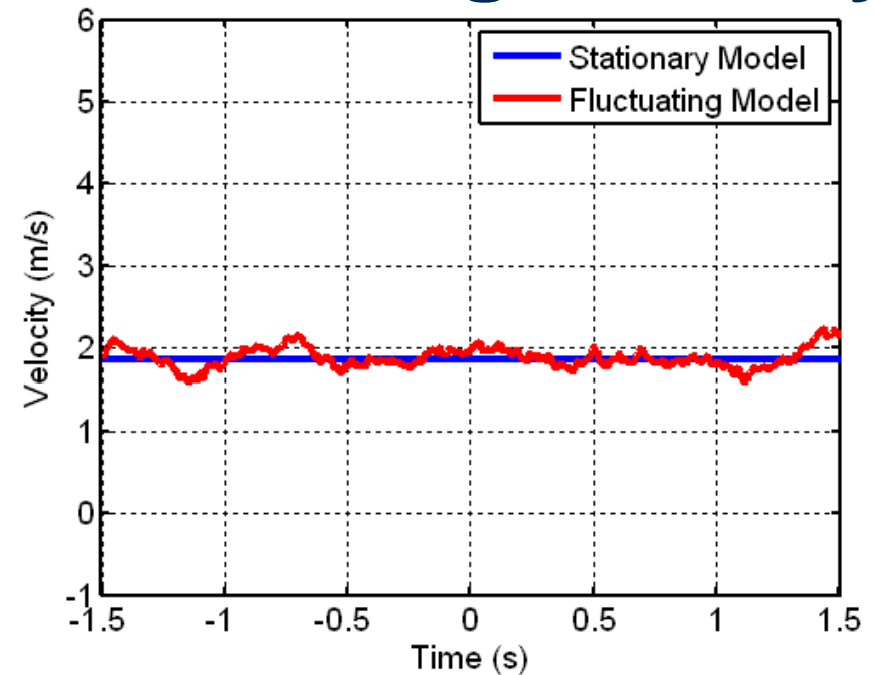
Measurement



- Simulated and measured spectrograms shows good agreement in the amplitude and frequency of the oscillations
- Simulated spectrogram shows a smoother and more periodic structure than the measured spectrogram

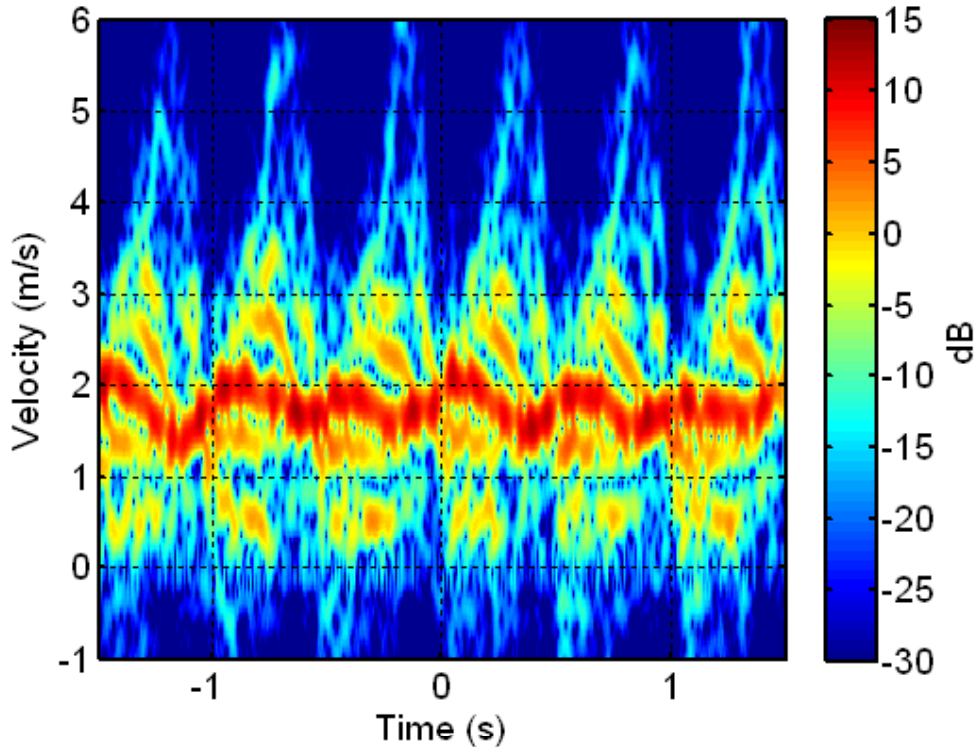
Dismount Model with a Fluctuating Velocity

- In the real world, people do not walk in a perfectly smooth manner as described by the previous dismount model
- Adjust model to allow time-varying fluctuations in the model parameters
- Velocity parameter controls many aspects of the dismount walking model
 - Average walking speed
 - Amplitude and frequency of the oscillations of the body parts
- Fluctuating velocity model allows time-varying fluctuations of the average velocity
 - $v(n) = \sqrt{1 - \alpha^2} v(n-1) + \alpha \Delta v_n$
 - Δv_n is a normally distributed random variable with a zero mean and unit variance
 - The decorrelation factor α controls the degree of the fluctuations

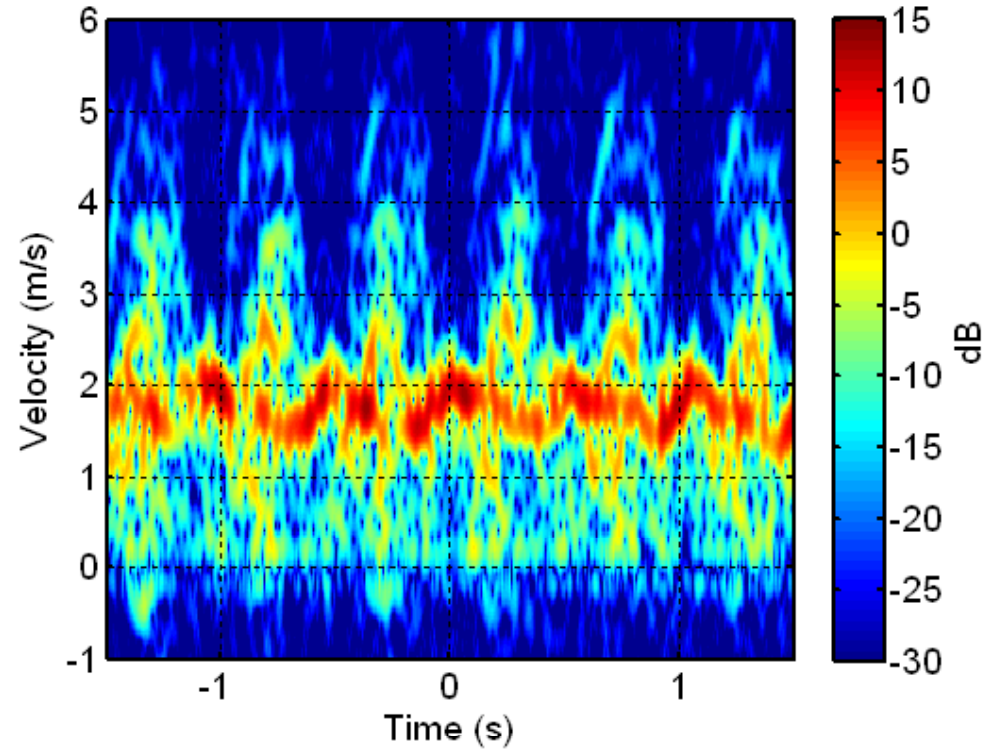


Comparison of Measurements and Simulation

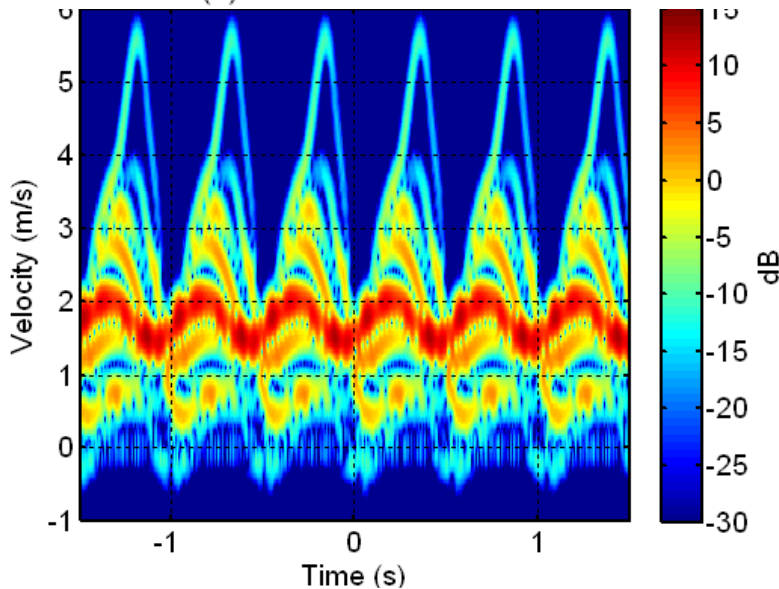
Fluctuating Simulation



Measurement



Stationary Simulation



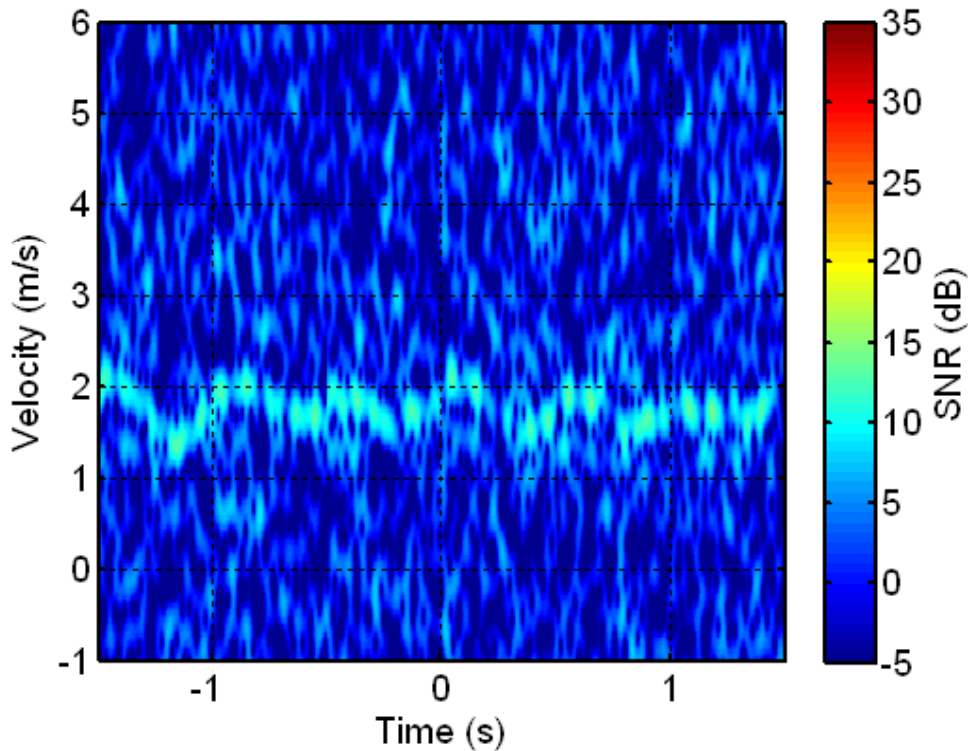
Measurements and Simulation Show an Apparent Match

Outline

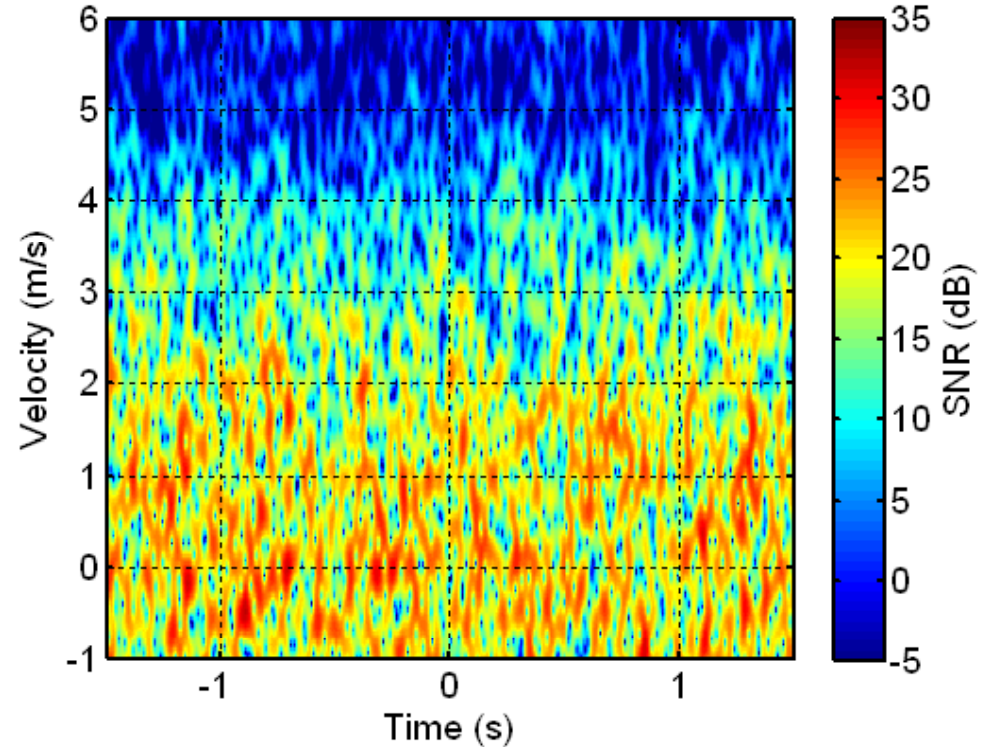
- Introduction
- Dismount modeling and measurements
- **Dismount algorithm development**
- Summary

Dismount Spectrograms for the Small UAV Radar System

Simulated Dismount with Noise

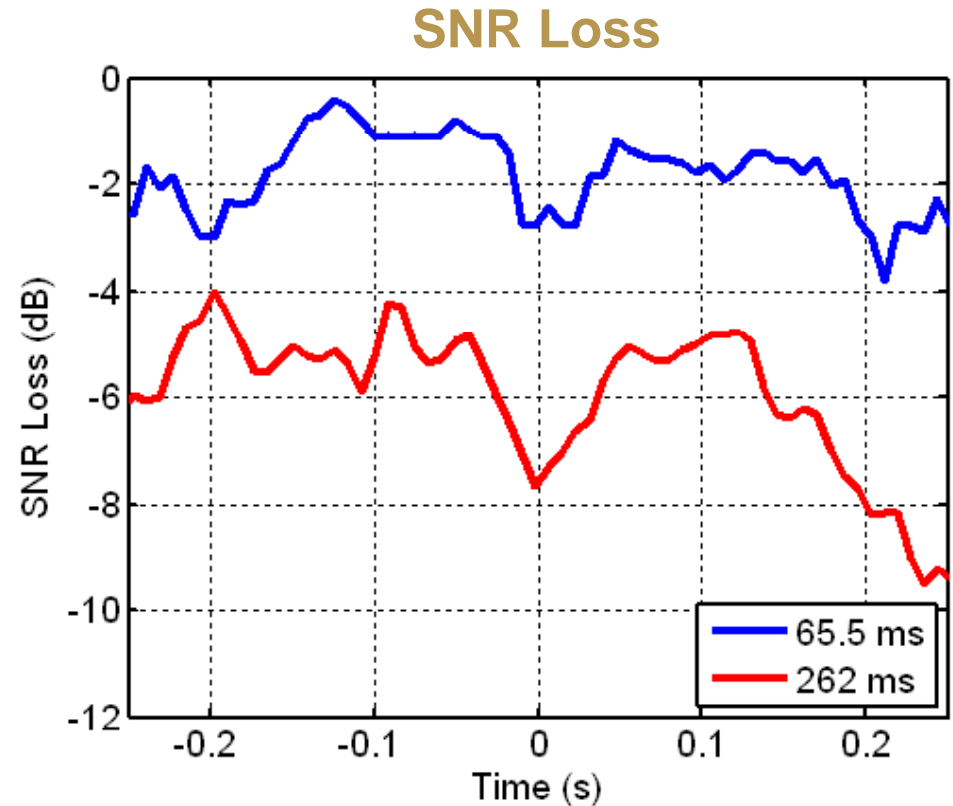
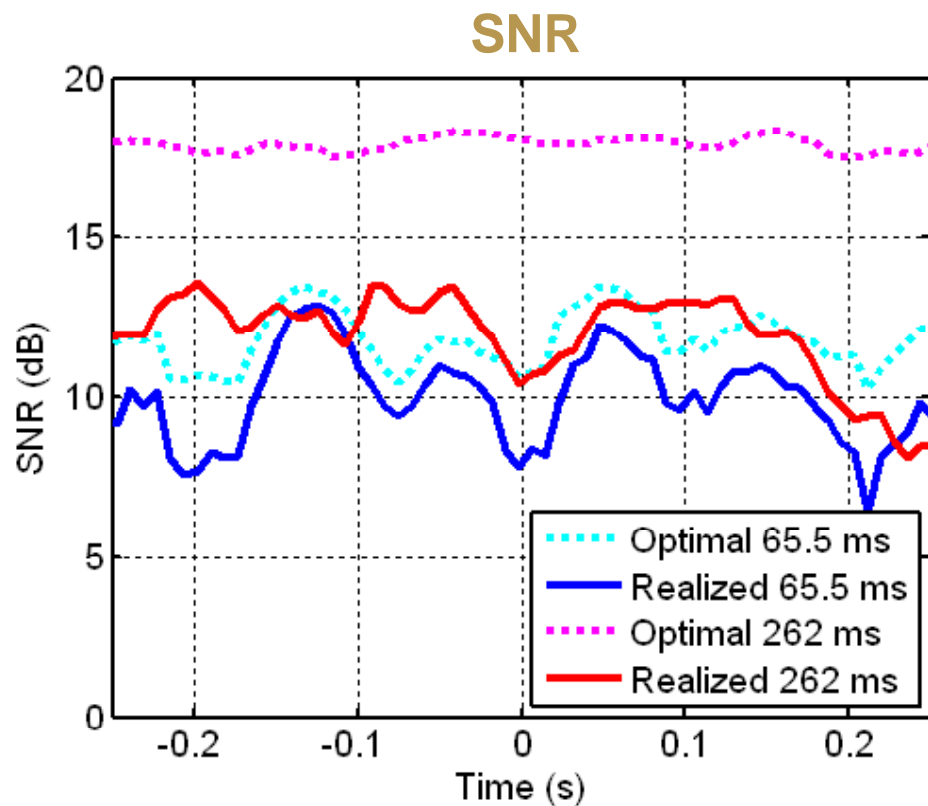


Simulated Dismount with Noise and Ground Clutter



Realized vs. Optimal SNR Using Linear Phase Filters

- Processing dismount signals with linear phase filters is generally sub-optimal
- The full integrated SNR is not achieved
- The loss between the optimal SNR and the realized SNR is defined as the SNR loss



Higher-Order Phase Filters

• Algorithms

- Linear phase (FFT)
- Quadratic phase

$$\phi(t) = \frac{4\pi}{\lambda} \left(v_0 t + \frac{a t^2}{2} \right)$$

- Cubic phase

$$\phi(t) = \frac{4\pi}{\lambda} \left(v_0 t + \frac{a_0 t^2}{2} + \frac{a_j t^3}{6} \right)$$

- Sinusoidal phase

$$\phi(t) = \frac{4\pi}{\lambda} \left(v_0 t + A \sin(2\pi f t + \alpha) \right)$$

• Dwell lengths

- 1/8 step (64 pulses, 65.5 ms)
- 1/2 step (256 pulses, 262 ms)
- 2 steps (1024 pulses, 1.05 s)

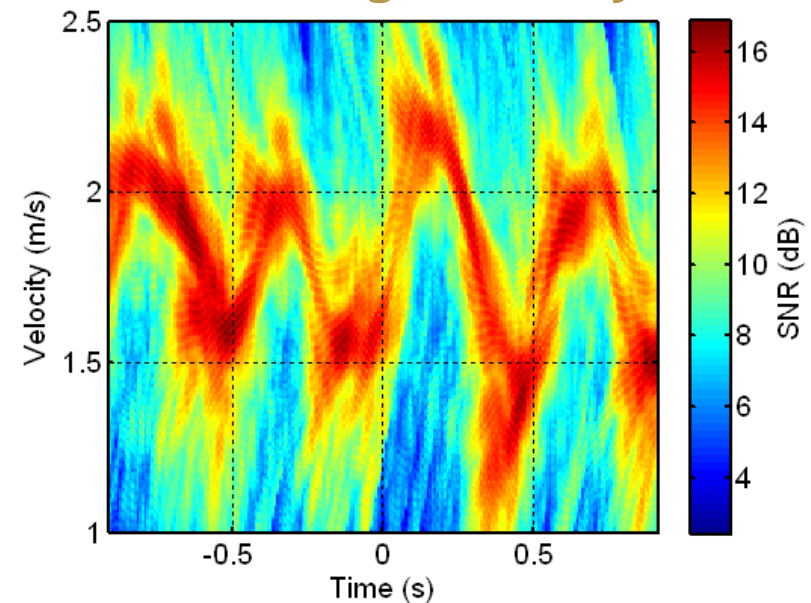
• Noise-limited performance (no ground clutter)

• Assume perfect motion compensation and no target migration through range cells

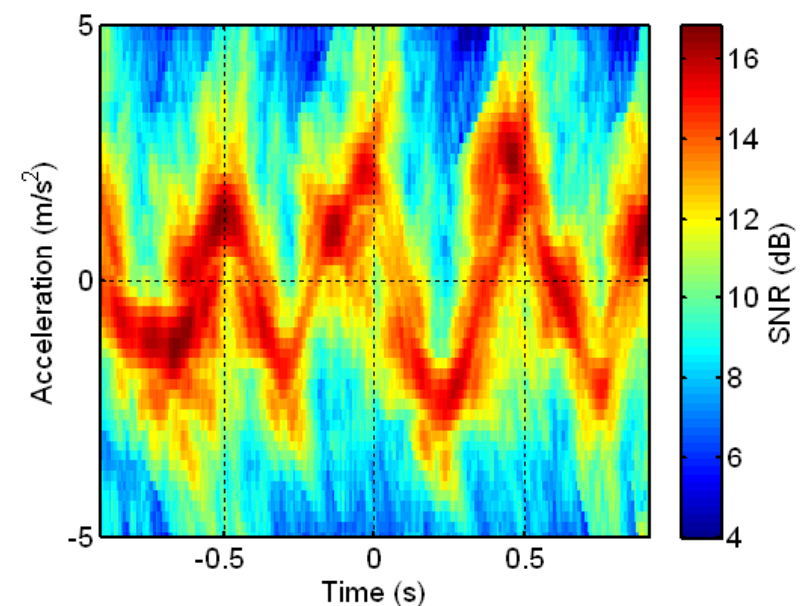
Quadratic Phase Filtering Example

- Example results for the 262 ms dwell (1/2 step)
- Requires testing over 2 parameters
 - Average velocity
 - Acceleration
- Current implementation tests all possible combinations of velocity and acceleration

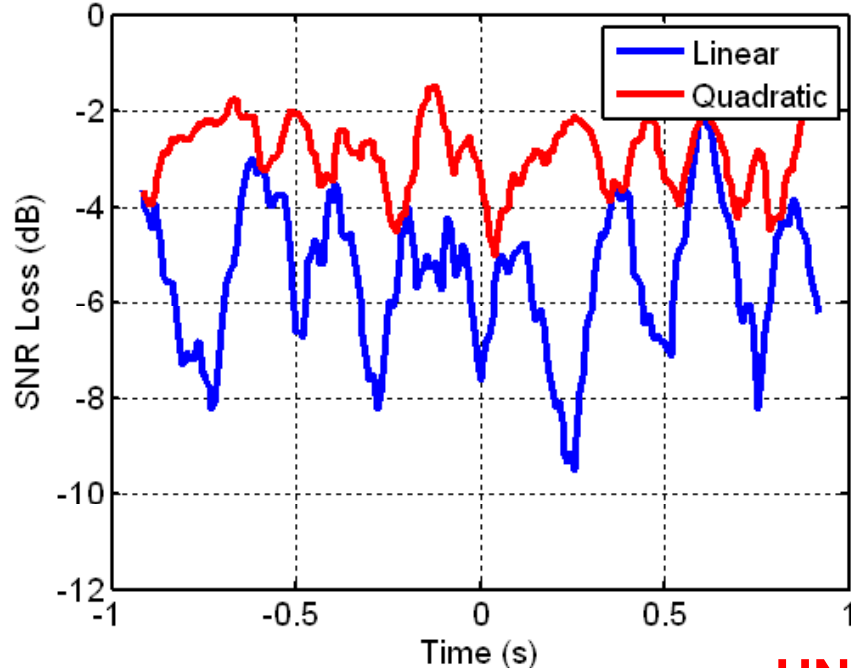
SNR over Average Velocity and Time



SNR over Acceleration and Time

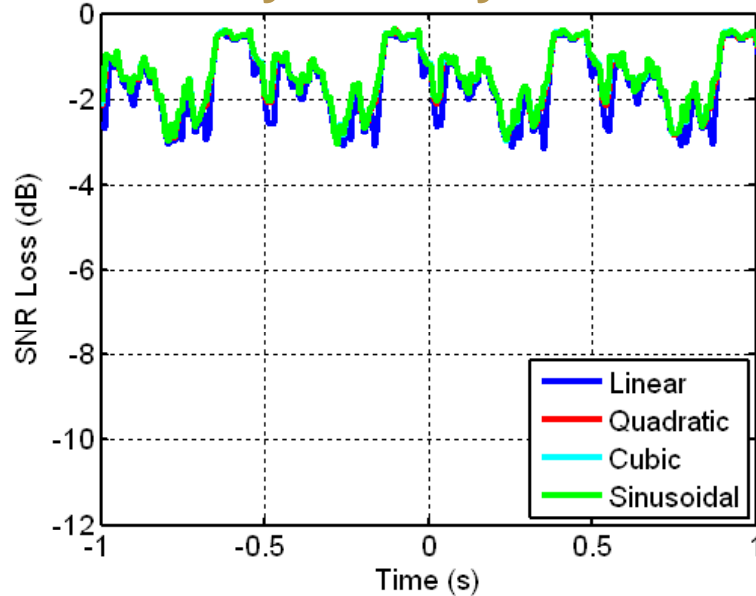


SNR Loss

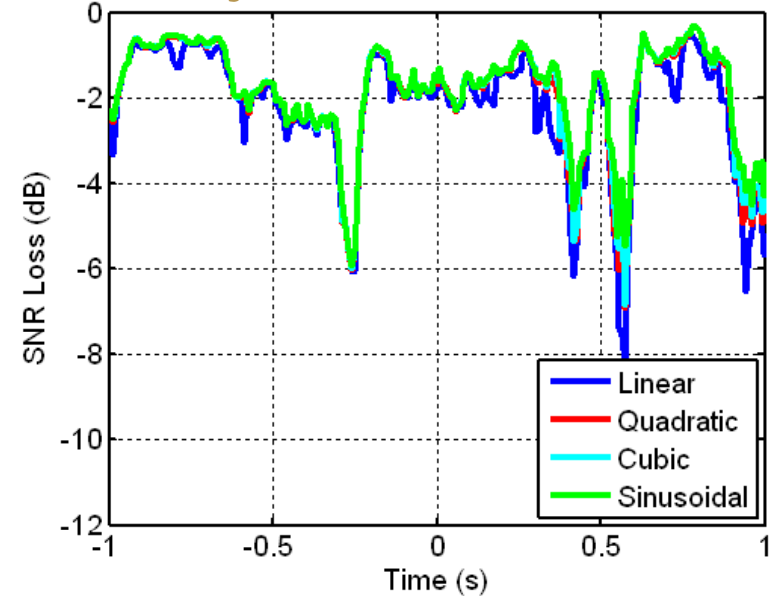


SNR Losses for a 1/8 Step Dwell (65.5 ms)

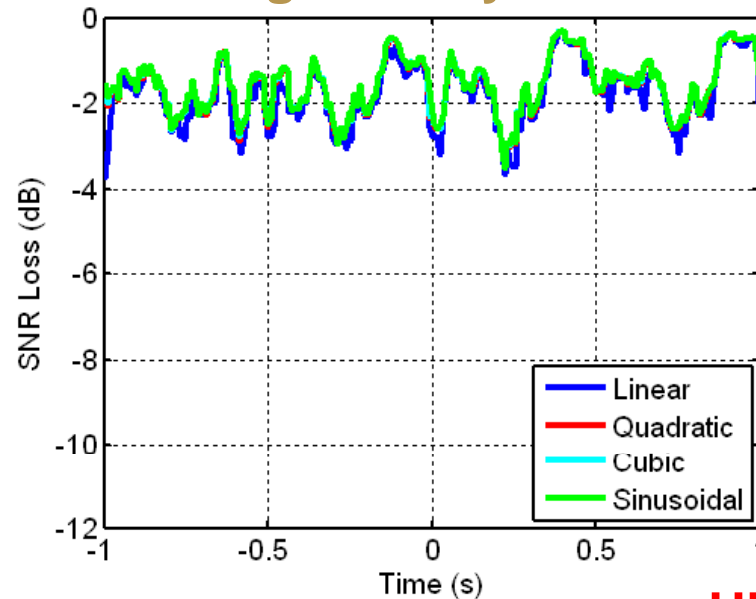
Stationary Velocity Simulation



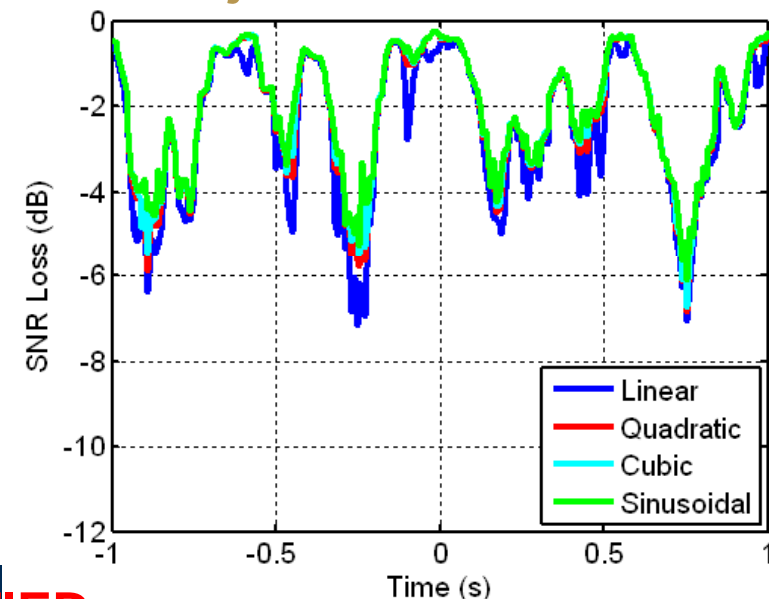
Subject 1 Measurement



Fluctuating Velocity Simulation

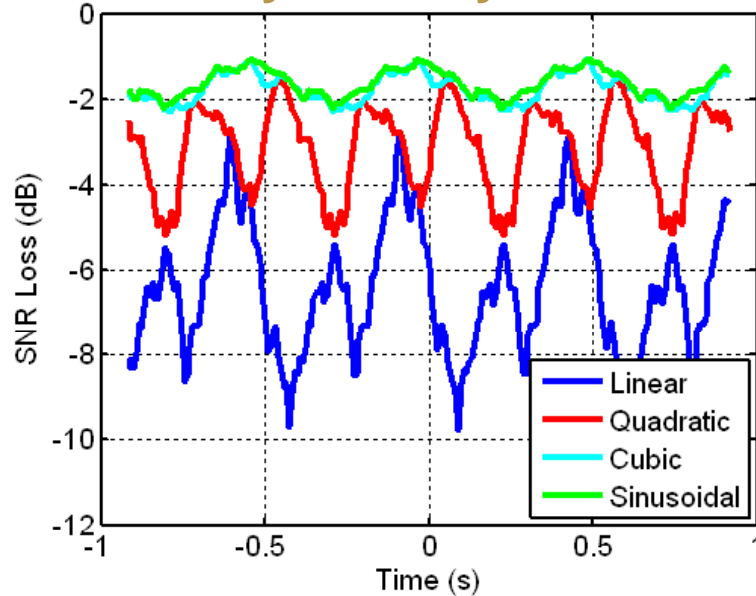


Subject 2 Measurement

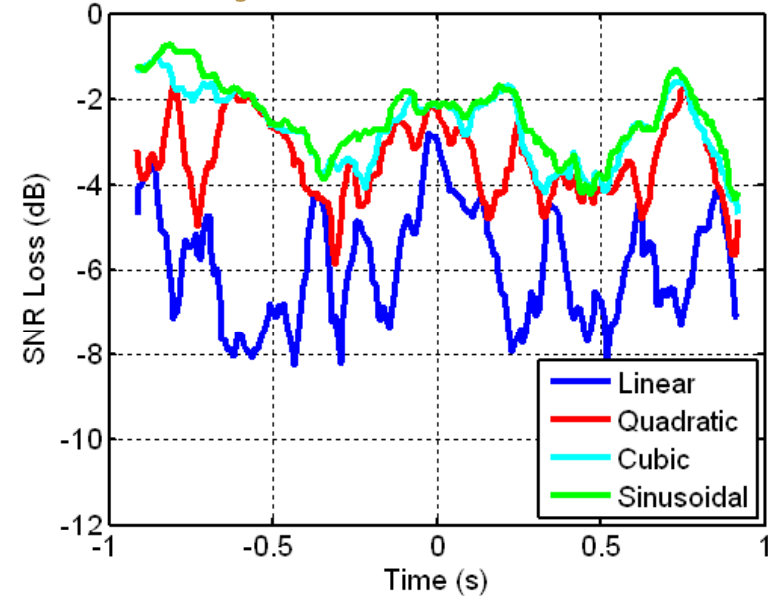


SNR Losses for a 1/2 Step Dwell (262 ms)

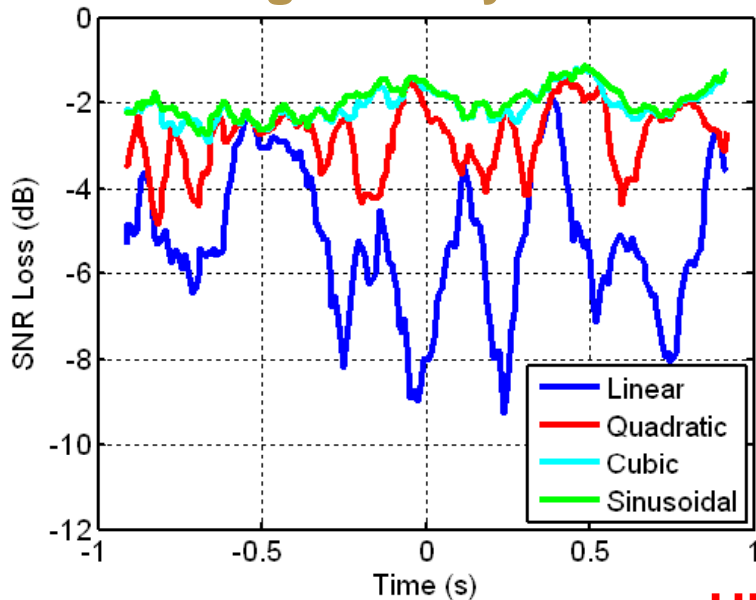
Stationary Velocity Simulation



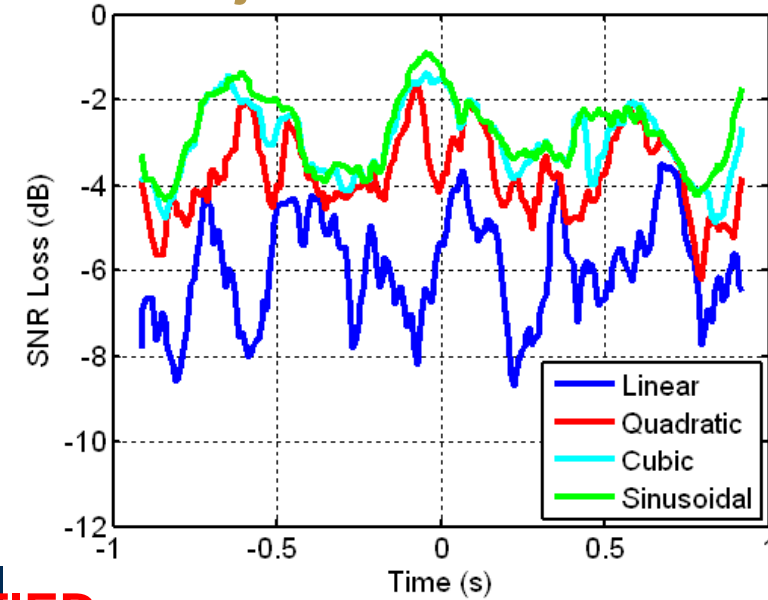
Subject 1 Measurement



Fluctuating Velocity Simulation

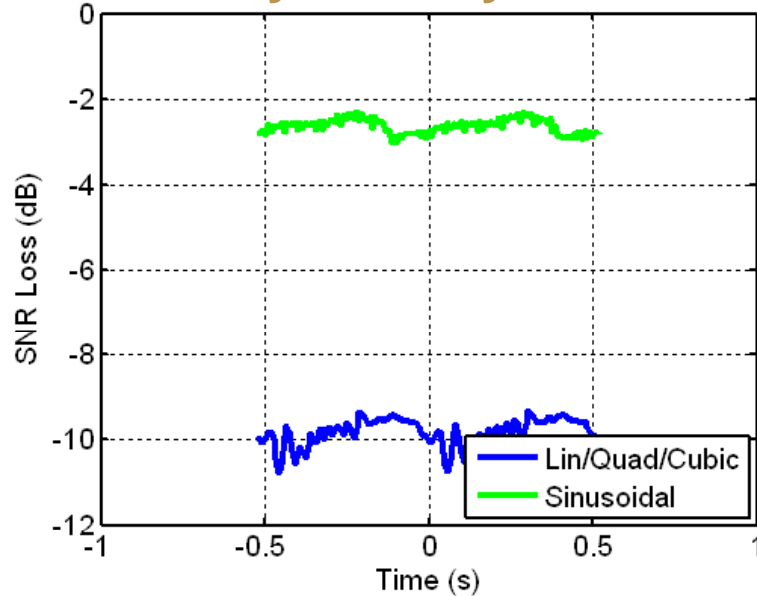


Subject 2 Measurement

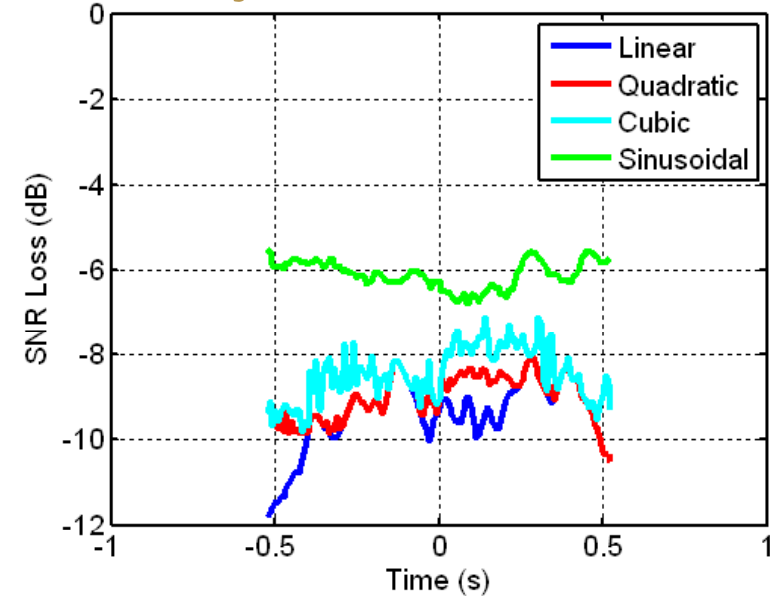


SNR Losses for a 2 Step Dwell (1048 ms)

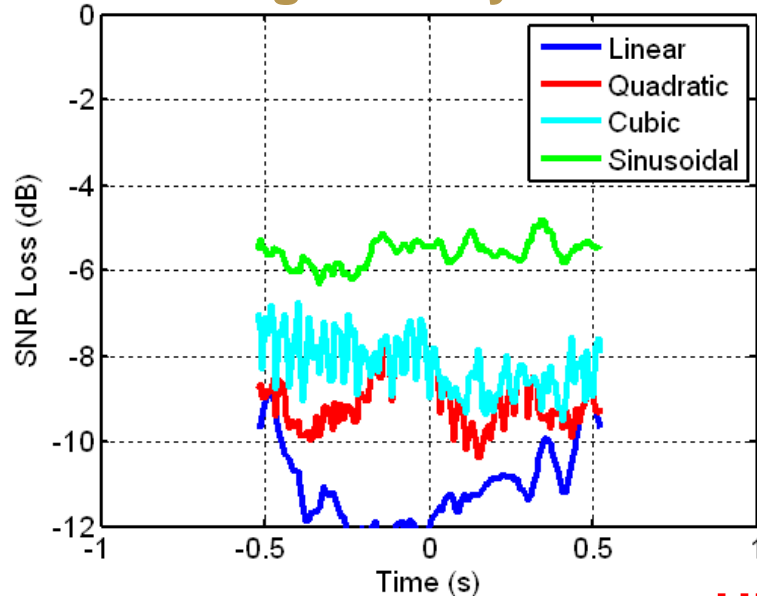
Stationary Velocity Simulation



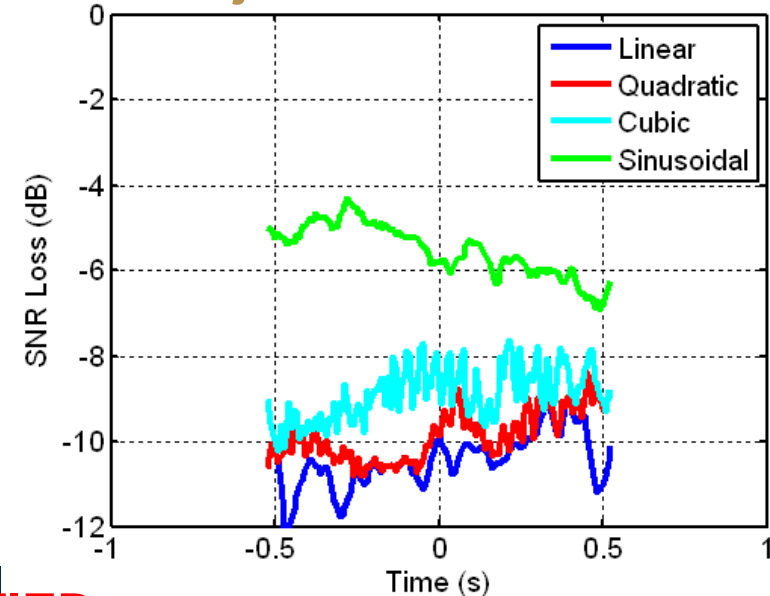
Subject 1 Measurement



Fluctuating Velocity Simulation



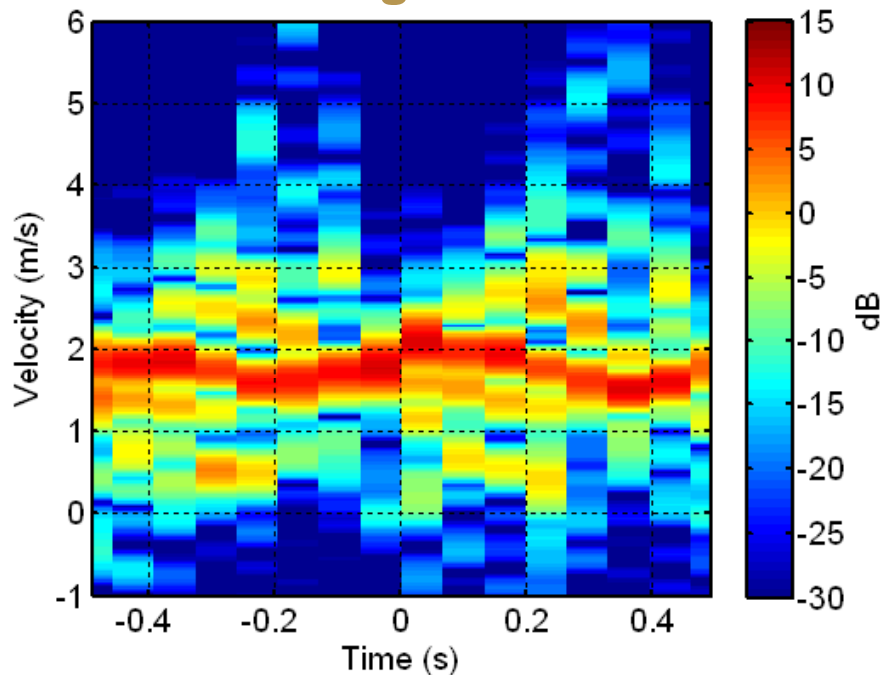
Subject 2 Measurement



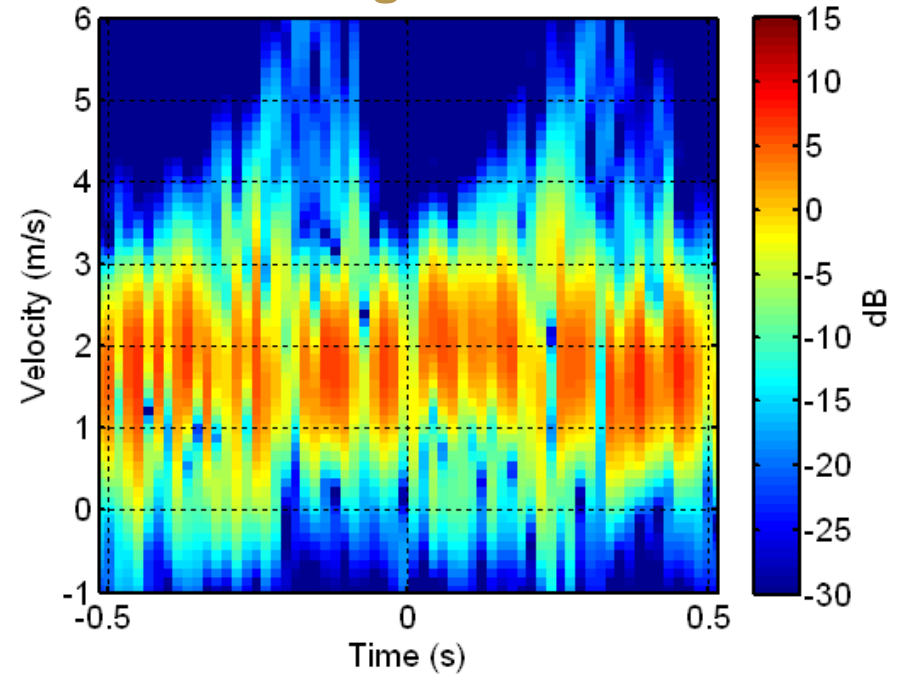
Post-Detection Integration (PDI)

- The complex phase histories of dismounts are difficult to match to using fully coherent integration techniques over long dwells
- PDI is a potential solution to this problem
 - Divide long dwell data into a series of short CPIs
 - Non-coherently combine Doppler bins from the short CPIs

CPI length: 64 pulses
PDI length: 16 CPIs



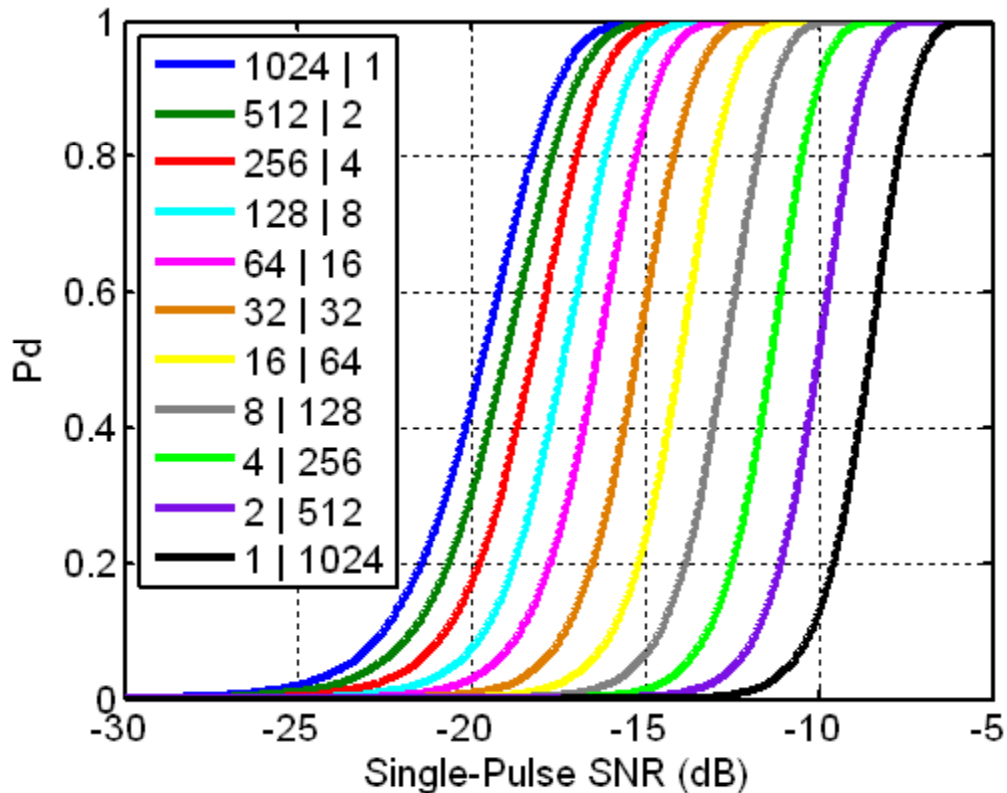
CPI length: 16 pulses
PDI length: 64 CPIs



PDI on a DC Signal

- PDI is less efficient than full coherent processing for a linear phase signal
- The greater the number of PDIs, the less efficient the PDI becomes
- PDI “steepens” the ROC curves

1024 Pulse DC Signal



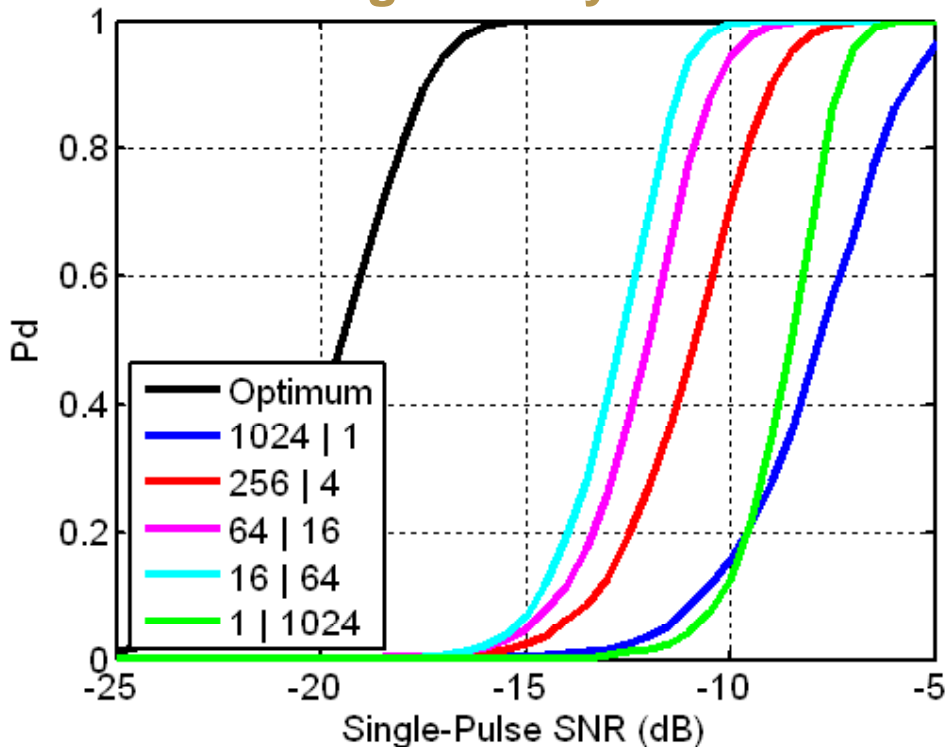
PDI Length	Optimal Gain (dB)	PDI Gain (dB)	PDI Loss (dB)	Efficiency
2	3.0	2.5	0.5	82%
4	6.0	4.8	1.2	80%
8	9.0	7.0	2.0	78%
16	12.0	9.2	2.9	76%
32	15.1	11.1	3.9	74%
64	18.1	13.0	5.1	72%
128	21.1	14.8	6.3	70%
256	24.1	16.5	7.5	69%
512	27.1	18.2	8.9	67%
1024	30.1	19.8	10.3	66%

*PDI gains calculated at a Pd of 90%

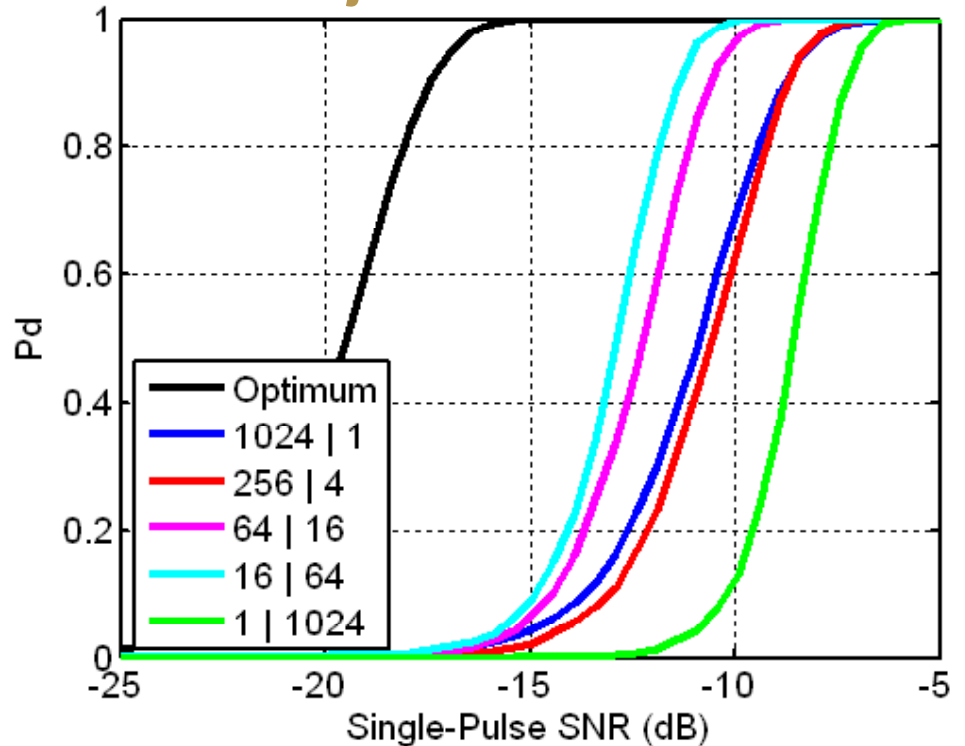
PDI on Dismount Signals

- Dwell length of 2 steps (1024 pulses)
- Best performance at a CPI length of 16 pulses (PDI length of 64)
 - 7 dB loss from optimum (2 dB from CPI, 5 dB from PDI)
- Similar performance for CPI length of 64 pulses (PDI length of 16)
 - 8 dB loss from optimum (2 dB from CPI, 3 dB from PDI, 3 dB from ??)

Fluctuating Velocity Simulation



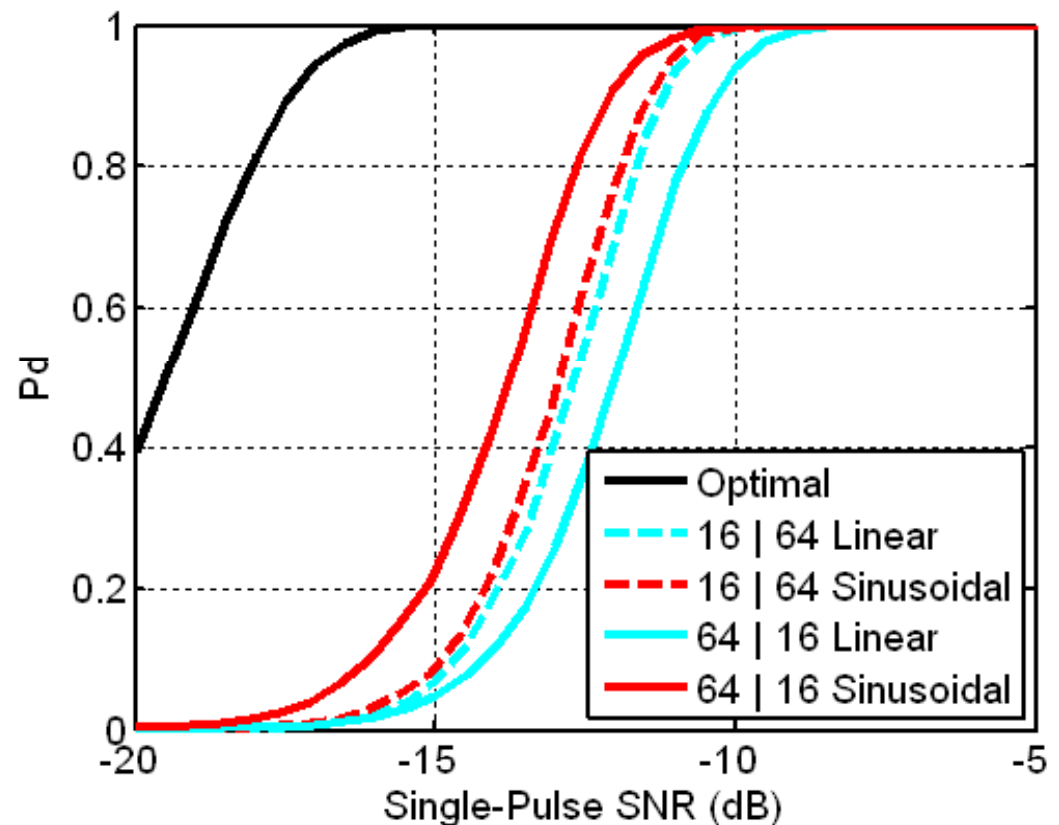
Subject 2 Measurement



Sinusoidal PDI

- Because of the time-varying properties of a dismount signal, the peak dismount power can move through Doppler bins over time
- Linearly combining Doppler bins is not optimal
- Potential solution: Use a sinusoid to combine Doppler bins
- 16 pulse CPI
 - Dismount does not significantly move through Doppler bins over time
 - Sinusoidal PDI shows little improvement in performance
- 64 pulse CPI
 - Dismount does move through Doppler bins over time
 - Sinusoidal PDI results in a 2 dB improvement in performance

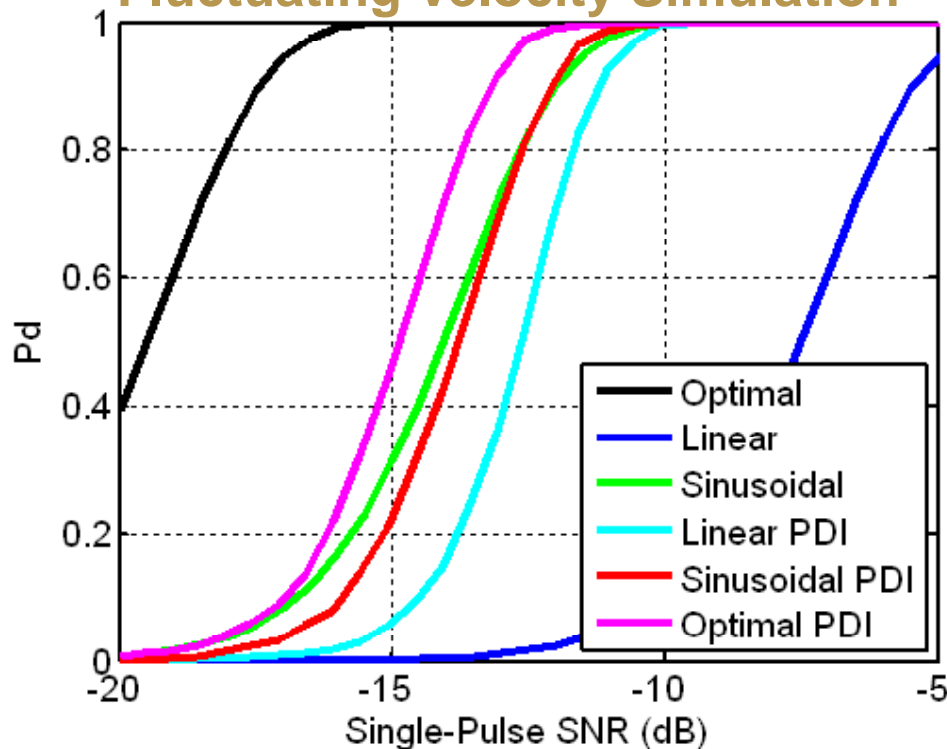
Fluctuating Velocity Simulation



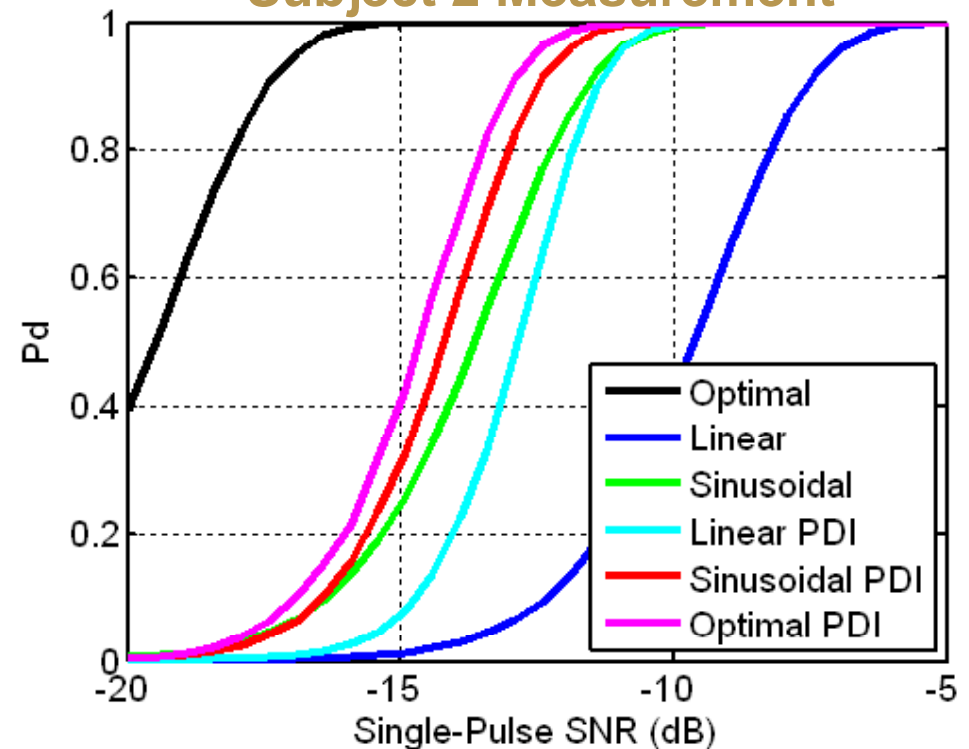
Overall Detection Performance

- Dwell length of 2 steps (1024 pulses)
- Sinusoidal PDI
 - Simulated data: Losses similar to the coherent sinusoidal filter
 - Measured data: Performance slightly better than sinusoidal filter
- Optimal PDI
 - Calculated using known peak Doppler bins
 - Losses of about 5 dB from optimum (2 dB from FFT and 3 dB from PDI)

Fluctuating Velocity Simulation



Subject 2 Measurement



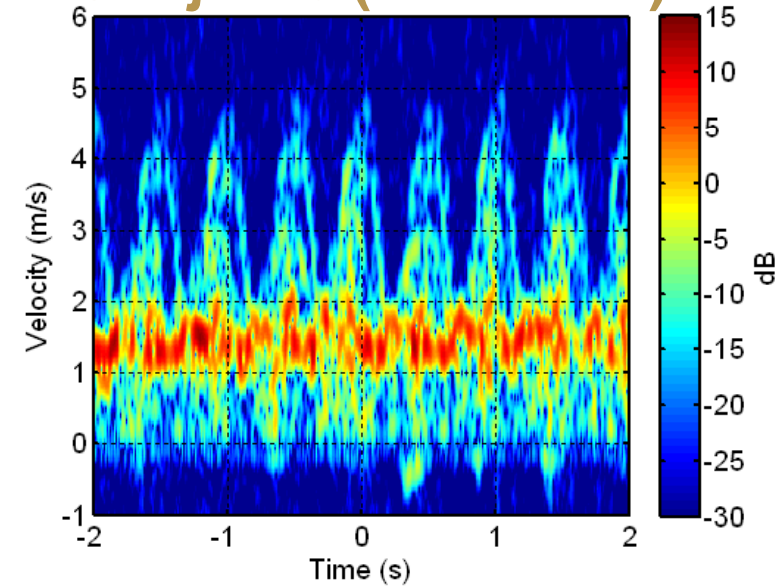
Characterization Potential

- **Gait characteristics vary from person to person**
- **The dismount torso moves with a nearly sinusoidal motion**
 - **Gait frequency: How often subject takes steps**
 - **Gait phase: When target starts taking steps**
 - **Gait amplitude: Degree of oscillation of the torso**
- **Micro-Doppler of arms and legs can also be considered**
- **Detection algorithms naturally estimate some gait parameters**
 - **Sinusoidal phase filtering**
$$\phi(t) = \frac{4\pi}{\lambda} (v_0 t + A \sin(2\pi f t + \alpha))$$
 - **Improves detection performance**
 - **Characterizes target**

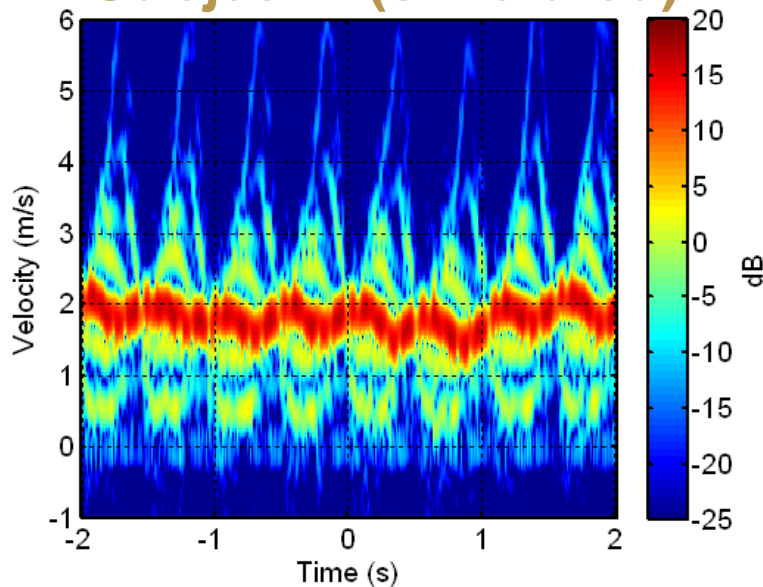
Gait Estimation of Different Subjects (1 of 3)

- Measured data from 3 different subjects
 - Subject 1: Simulated
 - Subject 2: Measured (similar to subject 1)
 - Subject 3: Measured
- Subjects walk radially toward the radar
- Subjects walk at their normal walking speed

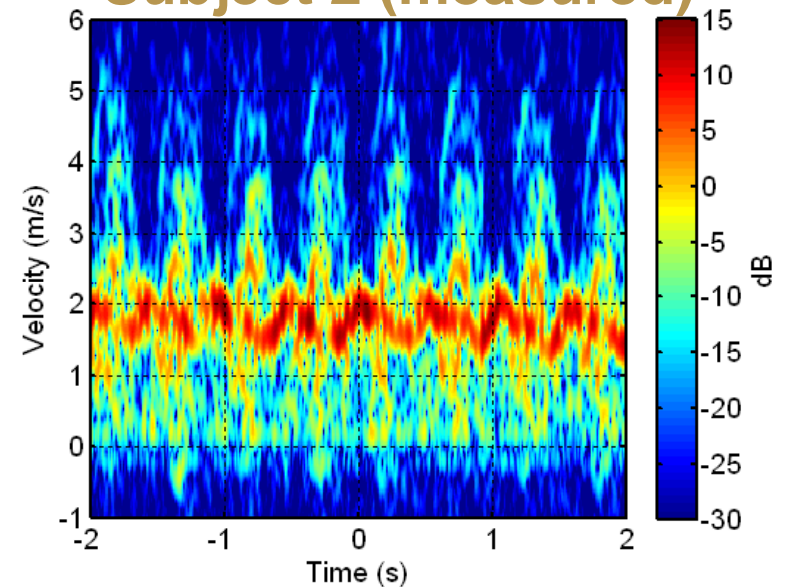
Subject 3 (measured)



Subject 1 (simulated)



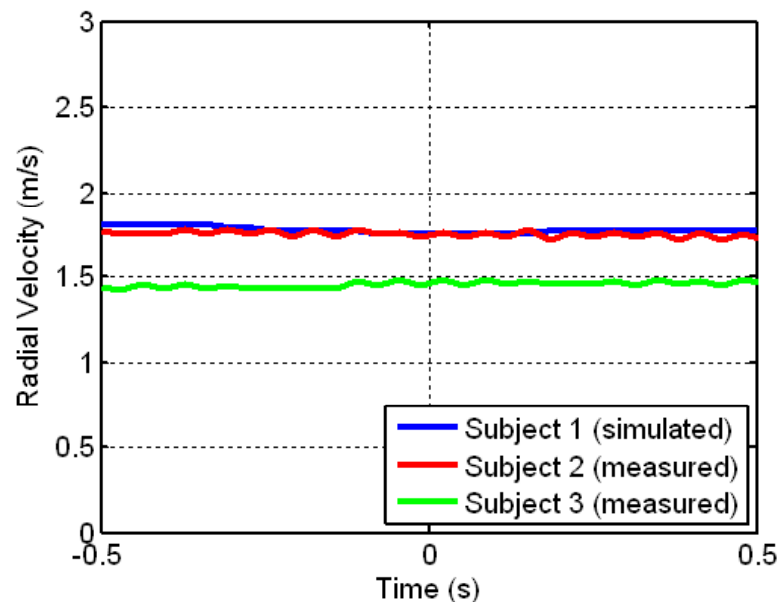
Subject 2 (measured)



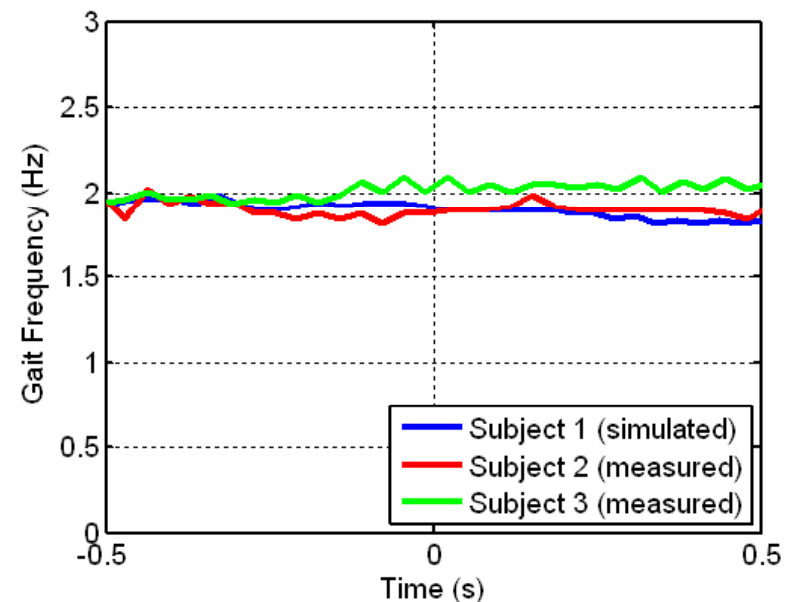
Gait Estimation of Different Subjects (2 of 3)

- Subjects 1 and 2 walk at similar speeds and have similar gait frequencies
- Subject 3 walks slower, but with a higher gait frequency
- Subject 3 is shorter than subjects 1 and 2 and must take faster steps to maintain the same walking speed

Radial Velocity Estimates

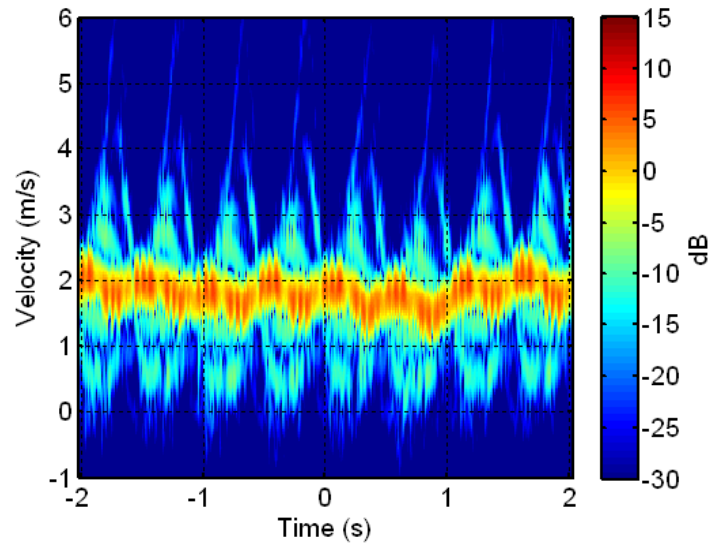


Gait Frequency Estimates

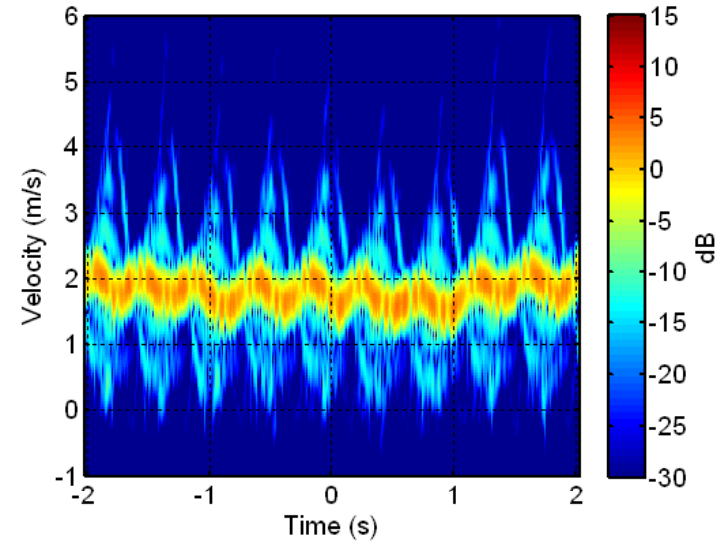


Gait Estimation of Different Subjects (3 of 3)

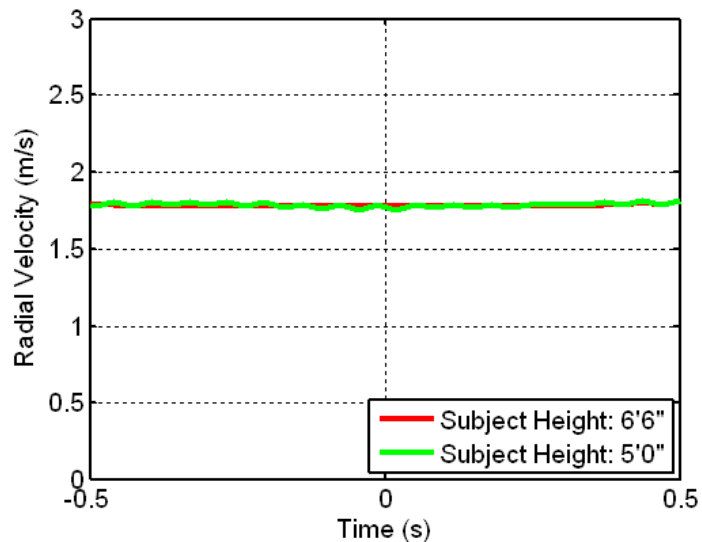
6'6" Height Subject



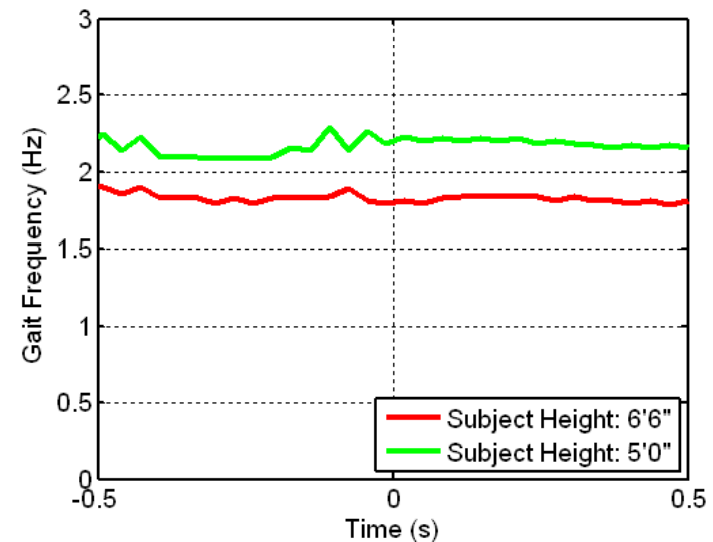
5'0" Height Subject



Radial Velocity Estimates



Gait Frequency Estimates



Outline

- Introduction
- Dismount modeling and measurements
- Dismount algorithm development
- **Summary**

Summary

- **Developed a physiological dismount model and integrated the model into a legacy simulation environment**
- **Losses due to linear phase filtering – a standard radar fare – are significant → 2 dB for 66 ms dwell, 6 dB for 262 ms dwell, 10 dB for 1.1 s dwell**
- **Effective dismount detection schemes must maximally aggregate energy**
- **Nonlinear phase filters show significant detection performance improvement → better match to actual target phase history**
- **Sinusoidal PDI shows improved detection performance and robustness**
- **Detection algorithms also show potential for characterization**

Lattice Boltzmann Method

Keith Prussing

Electro-Optical Systems Laboratory

Georgia Tech Research Institute

School of Physics

Keith.Prussing@gtri.gatech.edu

September 14, 2010

The Current Thermal Model

- The current method of solving the heat equation is based on a semi-implicit finite difference method.
- Modeled object is placed in isolation and may exchange radiatively with an implicit sky and ground.
- Multiple objects are modeled separately and do not thermally interact.
- Computation of radiation exchange factors is memory and time consuming.
- Solution procedure is not directly linked to the Scene-Simulation Database.

Radiative Exchange

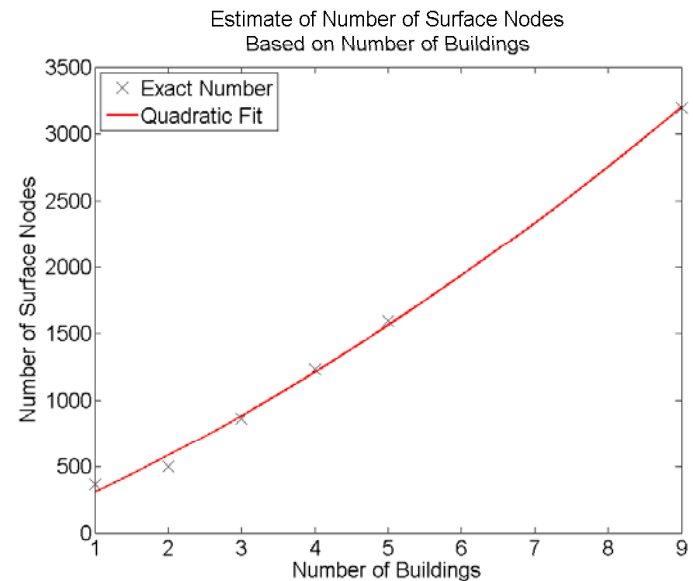
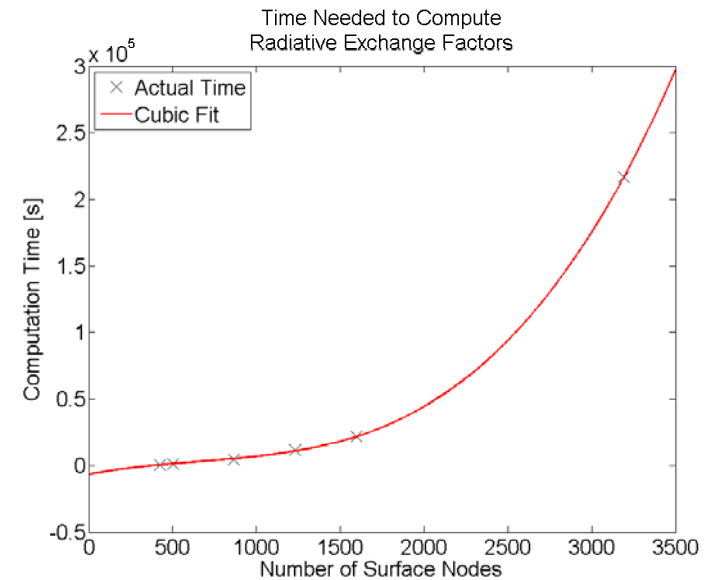
- **Current method uses ray tracing algorithm to find the geometric view factors.**
 - For each surface node, every other surface node must be visited.
 - If geometry changes, the exchange factors must be recomputed.
- **Current software has an upper limit on the number of surface nodes.**
 - Limit was placed when available memory was much smaller.
 - Simply changing the limit value and recompiling does not produce a working executable.
 - Limits the complexity of the scene making urban environments difficult to model.

Solutions Explored

- **Rewriting finite difference method:**
 - To handle complex scenes, implicit ground was removed.
 - Computation of radiative exchange was based on ray tracing algorithm just with more surface nodes.
- **Lattice Boltzmann method:**
 - Has been successfully applied to combined conduction and radiation problem in two dimensions (Asinari, 2010).
 - Since the formulation is readily adaptable to parallel processing, it has the potential to require dramatically lower computation time (Mishra, 2006).
 - This could become the numerical basis for a new solution procedure for the combined conduction, convection, and radiation problem in complex urban environments.

Increasing Allowed Number of Surface Nodes

- Time needed to compute total view factor increases as the number of nodes to the third power.
- Exchange with sky is over estimated due to ground not extending to infinity.
- Scene complexity is increased by requiring explicit ground to determine thermal shadows.



Lattice Boltzmann Method

- **Developed as a computational fluid dynamics algorithm by discretizing the Boltzmann equation (Succi, 2001).**
- **Physical properties are modeled by particle distribution function (PDF) (Mishra, Lankadasu, & Beronov, 2005).**
- **Macroscopic limit equations are recovered by examining how the PDFs behave when disturbed by a small perturbation (Ho, 2002).**
- **Solution procedure is alternating interaction and propagation steps (Mishra, Lankadasu, & Beronov, 2005).**

Lattice Boltzmann Method Advantages

- **Simple implementation (Wolf-Galadrow, 2005).**
- **Readily adaptable to parallel processing (Wolf-Galadrow, 2005).**
- **Handles complex geometry and boundary conditions (Wolf-Galadrow, 2005).**
- **Simple physical interpretation (Wolf-Galadrow, 2005).**
- **Unconditionally stable in linear regime when physical phenomena do not propagate faster than the speed of sound in the medium (Succi, 2001).**
- **Transient problem is solved directly with no iterations at each time step (Asinari, 2010).**

Lattice Boltzmann Method

$$\frac{\partial f_i}{\partial t} + \vec{e}_i \cdot \nabla f_i = \Omega_i$$

Kinetic Equation

f_i

Particle Distribution Function

\vec{e}_i

Lattice Velocity

$$|\vec{e}_i| = c_v$$

Lattice Speed

Ω_i

Collision Operator

$$T = \sum_i f_i$$

Temperature

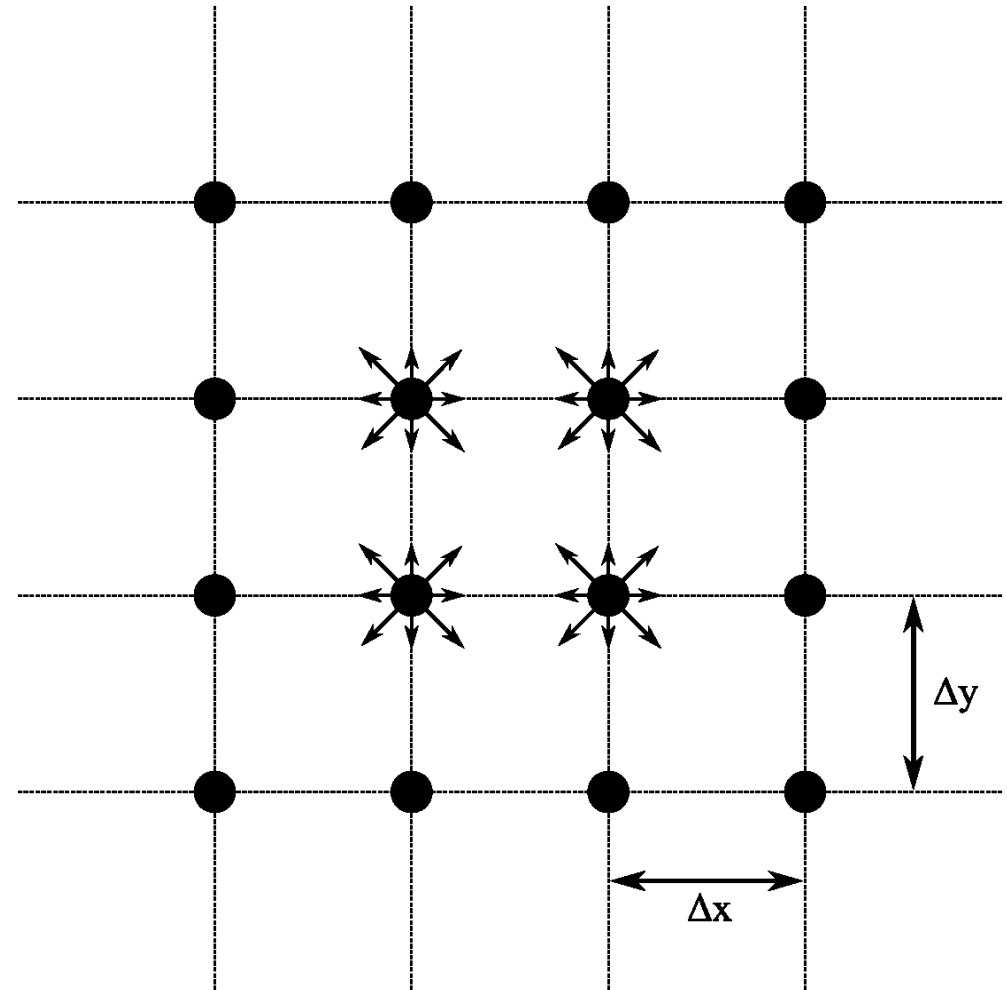
$$\vec{q} = \sum_i f_i \vec{e}_i$$

Heat flux

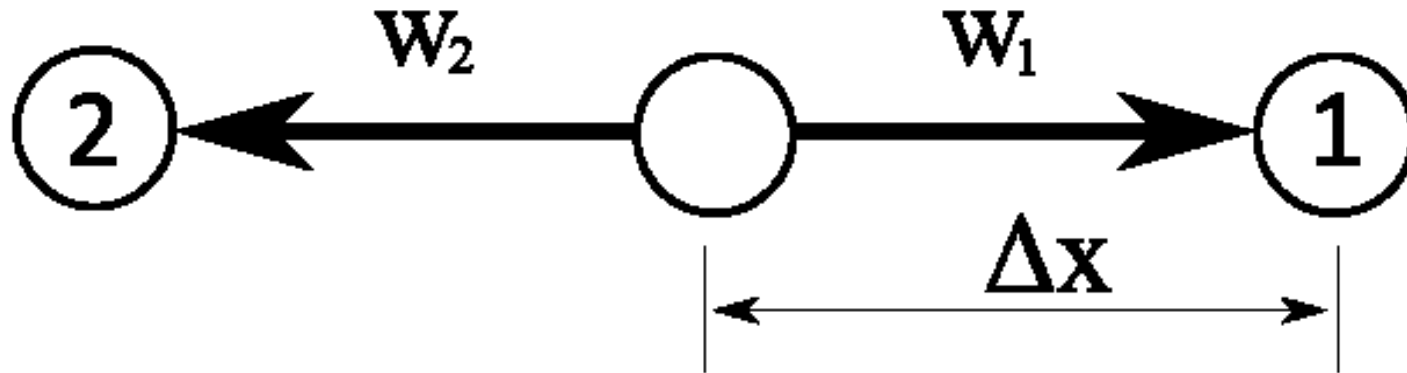
- Domain of interest is mapped onto a lattice.
- PDFs (Mishra, Lankadasu, & Beronov, 2005):
 - Exist at the lattice sites and evolve according to kinetic equation.
 - Propagate along lattice direction at lattice speed.
 - Interact at lattice sites according to collision operator.
- Temperature and heat flux are related directly to PDFs
- Collision operator contains all of the physics (Mishra, Lankadasu, & Beronov, 2005).

The Lattice

- **Explicit discretization of phase space.**
- **Physical phenomena are smoothed out over many lattice spacings (Wolf-Galadrow, 2005).**
- **Has particular velocities and weights regardless of physical phenomena modeled (Wolf-Galadrow, 2005).**
- **Must have sufficient symmetry to reproduce macroscopic phenomena (Wolf-Galadrow, 2005).**



D1Q2 Lattice



$$c_v = \frac{\Delta x}{\Delta t}$$

Lattice Speed

$$\tau = \frac{\alpha}{|c_v|^2} + \frac{\Delta t}{2}$$

Relaxation Constant

i	\vec{e}_i	w_i
1	$(c_v, 0)$	1/2
2	$(-c_v, 0)$	1/2

(Mishra & Roy, 2007)

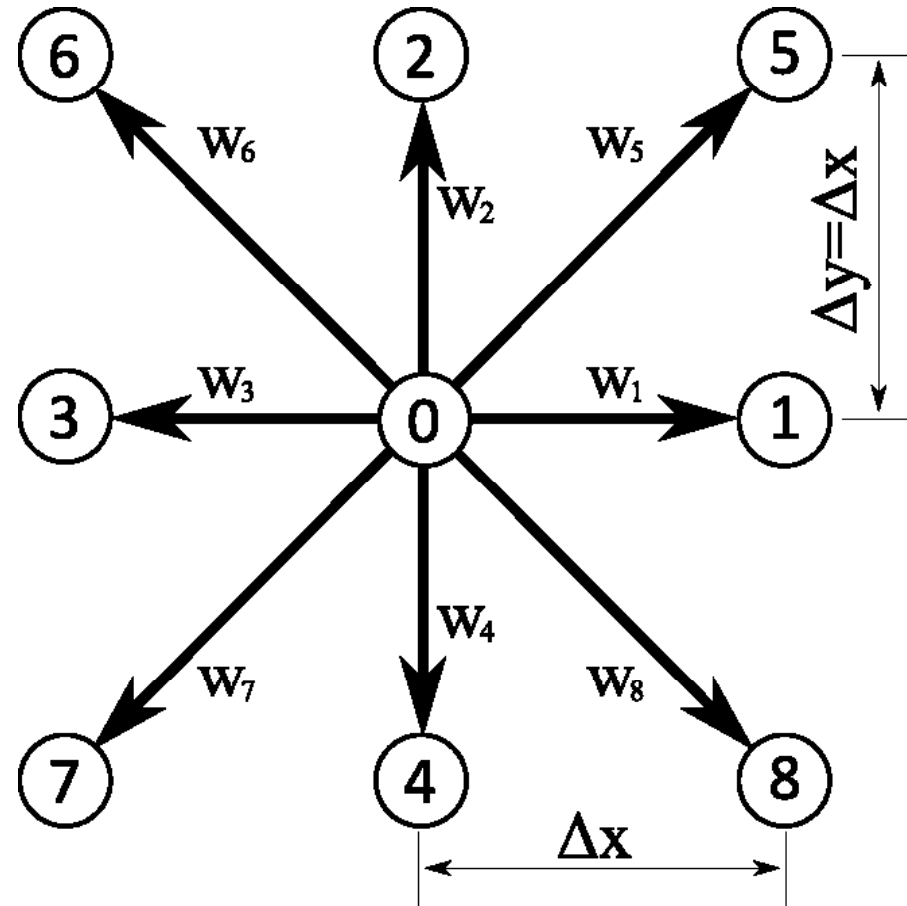
D2Q9 Lattice

$$c_v = \frac{\Delta x}{\Delta t}$$

$$\tau = \frac{3\alpha}{|c_v|^2} + \frac{\Delta t}{2}$$

Lattice Speed

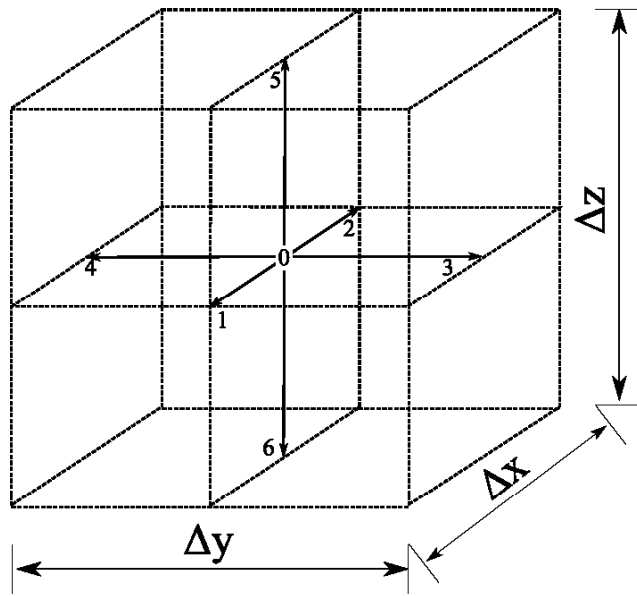
Relaxation Constant



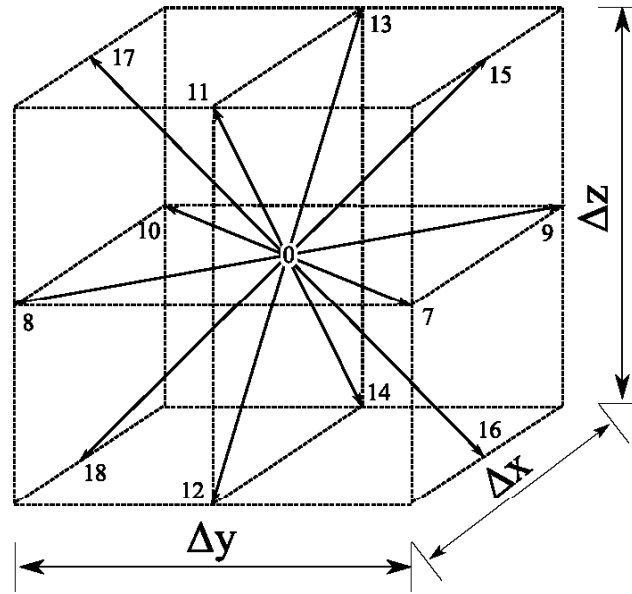
i	\vec{e}_i	w_i	i	\vec{e}_i	w_i	i	\vec{e}_i	w_i
0	(0, 0)	4/9	1	(c_v , 0)	1/9	2	(0, c_v)	1/9
3	(- c_v , 0)	1/9	4	(0, - c_v)	1/9	5	(c_v , c_v)	1/36
6	(- c_v , c_v)	1/36	7	(- c_v , - c_v)	1/36	8	(c_v , - c_v)	1/36

(Mishra & Roy, 2007)

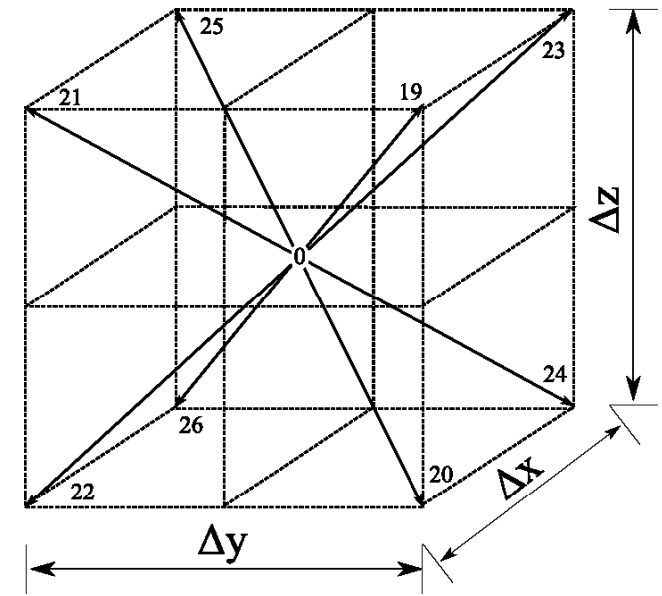
D3Q27 Lattice



c_v velocities



$c_v\sqrt{2}$ velocities



$c_v\sqrt{3}$ velocities

$$c_v = \frac{\Delta x}{\Delta t}$$

Lattice Speed

$$\tau = \frac{3\alpha}{|c_v|^2} + \frac{\Delta t}{2}$$

Relaxation Constant

i	\vec{e}_i	w_i	i	\vec{e}_i	w_i	i	\vec{e}_i	w_i
0	(0, 0, 0)	8/27	1	($c_v, 0, 0$)	2/27	2	($-c_v, 0, 0$)	2/27
3	(0, $c_v, 0$)	2/27	4	(0, $-c_v, 0$)	2/27	5	(0, 0, c_v)	2/27
6	(0, 0, $-c_v$)	2/27	7	($c_v, c_v, 0$)	1/54	8	($c_v, -c_v, 0$)	1/54
9	($-c_v, c_v, 0$)	1/54	10	($-c_v, -c_v, 0$)	1/54	11	($c_v, 0, c_v$)	1/54
12	($c_v, 0, -c_v$)	1/54	13	($-c_v, 0, c_v$)	1/54	14	($-c_v, 0, -c_v$)	1/54
15	(0, c_v, c_v)	1/54	16	(0, $c_v, -c_v$)	1/54	17	(0, $-c_v, c_v$)	1/54
18	(0, $-c_v, -c_v$)	1/54	19	(c_v, c_v, c_v)	1/216	20	($c_v, c_v, -c_v$)	1/216
21	($c_v, -c_v, c_v$)	1/216	22	($c_v, -c_v, -c_v$)	1/216	23	($-c_v, c_v, c_v$)	1/216
24	($-c_v, c_v, -c_v$)	1/216	25	($-c_v, -c_v, c_v$)	1/216	26	($-c_v, -c_v, -c_v$)	1/216

Kinetic Equation

- **BGK single relaxation time model used (Mishra, Lankadasu, & Beronov, 2005):**
 - Simplest representation of collision operator.
 - Uses largest relaxation time of system a relaxation constant, τ .
- **PDFs evolve to a simple form of the kinetic equation (Mishra, Lankadasu, & Beronov, 2005).**
- **A Chapman-Enskog expansion will recover the heat equation**

$$\Omega_i = \frac{1}{\tau} (f_i^{eq} - f_i)$$

$$f_i^{eq}(\vec{r}, t) = w_i T(\vec{r}, t)$$

$$f_i(\vec{r} + \vec{e}_i \Delta t, t + \Delta t) = f_i(\vec{r}, t) \dots$$

$$- \frac{\Delta t}{\tau} (f_i(\vec{r}, t) - f_i^{eq}(\vec{r}, t))$$

Ω_i	Collision operator
f_i	PDF
f_i^{eq}	Equilibrium PDF
τ	Relaxation constant
w_i	Direction weight
\vec{r}	Position
\vec{e}_i	Lattice velocity
T	Temperature

Chapman-Enskog Expansion

- PDFs are expanded to first order in a small parameter, ε (Ho, 1997).

$$f_i = f_i^{(0)} + \varepsilon f_i^{(1)} + O(\varepsilon^2)$$

- PDF at new time step is Taylor expanded about time t (Ho, 1997).

$$f_i(\vec{r} + \vec{e}_i \Delta t, t + \Delta t) = f_i(\vec{r}, t) + \Delta t \partial_t f_i + \Delta t e_{i\alpha} \partial_\alpha f_i + O(\Delta t^2)$$

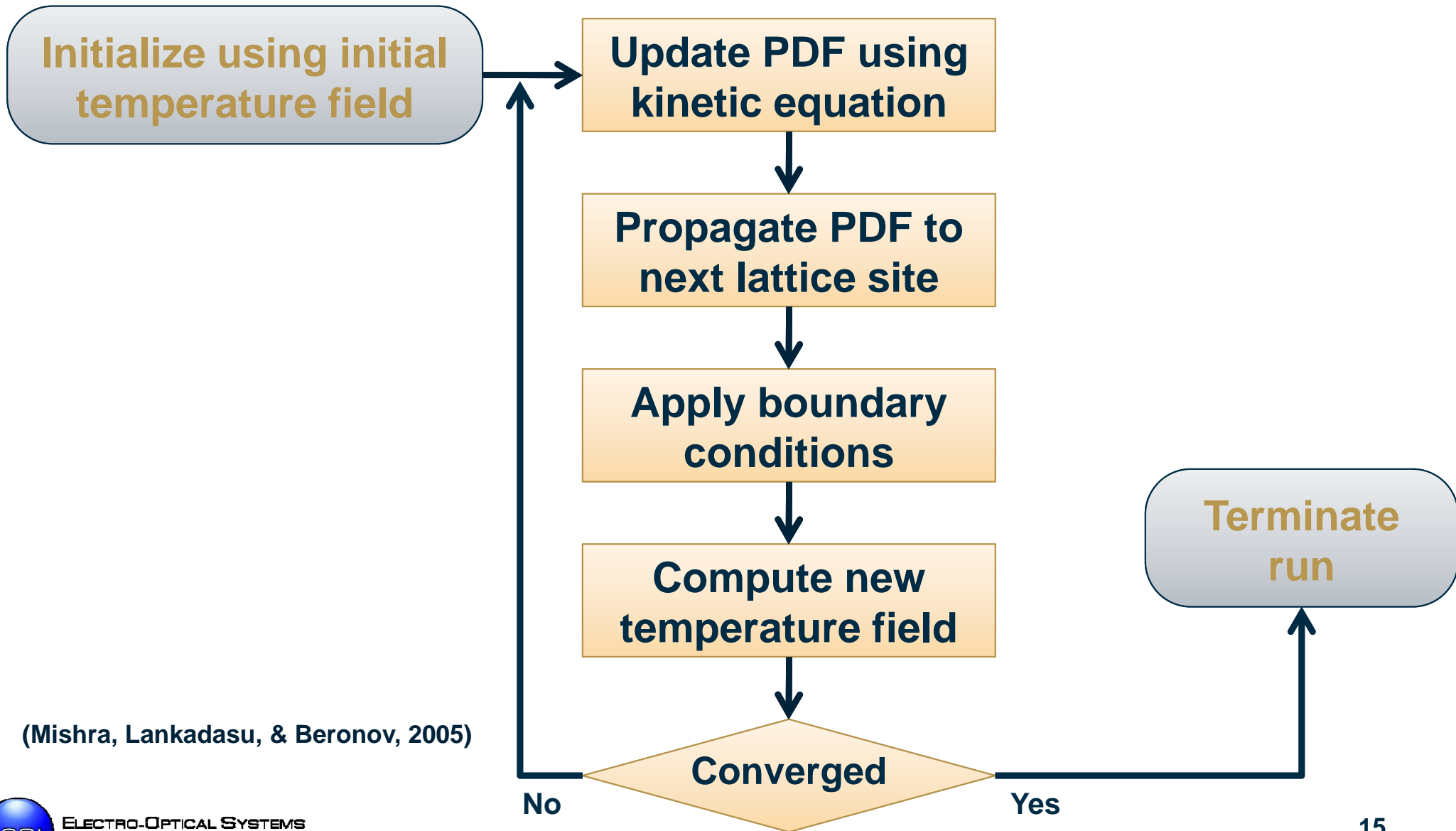
- Plugging both expansions into the kinetic equation and summing over the lattice velocities reproduces the heat equation (Ho, 1997).

$$\partial_t \sum_i f_i^{(0)} + \partial_\beta \sum_i e_{i\beta} f_i^{(0)} = -\varepsilon \partial_t \sum_i f_i^{(1)} - \varepsilon \partial_\beta \sum_i e_{i\beta} f_i^{(1)} - \frac{\varepsilon}{\tau} \sum_i f_i^{(1)}$$

$$\partial_t T + \nabla \cdot \vec{q} = 0$$

$$\partial_t T - \alpha \nabla^2 T = 0$$

The Algorithm



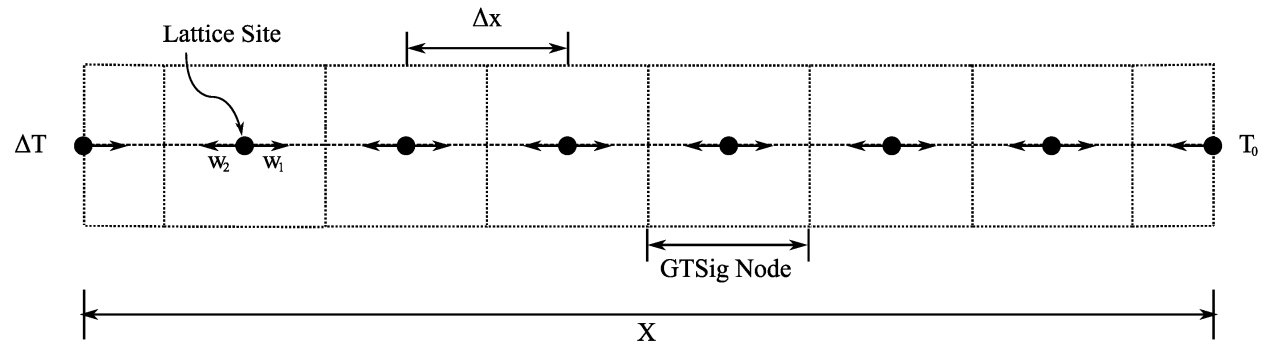
(Mishra, Lankadasu, & Beronov, 2005)

Evaluation of Lattice Boltzmann Method

- **Serial version of lattice Boltzmann method was implemented to evaluate accuracy**
- **Analytic solutions were found for simple conduction problems in one, two, and three dimensions**
- **Results from lattice Boltzmann method were compared to analytic and GTSig numerical results**
- **Numerical results were found to be consistent with GTSig; however, computation time was high due to poor implementation of the algorithm**

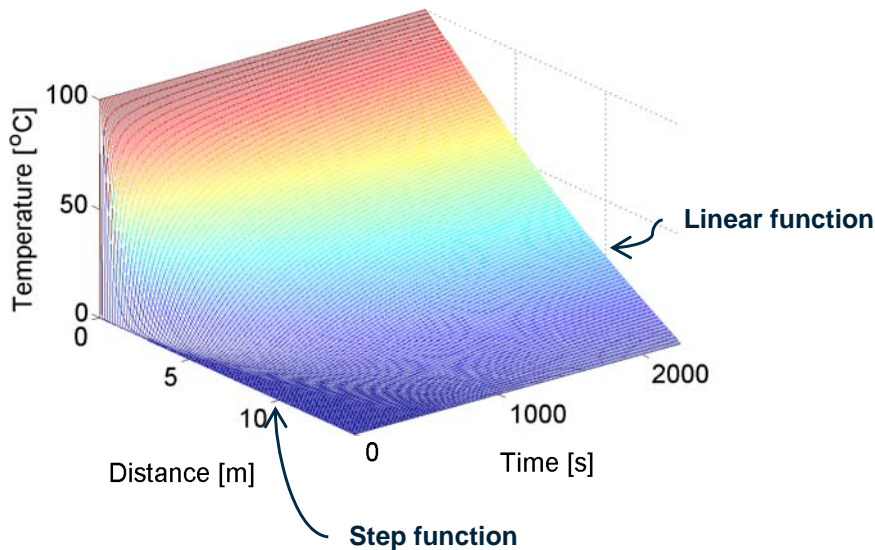
Fixed Wall Temperature in One Dimension

- Homogeneous object with uniform cross section
- West wall, $x=0$, temperature suddenly elevated



Analytic Solution as a Function of Space and Time

$$T(x,t) = T_0 + \Delta T \left\{ 1 - \frac{x}{X} - \sum_{n=1}^{\infty} \frac{2}{n\pi} \sin\left(n\pi \frac{x}{X}\right) e^{-\alpha n^2 \pi^2 t / X^2} \right\}$$

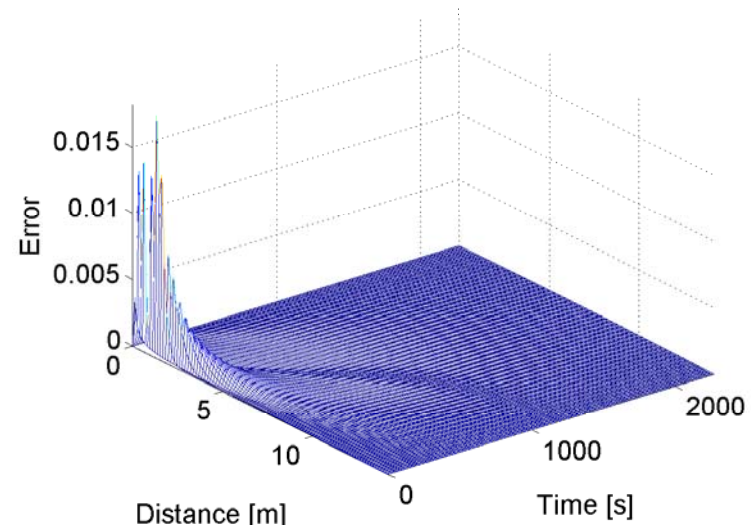


$\alpha = 0.02$	$\frac{\text{m}^2}{\text{s}}$	Thermal Diffusivity
$X = \sqrt{200}$	m	Domain Size
$T_0 = 0$	$^{\circ}\text{C}$	Initial Temperature
$\Delta T = 100$	$^{\circ}\text{C}$	West wall Temperature
$\Delta x = \sqrt{0.02}$	m	Lattice Separation
$\Delta t = 1$	s	Time Step

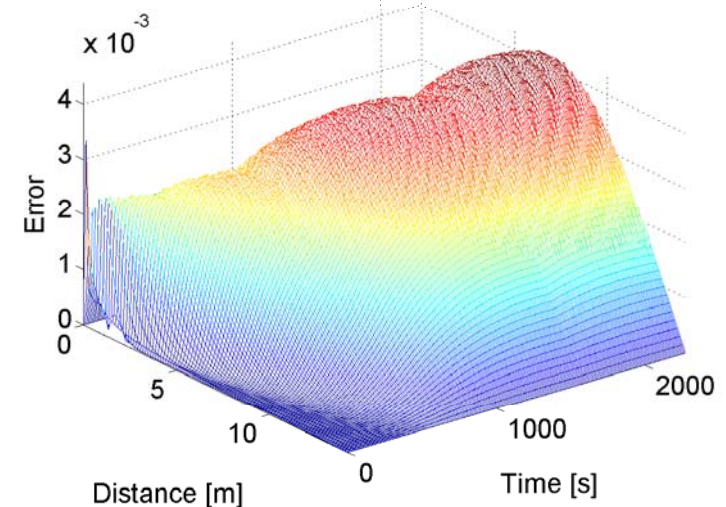
Fixed Wall Temperature in One Dimension

- Absolute error between analytic and numerical solutions was taken to be the difference between the analytic solution and the numerical normalized by the west wall temperature.
- We can see that the lattice Boltzmann method starts the simulation with a higher error but rapidly decreases.
- We anticipate that error is due to the step function nature of the boundary conditions.

LBM Absolute Error

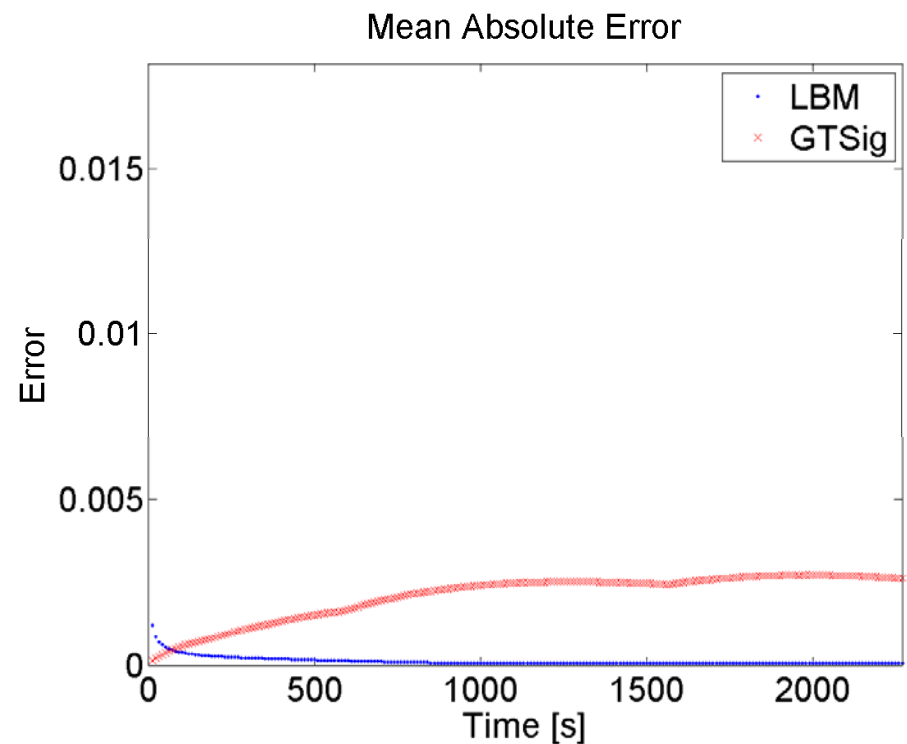


GTSig Absolute Error



Fixed Wall Temperature in One Dimension

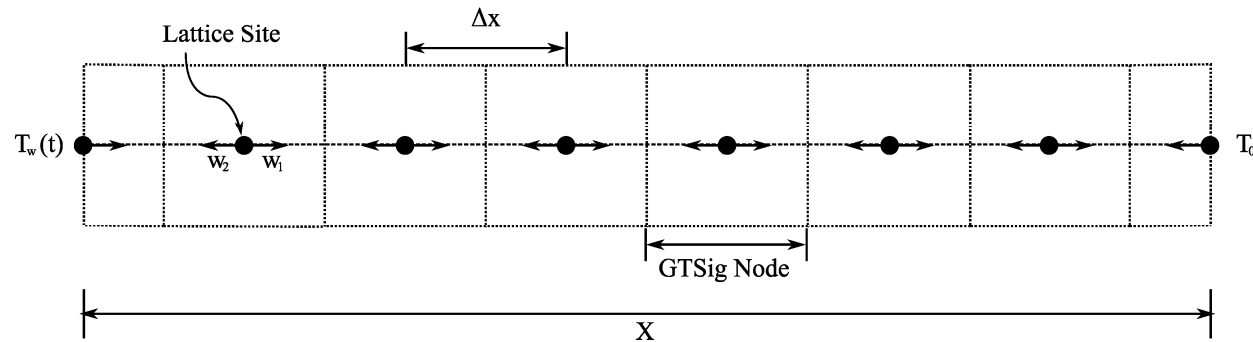
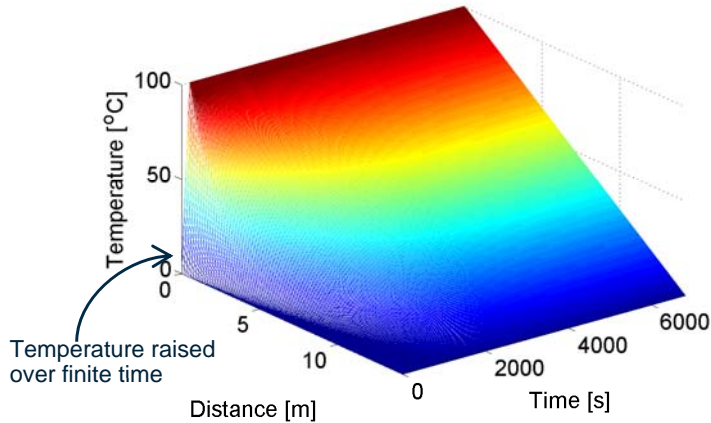
- Taking the mean absolute error over the domain at each time point, we see that the error in the lattice Boltzmann method quickly approaches zero; whereas the error in the GTSig results increases.
- We expect the error to be due to the unphysical nature of the west wall boundary condition.



Numerical Method	Time Elapsed [s]
GTSig: Finite Difference	1.529976
Lattice Boltzmann Method	2.963014

Time Dependent Wall Temperature in One Dimension

Analytic Solution as a Function of Space and Time



$$T(0, t) = \begin{cases} \Delta T \frac{t}{t_0} + T_0 & 0 \leq t \leq t_0 \\ \Delta T + T_0 & t_0 \leq t \end{cases}$$

West wall, x=0, temperature raised over finite time

- $\alpha = 0.02$ $\frac{\text{m}^2}{\text{s}}$ Thermal Diffusivity
- $X = \sqrt{200}$ m Domain Size
- $T_0 = 0$ °C Initial Temperature
- $\Delta T = 100$ °C West wall Temperature
- $\Delta x = \sqrt{0.02}$ m Lattice Separation
- $\Delta t = 1$ s Time Step
- $t_0 = 200$ s Turn on time

$$T(x, t) = T_0 + \Delta T \left[\left(1 - \frac{x}{X}\right) \theta_I(t) - \sum_{n=1}^{\infty} \frac{2X^2}{n^3 \pi^3 \alpha t_0} f_n(t) \sin\left(n\pi \frac{x}{X}\right) \right]$$

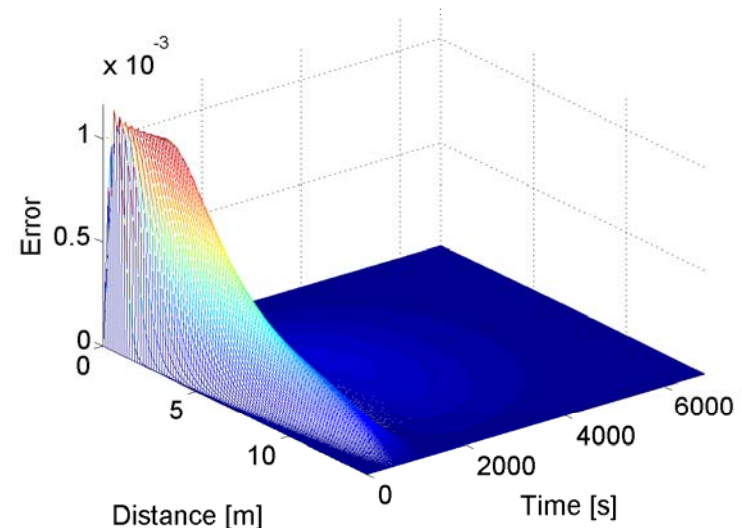
$$\theta_I(t) = \begin{cases} \frac{t}{t_0} & 0 \leq t \leq t_0 \\ 1 & t_0 \leq t \end{cases}$$

$$f_n(t) = \begin{cases} 1 - e^{-cn^2 \pi^2 t / X^2} & 0 \leq t \leq t_0 \\ e^{-cn^2 \pi^2 (t_0 - t) / X^2} - e^{-cn^2 \pi^2 t / X^2} & t_0 \leq t \end{cases}$$

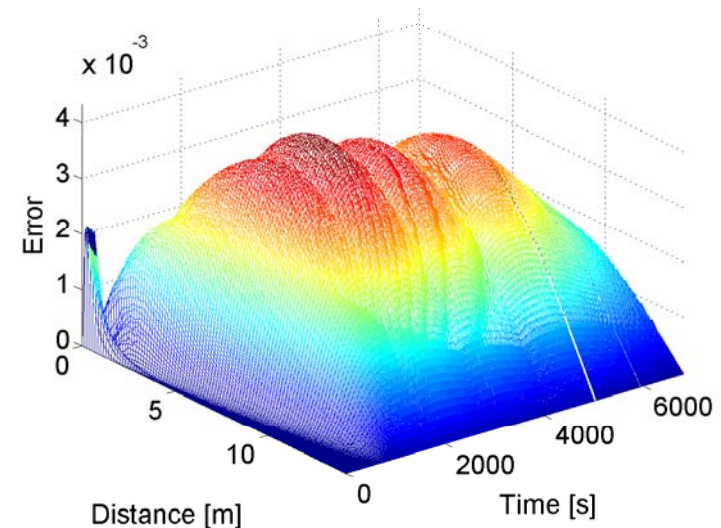
Time Dependent Wall Temperature in One Dimension

- We note that the error in the lattice Boltzmann method has dropped by an order of magnitude to the same level as the GTSig results.
- Comparing the GTSig results with the previous example, we see that the error is approximately the same.
- We can also see that the oscillatory behavior has been removed from the lattice Boltzmann results as expected.

LBM Absolute Error

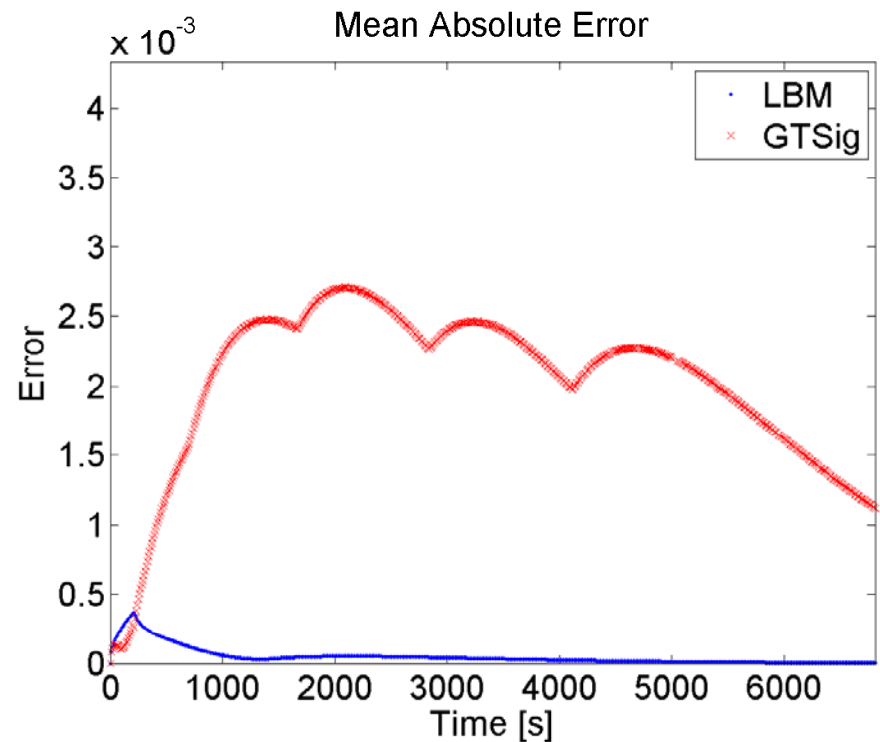


GTSig Absolute Error



Time Dependent Wall Temperature in One Dimension

- Looking at the mean error, we again see that the lattice Boltzmann method rapidly approaches the analytic solution.
- We also see that the GTSig solution is approaching the analytic solution but at a slower rate than the lattice Boltzmann method.

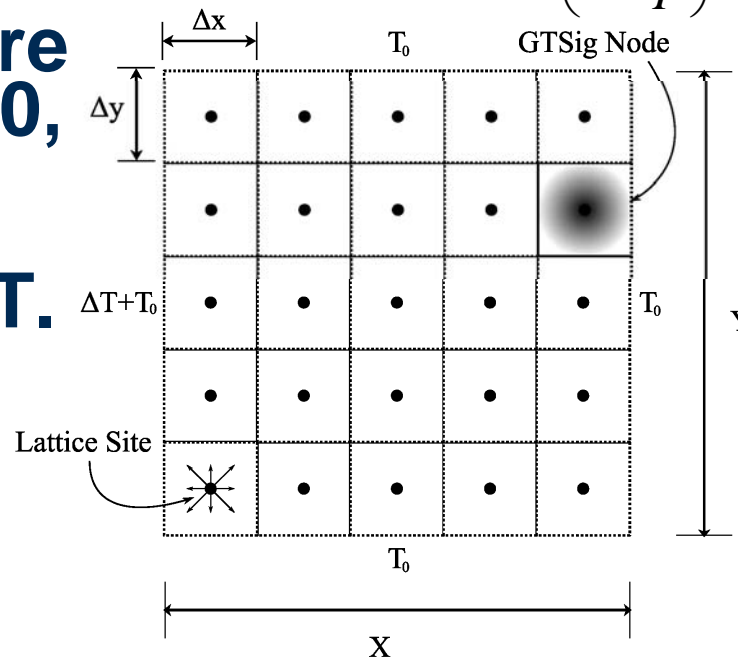


Numerical Method	Time Elapsed [s]
GTSig: Finite Difference	2.157286
Lattice Boltzmann Method	8.544141

Fixed Wall Temperature in Two Dimensions

- In two dimensions, we considered a homogeneous object initially at temperature T_0 .
- The temperature of the west, $x=0$, wall was then raised by a temperature ΔT .

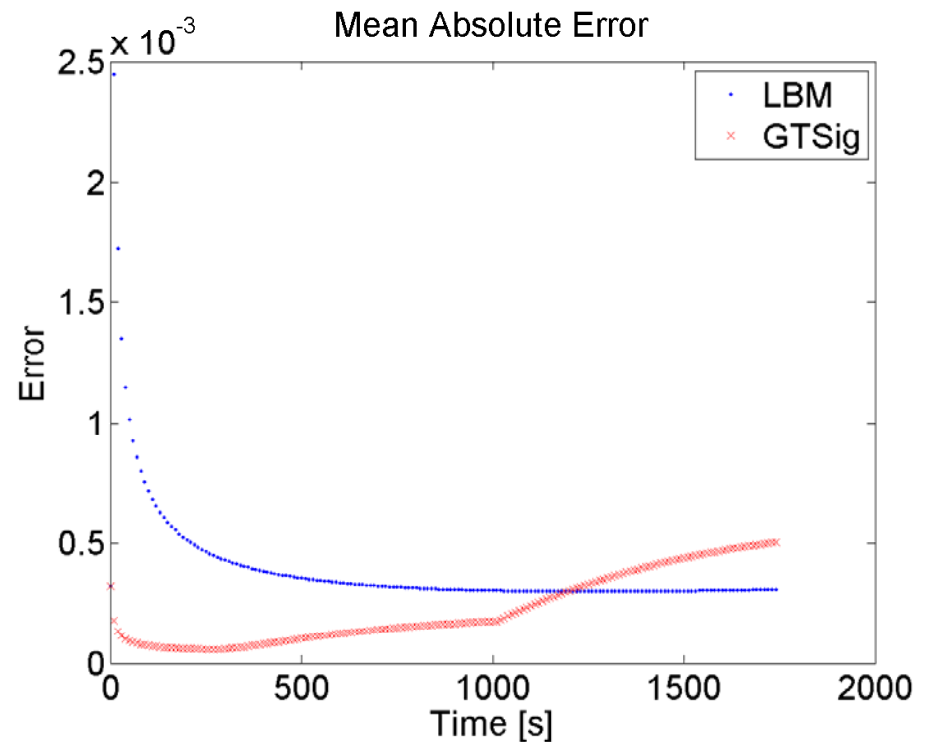
$$T(x, y, t) = T_0 + \Delta T \sum_{n=1,3,5,\dots}^{\infty} \frac{4}{n\pi} \sin\left(n\pi \frac{y}{Y}\right) \left\{ \frac{\sinh\left(n\pi \frac{X}{Y} \left(1 - \frac{x}{X}\right)\right)}{\sinh\left(n\pi \frac{X}{Y}\right)} \dots \right. \\ \left. - \sum_{m=1}^{\infty} \frac{2m\pi \sin\left(m\pi \frac{x}{X}\right)}{\left(n\pi \frac{X}{Y}\right)^2 + (m\pi)^2} e^{-\left(\left(n\pi \frac{X}{Y}\right)^2 + (m\pi)^2\right) \alpha t / X^2} \right\}$$



$\alpha = 0.02$	$\frac{m^2}{s}$	Thermal Diffusivity
$X = \sqrt{600}$	m	X - Dimension
$Y = \sqrt{600}$	m	Y - Dimension
$T_0 = 0$	$^{\circ}C$	Initial Temperature
$\Delta T = 100$	$^{\circ}C$	West wall Temperature
$\Delta x = \sqrt{0.06}$	m	Lattice Separation
$\Delta y = \sqrt{0.06}$	m	Lattice Separation
$\Delta t = 1$	s	Time Step

Fixed Wall Temperature in Two Dimensions

- Again the mean error over the domain was found to decrease rapidly for the lattice Boltzmann method while the GTSig error increased.
- While the numerical results are encouraging, the computation time was high due to serial implementation of the algorithm.
- Maximum errors for both GTSig and the lattice Boltzmann method were confined to the corners where the temperature is ill defined



Numerical Method	Time Elapsed [s]
GTSig: Finite Difference	27.208329
Lattice Boltzmann Method	387.877949

Fixed Wall Temperature in Three Dimensions

- The D3Q27 lattice was implemented and a numerical solution was found using the lattice Boltzmann method.
- Due to the extended time investment to compute the analytic solution, comparisons of the numerical solutions were not available.

$$T(x, y, t) = T_0 + \Delta T \sum_{m=1,3,5,\dots}^{\infty} \sum_{n=1,3,5,\dots}^{\infty} \frac{16}{mn\pi^2} \sin\left(m\pi \frac{y}{Y}\right) \sin\left(n\pi \frac{z}{Z}\right) \dots$$

$$\left[\frac{\sinh\left(\sqrt{\left(\frac{m\pi}{Y}\right)^2 + \left(\frac{n\pi}{Z}\right)^2} (X-x)\right)}{\sinh\left(X \sqrt{\left(\frac{m\pi}{Y}\right)^2 + \left(\frac{n\pi}{Z}\right)^2}\right)} \dots \right]$$

$$\cdot \sum_{m=1}^{\infty} \frac{2 \frac{k\pi}{X^2} \sin\left(m\pi \frac{x}{X}\right)}{\left(\frac{k\pi}{X}\right)^2 + \left(\frac{m\pi}{Y}\right)^2 + \left(\frac{n\pi}{Z}\right)^2} e^{-\left(\left(\frac{k\pi}{X}\right)^2 + \left(\frac{m\pi}{Y}\right)^2 + \left(\frac{n\pi}{Z}\right)^2\right) \alpha t / X^2}$$

$\alpha = 0.02$	$\frac{m^2}{s}$	Thermal Diffusivity
$X = \sqrt{600}$	m	X - Dimension
$Y = \sqrt{600}$	m	Y - Dimension
$Z = \sqrt{600}$	m	Z - Dimension
$T_0 = 0$	°C	Initial Temperature
$\Delta T = 100$	°C	West wall Temperature
$\Delta x = \sqrt{0.06}$	m	Lattice Separation
$\Delta y = \sqrt{0.06}$	m	Lattice Separation
$\Delta z = \sqrt{0.06}$	m	Lattice Separation
$\Delta t = 1$	s	Time Step

References

1. Asinari, P., Mishra, S. C., & Borchellini, R. (2010). A Lattice Boltzmann Formulation for the Analysis of Radiative Heat Transfer Problem in a Participating Medium. *Numerical Heat Transfer, Part B: Fundamentals*, 57(2), 126-146.
2. He, X. & Luo, L. (1997). Theory of the Lattice Boltzmann Method: From the Boltzmann Equation to the Lattice Boltzmann Equation. *Physical Review E*, 56(6), 6811-6817.
3. Ho, J., Kuo, C., Jiuang, W., & Twu, C. (2002). Lattice Boltzmann Scheme for Hyperbolic Heat Conduction Equation. *Numerical Heat Transfer, Part B: Fundamentals*, 41(6), 591-607.
4. Mishra, S. C., Lankadasu, A., & Beronov, K. N. (2005). Application of the Lattice Boltzmann Method for Solving the Equation of a 2-D Transient Conduction-Radiation Problem. *International Journal of Heat and Mass Transfer*, 48(17), 3648-3659.
5. Mishra, S. C. & Roy, H. K. (2007). Solving Transient Conduction and Radiation Heat Transfer Problems Using the Lattice Boltzmann Method and the Finite Volume Method. *Journal of Computational Physics*, 223(1), 89-107.
6. Succi, S. (2001). *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond*. Oxford: Clarendon Press.
7. Wolf-Galadrow, D. A. (2005). *Lattice-Gas Cellular Automata and Lattice Boltzmann Models: An Introduction*. Berlin: Springer.

MURI Final Report

**Human Biomechanical Model and
Applications**

Brian Kocher

September 14, 2010

Introduction

- **Development of a biomechanical model was central to our handling of the human detection problem**
- **Additionally, a generalized biomechanical model allows us to tackle problems in a variety of fields and applications, including tracking, identification, and medical analysis.**

Georgia Tech Research Institute Georgia Institute of Technology

Model Structure

- **Physical model – computer representation of a human being**
- **Mechanical model – replicates human motion, current form is based on GTRI-acquired motion capture data**
- **Signature Model – generated via a database system, signatures are based on calculated radiance and can be rendered in multiple modalities**

EOBL ELECTRO-OPTICAL SYSTEMS
GEORGIA TECH RESEARCH INSTITUTE

3

The model is constructed through three primary phases of development.

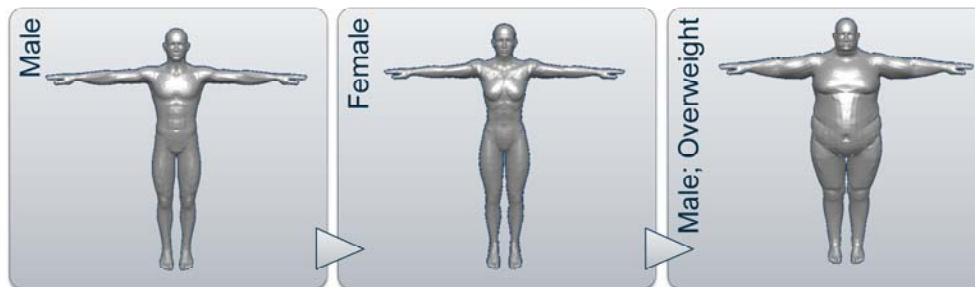
The first phase is the physical model, during which we construct a digital representation of a human, we typically call it the 'mesh.'

The 2nd phase is a combination of two parts, the first of which is acquisition of motion capture data and a digital model of an articulated skeleton. The 2nd part is the pairing of the digital skeleton to the target mesh. This allows motion of the skeleton to be projected upon the vertices which comprise 'control points' of the mesh.

The 3rd and final phase is only concerned with signature generation. Once the mesh is articulated and accurately reproducing skeletal motion, it is fed into a program where we classify groups of faces into 'surface nodes.' These surface nodes are comprised of faces which share similar position, orientation, and underlying structure(s). Once the mesh has been 'noded,' material properties such as reflectivity, thermal conductivity, and similar properties are assigned to the nodes and a signature is generated using those values.

Physical Model - Model Generation

- The physical model is generated via user manipulation of input parameters. Age, gender, tone, weight, and stature can all be tuned according to user specifications



The physical model is generated in an open-source program called 'Make Human.' The Make Human program is an active project in the open-source community and is regularly updated with features on a monthly basis. Nightly builds of the program are available for download at www.makehuman.org

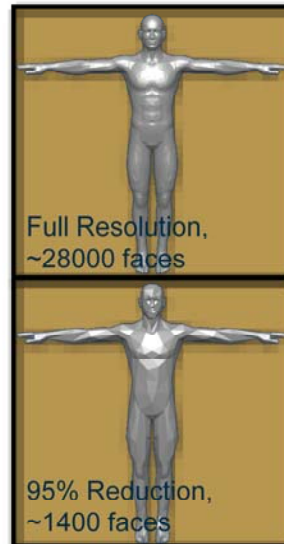
Within makehuman, a user may manipulate the model at multiple levels of detail: global → region → local

There are 5 primary global variables which can be manipulated and applied uniformly across the entire model: Age, Gender, Tone, Weight, and Stature.

Once the generic properties of a model have been customized, additional changes can be performed on a regional basis, so an arm, the face, the chest or other body regions may be customized independent of the rest of the body. A user has the flexibility to create a more "realistic" human, with uneven distribution of weight, muscle mass, or similar features. The local settings allow a user to change the appearance of individual muscle groups, so you can change the location of the knee, size/shape of the nose, ears, and similar details.

Physical Model - Optimization

- **Model originally generated with approximately 28,000 faces, however, detail can be tailored to user-specific needs.**
- **Models of fewer than 1000 faces are still viable and offer reduced computational load while maintaining general features**



We export a make human model as a .obj file (generic 3d model file-type) at full detail. Full detail typically means approximately 16 – 28000 faces, an excessive amount of detail. This level of detail is entirely unnecessary, particularly when simulating sensors with resolutions on the order of centimeters. We reduced the number of faces with tools in the 3d modeling program, Blender. Blender is another open source program, which is also currently under active development. Blender allows one to reduce the number of vertices by a specific percentage whilst maintaining the original shape of the object as faithfully as possible.

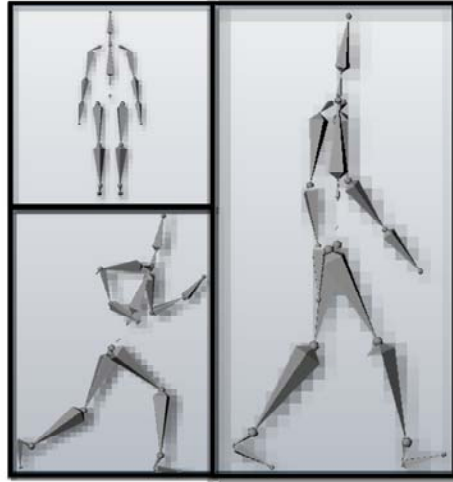
Once the level of detail is reduced to an 'appropriate' level the 1st phase of model generation is complete.

Mechanical Model - Development

- **Initial model used a skeletal-based inverse kinematics system.**
 - Inverse kinematics systems are equation solvers whose parameters are set by the position and orientation of an end point.
 - Model was set at key positions and a full motion cycle generated via an interpolation scheme
- **Adequate for basic signature generation**
- **Model was highly cyclical and contained insufficient detail for realistic gait analysis**

Mechanical Model – Motion Capture

- Record human motion and project that motion onto a digital model or skeleton
- Several methods of motion capture:
 - Active Optics
 - Passive Optics
 - Mechanical Capture Suit
 - Inertial Capture Suit



Passive Optical Motion Capture

- Markers are placed on the target near joints and similar key locations to indicate joint locations and the relative location of each joint to its neighbor
- Passive optical systems rely upon markers coated in a retro-reflective material and a network of cameras equipped with bright lamps
- Each marker in the system *must* be observed at all times by a minimum of 2 cameras for proper function. As such, optical systems are limited and unsuited for any recordings involving
 - Environmental Obstructions
 - Motions involving covering or grounding of the subject
 - Outdoor motions
 - Large capture volumes
 - Subjects wearing anything other than a spandex suit

Passive markers are entirely indistinguishable from one another within a camera's field of view. As such, if a marker is "lost" from view and later recovered, the system regards those instances as two distinct markers.

Furthermore, if one marker is occluded by another marker during a motion there is no means of distinguishing the two markers from one another after the occlusion.

Range is severely limited in this system. Since the light has to travel to the marker and return to the camera(s), the lamps have to be, initially, very bright and the environment very dark to ensure enough contrast to properly track *just* the markers and not any bright environmental reflectors (clothes, skin, etc)

Markers are mounted on the surface of the target and need to be as close to the joints as possible. Since they must remain visible at all times the system is unsuitable to be used over top of body armor or street clothes.

Active Optical Motion Capture

- Active systems, like their passive counterparts rely on tracking markers clustered on or near the joints of a subject
- Active markers are powered and emit their own light which is in turn picked up by a set of cameras surrounding the capture area
- Two cameras are required to observe each marker at all times, however, active systems have several distinct advantages over passive
 - Light sources may operate in the infrared or specific color ranges, allowing finer tuning of the cameras to avoid tracking of non-marker objects or materials
 - Markers may be time-modulated to strobe at different rates or in unique patterns relative to other markers, allowing the system to identify them and correct for marker occlusion and overlap
 - Marker-sourced light covers only half the distance of camera-sourced light, giving a 4-fold increase in energy and allowing for a larger capture area with fewer cameras

Note however, that all of these advantages do nothing to help solve one of the primary difficulties with optical systems, the requirement of line of sight by two cameras on every marker to allow for adequate tracking of those markers. As such, obscuring of markers by any environmental objects or the subject's body and motion represents loss of data and comprises the integrity of the captured motion.

Markers cannot be worn underneath clothing, and any additional layers between the marker and the body opens the system up to noise and problems with picking up movement of the garments rather than the actual motion of the subject

Mechanical Motion Capture

- The subject wears a mechanical exoskeleton which records the relative motion of its articulated parts
- Has no dependence on cameras, data is sent to a small base-station worn by the user and transmitted via tether or wireless signal to a workstation
 - Range is determined by wireless signal strength, battery life, and/or tether length, depending upon the configuration
- As with optical systems, the exoskeleton cannot be worn beneath clothing
 - depending upon the configuration, the suit may limit range of motion of the user and produce non-realistic data

Owing to the restrictions of the suit, there might still be great difficulty in recording any motion where the user leaves his/her feet to simulate crawling or similar motions. Additionally, suit bulk could also hamper any interaction with external objects, particularly with vehicle mount/dismount.

Some systems use a kind of 'elastic' sensor, which returns information that is partially based on the stretch felt by the sensor. These sensors are sensitive to wear and tear and data quality is likely to degrade over time and multiple uses.

Inertial/Magnetic Motion Capture

- Inertial and magnetic motion capture systems rely upon a system of integrated accelerometers, gyroscopes, and/or magnetometers to derive motion information
- Each sensor is approximately the size of a small computer mouse and placed, once again, on or near the joints of the body and linked to a small base station which in turn transmits data to a computer via wireless or wired signal
 - As with the mechanical system, range is determined only by battery life and signal strength
- Sensors can easily be worn beneath clothing and are lightweight, causing little or no interference in the natural motion of a subject

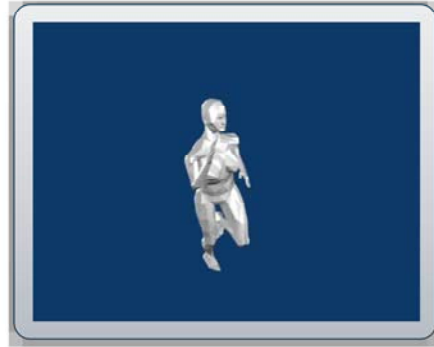
The only drawback with this system is short battery life and sensitivity to objects with large amounts of ferrous material. Vehicle mounting and dismounting is easily recorded, provided the vehicle does not contain large amounts of metal. We contracted specifically with an individual who uses this type of system and he was currently working on solving the problem of recording motion in and around large trucks and hummvees.

In general, however, magnetic interference is non-existent so long as the user does not make physical contact with a large metal object.

The gyroscopes and accelerometers are used to determine relative joint locations and orientations, while the magnetic sensors are used to orient and locate the subject in global space using the Earth's magnetic field as the basis of measurement.

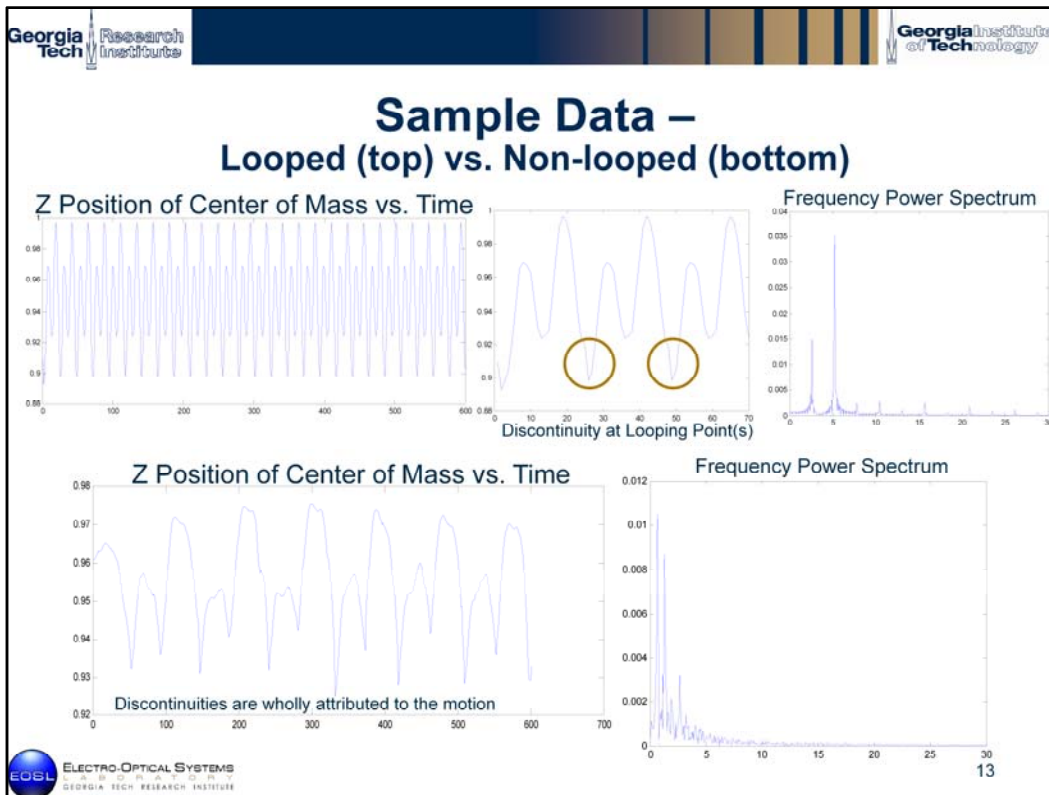
Mechanical Model – Justification for GTRI Internal Acquisition of Motion Capture

- Freely available motion capture data is of short duration and generally insufficient variety.
- Short duration motion capture requires extensive looping to generate motions of relatively short duration (5 to 10 seconds)
 - End result is highly repetitive and generally unnatural looking motions



There are limited datasets of motion capture available for public consumption. The two primary sources are the Advanced Computing Center for the Arts and Design (ACCAD) of Ohio State University and the Carnegie Mellon University (CMU) data sets. Both of these datasets were created with optical systems and are of variable quality. The CMU dataset is much larger than the data available at ACCAD, though it covers a wide range of completely irrelevant motions, including human impressions of dinosaurs and animals. Furthermore, descriptions of subject characteristics are entirely unavailable for the data, making it impossible to draw conclusions regarding gait's relationship to weight, height, age, and sex.

Furthermore, all the data is of a very short duration, requiring extensive looping to construct motions on time-scales greater than 5 seconds.



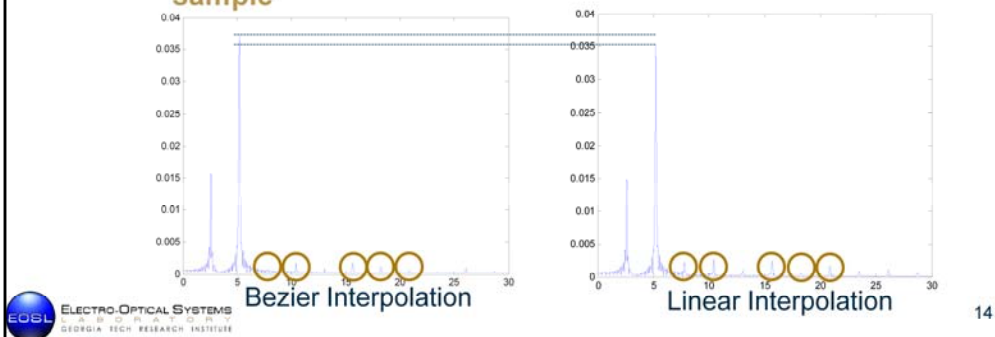
Note the difference in the upper graph vs. the lower graph. The left graph of the looped motion is the z-position of the center of mass throughout the entire motion. Every peak is of an identical magnitude and periodicity. As a result, the frequency power spectrum is incredibly clean. However, there exists a discontinuity in the loop, as the interpolation scheme is not perfect, and can create sharp changes in the motion which can distort the frequency power spectrum.

In contrast, the lower graphs represent an entirely un-looped motion over 600 frames (10s). Qualitatively, every stride is approximately the same, though the relative heights of the peak, sharpness of the lower peaks, and depth of the valleys all vary considerably across the motion. Furthermore, there is an apparent cresting in z-position throughout the motion. The resulting frequency power spectrum is considerably dirtier and of a lower magnitude, however, it contains substantially more low-frequency motion. This includes a 3rd frequency of considerable magnitude at approximately 3.5 Hz. Such a peak is non-existent in the upper data and could be exploited as a motion unique to the individual, considered separate from the primary peaks which will follow the frequency of motion of the arms and legs.

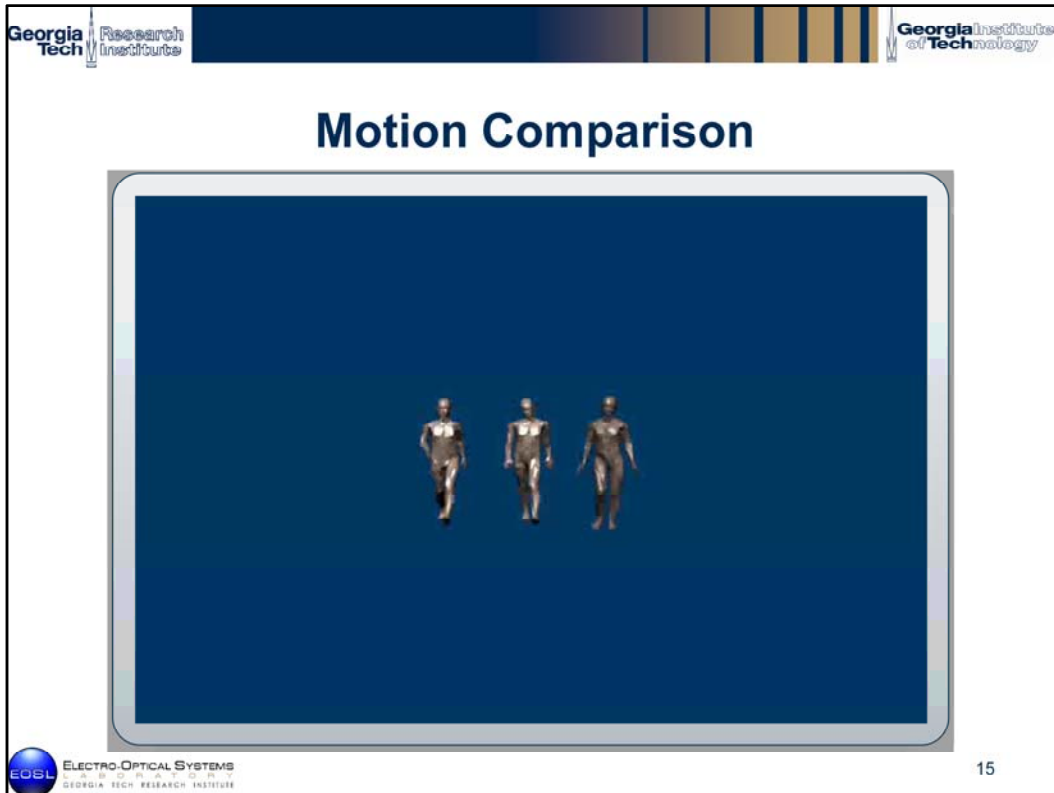
Furthermore, the lower motion reveals the trend to favor one side throughout the entire recording, indicating it as a real phenomenon particular to this individual. The upper data, since it is merely a repetition of an individual going through a single stride, can only be assumed to accurately reflect the motion of this individual in the long-term. However, this effect may fade over time, become more pronounced, or disappear altogether depending upon running surface and other externalities.

Example of Spurious Signals in Fourier Analysis of Looped Signals

- By altering the method of interpolation for the joint, the frequency power spectrum noticeably changes for the heavily looped motion
- Maximum peak decreasing by approximately 0.005 and substantially greater high frequency content in the linear sample



By simply altering the method of interpolation which computes the brief 2 frame interval between loops we have non-trivially affected the frequency power spectrum of the motion signal. Given this change, these small signals can be attributed primarily to the 2 artificially produced frames, making them poor candidates for exploitation with regards to human identification and signature determination. Unfortunately, these smaller high frequency features would be the source of most of the difference between two independent motions, as the largest two components will, for a given rate of stride, be nearly identical. As such, by compromising these small differences, we are self-limiting the “useful” portion of the signal and compromising methods derived of this data set.



This video is a comparison between our original method of constructing motion via a constrained inverse kinematics (IK) system, downloaded and looped motion capture data, and GTRI acquired motion capture data.

Subject Left – IK motion

Subject Center – Free and looped motion capture data (CMU)

Subject Right – GTRI acquired motion capture data

Note the 'stiffness' of the figure at left relative to the captured motions. The two captured motions have similar fluidity and natural appearance, however, the center motion is clearly looped over one or two strides. Also note the head position of the center subject, pointed down and to our left.

Mechanical Model – Motion Capture Acquisition Review

- **Given the limitations of the available datasets, we recorded our own series of motions to build an internal motion database**
- **Recorded 400+ motions from several broad categories:**
 - **Simple Motions – walking, running, jogging, etc...**
 - **Motion Transitions – breaking into a run from a walk**
 - **“Social” Motions – Waving, gesturing**
 - **Object Interactions – Carrying assorted weights, entering/exiting a vehicle**

The data was collected from a single male subject over a period of 3 days. Prior to the data collection a list of scenarios and actions was compiled to provide a framework to guide the collection of data, and ensure the recording of motions relevant to our research interests. Every action or interaction was recorded a minimum of three times to ensure data quality and to capture natural variation in the subject's motion.

Motion Capture Acquisition Continued

- We worked with Motionwerks to collect all the motion-capture data with an inertial suit.
- Data was collected from a single male of average height and weight over 3 days
 - Day 1 – Familiarization and basic motions
 - Day 2 – Object interactions, vehicle mount/dismount, etc
 - Day 3 – Continued simple motions and motions with weights
- Each motion was recorded a minimum of three times to reduce noise in the sample



Persons in the figure are, from left to right: Dr. Michael Cathcart, Brian Kocher (Tech Temp), and Roger Nelson (Motionwerx). Brian Kocher is wearing the suit which mounts all the sensors. Clothes can be and were worn overtop of the suit and sensors during the recording sessions. The yellow/black combination are meant to make it easier to visually identify parts of the body and their motions in video recorded alongside the motion capture. Velcro patches are used to hold on the sensors. The gray box at the center of the torso is the base-station to which all the sensors are wired. It is capable of transmitting data via a tethered or wireless signal.

Motion Capture Format

- **Data is recorded and saved directly as a bvh file. Structure of the bvh file is as follows:**
 - **Header – Outlines hierarchy, starting position and rotation for each bone, relative to the parent bone(s), and**
 - **Motion – In order of the recorded hierarchy, documents the rotation and position of each bone from frame to frame, where each row in the file represents a single frame**
 - **Format is in plaintext and can be read via any text editor**
- **Blender supports import of bvh data and constructs the skeleton according to the specifications outlined in the bvh file**

The hierarchical structure of the skeleton is based upon each bone having a single parent. However, a parent may have multiple children and all chains typically end at a single master bone which dictates the motion of the skeleton in global space. In our case, this master bone is located at the waist. A sample chain is organized as follows:
Hips → LeftUpLeg → LeftLeg → LeftFoot → LeftFootHeel → 'End Site'
The 'End Site' marks the end of the final bone in the chain. The bones are linked in that the "top" of a child bone is located at the "bottom" of its parent. Bones are drawn in the program between these endpoints, labelled head and tail, respectively.

All the motions we recorded were taken at a sampling rate of 60Hz, though there is some flexibility as to the sampling rate, with 30Hz being another "standard" rate. Typically, data is recorded at the highest efficient rate and may be down-sampled if some reduction in data density is required.

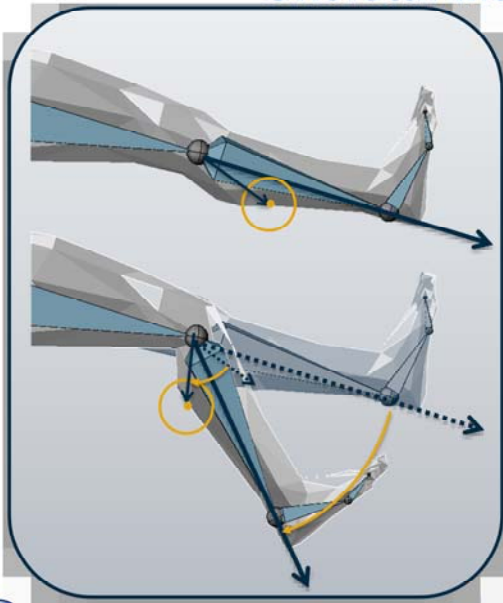
Motion Capture Processing

- **Motion capture data is very dense, at 60 frames per second, and every bone (21 in our skeleton) having a unique data point**
- **In Blender, I parse down the data to the desired segment of the recording, omitting starts and stops, or extraneous motions before and after the relevant recording**
- **Some recordings of 1600+ frames can be reduced to approximately 1000 by eliminating extraneous frames**

There is little/no post-processing aside from removal of extra frames. The system employed by Motionwerx always makes sure the recorded subject is 'grounded' and prevents a recording from drifting in the vertical. This is done via an algorithm which locates the lowest control point of the recorded subject and using it as the ground reference. This system can be disabled if recordings of stair climbing or other activities during which no limbs are grounded (run/jog).

Once I am happy with the motion I can bind it to a skeleton, a process outlined in the next slide

Skeletal Manipulation



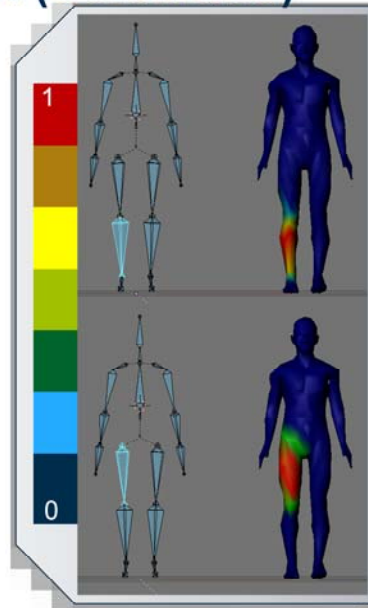
- The virtual skeleton is mated to the human geometry
- The skeleton controls the model geometry by transferring the bone transforms onto the mesh vertex locations

This slide explains the basic rules which dictate how the vertices of the mesh are influenced by the motion capture derived skeleton.

Vertices are assigned to at least one bone with a specific weight parameter which ranges between 0 and 1. This weight parameter only comes into play when more than one bone has influence over a vertex. In these cases, the resulting motion is a result of a weighted linear combination of the bones' motions projected onto the vertex. The coefficients of the linear combination are determined by the weight parameter of each bone.

Skeletal Manipulation (Continued)

- Bones are assigned influence over specific vertices
- Strength of influence of each bone is set by a weighting parameter
- Weighting parameter assignment is informed by physiological constraints



This slide provides a visual representation of the binding process, displaying one of the interfaces by which bone weights are assigned.

The top segment of the image has the lower right leg selected in the skeleton at image left paired with an image of the subsequent vertices assigned to that bone colored by weight. The colorbar at the left provides information of the weights displayed. Red is maximum, green lies at approximately $\frac{1}{2}$ weight, and the lighter/darker shades of blue lie around approximately $\frac{1}{4}$ to zero.

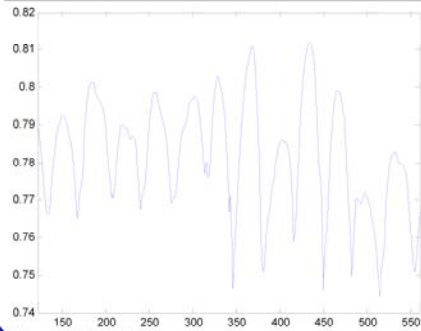
The lower segment of the image is of a similar pairing, this time of the upper-right leg of the skeleton and corresponding colored vertices.

In both cases, the dark blue is a weight of zero.



Complex Motion Sample

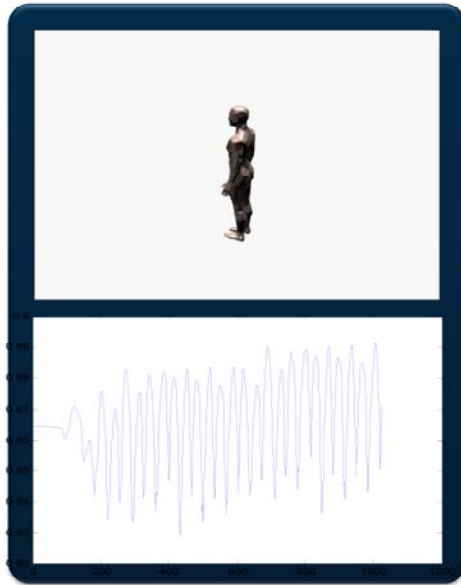
- Longer motions or scenes may be constructed by piecing together smaller components to form chains of motion
- Each link of the chain consists of motion capture data, preserving physical accuracy



This video illustrates the combination of a straight line walk with a 90 degree turn to the subject's right. Once the turn is complete the movement is patched once again with a simple straight line walk. This represents a chain of 3 motions strung together

The plot below is the z position of the center of mass of the person throughout the walk. The joints are not entirely apparent, though the different phases of the motion are clearly visible, with the turn having much larger vertical variation than the straight walking sections. While artifacts are produced by the combining of motions, they are much less likely to have an effect on any statistical or fourier analysis of the data because of their rarity (2 points in this case). By choosing to match the motions at points where the posture is nearly identical for both frames the transition may be further improved. The greatest difficulty in matching motions from different data-sets lies in the general "angle" of the body, as a subject tends to lean slightly to one side over time. In the turning segment, the subject is clearly leaning over into the turn.

Simple Motion Sample



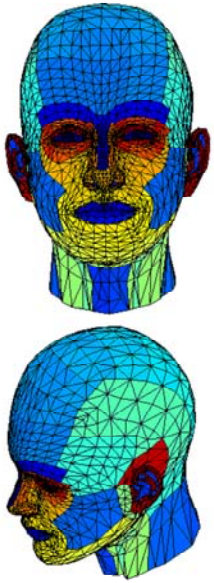
- Motion at the left consists of approximately 10s of walking duplicated a single time, resulting in approximately 1200 frames of motion (20s)
- Since there are only two links in the chain, there is only a single possible instance for the introduction of data artifacts, in contrast with repeated artifacts inserted in more heavily looped motions

This slide presents an example of a simple motion, consisting simply of a walk in a straight line duplicated for a single instance, producing a walk in a straight line of approximately twice the length of the original. There is still some limited introduction of artifacts in the motion, however, the effects of these artifacts is minimized by their rarity (one instance for this particular example). Furthermore, by using a bezier interpolation scheme and choosing suitable frames to “match” for the looping, the transition between the original and duplicate phases may be greatly improved, mostly eliminating any visual or statistical artifacts in the dataset.

Suitable frames in this case mean frames where the posture of the individual is approximately identical. In this case the individual had the left leg planted with the right leg just beginning to come forward on a swing phase. The arms are typically in nearly identical positions and are generally simple to match. The difficulty comes in “timing” the transition properly so the leg swing appears smooth and natural without a sudden jerk forward or slowing down.

Georgia Tech Research Institute Georgia Institute of Technology

Signature Model



- Mesh facets are divided into groups of surface nodes
- Surface nodes are assigned material properties which determine reflectance and other characteristics required for signature prediction

EOEL ELECTRO-OPTICAL SYSTEMS GEORGIA TECH RESEARCH INSTITUTE

24

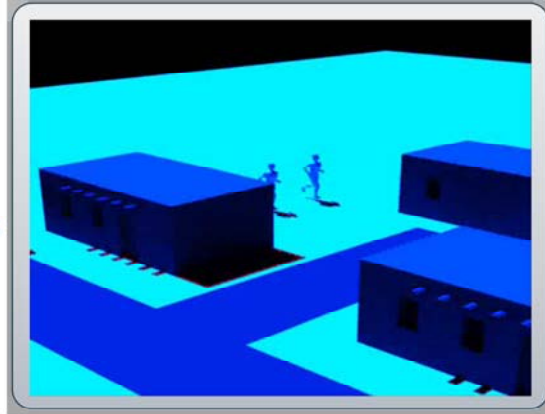
Once the motion is complete, the model may be exported and linked to a signature model. The first step of this process involves exporting the exterior mesh of the individual as a simple OBJ file, which is a standard computer modeling format written in plaintext and readable by any text editor. The 3d model is loaded into a face-selection program developed at GTRI and the faces of the model are grouped into “surface nodes.” Surface nodes are created based on a number of criteria, including but not limited to orientation, location, underlying structure (blood vessels), covering (clothing), and material composition (walls of a building vs. windows).

The image above illustrates the grouping of the face, with patches representing the large carotid arteries which flow up the neck and across the jaw line. The nose is divided into several portions, and the eye sockets are grouped together but not lumped in with the eyes themselves as the eyes will have a vastly different thermal signature than the skin surface of the face.

It is worth noting that the same mesh may be used for any number of motions and the surface nodes of that mesh may be used interchangeably. So long as the number of faces and vertices remains the same and the vertex associations not altered (vertex assigned to face A not reassigned to B), the list of surface nodes will remain accurate. This remains true even if the location of the vertices is changing, as is the case with motion. In that case, a single surface node list will apply across all frames of the motion, regardless of the deformation of the mesh.

Signature and Scene Generation

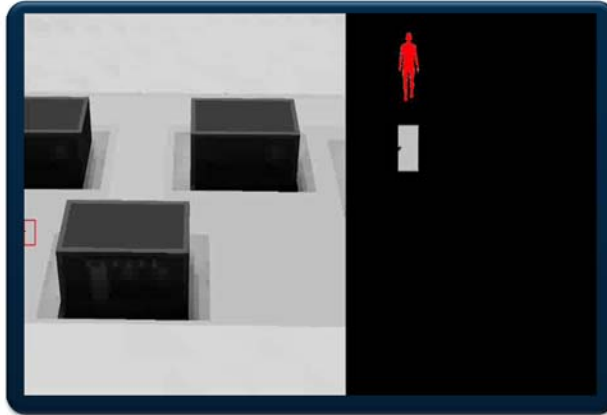
- A scene is constructed from stored database information. Human geometry is inserted into user specified locations
- The renderer accesses object characteristics and uses ray-tracing techniques to visualize a scene in a specific band and for a specific camera location



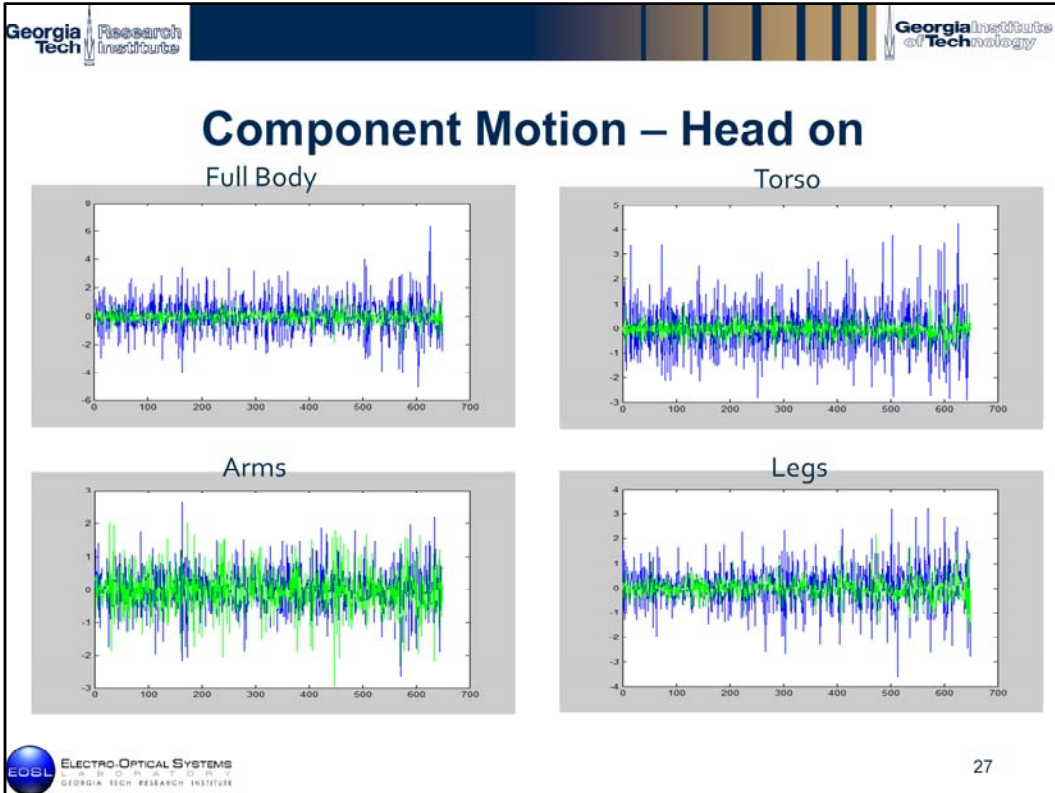
This is a sample scene generate in optical wavelengths with two humans running between buildings. Taking 3 independent samples of motion capture data solves the issue of “stepping in time” displayed in the sample image. An additional step to disguising repeated instances of the same motion in a scene include changing the relative phase of the motion in different subjects. Using this phasing method, a relatively small number of motions may be used to generate a scene with many humans present.

Applications – Motion Tracking

- Motion tracking is a natural application of the model
- Video may be generated at will from any desired angle and illumination in multiple modalities
- Motion tracking software can be optimized for a variety of conditions and scenarios via computer simulation



See Alan Thomas' briefing for details, as it chronicles the development of this motion tracker in detail.



Motion Tracking Results – This data tracks the pixel motion for different sections of the body from the frontal aspect view. There are slight signals in the torso and full-body, but the periodic motion is relatively robust in the arms and the latter half of the leg data, possibly due to the subject’s approach to the camera.



Motion tracking results – Side aspect
results in this view are much more robust, displaying a clear oscillatory pattern for both horizontal and vertical signals

Applications – Human Identification

- **Gait is an ideal trait to base human tracking and identification upon**
 - Can be studied remotely and unobtrusively
 - Cameras are small, mobile, and do not require human operators for data collection
- **However, gait recognition is still an open research problem and faces some of the following issues**
 - Current methods are based upon analysis of a target's silhouette, which requires a side-aspect view of an individual
 - Performance degrades with range and sensor resolution
 - No methods attempt identification from moving platforms

Human ID – Current Methods

- **Current methods are based upon shape analysis of an individual's motion**
- **Motion is broken down into a series of images (camera frames) and various techniques applied to detect a pattern of the motion**
- **All techniques depend upon a side-aspect view of the individuals in a relatively low clutter environment to successfully acquire and identify the target**

Human ID – Our work

- **Our model can contribute to the human identification problem in a number of ways:**
 - **Motion capture data may be dissected to locate motion features that can be exploited for identification**
 - **Once identified, methods of feature extraction may be practiced in simulated environments to optimize methods and identify failures**
 - **With a generic framework in place, more in depth work can be performed by adding clutter, camera movement or other complications which might exist in an active environment**

Georgia Tech Research Institute Georgia Institute of Technology

Human ID – Sample Camera Configuration

- This diagram is one possible camera configuration for a rendering. Each camera represents a distinct point of view for which a motion tracker may be tested
- Resolution and field of view for each camera may be changed depending upon user preferences

EOBL ELECTRO-OPTICAL SYSTEMS
G E O R G I A
G E O R G I A T E C H R E S E A R C H I N S T I T U T E

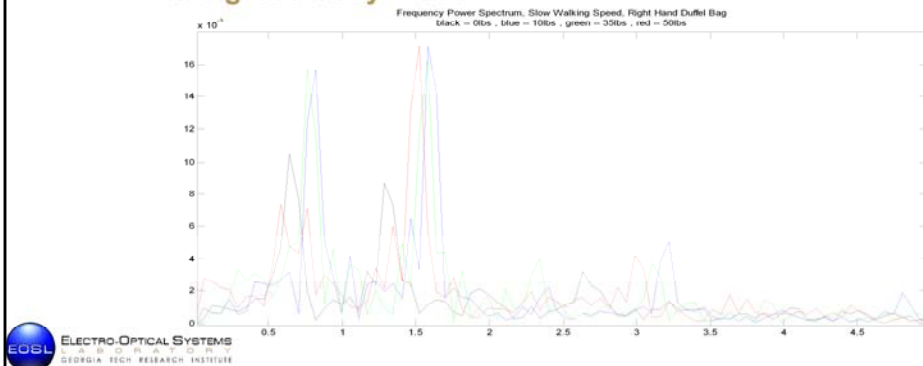
32

These images represent one possible camera configuration within the digital environment. Using this configuration, we can generate the same motion viewed from multiple vantage points to test existing motion detection and tracking methods without leaving the computer or purchasing a single camera.

The lower right diagram illustrates cameras (blue polygons) positioned about a circle of a specific range centered upon the mid-point of the subjects path. The configuration of cameras at 45' increments was used for Dr. Alan Thomas' work of the motion tracker.

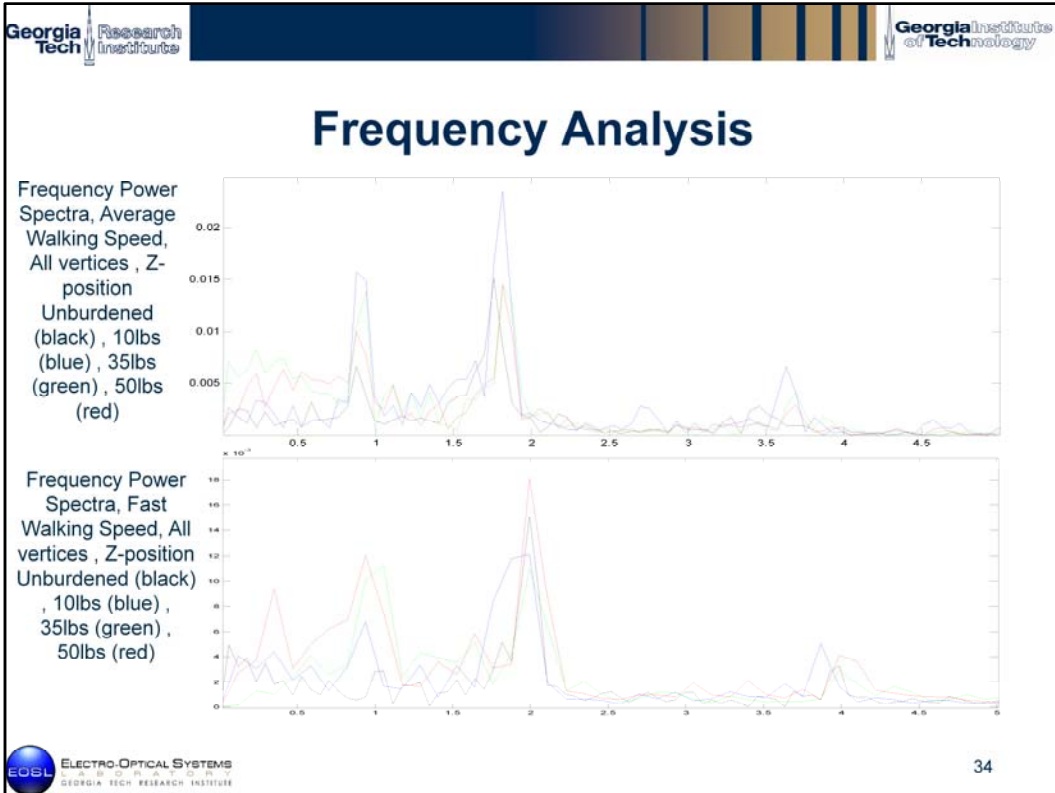
Frequency Analysis

- Frequency analysis
 - Identifying the fundamental and secondary frequencies of a motion might allow us to pick out “tics” specific to individuals
 - Analysis need not be limited to simple leg/arm motion. Body sway, head bounces, and arm swing can all be isolated from the gross body motion

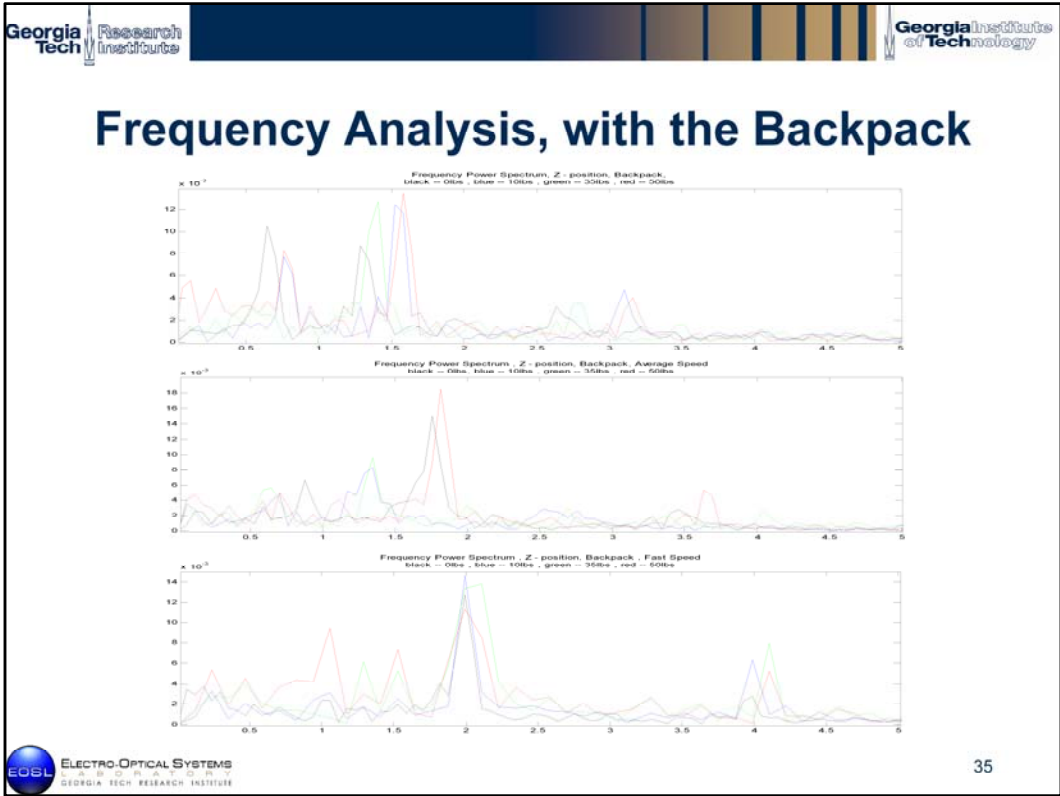


33

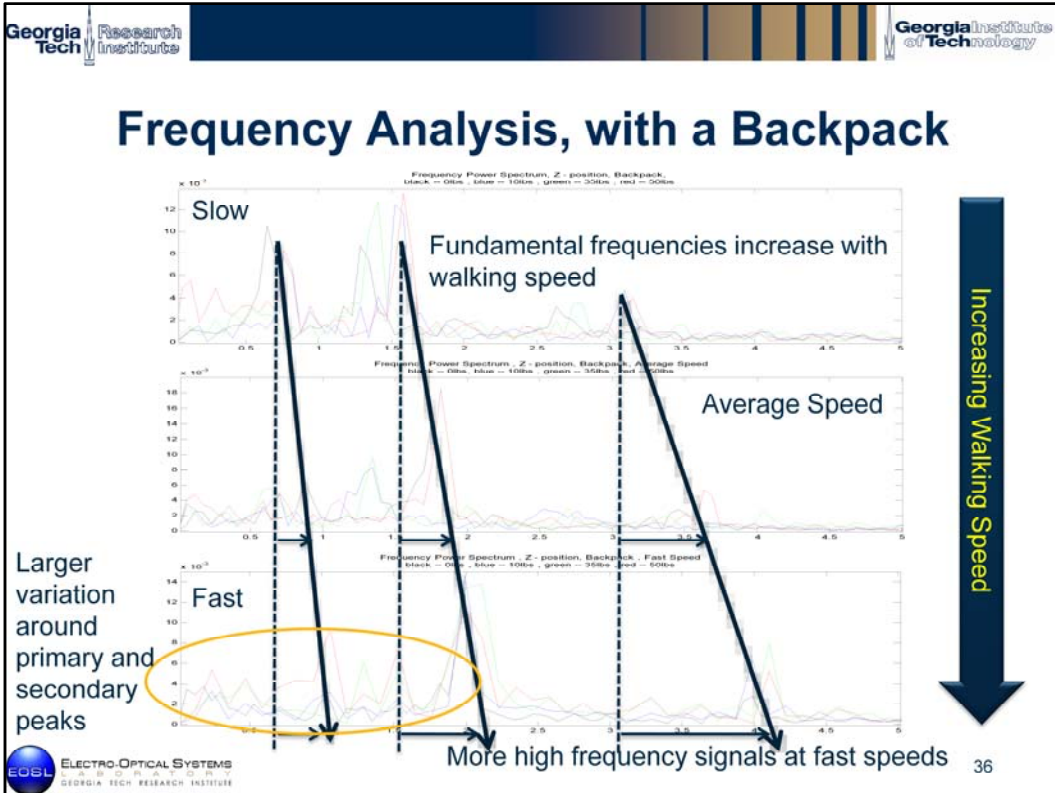
Take special note of the power spectrum in this image. This graph is the Frequency Power Spectrum of the z – motion of the centroid of the full body. For the 0lbs walk, The primary frequency component is approximately 0.6 hz, followed by a lower 2ndary component at approximately 1.3-1.4 hz. On the burdened samples, The primary component is located at approximately 1.5 Hz, with a secondary component nearly equal to the primary at 0.8 Hz, or slightly greater than $\frac{1}{2}$ of the primary. It is the relative strength of these components that should draw your attention, as it reflects a *decrease* in frequency attributable to the right arm hanging at the target’s side instead of swaying naturally.



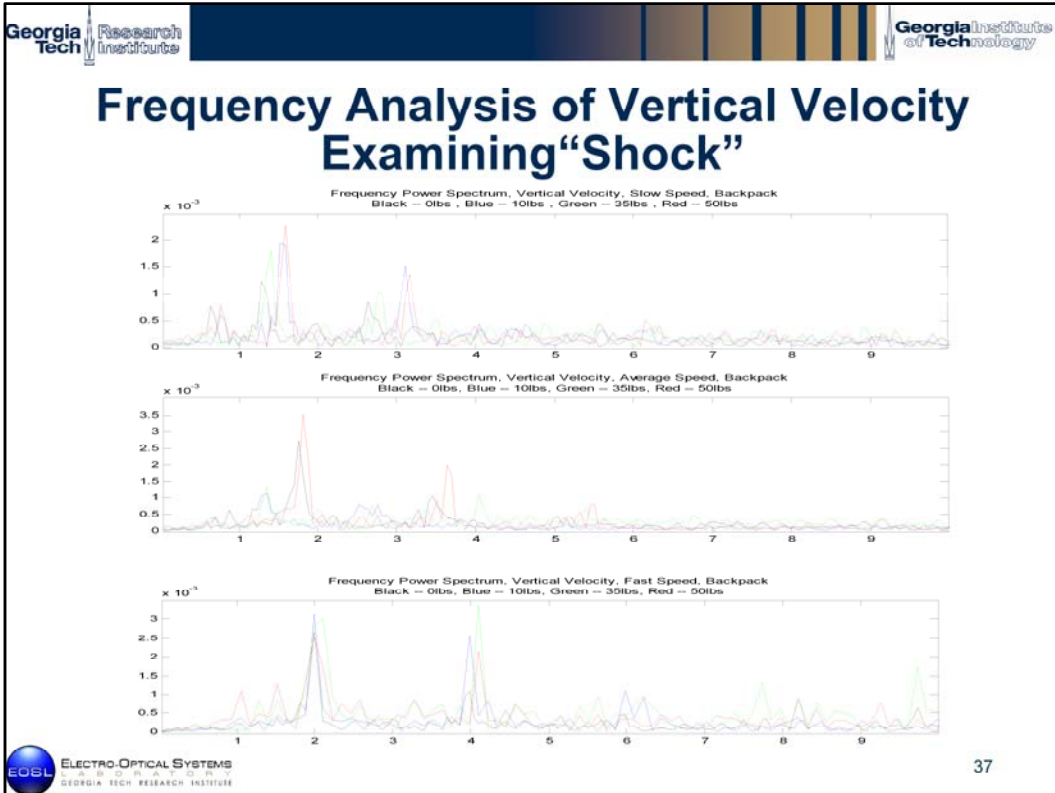
The relationships between the unburdened and burdened datasets appears to have changed relative to the slower walking speed. Now all datasets have a primary spike at a higher frequency component, albeit this frequency has shifted higher, to approximately 1.8 Hz and 2 Hz, respectively. However, note the increase in magnitude of the components. This trend is *not* apparent in the backpack data, nearly disappearing altogether in the fast speed.



Results for the backpack at different weights are inconclusive. Additional data may provide a better basis for comparison



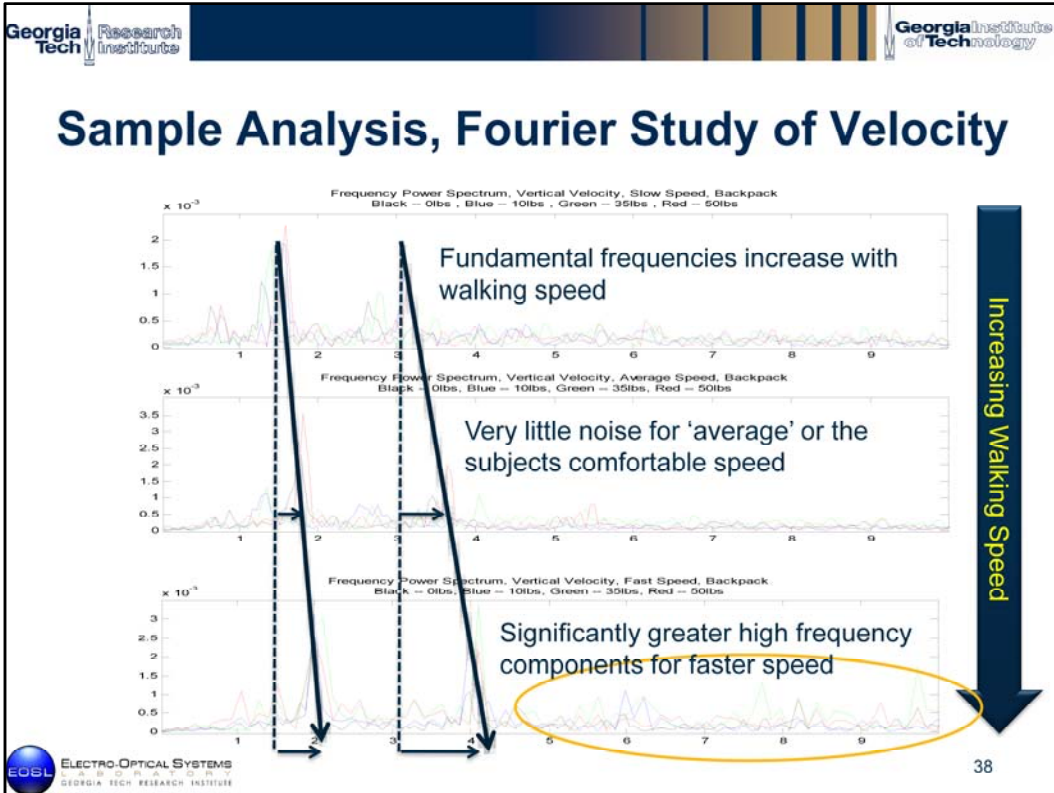
There is a clear trend for increasing fundamental frequencies alongside increasing walking speeds. Furthermore, there appears to be greater noise in the signal from the “fast” walk, indicating energy which is being misdirected into parts of the motion which are un-related to the base gait. These tertiary components indicate instabilities and energy which is wasted because of strain as a result of a difficult pace or possible burden on the individual. Furthermore, note that while the black, blue, and green lines may have higher frequency components, the red line, which represents the largest burden, *always* has high frequency components and a slightly higher amount of noise clustered around the primary peaks. This particular brand of noise could be used to detect concealed burdens if it is observed in recordings of other individuals.



For each graph, the colors of the line correspond to weights in the following manner:

- Black – 0lbs
- Blue -- 10lbs
- Green -- 35lbs
- Red -- 50lbs

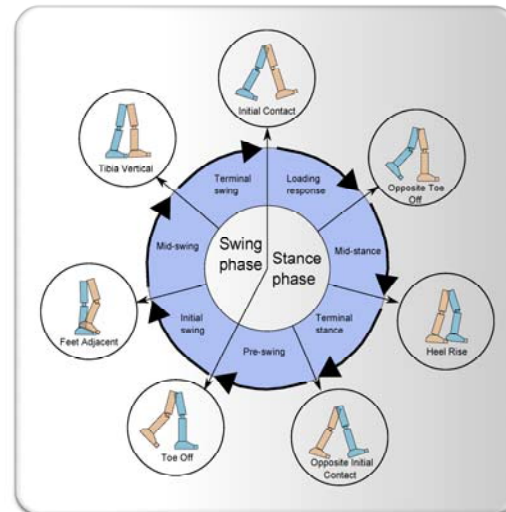
Generally speaking, increased magnitudes of power spectra are observed for all additional weight trials (with the exception of two of the average walk trials). Perhaps the walk is “stiffer” with weight in the backpack compared to a walk without the backpack.



There is a nearly linear increase in fundamental frequencies which corresponds to an increase in walking speed. The fastest walking speeds also lead to the greatest amount of energy concentrated in the highest frequency regime. Such high frequency components indicate increasing instabilities in the motion as a result of energy being shunted off of the fundamental motion and frequencies to inefficient and generally non-vital components. These non-vital components are attributable to tics, trips or other errors in gait as a result of a large departure from the natural cycle/pacing of the individual.

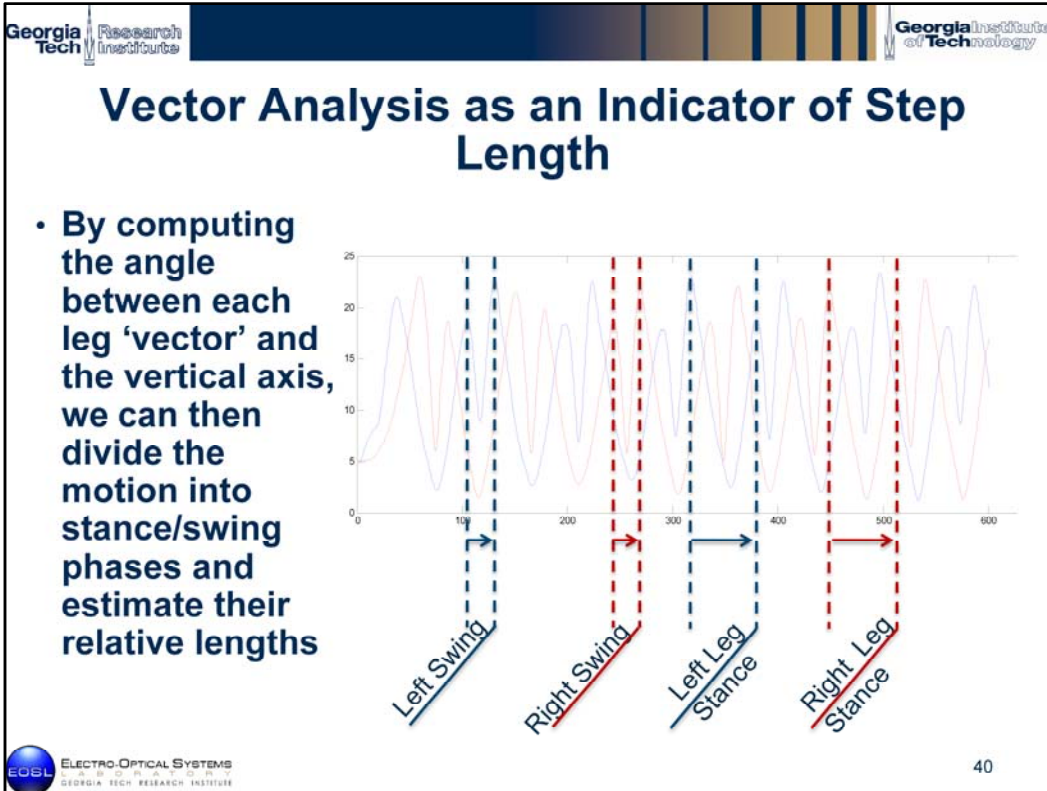
Analysis of Step Length

- There are numerous studies examining the effect of body weight support for physical therapy and rehabilitation
- Studies on individuals with no injuries indicate body weight support leads to a dramatic decrease in the 'stance phase' of the gait cycle



L Finch, H Barbeau, B Arsenault,. Physical Therapy/Volume 71, Number 11/November 1991, 842 → 855

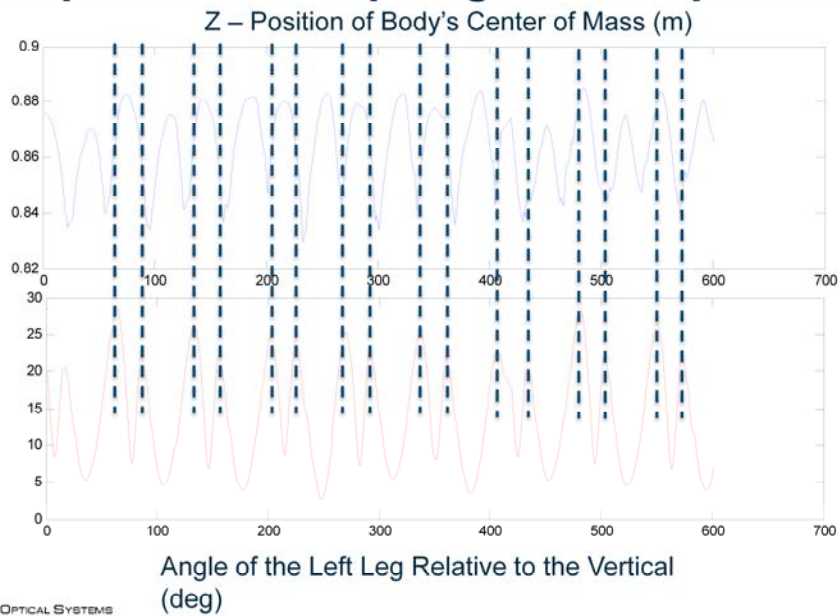
Image to be added here from the book from the library: "Gait Analysis, An Introduction"
Michael W. Whittle



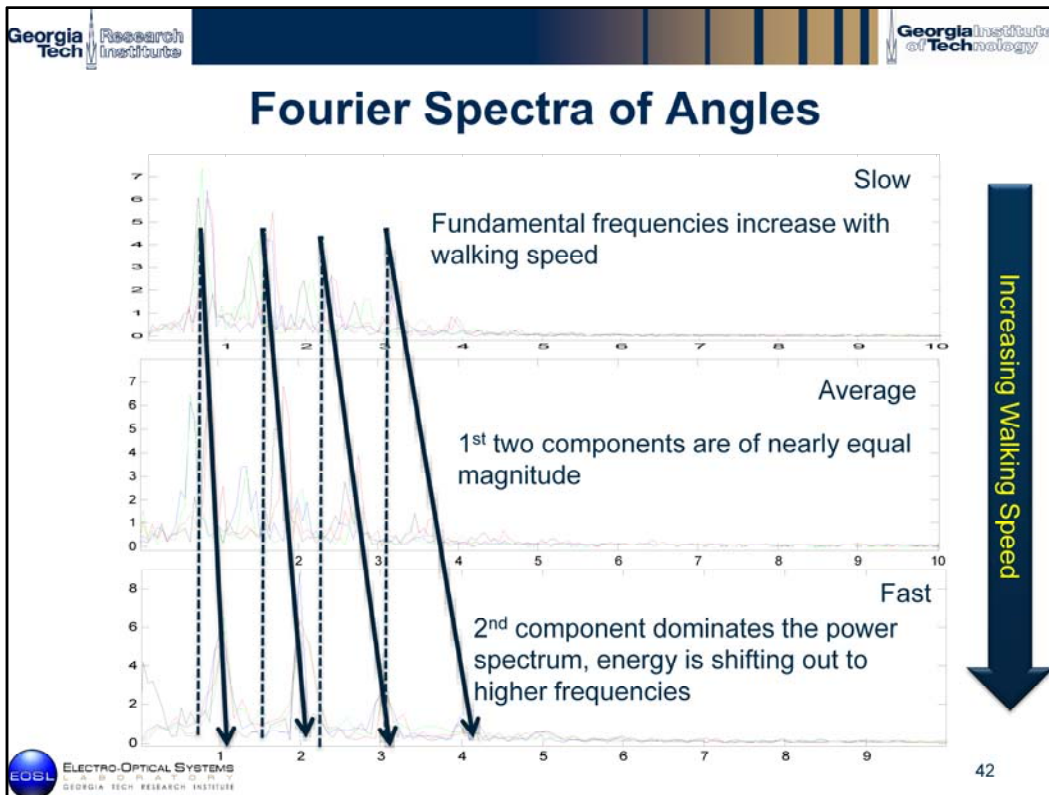
The graph represents the angle between the vector representing the left (blue) and right (red) legs and vector pointing up the z-axis. The phase of motion can be determined through examination of the slope of each line. The pattern that emerges starts at a peak to a sharp, albeit "shallow" valley, peaks again, then a much deeper valley over a longer period of time, followed by another slow rise to a peak. The slope of the line represents the rate of change of the angle with the vertical.

The sharp/shallow peak represents a time when the angle with the vertical rapidly changes through a minimum back to an extrema. This period of motion represents the swing phase of that leg, when the corresponding foot has left the ground and is being rapidly brought forward of the body. The valley is shallower because the airborne leg is swung slightly outward as it passes the midline so as not to collide with the planted leg. Once the leg is planted it goes through a slower, steadier phase of motion which corresponds to the stance phase, when it is supporting the body and moving an individual smoothly forward.

Comparison of step angle with z-position



As you can see, every swing phase of the left leg corresponds to jump in height of the center of mass of the subject. By examining the relationship between these areas of increased height, we may judge whether or not an individual favors the right or left side and infer a motivation for that perturbation of motion.



As more energy is shifted out to higher frequencies it could indicate instability in the gait of the subject being examined as a result of a greater departure from that individual's natural pace and/or stress and strain placed on the individual as a result of mass. In each of the above cases, the red line, which represents the data taken while carrying a backpack of 50lbs, has peaks at higher frequencies than the other data points. Furthermore, there appears to be a bit more noise around the peaks for the heavier backpack, possibly indicating instabilities as a result of the burden.

Effects of Weight on Walk Phasing

- The table at the right presents the ratio of the mean length of the stride phase vs. the mean length of the stance phase
- Results do not indicate a definite correlation with increasing burden and stance phase (decreasing ratio)

Burden	Slow	Average	Fast
0 lbs	0.3944	0.5027	0.5483
10 lbs	0.4114	0.4174	0.5191
35 lbs	0.3775	0.3818	0.5567
50 lbs	0.4293	0.4848	0.5204

By computing the mean length of the swing phase and mean length of the stance phase, we can estimate the proportion of each stride dedicated to each. Several studies of gait and obesity indicate that as an individual's weight increases, the amount of time dedicated to the stance phase increases for each leg. This in turn leads to a longer period of time for which both legs are planted, termed the "double support" phase.

By comparing the ratios for each phase, I wanted to see if the same was true of a 'normal' individual who was carrying loads of varying magnitudes, in this case 0, 10, 35, and 50 lbs. A solid positive result would mean an approximately linear decrease in the ratio concurrent with a steady increase in weight. While a majority of cases studied did show a decrease in ratio, the effect did not appear to scale with the weight. This researcher recommends a second look at the problem with a wider dataset.

Applications -- Medicine

- **Athletic programs are already taking advantage of motion capture to optimize technique and diagnose injuries**
- **Simulation would allow a doctor to take an individual's gait and view it from any conceivable perspective through manipulation of the camera**

Known Problems

- **Small sample size. Individual variation of gait could be far larger than previously anticipated**
- **Current stock of data was taken from a single individual. Taking data from additional subjects would establish a continuum of gait characteristics to examine**
- **Gait characteristics for a wide variety of subjects should be examined to determine a specific set of 'primary' factors which influence an individual's gait**

The biggest obstacle to more in-depth analysis and decisive conclusions is a limited sample size of the data. Currently, all the GTRI motion capture data is sourced from a single individual. It is currently not possible to evaluate the relative variability of gait in one subject versus another subject. As such, it is difficult to characterize if the observed changes in gait will be visible across multiple individuals. Additionally, development of a gait "profile" which can be applied to individuals based on their height, estimated weight, etc, would be a useful tool for identification. Departures from a predicated profile would act as a target signature. Such a profile would have to adequately predict the gait of a wide variety of individuals, and successful development depends upon a reasonably large sample of individual gaits with which to test and validate the profile(s).

Optimum Gait

- **Gait is a complex motion with many factors determining the final motion**
- **Most studies ask subjects to walk at their most “comfortable pace” which the body has naturally optimized its movement for**
- **Deviation from this optimum produces inefficiencies in the walk and greater movement of the body’s center of mass, particularly in the vertical direction**

This slide continues the discussion of an optimum gait or profile for each individual.

Remote Determination of Optimum Gait

- Further data analysis may indicate key factors in determining an individual's "optimum gait" via remote observation
- If an optimum gait is identified, any deviation from a generic profile may be used as an identifier assigned to any individual under observation
- Furthermore, deviation from a constructed profile could be used to infer any external perturbations on the gait, including a visible burden, concealed burden, or possible injury

Optimum Gait and Perturbations

- It is worth noting, however, that the literature indicates an individual's gait is the result of a series of optimizations and corrections based upon the body's condition.
- Any deviation from a 'standard' gait is a result of a perturbation of sufficient strength to show *through* the body's attempts to correct or compensate for that perturbation.
- Lack of conclusive markers in the current data could indicate the applied external weights were insufficient to overload the subject's compensations for that load

Conclusions

- **Our current method is effective at reproducing an individual's gait**
- **It is possible to recognize an individual through gait, though current methods are imperfect. A wider dataset covering more individuals would be ideal for further investigation**
- **Using simulated imagery as a foundation for algorithm development in tracking, identification, and analysis applications is valid and economical**

References

1. Richard W. Bohannon (1997). Comfortable and maximum walking speed of adults aged 20-79 years: reference values and determinants. *Age and Ageing* (26), 15-19.
2. de Souza S.A.F. , Faintuch J., Valezi A. C., Sant'Anna A. F., Gama-Rodrigues J. G., Senhorini R. C. (2005). Gait Cinematic Analysis in Morbidly Obese Patients. *Obesity Surgery* (15), 1238-1242.
3. Boulgouris N. V. , Plataniotis K. N. , Hatzinakos D. (2006). Gait recognition using linear time normalization. *Pattern Recognition* (39), 969-979.
4. Phillips P. J., Sarkar S., Robledo I., Grother P., Bowyer K. (2002). The Gait Identification Challenge Problem: Data Sets and Baseline Algorithm. *IEEE, 16th International Conference on Pattern Recognition* (2), 385-388.
5. Kale A., Sundaresan A. , Rajagopalan A. N., Cuntoor N. P. , Roy-Chowdhury A. K. , Krueger V., Chellappa R. (2004). Identification of Humans Using Gait. *IEEE Transactions on Image Processing* (13), No. 9, 1163-1173.
6. Finch L., Barbeau H., Arsenault B. (1991). Influence of Body Weight Support on Normal Human Gait: Development of a Gait Retraining Strategy. *Physical Therapy* (71) #11, 64-77.
7. Wearing S. C., Hennig E. M., Byrne N. M., Steele J. R., Hills A. P. (2006). Musculoskeletal disorders associated with obesity: a biomechanical perspective. *Obesity Reviews* (7), 239-250.

References 2

1. Buelthoff H., Little J., Poggio T. (1989). A Parallel Algorithm for Real-Time Computation of Optical Flow. *Nature* (337), #9, 549-553.
2. de Souza S. A. F., Faintuch J., Valezi A. C., Sant'Anna A. F., Gama-Rodrigues J. J., Fonseca I. C. B., de Melo R. D. (2005). Postural Changes in Morbidly Obese Patients. *Obesity Surgery* (15), 1013-1016.
3. Liu Z., Sarkar S. (2006, June). Improved Gait Recognition by Gait Dynamics Normalization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (28), #6, 863-876.
4. Hayfron-Acquah J. B., Nixon M. S., Carter J. N. (2003, September). Automatic Gait Recognition by Symmetry Analysis. *Pattern Recognition Letters* (24), #13, 2175-2183.
5. Foster J. P., Nixon M. S., Pruegel-Bennett A. (2003, October). Automatic Gait Recognition Using Area-Based Metrics. *Pattern Recognition Letters* (24, #14), 2489-2497.
6. Murase H., Sakai R. (1996). Moving Object Recognition in Eigenspace Representation: Gait Analysis and Lip Reading. *Pattern Recognition Letters* (17) 155-162.
7. Little J. J., Boyd J. E., (1998). Recognizing People by Their Gait: The Shape of Motion. *Videre: Journal of Computer Vision Research* (1, #2) 2-32.
8. Nixon M. S., Carter J. N., Cunado D., Huang P. S., Stevenage S. V. (2002). Automatic Gait Recognition. *Biometrics*. Norwell, Massachusetts, Kluwer Academic Publishers.

References 3

1. Kuo D. (2001, June). A Simple Model of Bipedal Walking Predicts the Preferred Speed-Step Length Relationship. *Journal of Biomechanical Engineering* (123 #3), 264-270
2. Clark-Carter D. D., Heyes A. D., Howarth C. I. (1986, June). The efficiency and walking speed of visually impaired people. *Ergonomics* (29, #6), 779-789.
3. Abe D., Yanagawa K., Niihata S. (2004, July). Effects of load carriage, load position, and walking speed on energy cost of walking. *Applied Ergonomics* (35 #4), 329-335.
4. Whittle M. W. (2007). *Gait Analysis: an Introduction* (4th ed.). Elsevier Ltd.

Urban Scene Modeling

**Keith Prussing, Sarah Lane, Shayla Otolorin,
and Spencer Nichols**

Electro-Optical Systems Laboratory

Georgia Tech Research Institute

September 14, 2010

Keith.Prussing@gtri.gatech.edu

Shayla.Otolorin@gtri.gatech.edu

Sarah.Lane@gtri.gatech.edu

Spencer.Nichols@gtri.gatech.edu

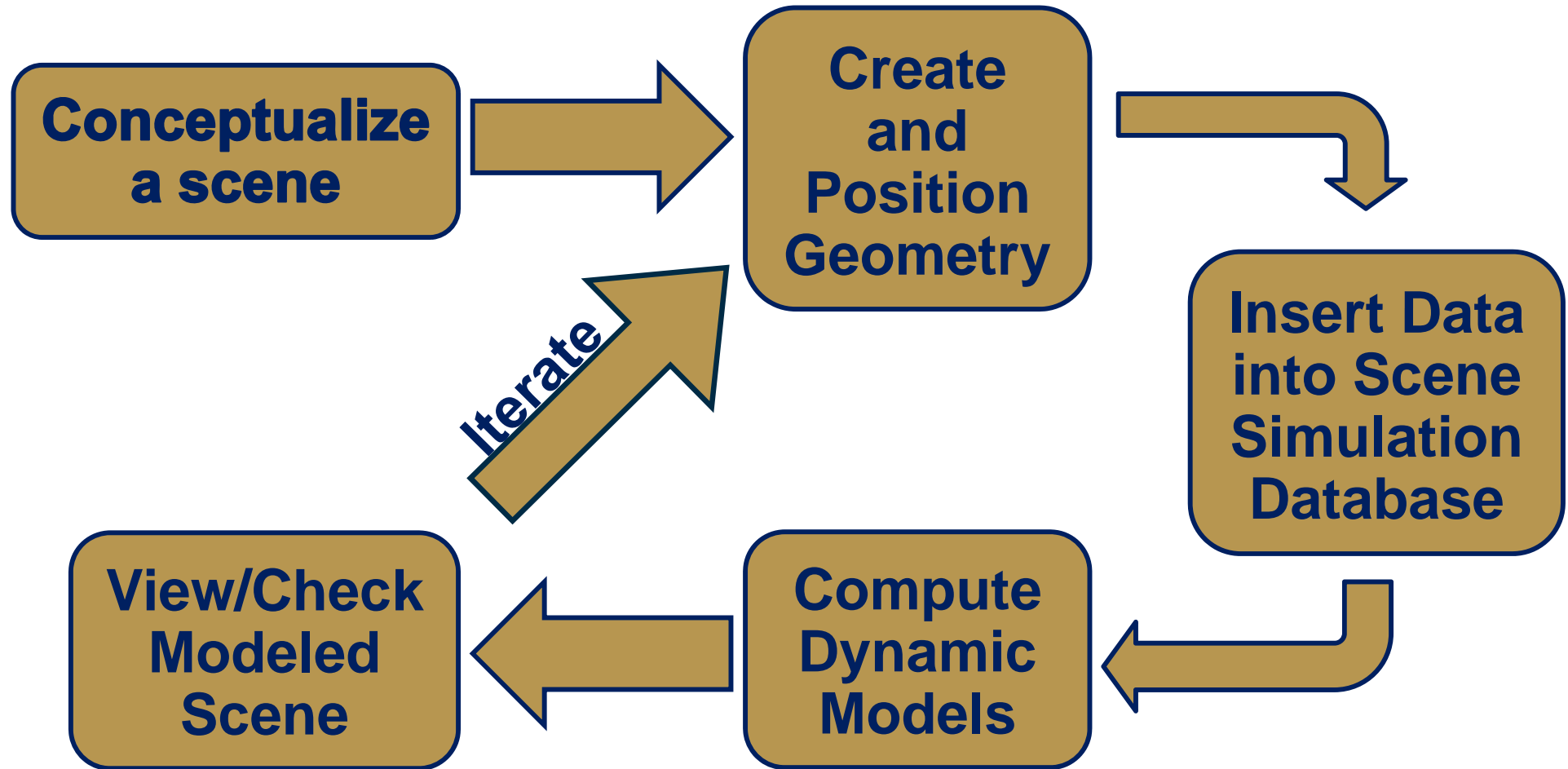
Outline

- **Scene Generation Process**
- **Scene Simulation**
- **Scene Visualization**
- **Signature Generation Procedure**
- **Urban Scene Development**
- **Physically Based Ray Tracing**

Scene Modeling

- **Interesting scenarios are not static**
 - We created a system that updates object positions and signatures as a function of time
- **Need a means to recreate dynamic situations**
 - We compute the motions and signatures at key moments of time and store the results in the scene simulation database for later use

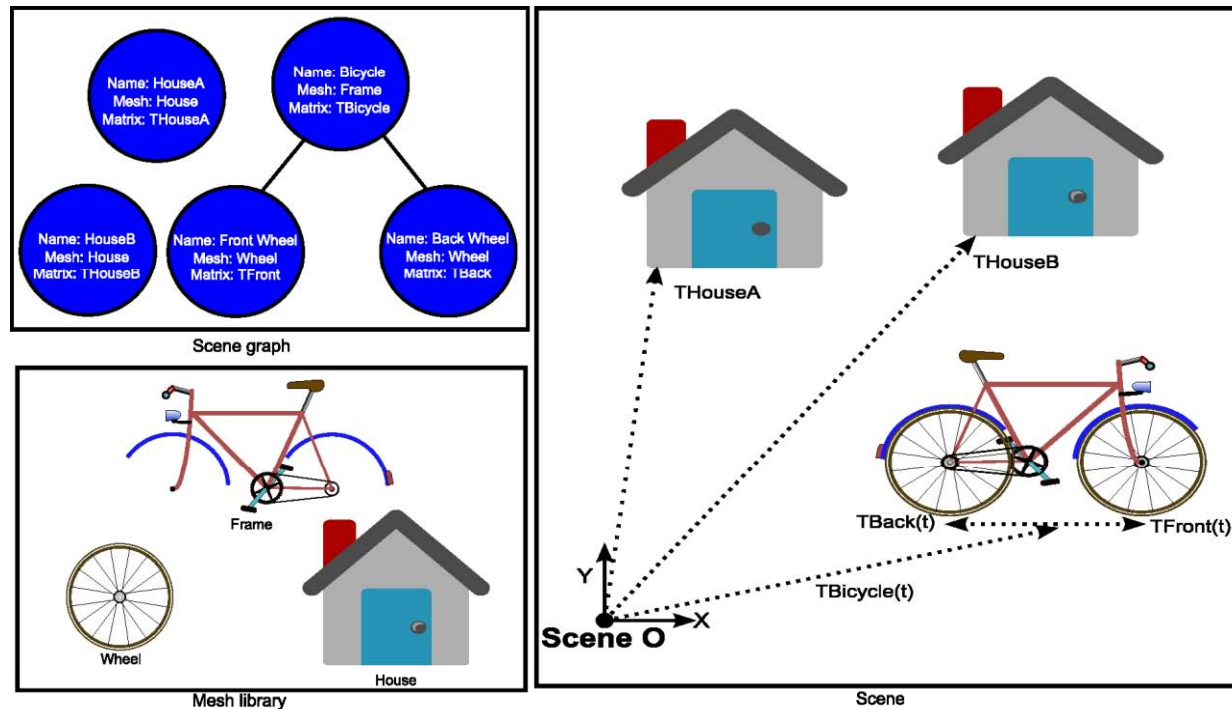
Scene Generation Process



Scene Creation

- **A Scene is**
 - **A collection of static elements**
 - Buildings, roads, trees
 - **Dynamic elements**
 - Cars, people, animals
 - **And their interactions**
 - Shadows, obscuration, wind induced motion

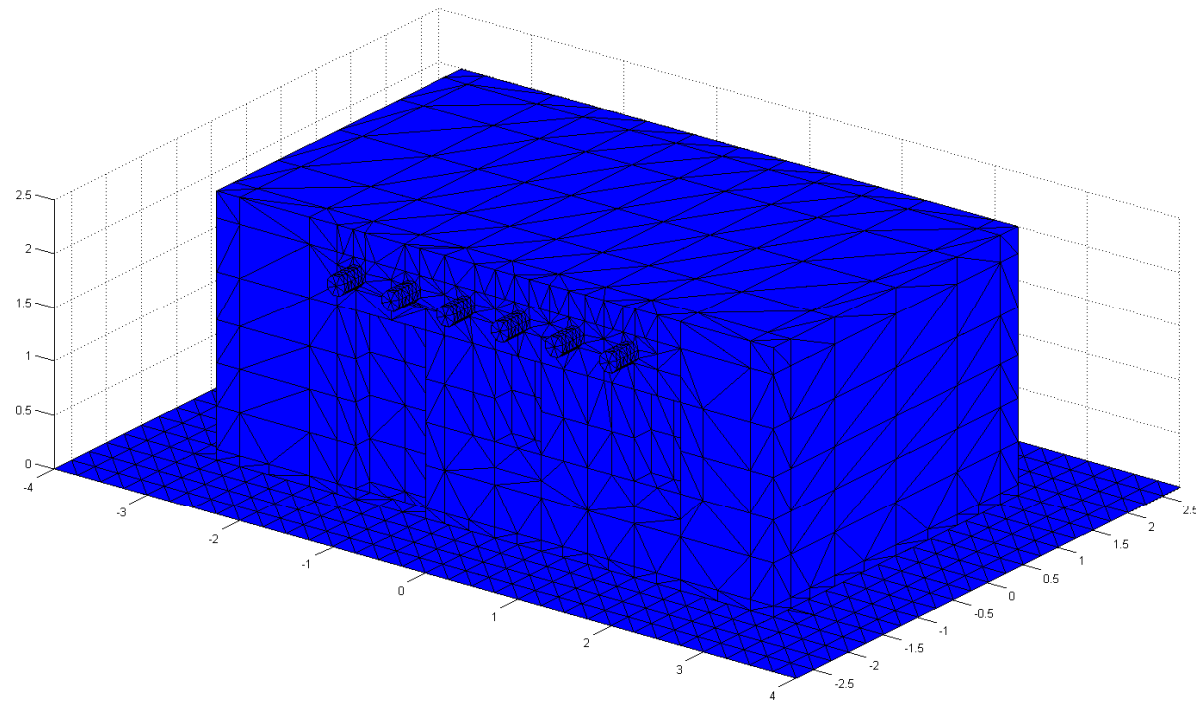
Scene Creation



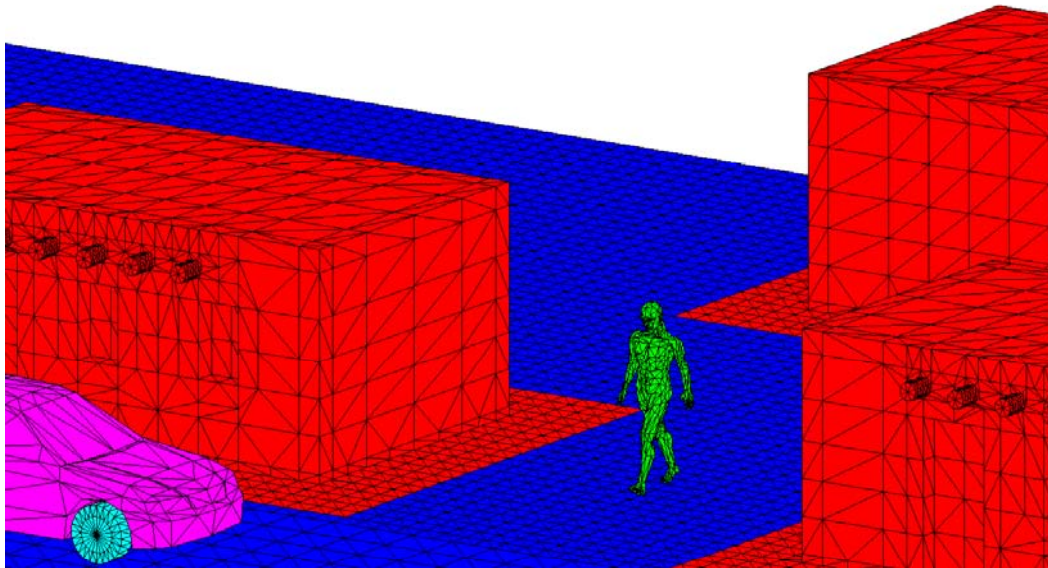
- A scene has a coordinate system
- A scene is associated with a dynamic aspect termed the “simulation”.

Geometry Representation

- Surfaces are represented by a polygon mesh
- Stored in Scene-Simulation Database as a “mesh”
- Vertices are in the local coordinate system of object



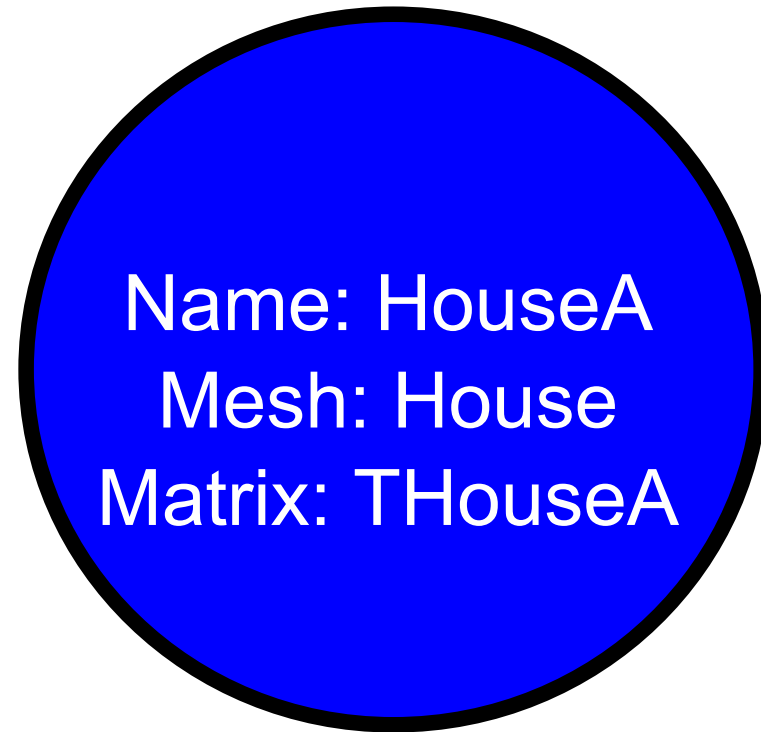
Scene Representation



- Scene is represented as a graph in the Scene-Simulation Database
- Contains all objects, “scene nodes,” and their relations
- Associated with a dynamic aspect termed the “simulation” that evolves the time dependant features

Scene Node

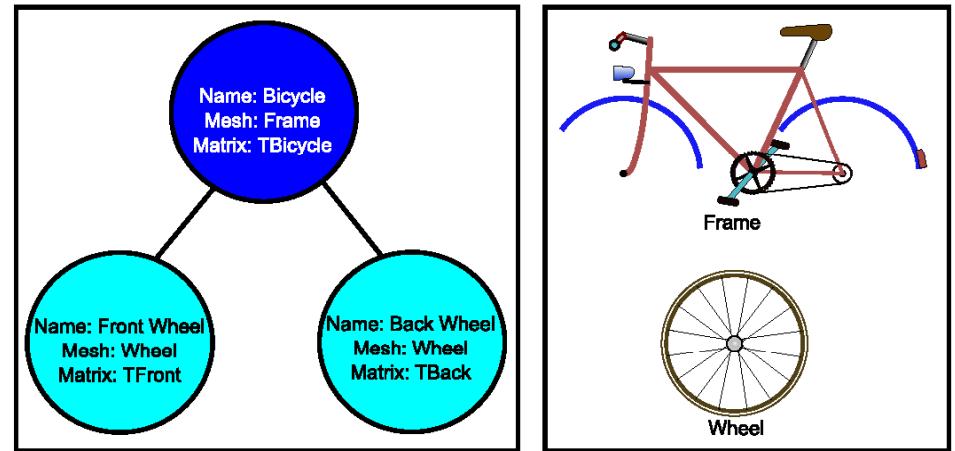
- Each scene node has:
 - A unique identifier, “name”
 - A geometric definition, “mesh”
 - A position and orientation, “matrix”
- A scene node is a unique instantiation of a mesh
- Each scene node has a transformation from local to global coordinates
- A scene node may have child scene nodes
 - Child inherits all transformations of the parent



A Scene node

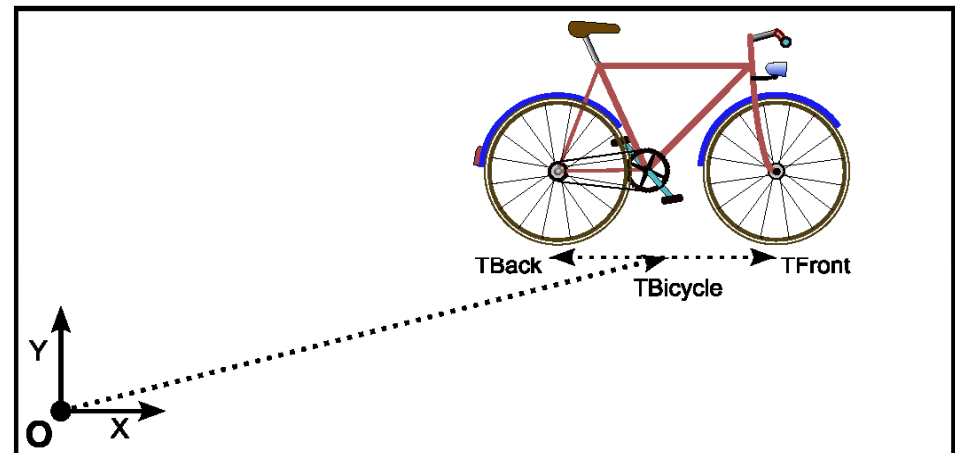
Parent-Child Scene Nodes

- A child scene node inherits all of the transformations of the parent
- This allows objects that move together to be easily computed



Scene graph

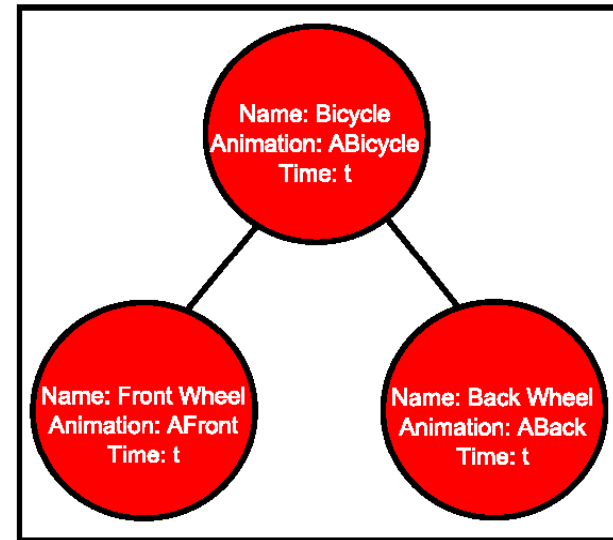
Mesh library



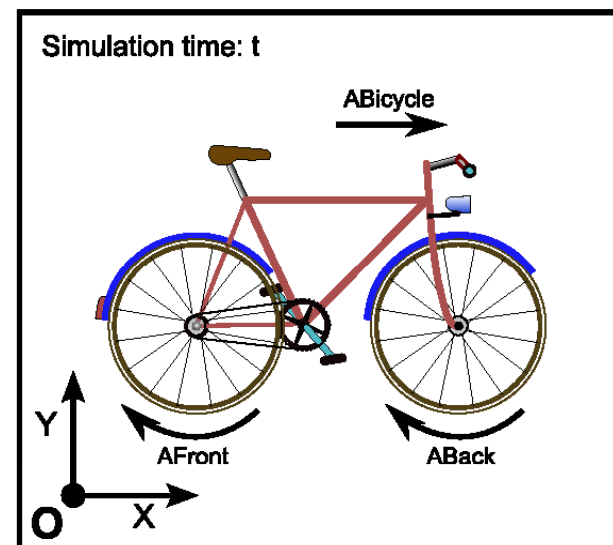
Scene described by Scene graph

Simulation

- The simulation is the time evolution of the scene
- The simulation tracks the time and updates the animations and signatures as a function of time
- The key frames are the times when locations and signatures of scene nodes are known exactly
 - Visualizer uses a linear interpolation if time is between two key frames
- The simulation is used for image generation



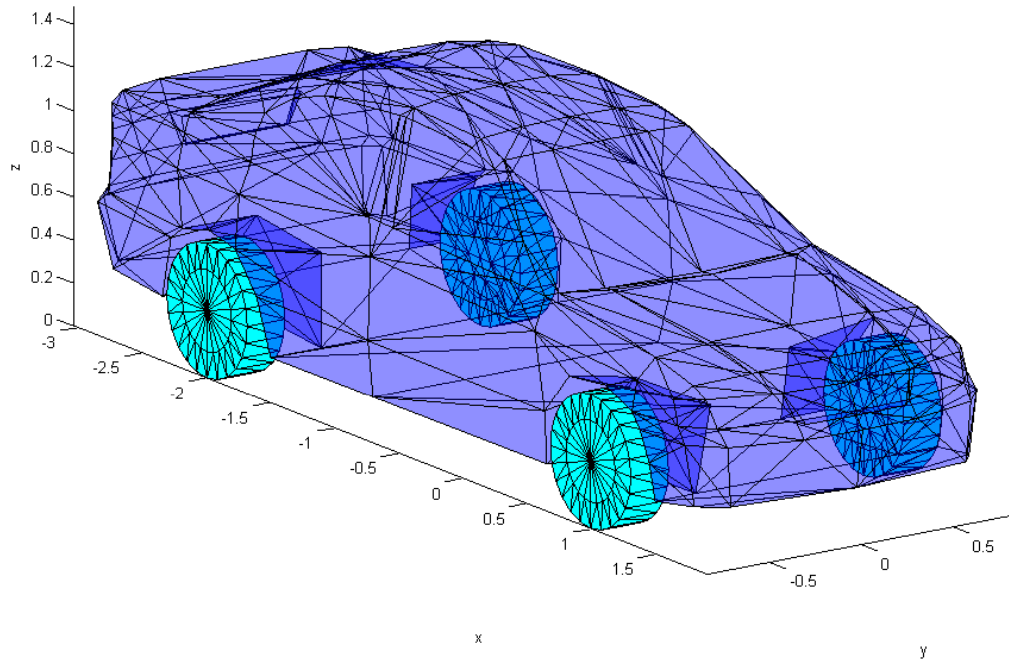
Simulation graph



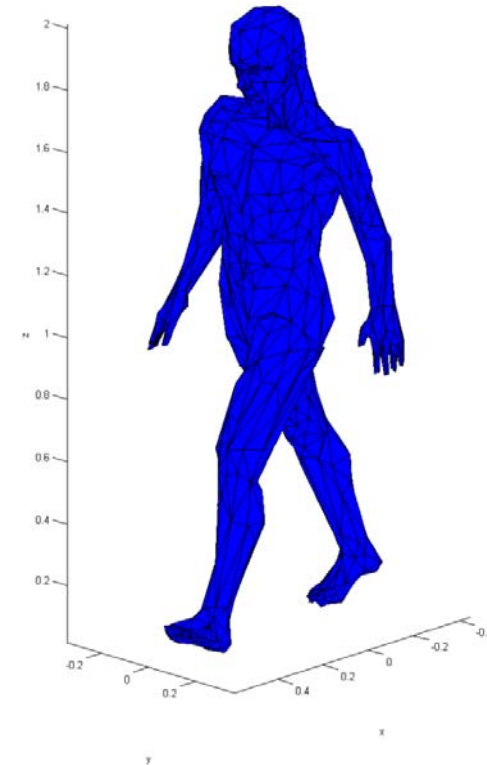
Scene described by Simulation graph

Object Motion

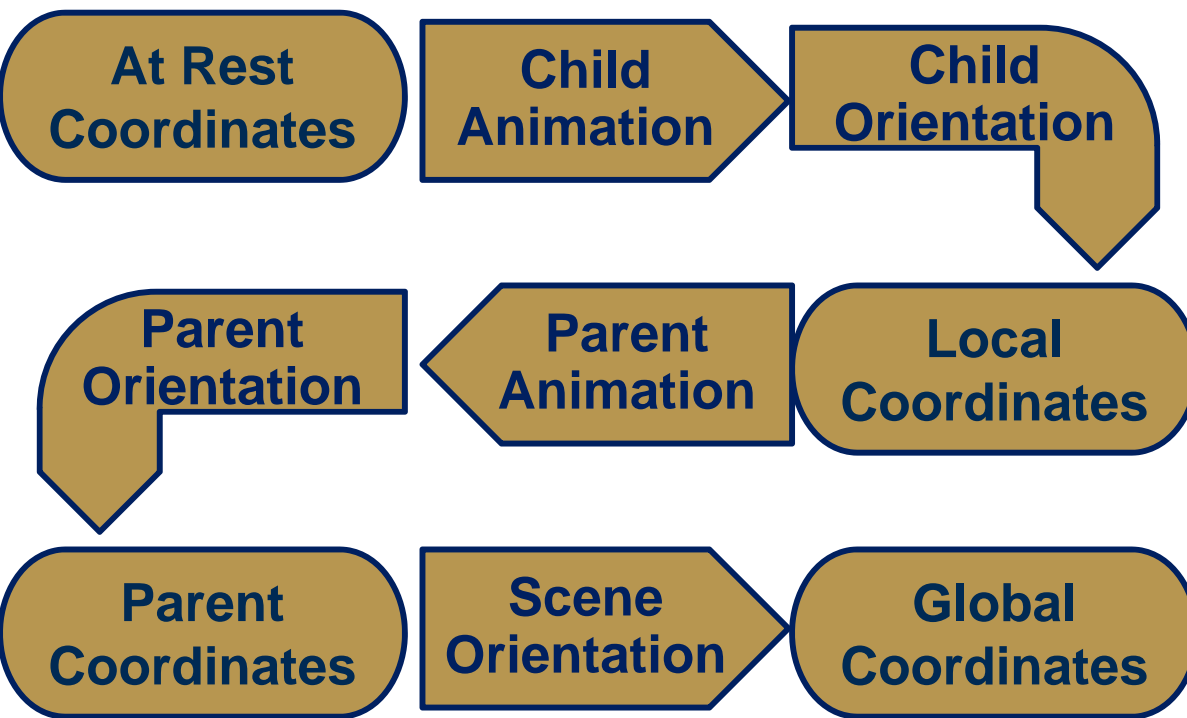
Independent parts



Deformable bodies

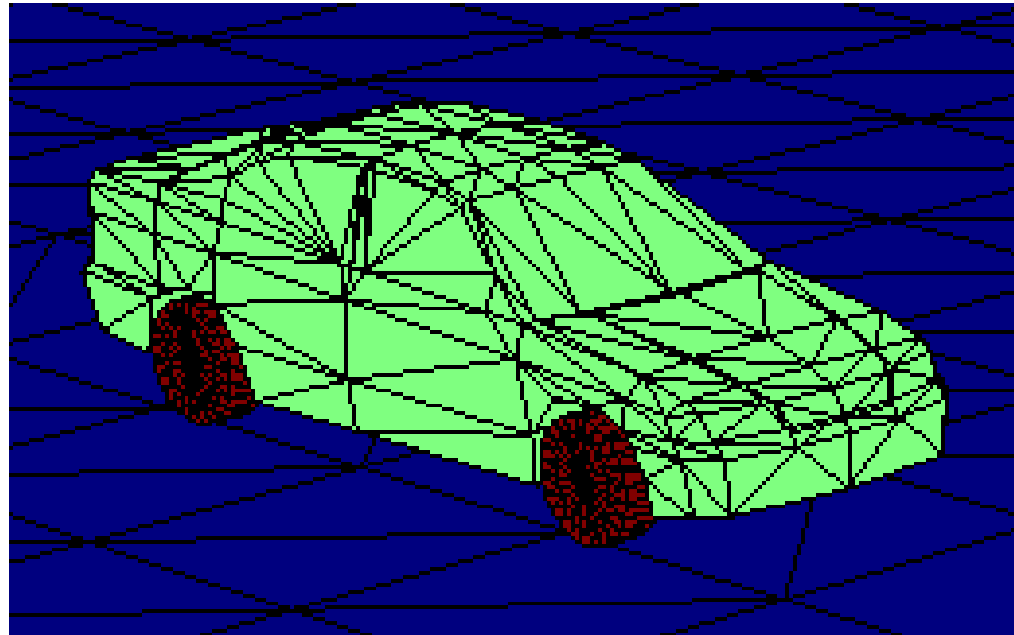


Independent Motion



- Utilize Parent-Child scene node relations
- The child scene node is animated in its local coordinate system and then receives the parent animation

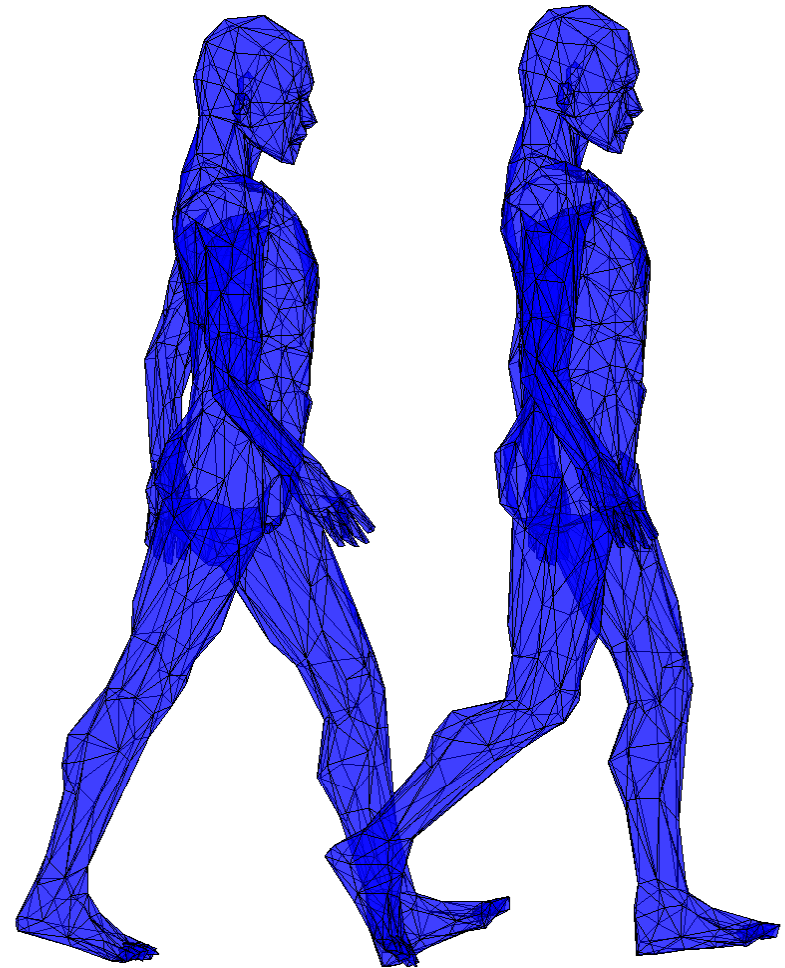
Independent Motion



- Vehicle motion is one example
- The vehicle is modeled as a vehicle body (parent scene node) and four wheels (child scene nodes)
- The physically based vehicle dynamics model and its interface with the Scene-Simulation Database is described in “*First Principles Based Vehicle Dynamics Model*” by Keith Prussing

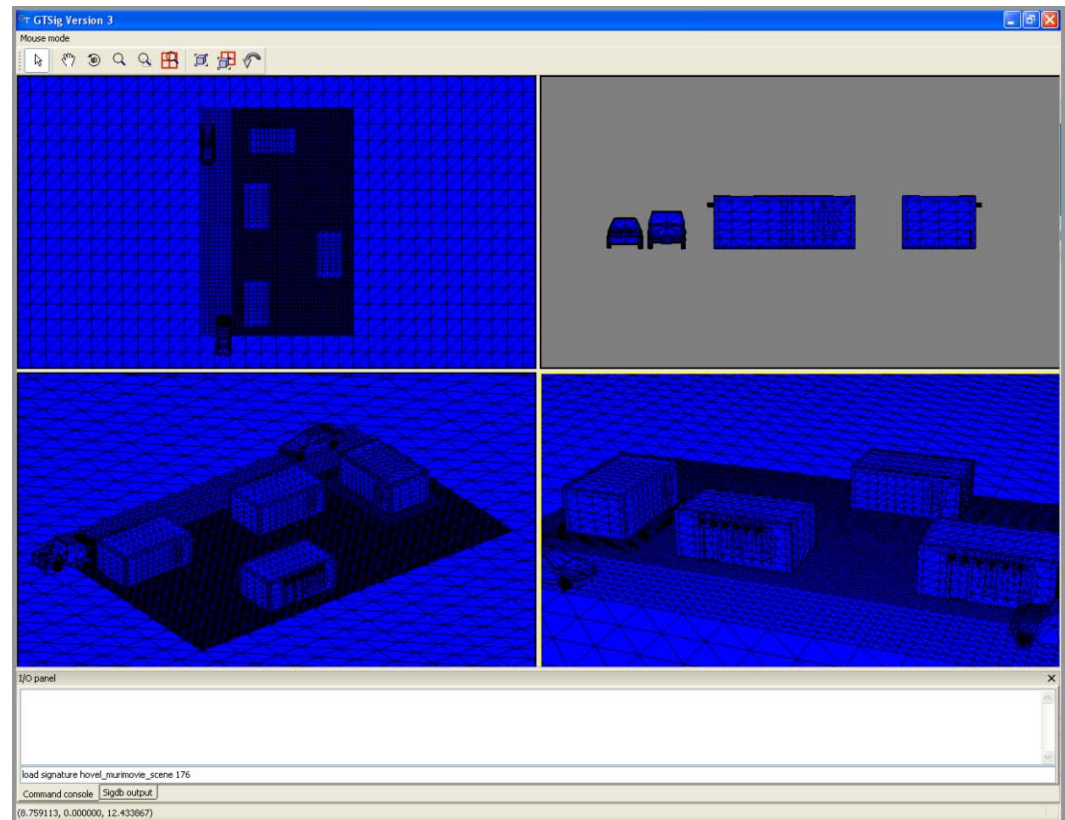
Deformable Bodies

- Pose for each key time is stored in Scene-Simulation Database
- At a given time, simulation determines which pose to draw
- Global position is determined by center of mass motion

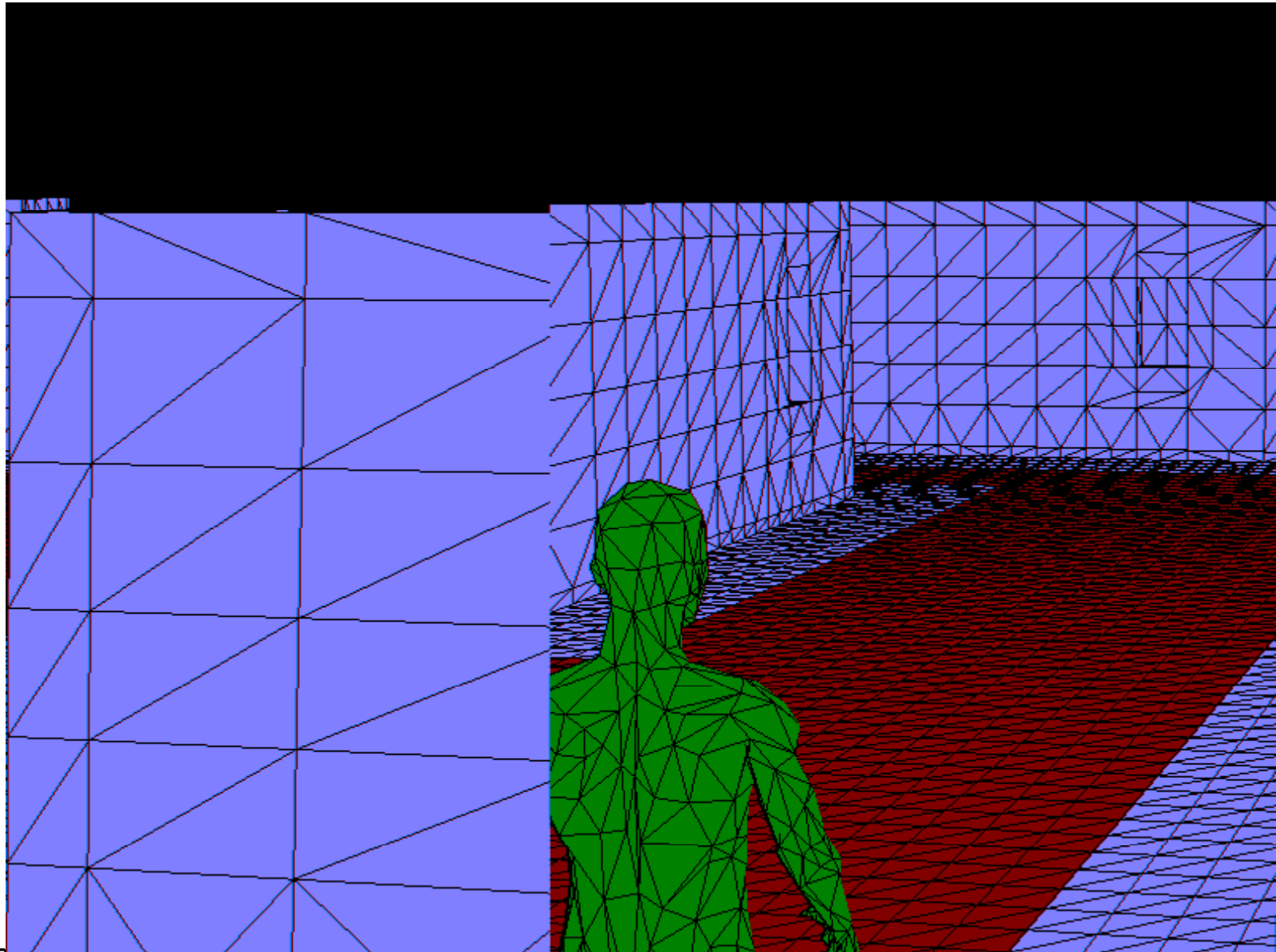


Putting it All Together

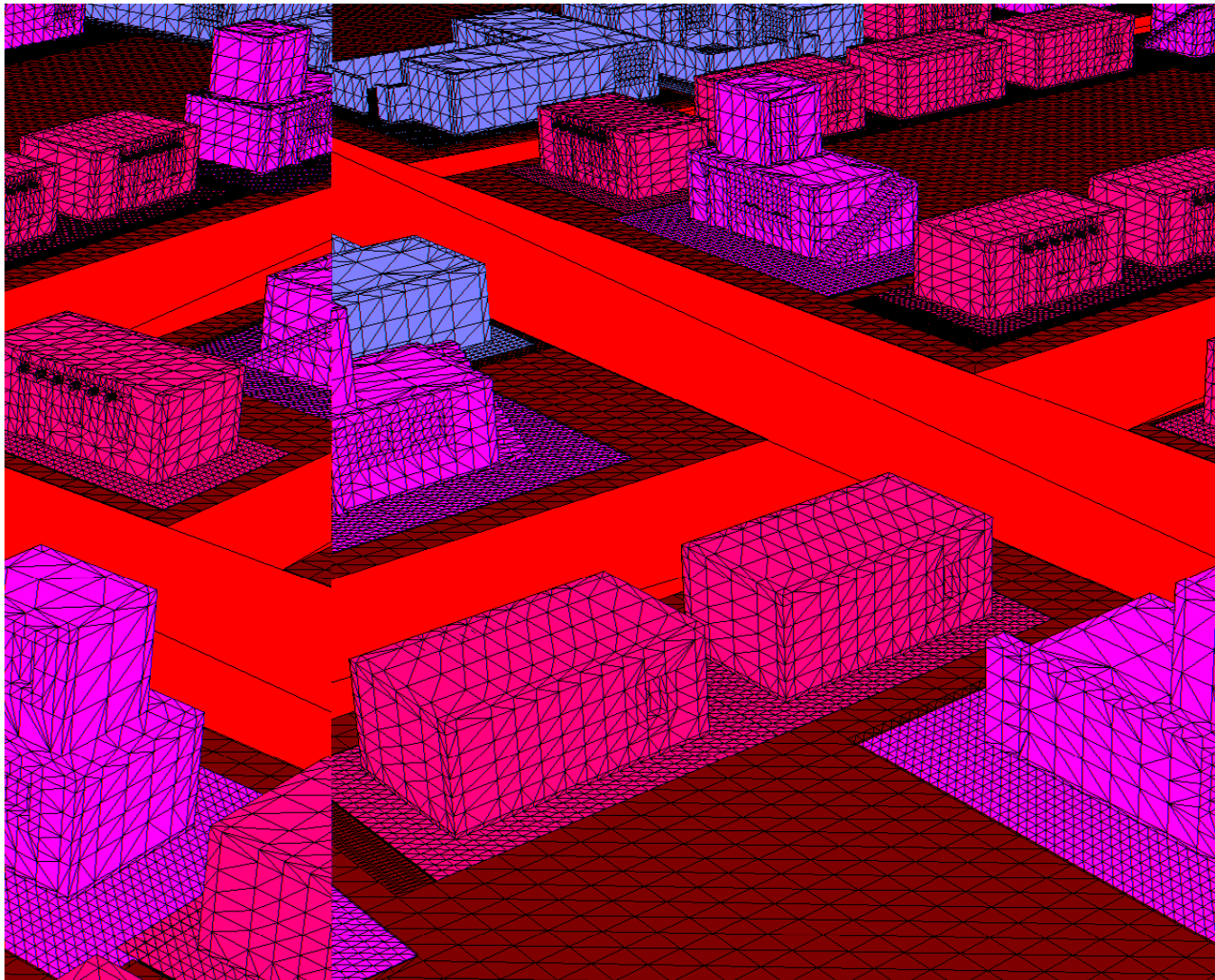
- We load a scene and simulation into the visualizer
- We produce a series of bitmap frames
- We compile the frames into a movie



The Wire Frame Product



The Wire Frame Product



Signature Generation Procedure

- **For signature generation one must first determine:**
 - **The geographic location of the scene**
 - **The date and time of the simulation**
 - **The current weather conditions of the scene**
 - **The sensor waveband(s)**
 - **The relationship of scene elements**

Signature Generation Procedure

- **With the scene scenario established, one must select the source of the signature data:**
 - **Measured data from representative sensor in the waveband of interest, or**
 - **Modeled data from model of choice**
- **This signature data is stored in the Scene-Simulation Database**
- **The simulation process accesses the Scene-Simulation Database to retrieve signature**
 - **The signature can change with time and/or change in scene geometry**

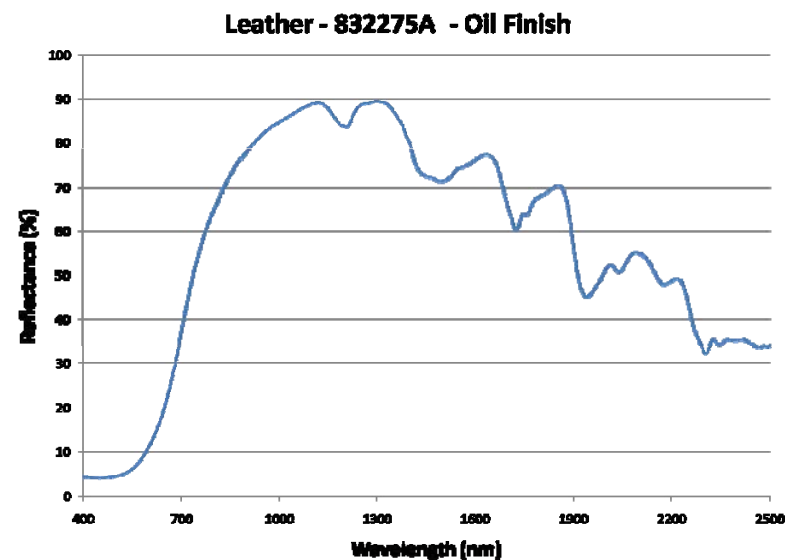
Signature Model Data Sources

- **External Sources**
 - **ASTER database (0.3 – 14 μm)**
 - **NEF – Non Conventional Exploitation Factors database**
- **Lack of sufficient multispectral data for signature model development (particularly for personnel detection issues) necessitates the development of the Scene-Simulation Database.**
- **This database approach provides flexibility for storing and accessing multi-source data.**

Signature Model Data Sources

- **Internal (EOSL) Sources**
 - **Spectrometers**
 - Cary (300 – 3000 nm)
 - Ocean Optics (200 – 1100 nm)
 - B&W Tek (900 – 1700 nm)
 - ASD (350 – 2500 nm)
 - **Radiometer**
 - D&P TurboFT (2.5 – 16 μ m)
 - **Hyperspectral Imager**
 - Telops Hyper-Cam (8 – 11 μ m)
 - **Other Imagers**
 - Sensors Unlimited (900 – 1700 nm)
 - FLIR ParthFindIR (8 – 14 μ m)
 - FLIR A40M (8 – 14 μ m)

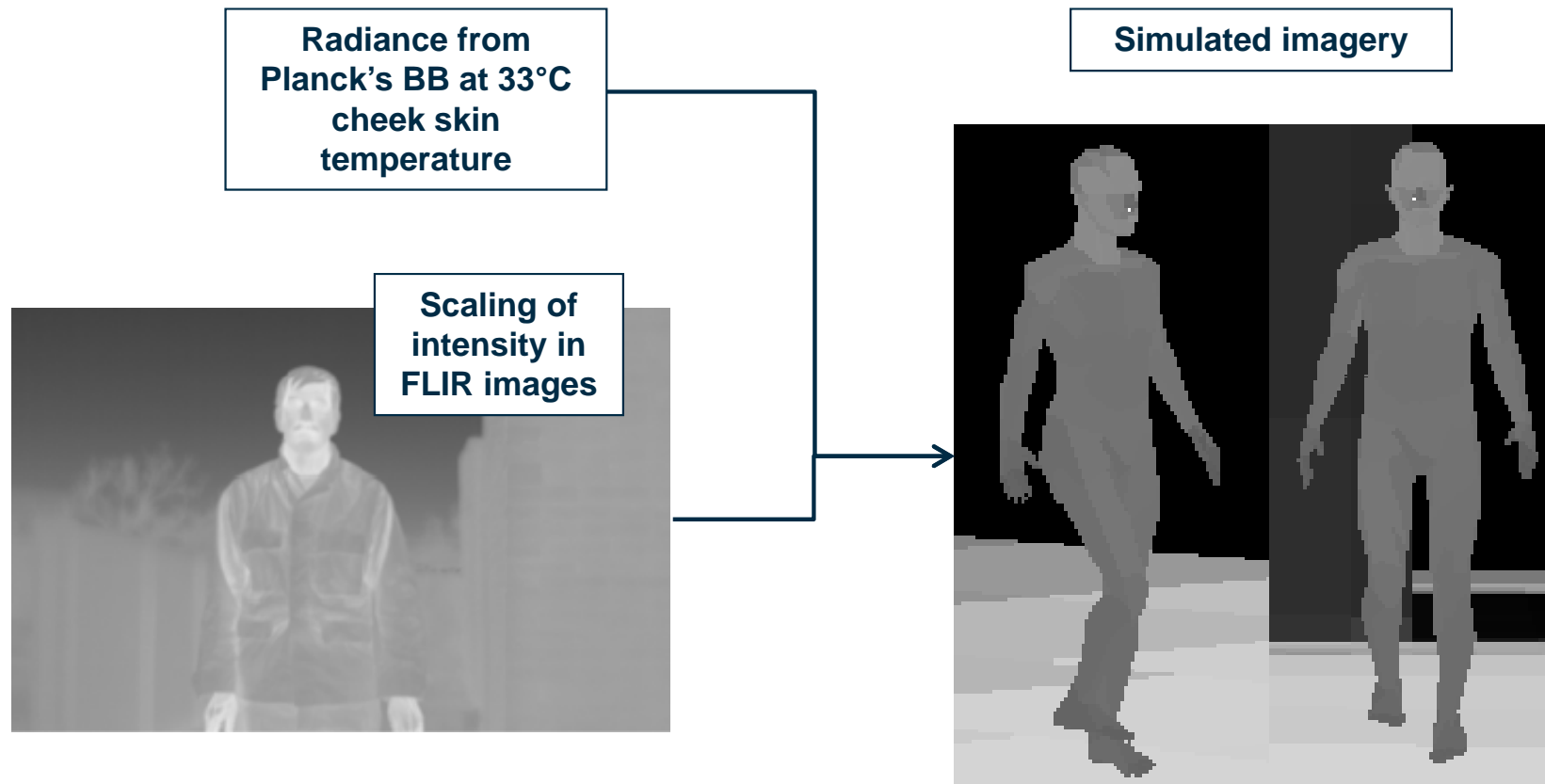
Example Cary data



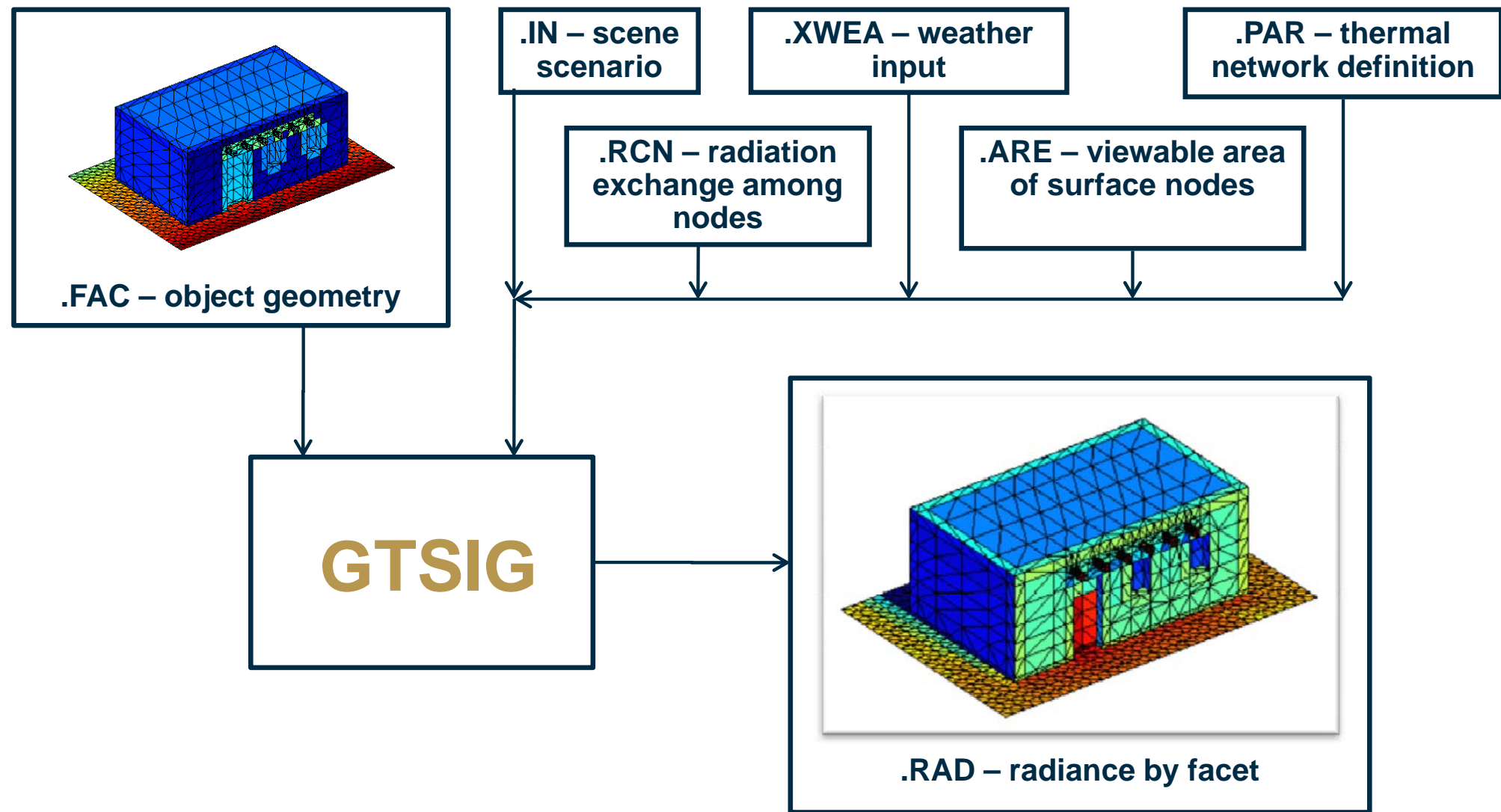
Measured spectral data is integrated over wavebands of interest to provide signature estimates for materials in the scene.

Measured Signature Data

- Human thermal signatures are estimated based on an average human skin temperature and available imagery.



Thermal Signature Modeling GTSIG



Thermal Network

- The thermal network defines the heat transfer through the entire model.
- The model itself is divided up into volumes (nodes) that are connected in the network.
- For each connection that participates in heat transfer, one must define conduction or convection.

$$COND = \frac{kA}{l} \quad CONV = hA$$

$$k = \text{conductivity} \left(\frac{W}{mC} \right)$$

$$A = \text{area} (m^2)$$

$$l = \text{length} (m)$$

$$h = \text{heat_transfer_coeff} \left(\frac{W}{m^2C} \right)$$

- For each node, one must define the capacitance for the volume.

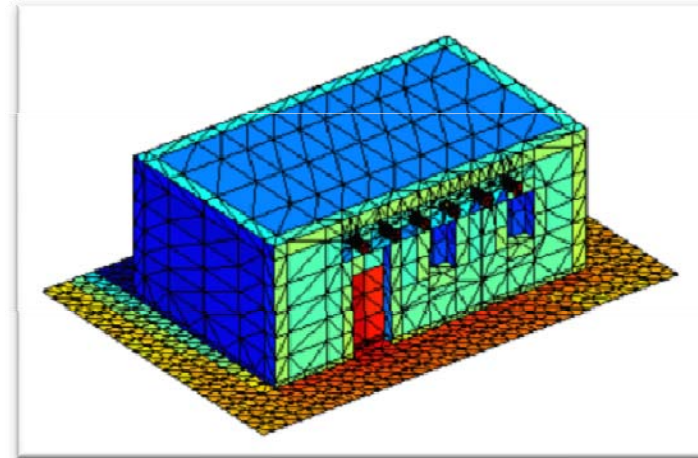
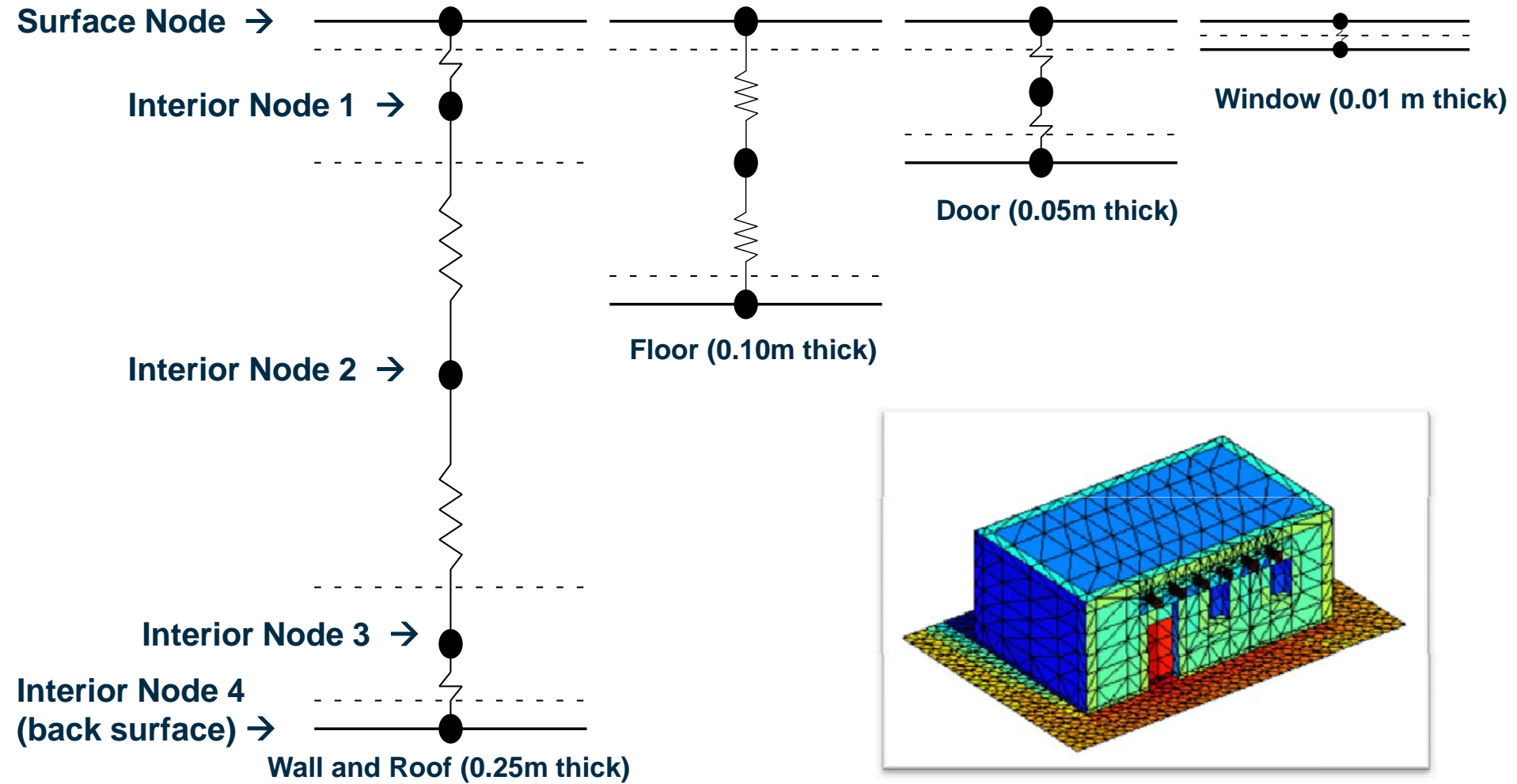
$$CAP = \rho C_p V$$

$$\rho = \text{density} \left(\frac{kg}{m^3} \right)$$

$$C_p = \text{specific_heat} \left(\frac{J}{kgC} \right)$$

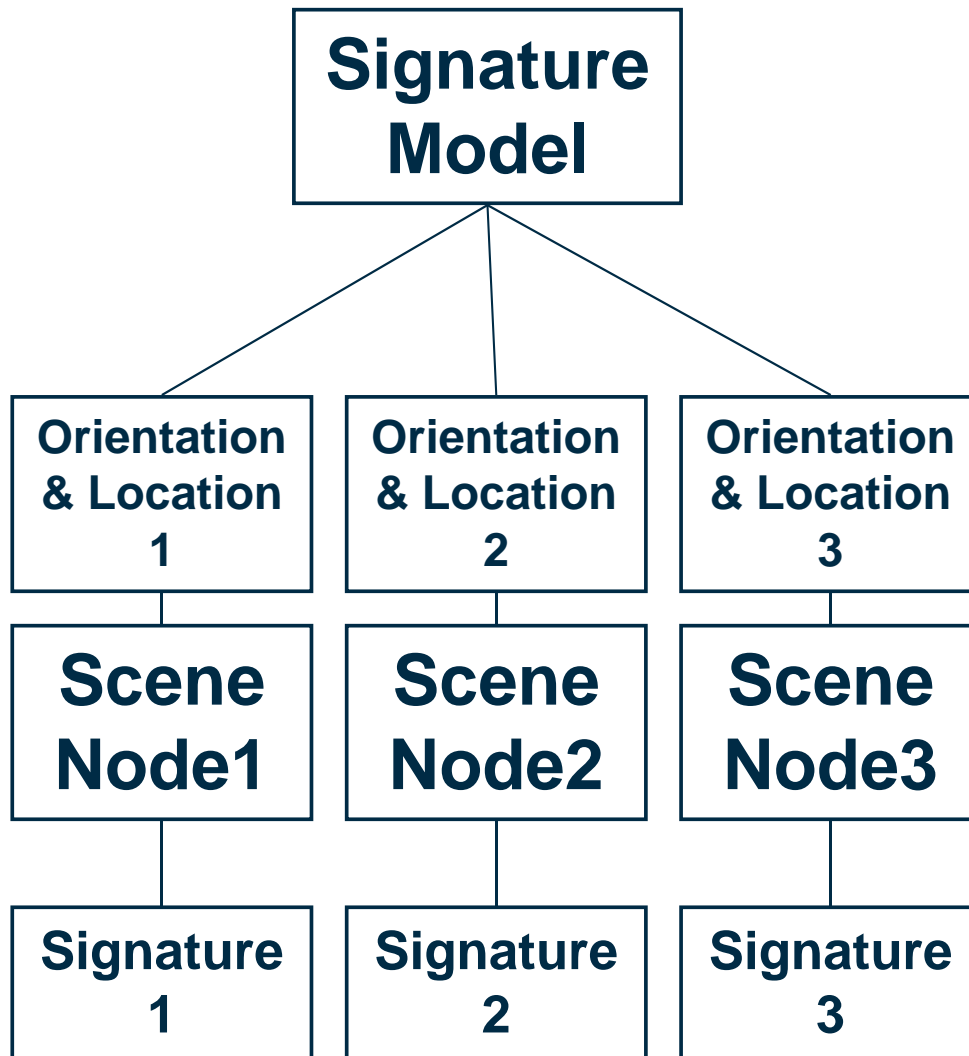
$$V = \text{volume} (m^3)$$

Thermal Network



The connection structure for the hovel is shown. The hovel is used for the test scene.

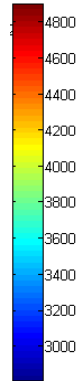
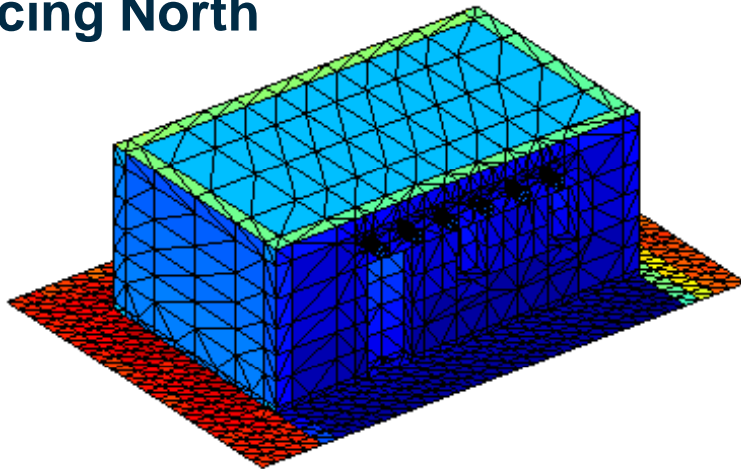
Signature Model



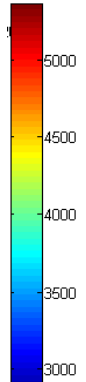
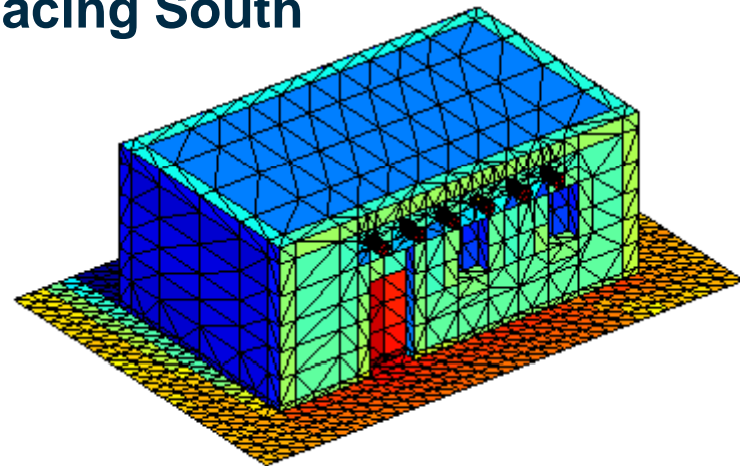
- The same geometry may be positioned at different locations and orientations (i.e. each scene node).
- For signature purposes, each scene node is treated as a unique object.
- The output radiances are painted on every facet on every scene node in the Scene-Simulation Database.

GTSIG output

Facing North

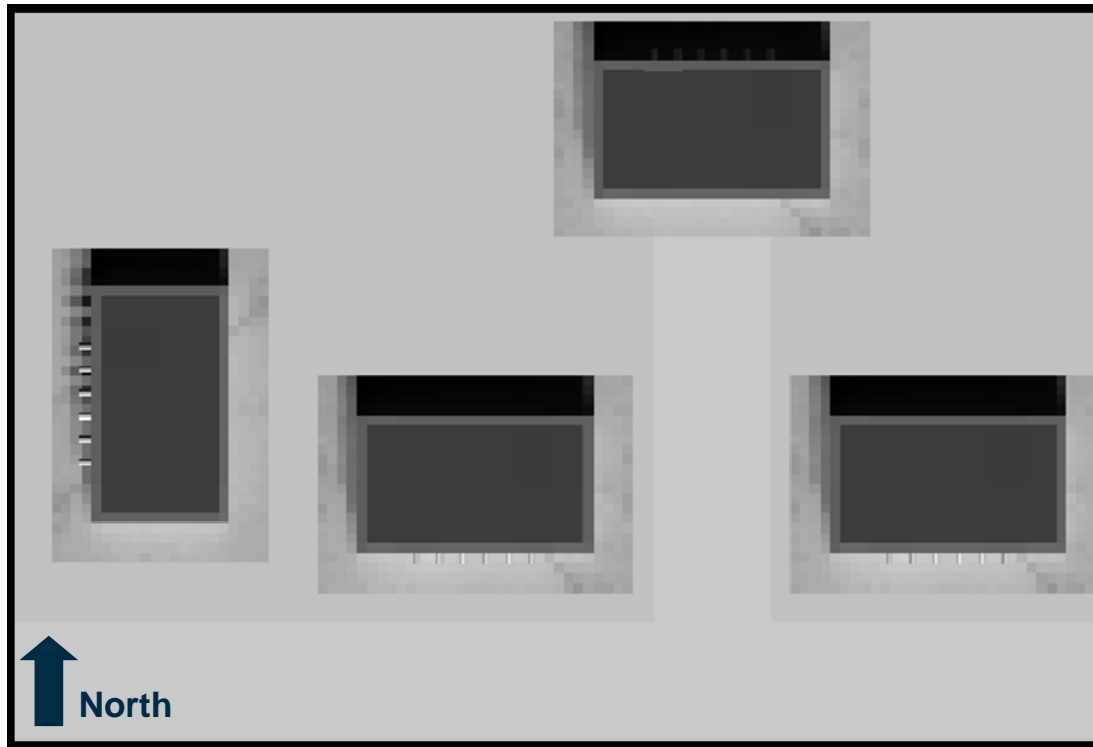


Facing South



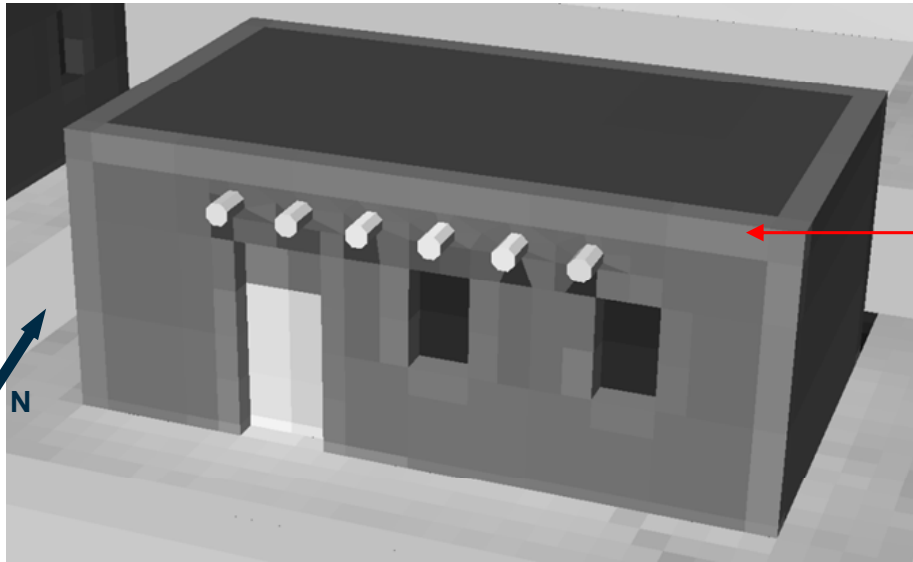
- Temperature ($^{\circ}\text{C}$) and radiance ($\mu\text{W}/\text{cm}^2\text{sr}$) are output for every facet at every time step.
- Signature of each facet is pre-computed using desired algorithm and stored at key times in the Scene-Simulation Database.
- Simulation interpolates to determine signature at specific rendering time.
- In addition to GTSIG calculated values, radiance, reflectance, or any other value from any source can be rendered on facets.

Test Scene – Hovels + Background

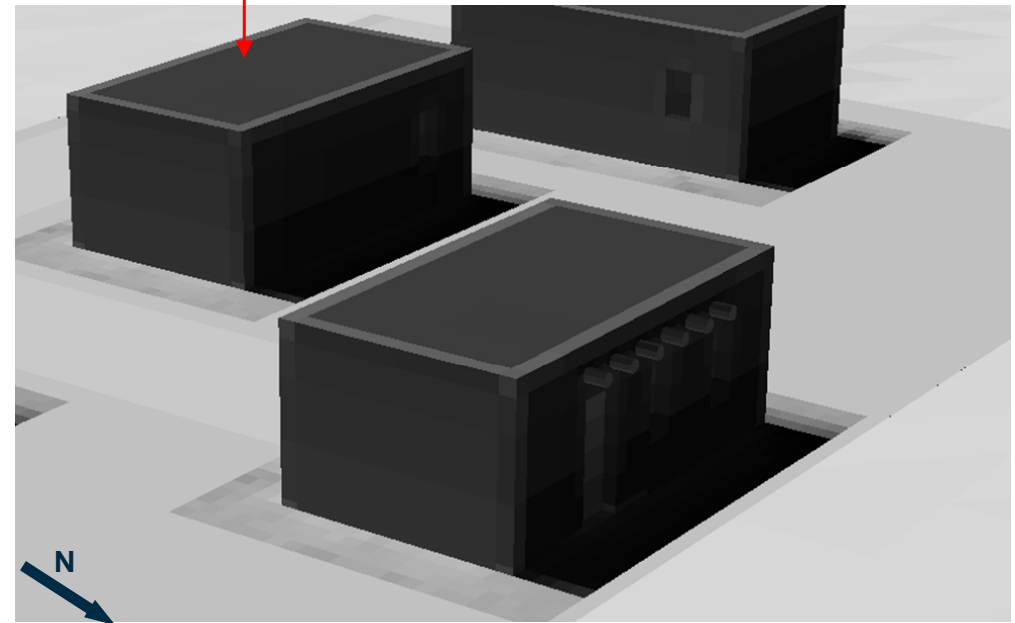


- The signatures are visualized on the scene geometry. They are scaled to a colormap (0 – 255 grayscale).
- Three geometric objects:
 - Hovel mesh
 - Background between hovels mesh
 - Background tile mesh
- 19 unique scene nodes:
 - Four hovels
 - One background between hovels
 - 14 background tiles
- Signatures calculated separately for each scene node with GTSIG for the following parameters:
 - Noon on February 23, 1986
 - Eglin AFB, FL
 - 8-12 waveband
- Signature colormap:
 - 2812 $\mu\text{W}/\text{cm}^2\text{sr} \rightarrow 0$
 - 5480 $\mu\text{W}/\text{cm}^2\text{sr} \rightarrow 255$

Test Scene – Hovels + Background

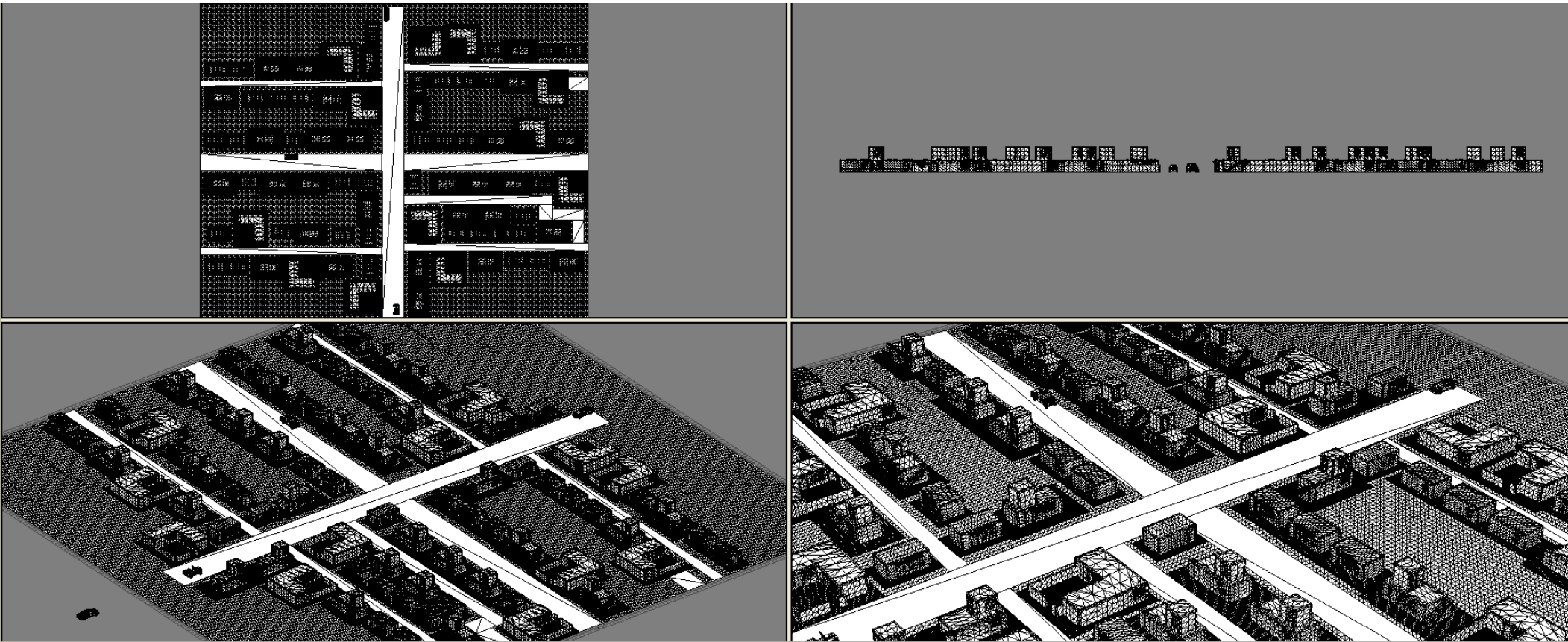


Scene Node 4, faces South



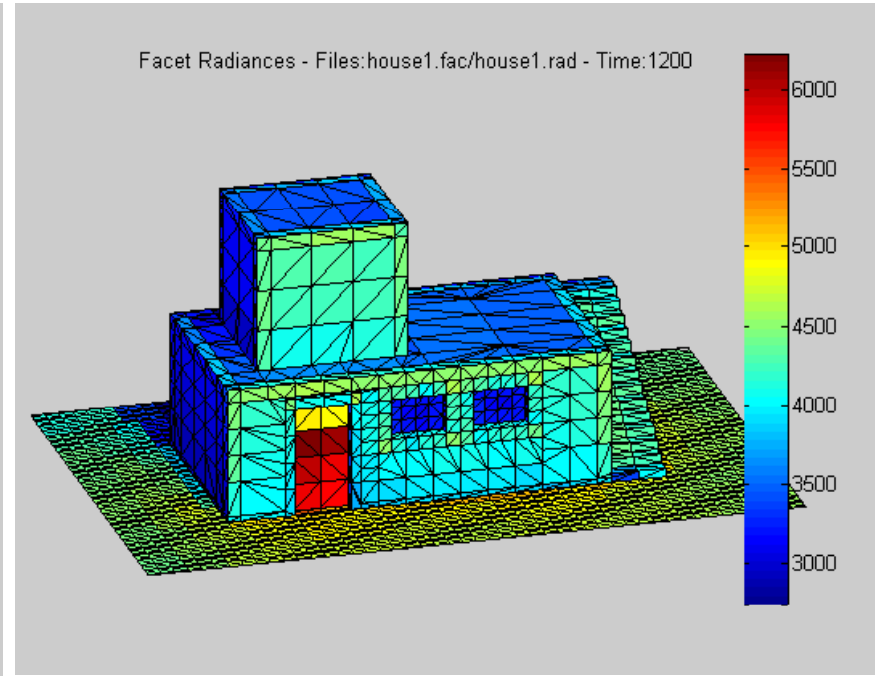
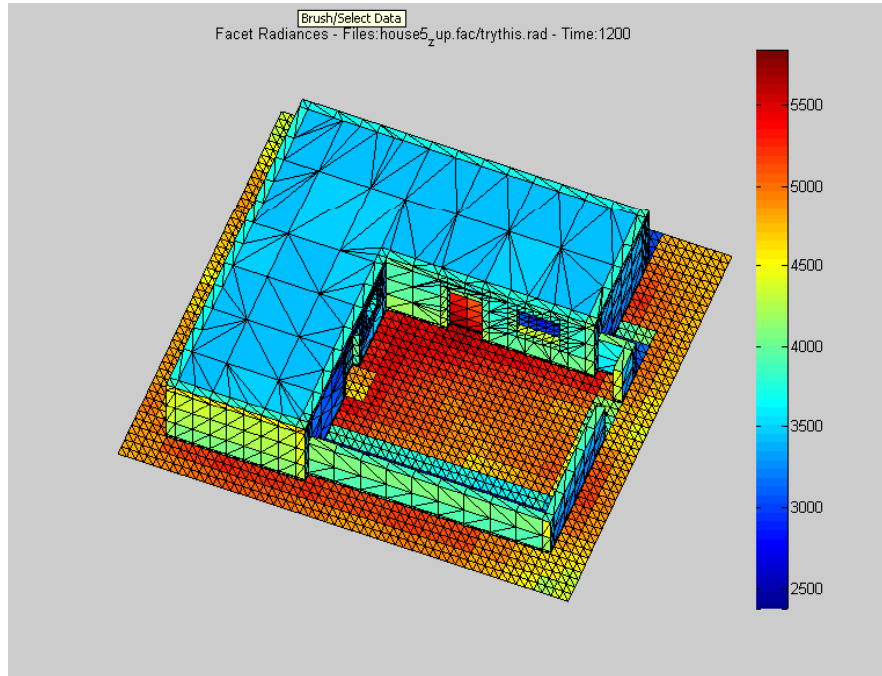
- Local ground around hovel allow us to see the interaction of the hovel with its background.
- We can see thermal shadows on the North sides of all the hovels, underneath the posts, and on the door.

Urban Scene



- The hovel test scene was expanded to include a variety of structures.
- In addition to the hovel, geometry was created for two houses, a sedan, a truck, and a jeep.

Urban Scene



- **GTSIG models were created for the two houses and the background.**
- **The radiance output from the GTSIG models were inserted into the Scene-Simulation Database for visualization.**

Urban Scene



Long wave IR visualization of urban scene at noon on February 23, 1986, Florida.

Visible Image Rendering

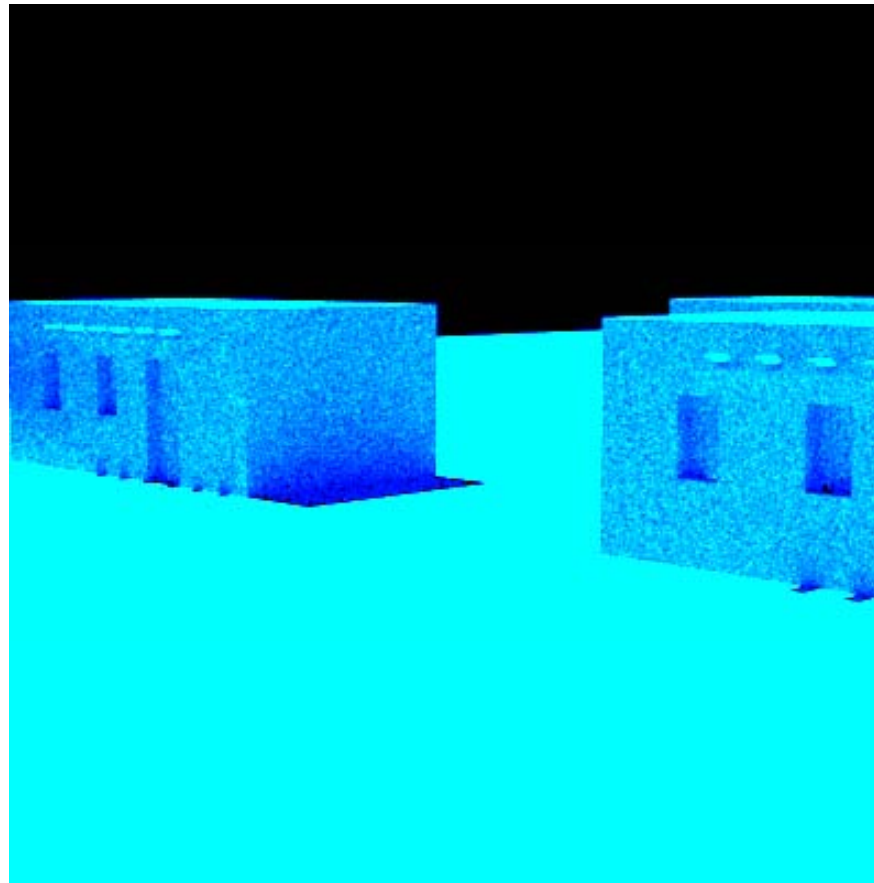


- LuxRender is a freeware renderer for generating visible band images.
- Materials and corresponding reflectances are assigned independently to the each facet group on the object.
- Input data files for LuxRender are exported from the scene simulation database.
- With these data LuxRender generates a color image.
 - The resulting image represents what the average human eye would see and not a specific sensor.

PBRT

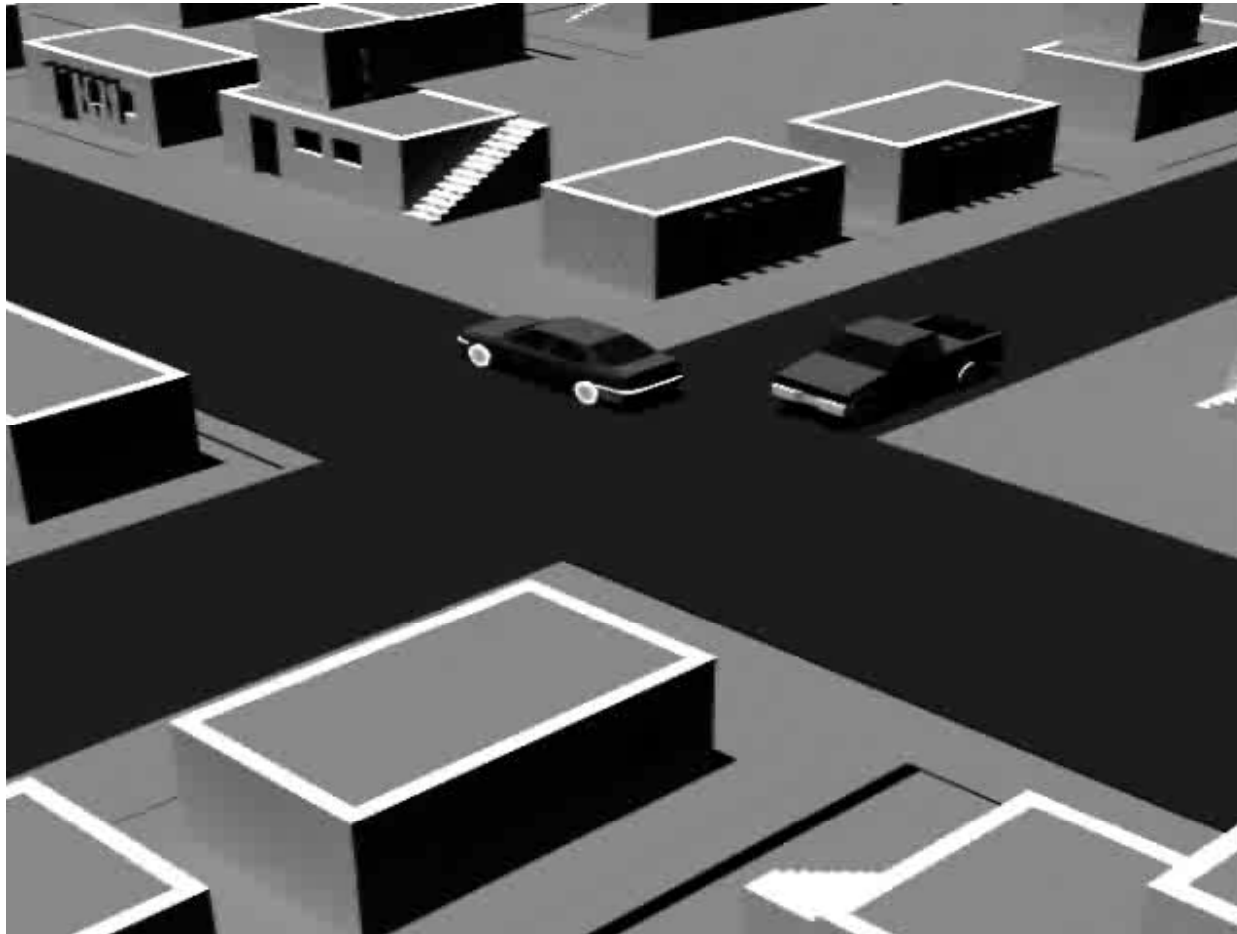
- Physically Based Ray Tracing¹
- Allows rendering of shadows much better than rasterization
- Can model emitting and reflecting objects
- Has a “sun model” that is a projection light source at the proper intensity in the waveband of interest.
- Many ray tracers, including PBRT and LuxRender, run in RGB space and use the CIE 1931 color space to create the proper dynamic range for the final image
- PBRT has been modified to run in only one band
 - The final images are thus the exact radiance values generated by the scene.
- PBRT interfaces with the Scene-Simulation Database via Matlab to retrieve signature information for rendering

PBRT Movie – Truck in Hovel Test Scene



- **Materials are assigned to every facet group in the scene and the definitions as well as spectral reflectance data is stored in the Scene-Simulation Database.**
- **Movie rendered at 400 nm**

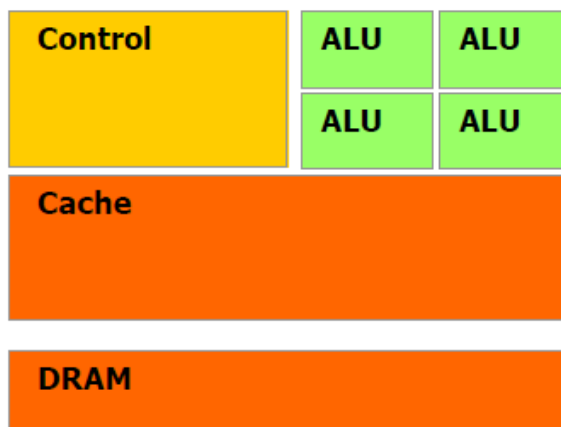
PBRT Movie – Vehicles in Urban Scene



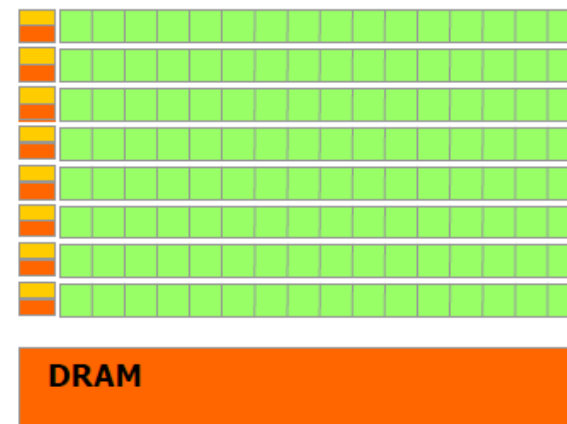
- Movie rendered at 400 nm

GPU Architecture

- GPUs (Graphical Processing Units) have hundreds of low precision ALUs (Algorithmic Logic Units) in comparison to the handful of high precision ALUs found on today's CPUs
- All of the GPU ALUs can operate in parallel inside parallel "Blocks" that form the GPU
- Current GPUs are optimized for floating point calculations
- GPU ALUs have been speeding up and becoming more powerful since NVIDIA's push for a GPGPU (General Purpose GPU) architecture



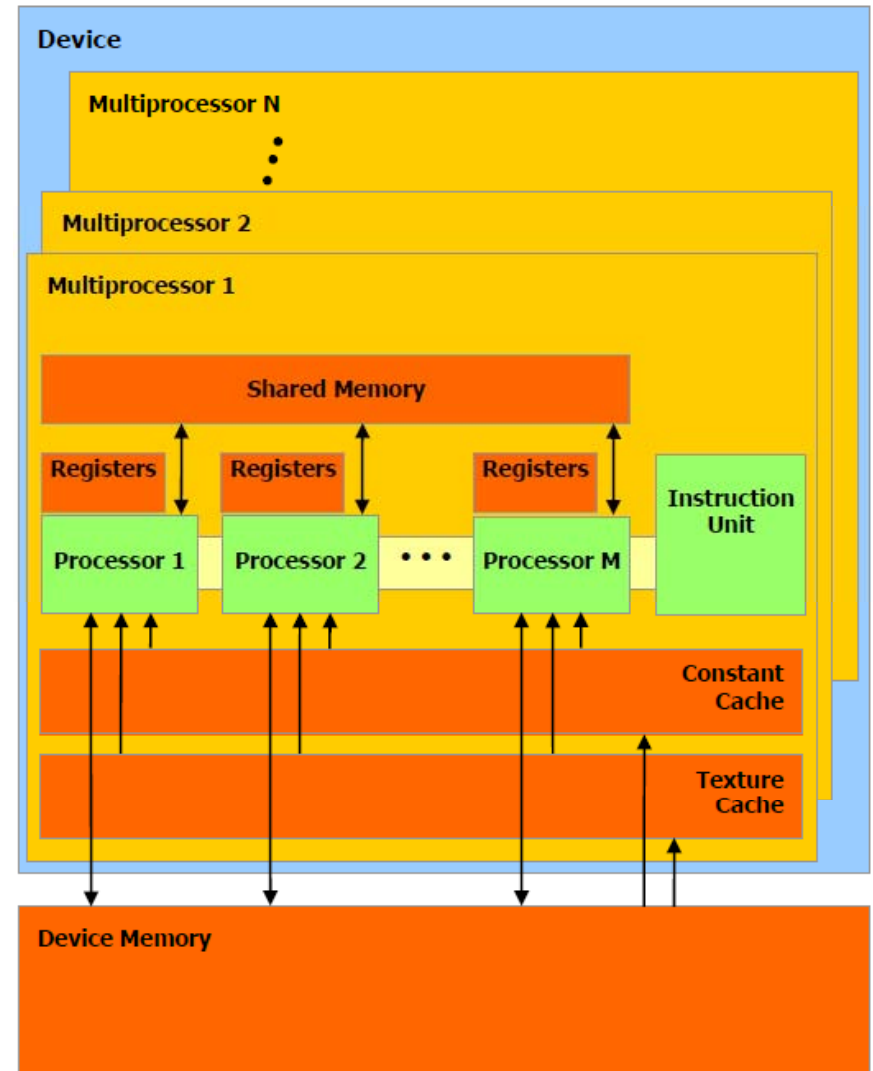
CPU



GPU

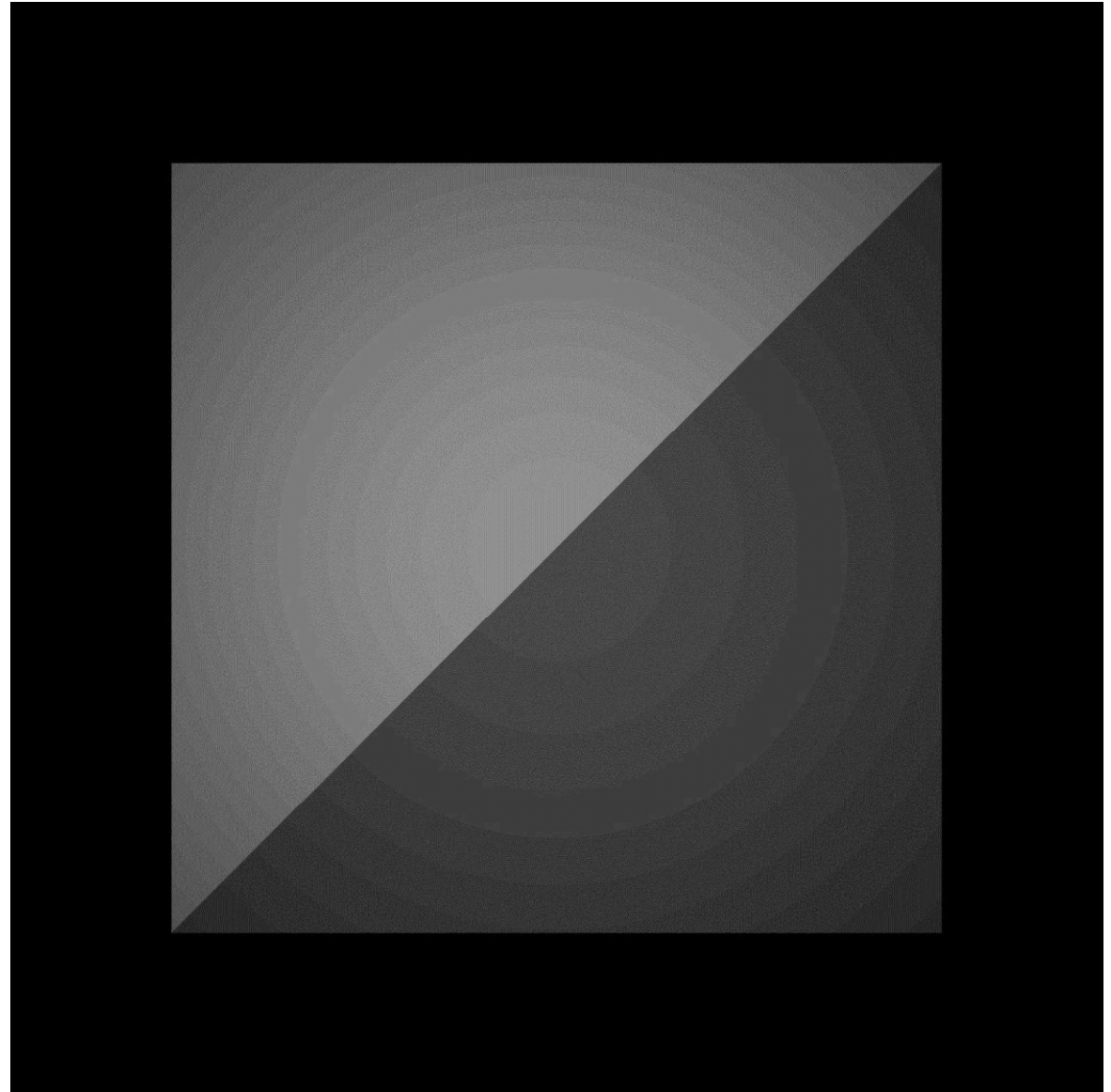
Implementing Code on the GPU

- There can be hundreds of Blocks (multiprocessors) on a GPU with at most 512 Threads (ALUs) per block (depending on the graphics card)
- Each Thread is capable of running in parallel with all other threads and can perform floating point calculations
- Depending on how memory is allocated, Threads and Blocks can cooperate to finish the same task
- Threads can also be chained together to form a processing “pipeline” to perform complex tasks



GRT: First Test

- Looking at two triangles with intensities of 2 W/m^2 and 1 W/m^2
- Rendering Time $< 1\text{s}$
- Rings due to floating point error – can be removed by changing code from single to double precision floating point data



Multimodal Data Collection

Edward Burdette

Electro-Optical Systems Laboratory

Georgia Tech Research Institute

September 2010

edward.burdette@gtri.gatech.edu

404-407-6037

Motivation

- **Demonstrate utility of mobile camera arrays**
 - Array cameras are small, lightweight, and inexpensive
 - Multi-band and multi-directional environmental monitoring
 - Data transmitted to a central collection unit for processing
- **Support multimodal signature modeling**
 - Motion detection algorithms
 - Simultaneous inter-band data verification
 - Validate scene models



Background

- **Inexpensive, high-resolution camera prevalence suggests usefulness of arrays**
 - 8 Megapixel cell phone camera resolution
 - Cell phone battery life of ~10 hours
 - Covert imaging device supported
- **Desired collection system capabilities include**
 - Mobility
 - Compatible with multiple sensors
 - Tunable data input rate and storage
 - Permit real-time recording views

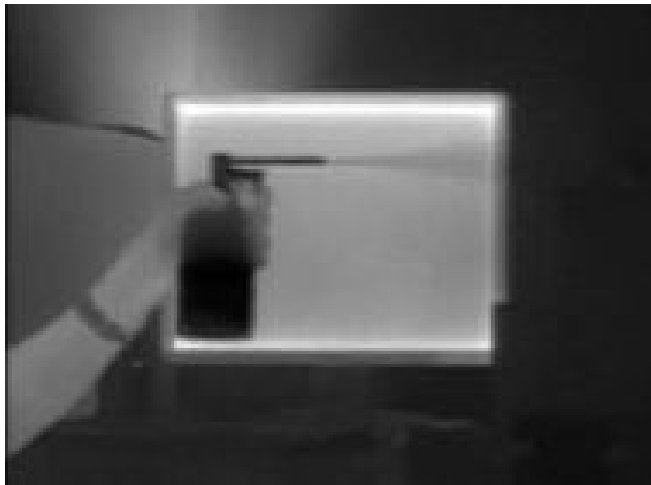


Camera Info (1 of 2)

FLIR PathFindIR



- 8 – 14 μm
- 320 x 240 pixel resolution
- Uncooled microbolometer
- 0.8 lb weight



Goodrich SWIR

- 0.9 – 1.7 μm
- InGaAs CCD
- 320 x 256 pixel resolution
- Size < 9.5 in³



Camera Info (2 of 2)

Visual Camera



- 0.4 – 0.7 μm
- 320 x 240 pixel resolution
- \$40



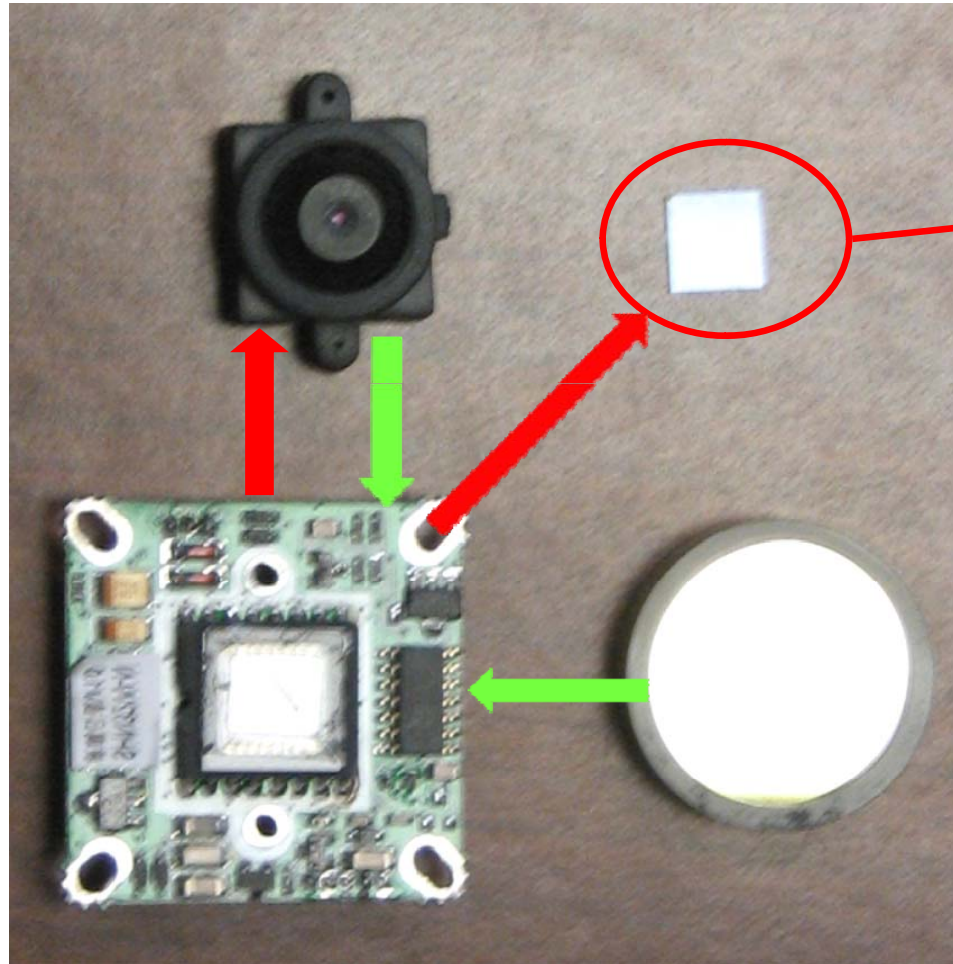
NIR Camera

- 0.75 – 1.05 μm
- 320 x 240 pixel resolution
- \$110 (includes longpass filter)

Constructing the NIR camera

1) Begin with typical VIS camera

2) Remove camera lens



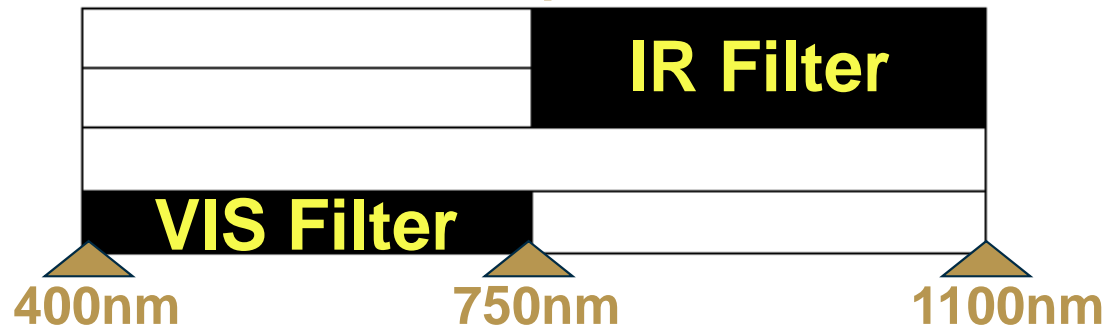
3) Remove IR filter

4) Reinstall lens and cover with VIS filter

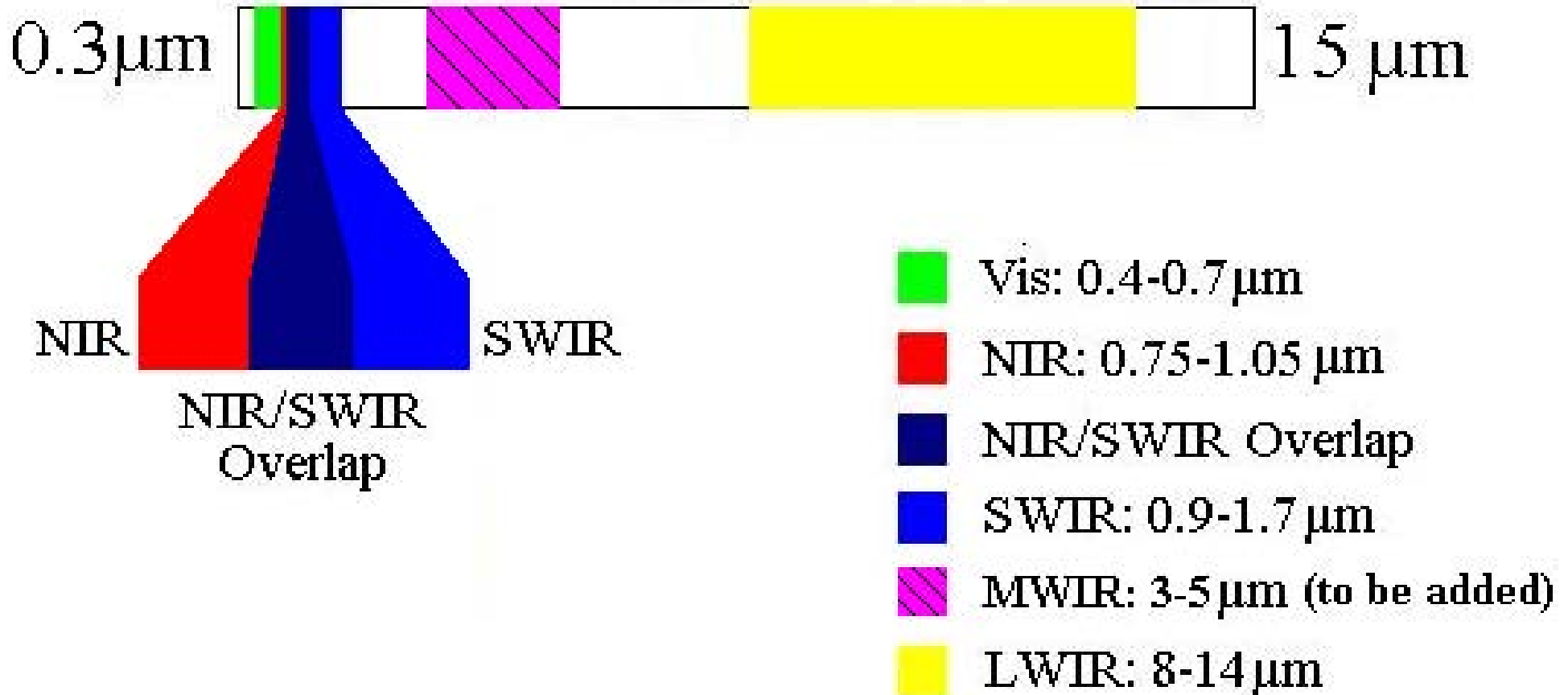
Step

1
2
3
4

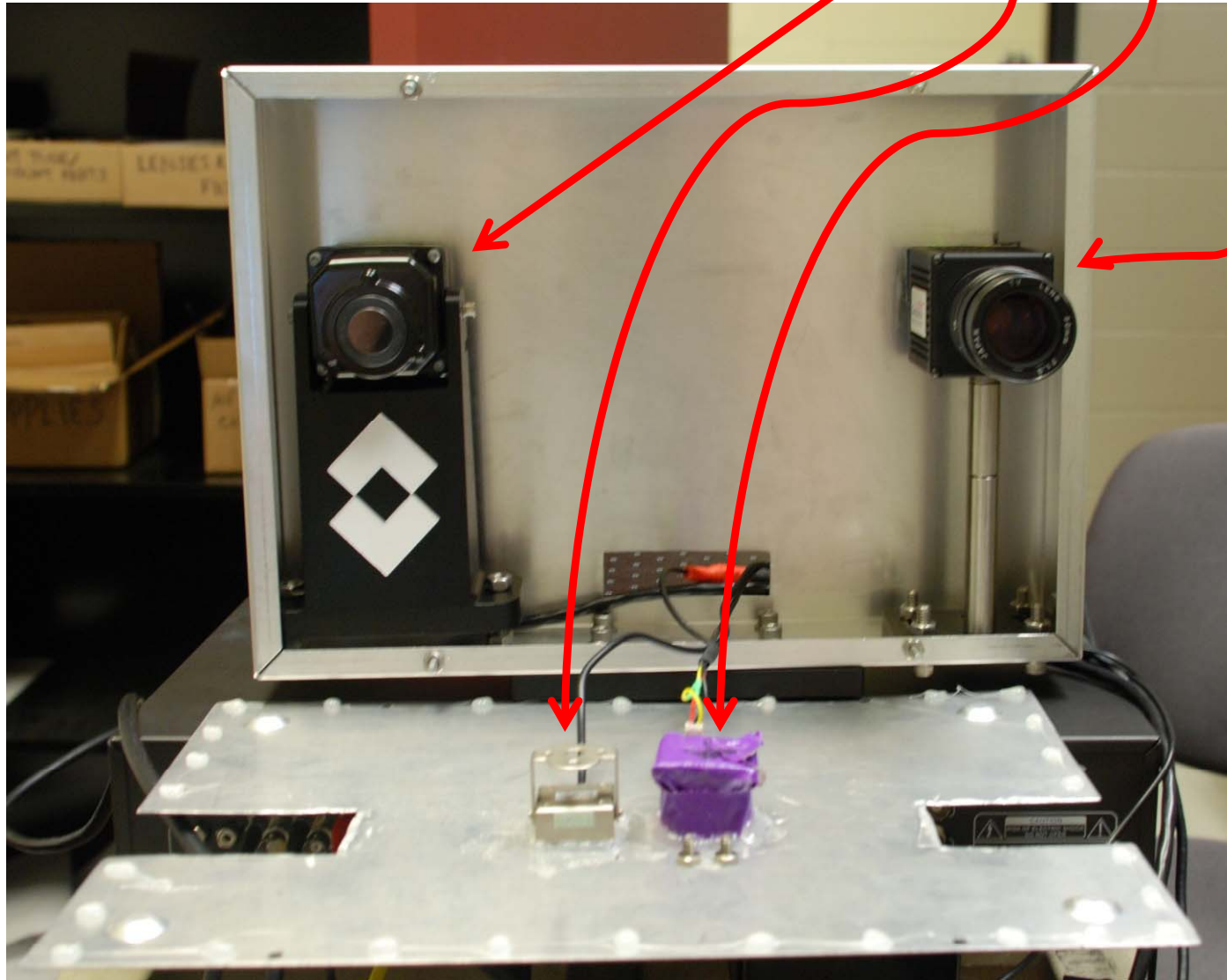
Sensor Spectrum



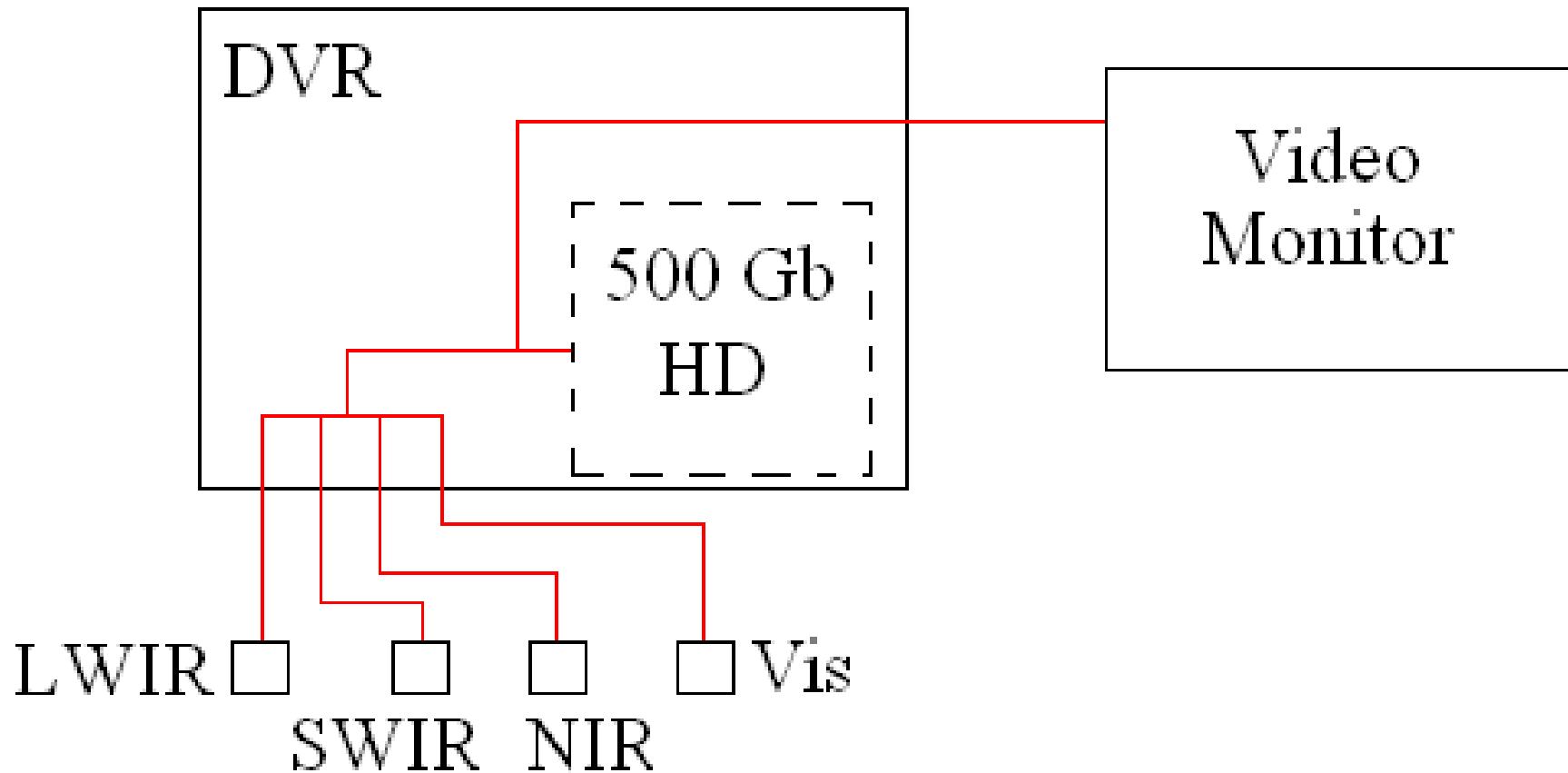
Spectral Coverage



Camera Arrangement (L to R): LWIR, Vis, NIR, SWIR



System Connection Diagram



Digital Video Recorder

Compatible with any composite video sensor

1.5 Terabytes of data stored on 3 internal hard drives

Up to 16 video and 4 audio input channels

D1 (704x480) resolution and 480 fps divided over the 16 inputs



DVR Front



DVR Back

Data Collection Setup

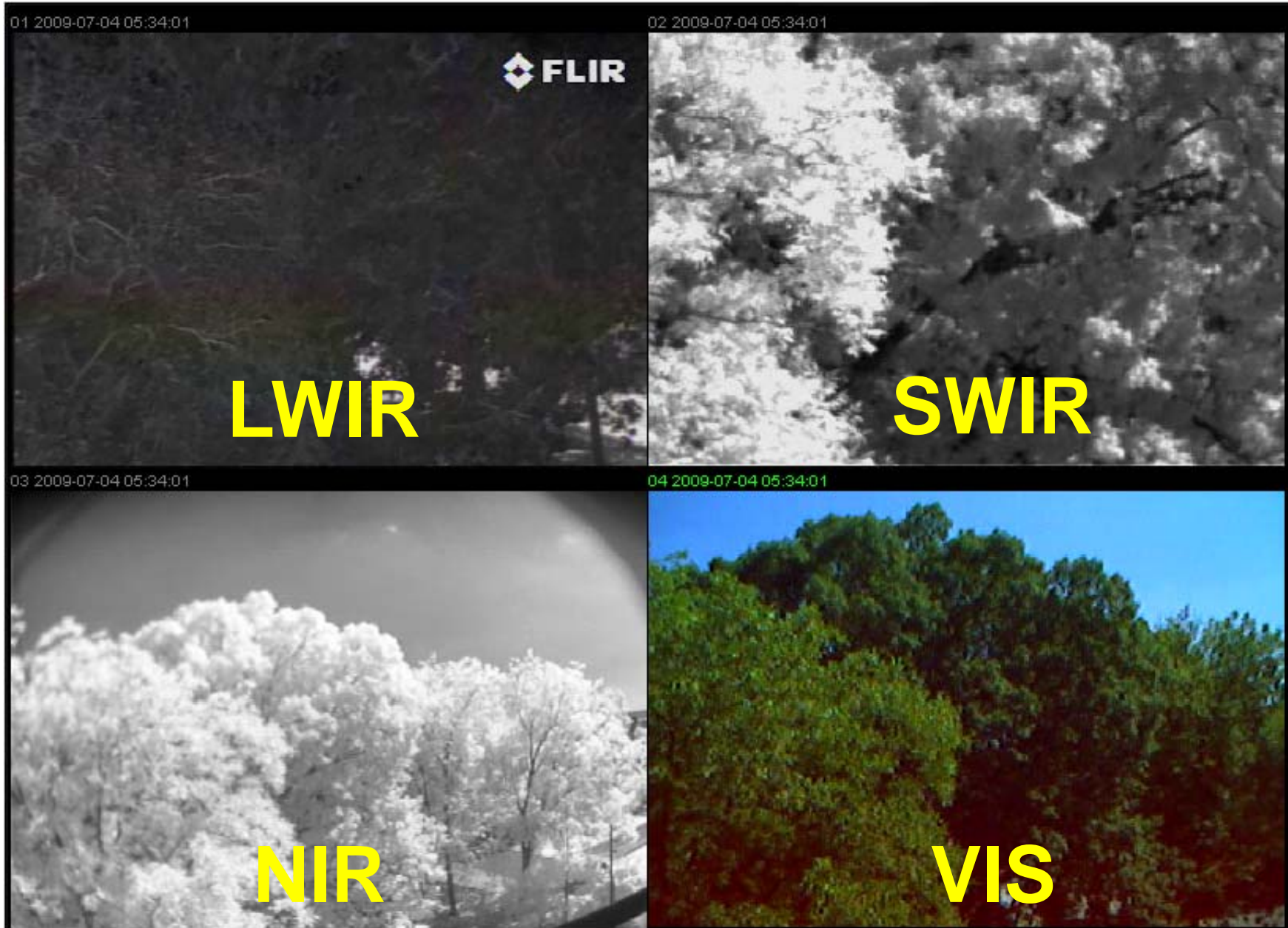
- Monitor/laptop port permits active view of camera data
- Tripod mount gives stability while enabling 360° views
- DVR processes up to 1 Gigabit of data per second
- Setup can accommodate more DVR units and cameras



Passenger Vehicle Imagery



Foliage Comparison



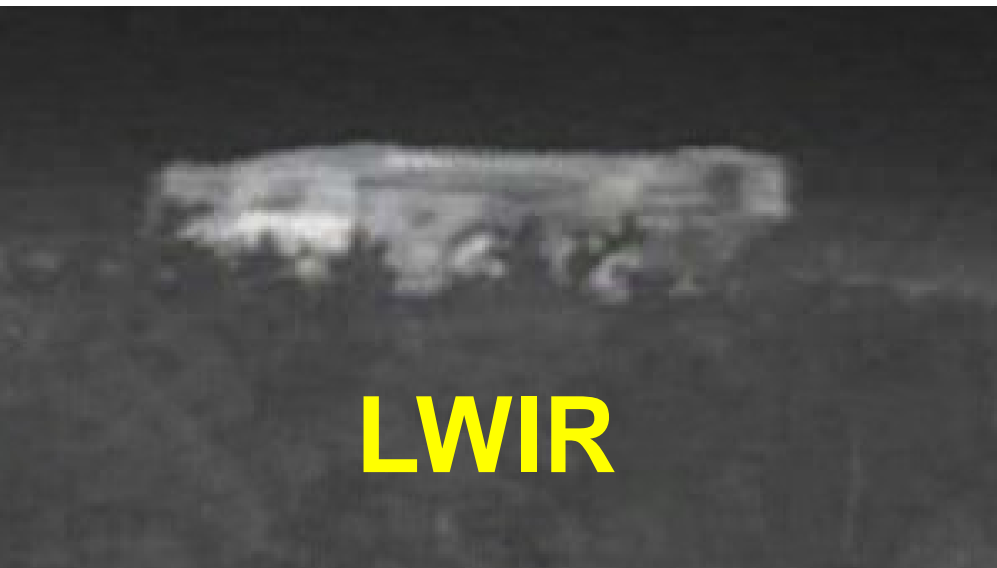
Traffic Video



Disturbed Earth after Watering



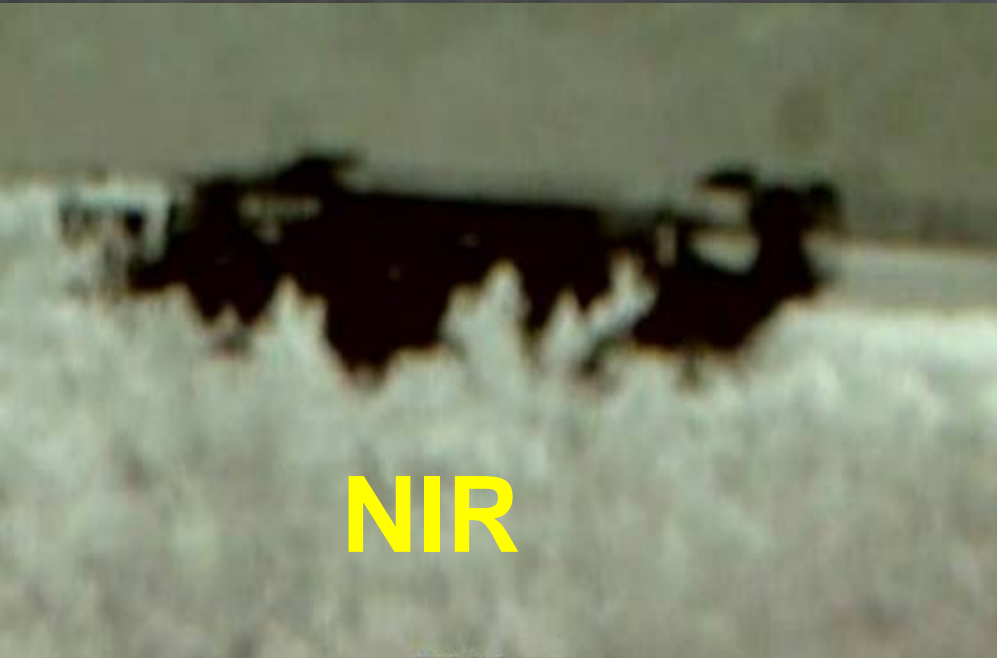
Camouflaged Military Vehicle



LWIR



SWIR

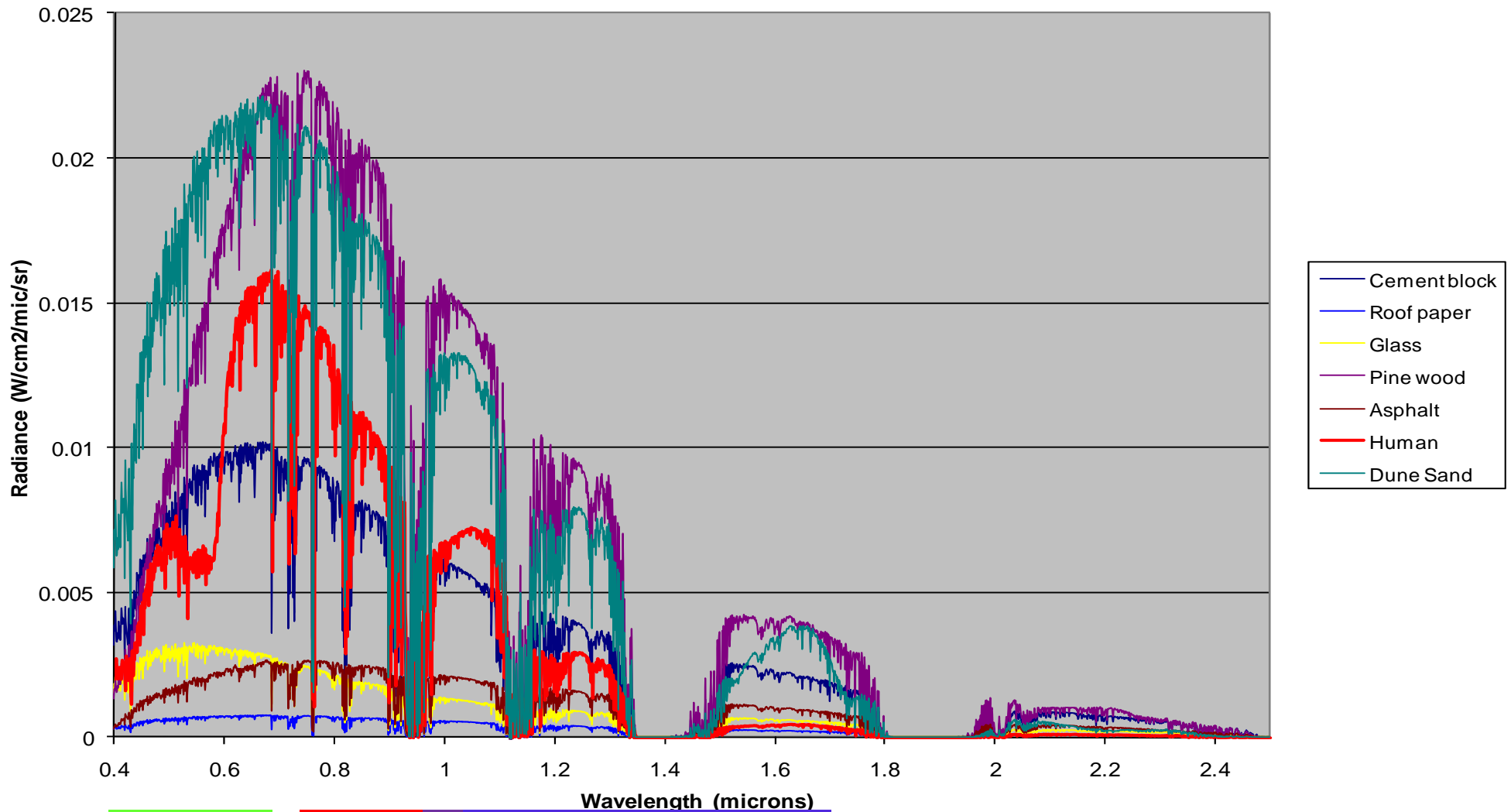


NIR



VIS

Sensor system + bandpass filters = material characterization



Accomplishments

- **Built material discrimination system foundation**
- **Recorded 704x480 resolution video at 30 fps**
- **Covered LWIR, SWIR, NIR, and Vis spectral bands**



Accomplishments (Cont.)

- Cheaply converted VIS cameras to NIR
- Demonstrated utility of portable MS sensor network
- Developed a 1st generation mobile sensor web



Motion Analysis of Video to Support Personnel Detection

Alan M. Thomas

Electro-Optical Systems Laboratory

Georgia Tech Research Institute

alan.thomas@gtri.gatech.edu

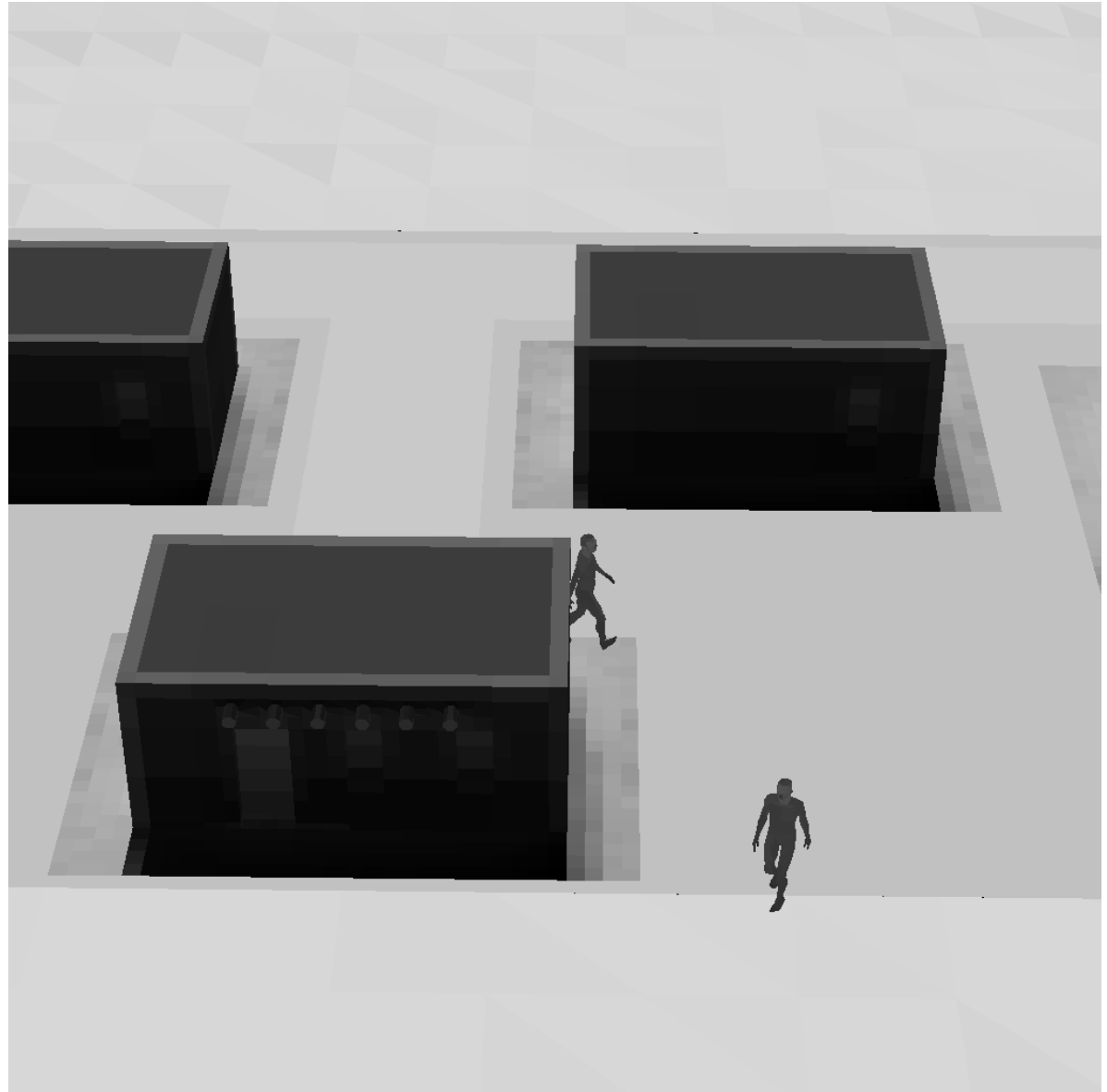
404-407-8223

Motivation

- **The ability to associate data related to a single target is a preliminary step to performing multi-sensor data fusion.**
- **We desire a method for associating data from multiple EOIR sensors as well as RF micro-Doppler signals, and ultra-sonic signals.**
- **The analysis of motion/gait may provide the mutual information necessary for performing the data association.**
- **We are also interested in exploring the possibility of performing identification based upon gait as observed in imagery.**
- **Furthermore, the ability to point out anomalous or hostile behavior is desirable.**

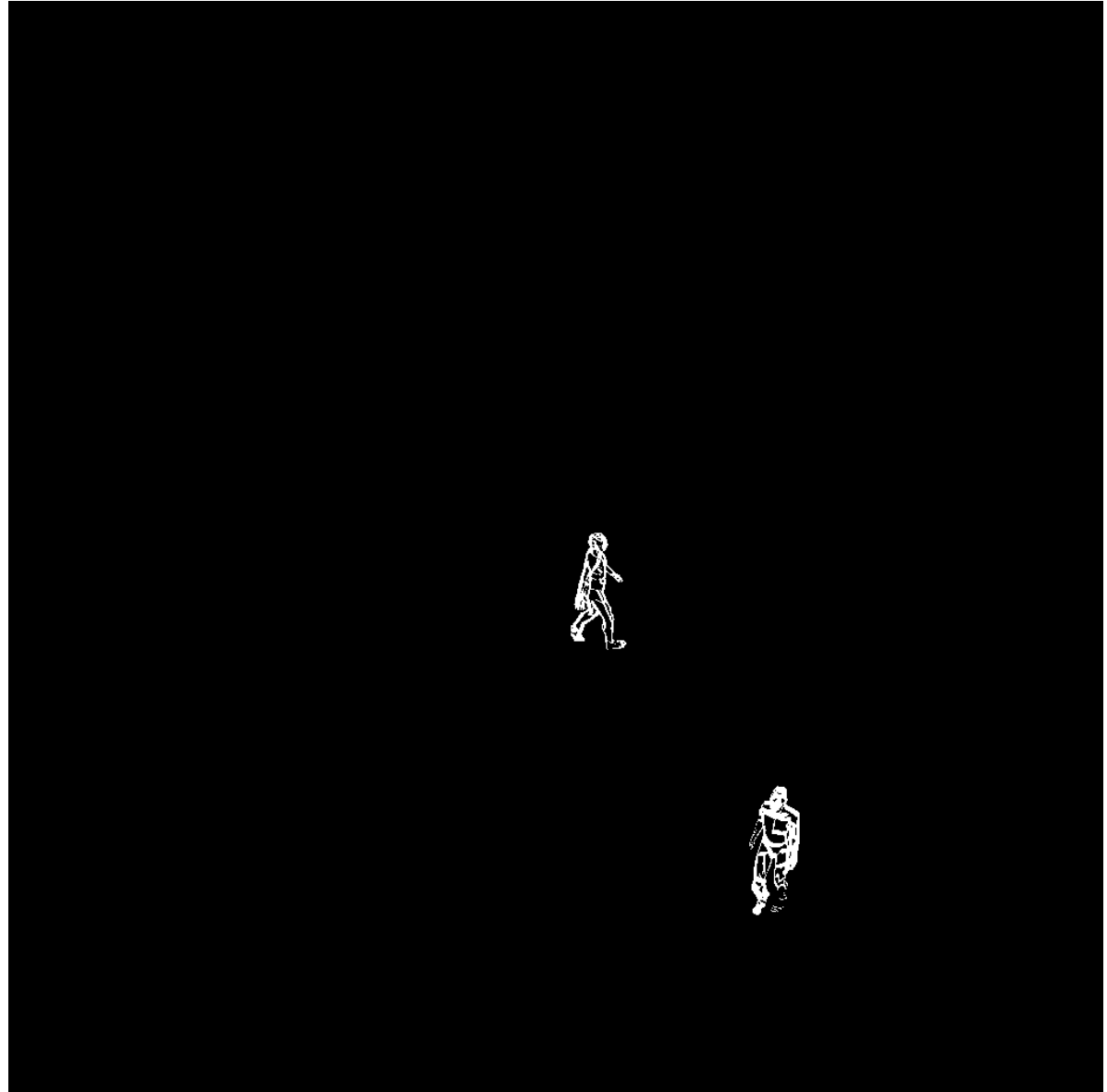
Change Detection

In developing our motion analysis methods the following video sequence was used as a test case.



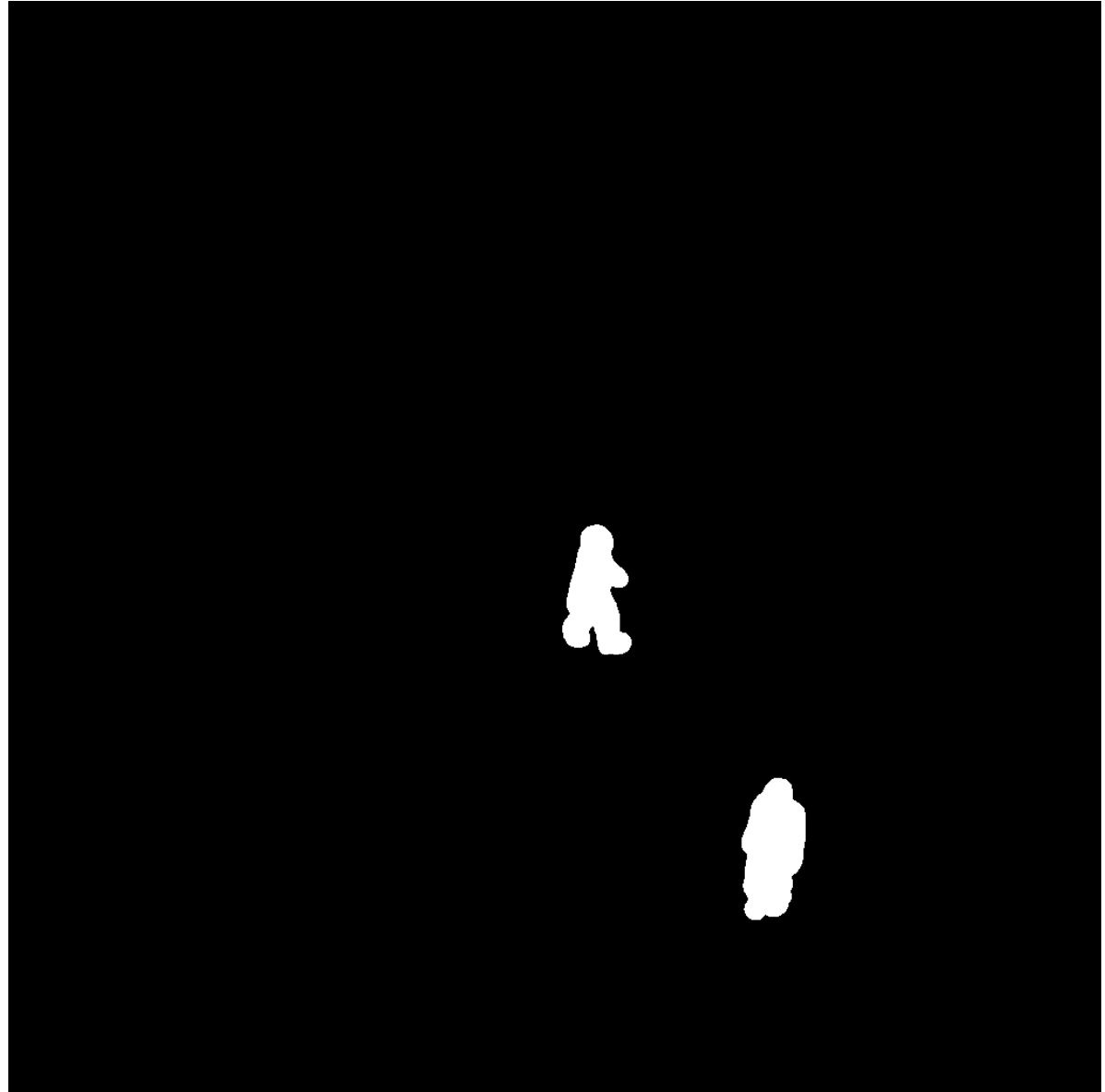
Change Detection

The pixel-wise difference between consecutive frames was used as the basis for detecting motion in video.



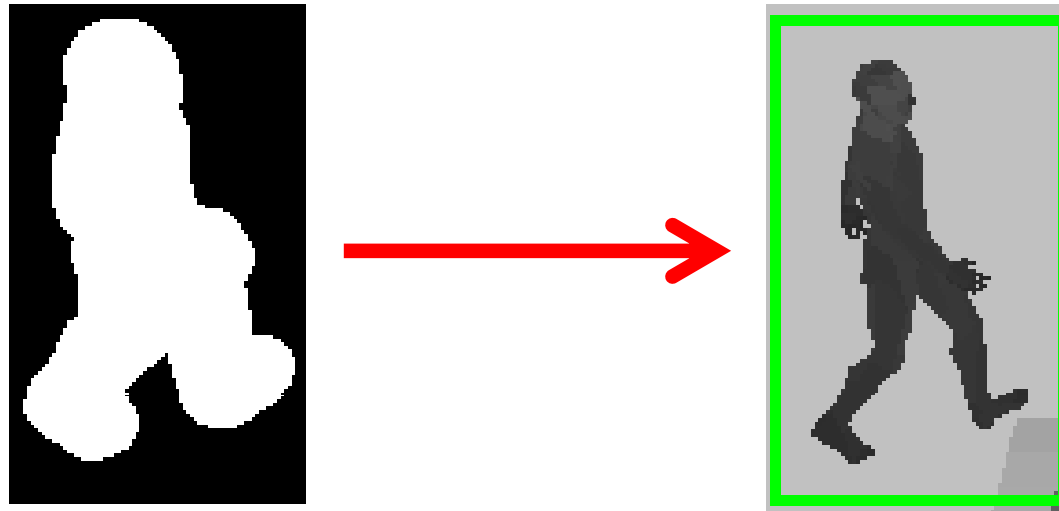
Change Detection

The resulting images are then blurred through convolution with a Gaussian filter and a threshold is applied ($1.3 * \text{mean}$) to yield an image with binary values.



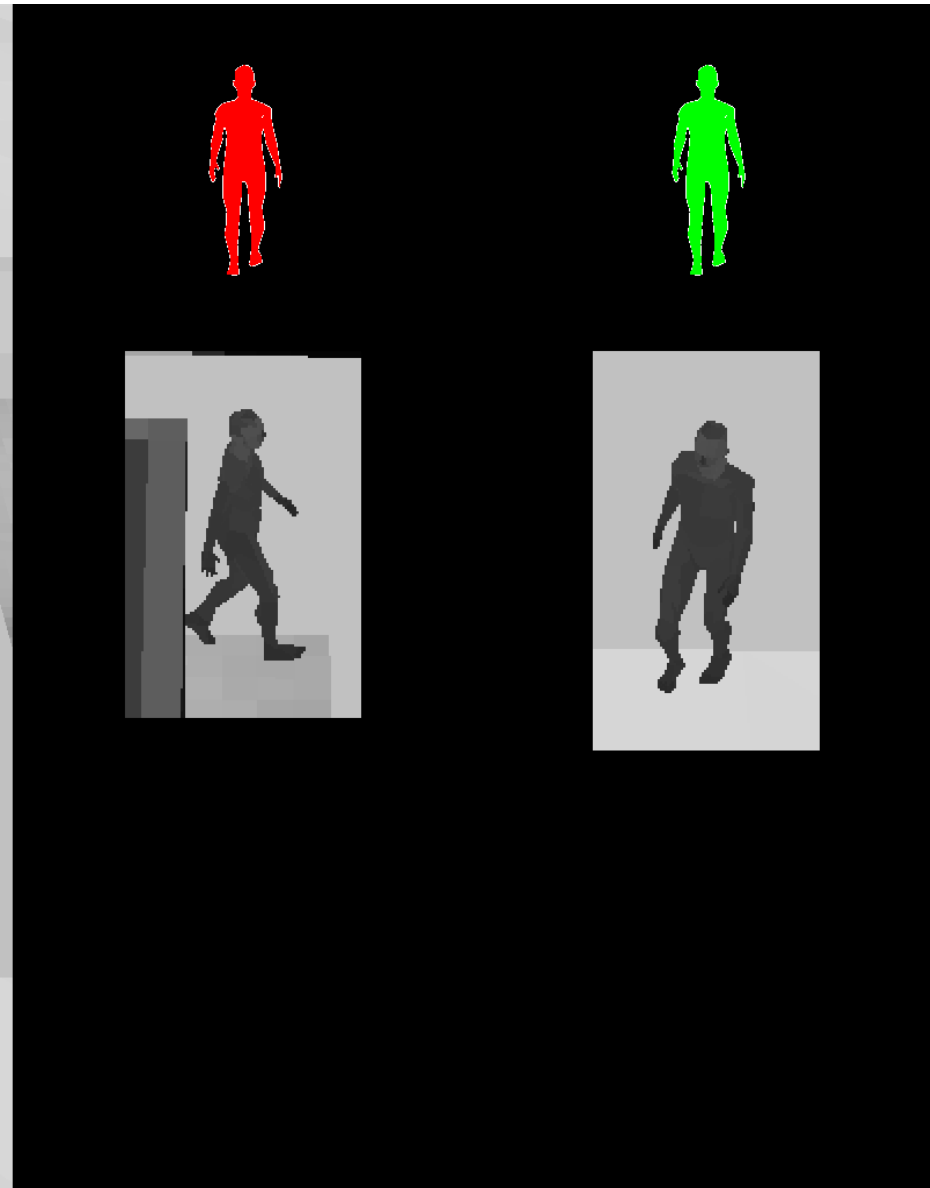
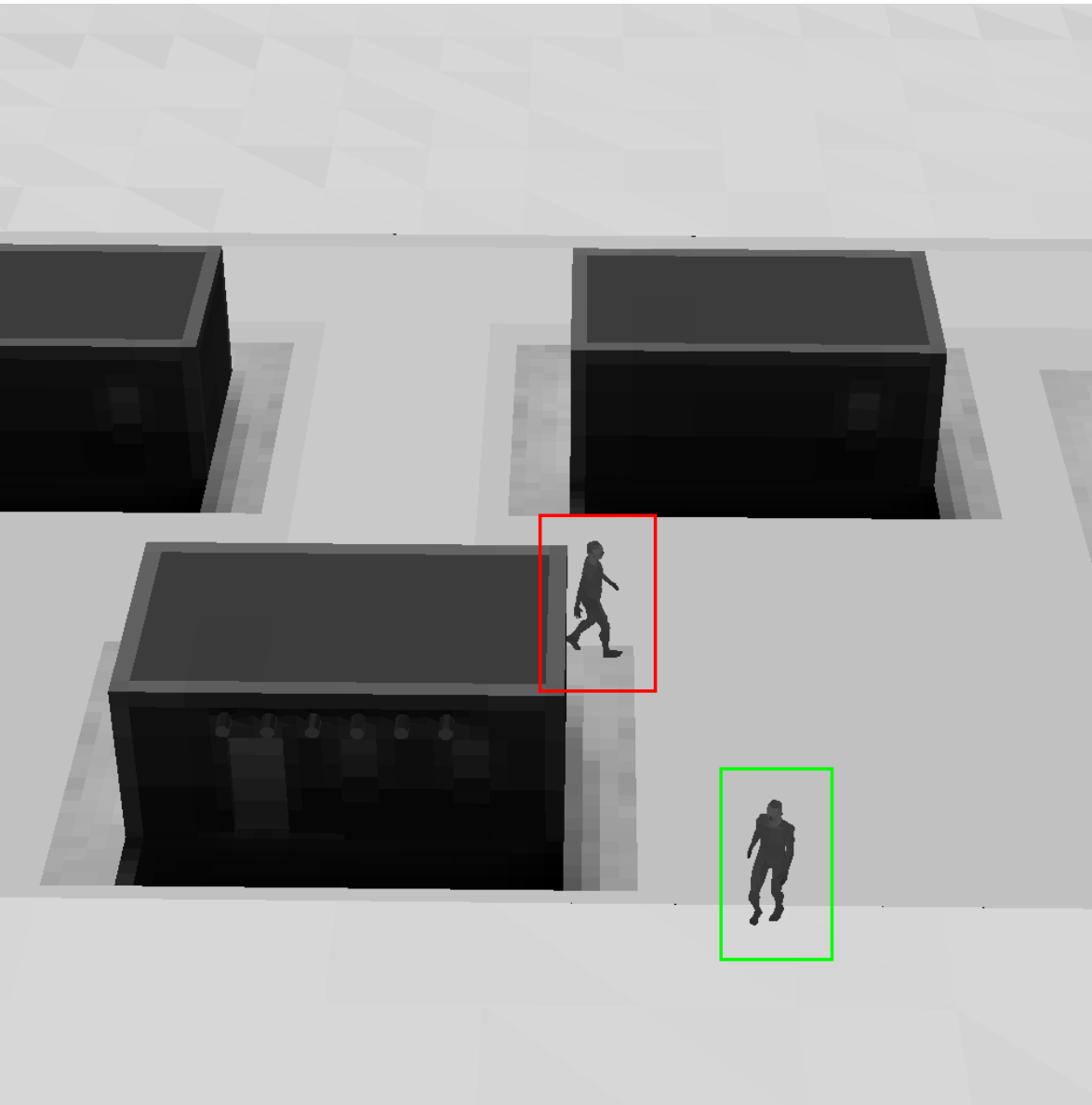
Track Formation

- Need a means for associating targets over time
- We begin by giving a target window to each blob.



- In the next frame, if the centroid of a blob falls within the target window of a target in the previous frame, then the two are associated together and a track is begun. The process is repeated iteratively.

Track Formation



Motion and Gait Analysis – Discrimination

- **The motion of moving objects has the potential to be used as a characteristic in discriminating between humans and other moving entities**
- **As a simple example, the speed of most vehicles is outside the bounds of realistic human movement.**
- **A more detailed look at motion in video may yield potential methods for discriminating humans from animals, other vehicles, and the natural movement of vegetation.**

Motion and Gait Analysis - Fusion

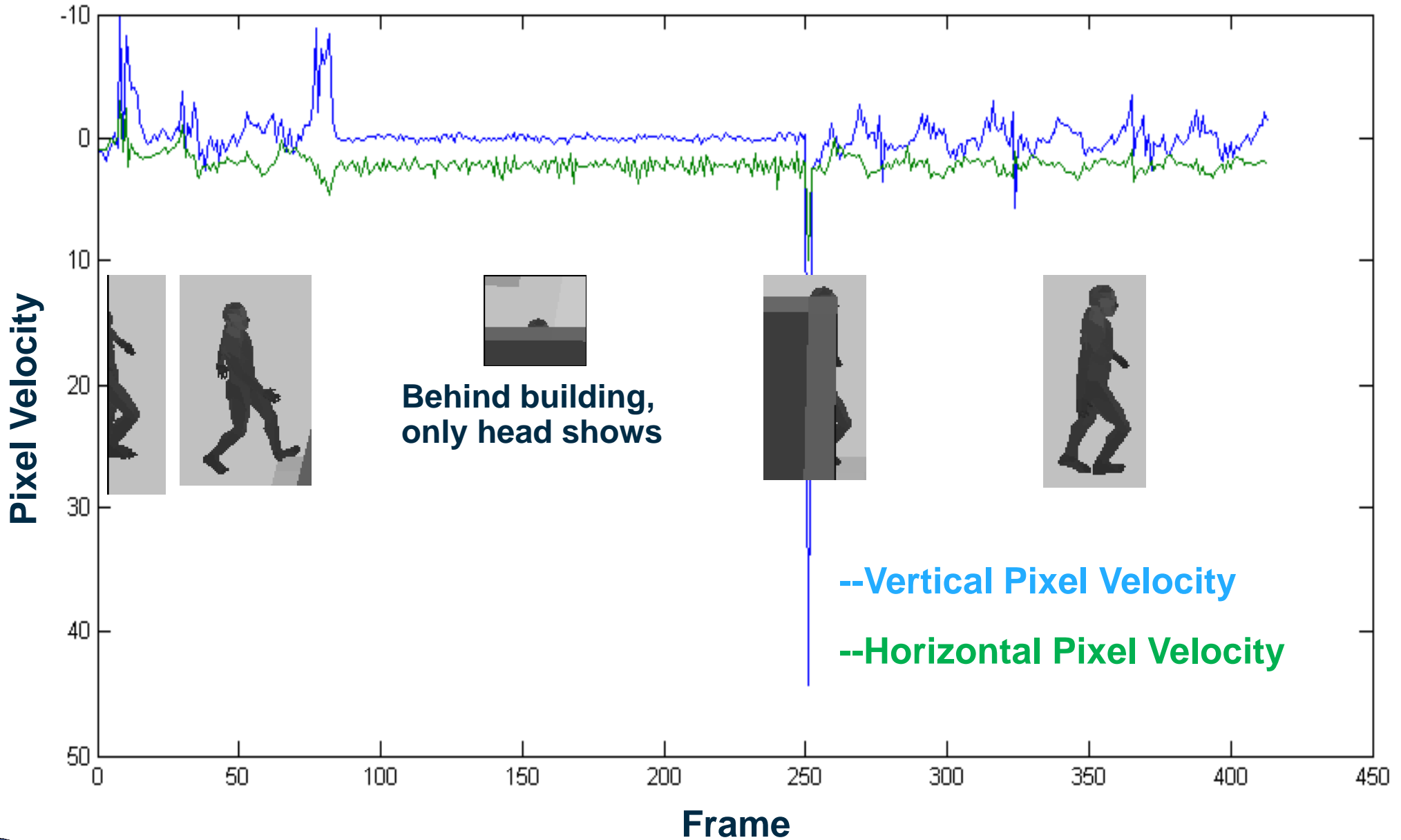
- If we can fuse from different sensors (and sensor modalities), then more information can be brought to bare on what is and is not human.
- The ability to associate data related to a single target is a preliminary step to performing multi-sensor data fusion.
- We desire a method for associating data from multiple EOIR sensors as well as RF micro-Doppler signals, and ultra-sonic signals.
- The analysis of motion/gait may provide the mutual information necessary for performing the data association.

Motion and Gait Analysis

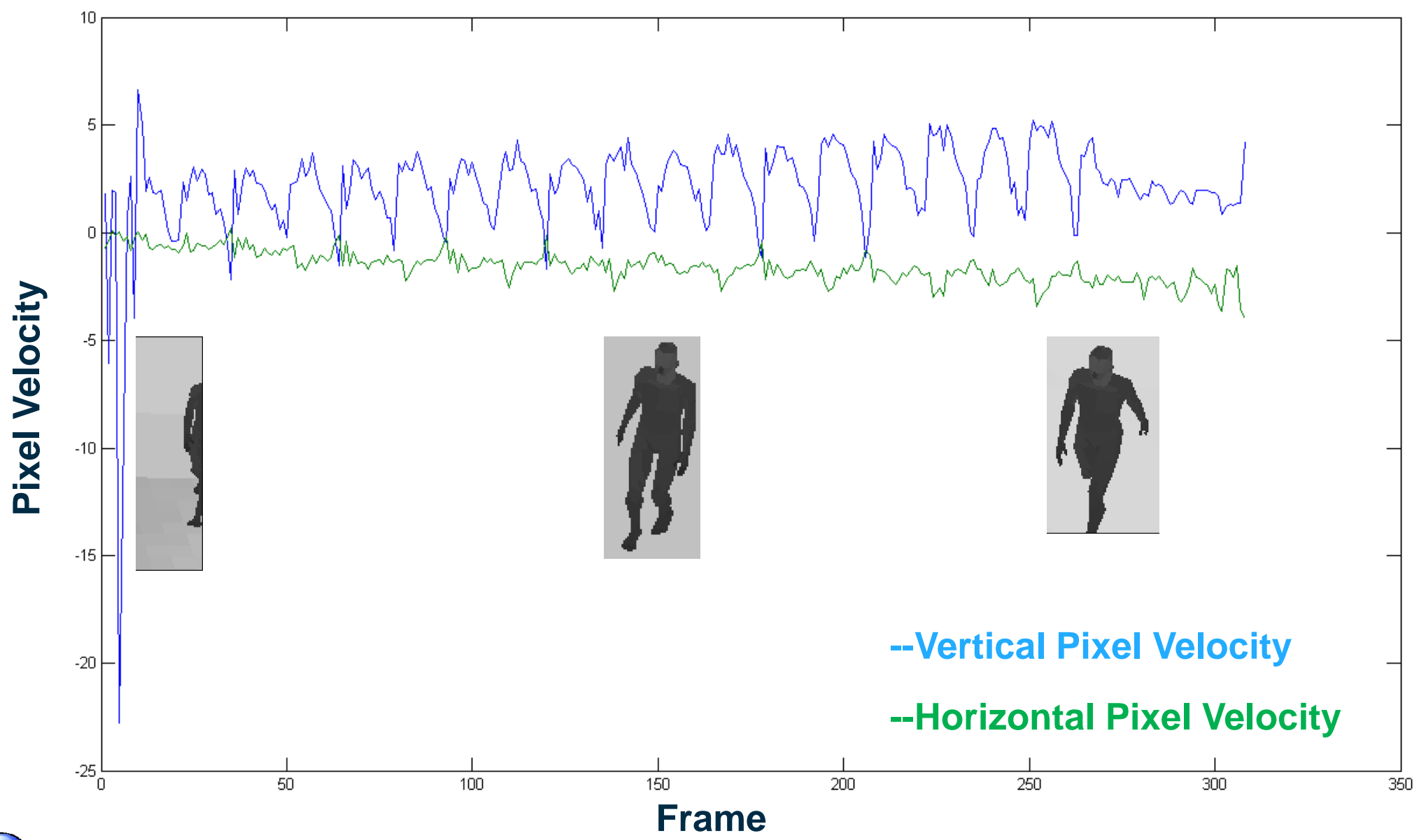
- For each frame a “pixel velocity” is formed by calculating the centroids of the respective blobs and taking their differences in consecutive images
- For a set of blob centroids $\{(x_j, y_j)\}$ the pixel velocity in frame j is calculated as $V_j = (x_j - x_{j-1}, y_j - y_{j-1})$.
- A history of such pixel velocities may then be considered as a signal over time.



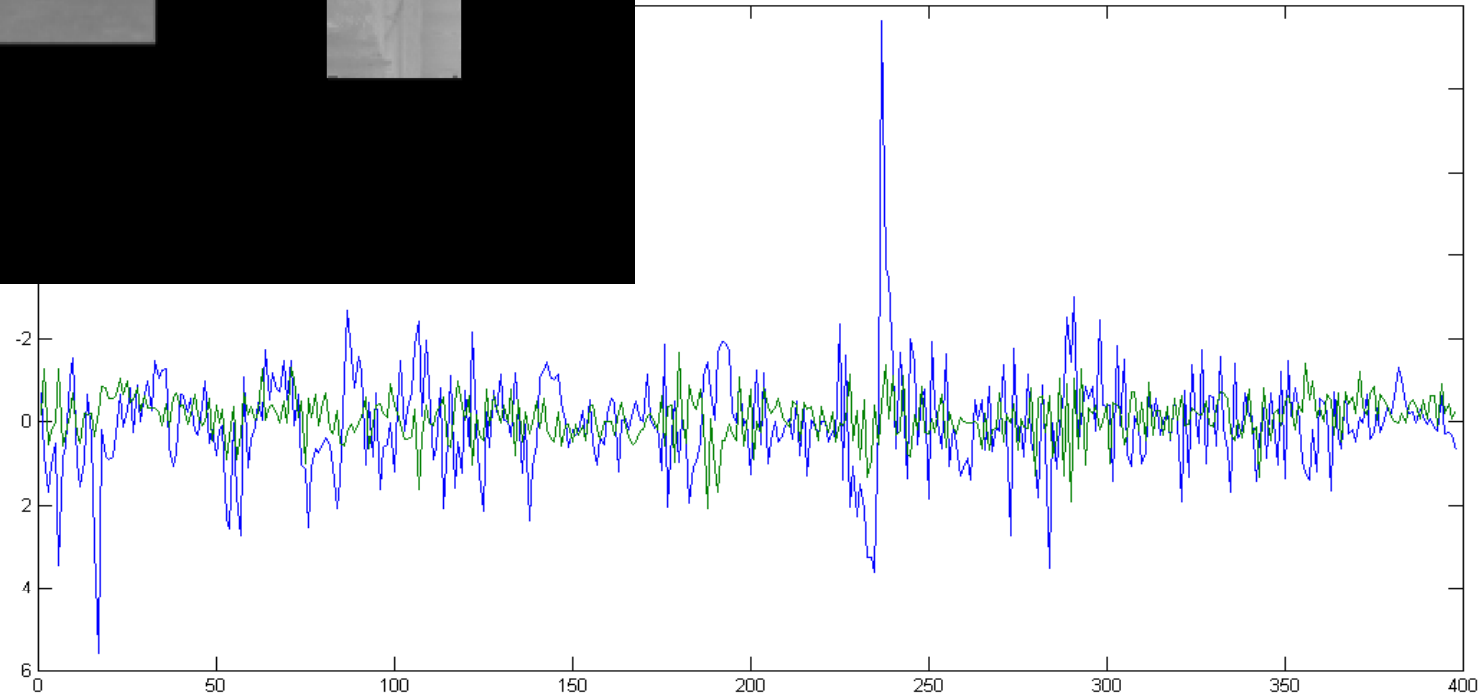
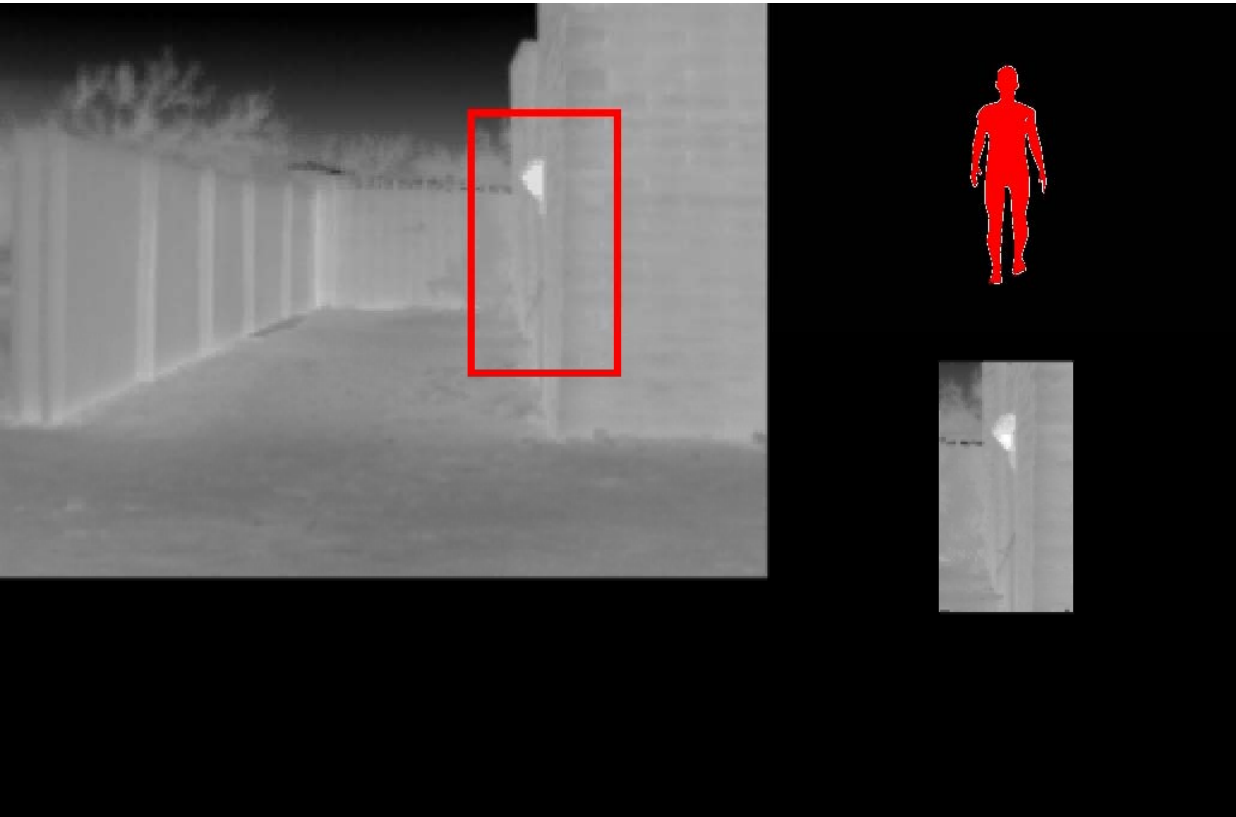
Motion and Gait Analysis



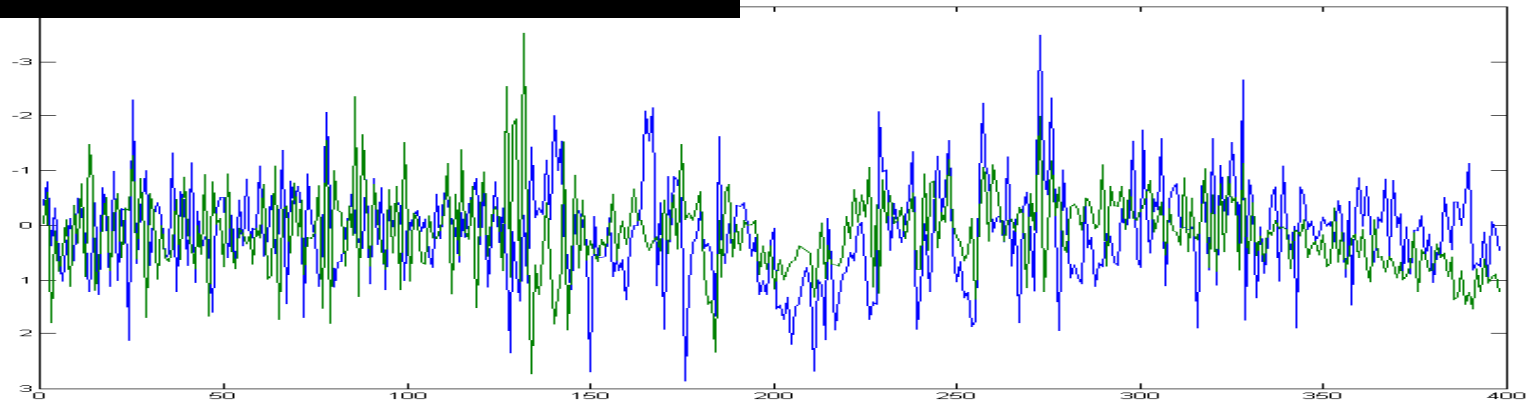
Motion and Gait Analysis



Motion and Gait Analysis



Motion and Gait Analysis

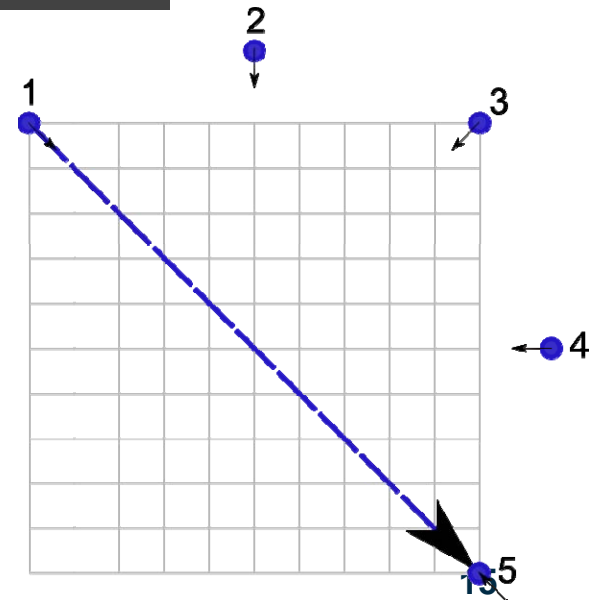


Different Viewpoints

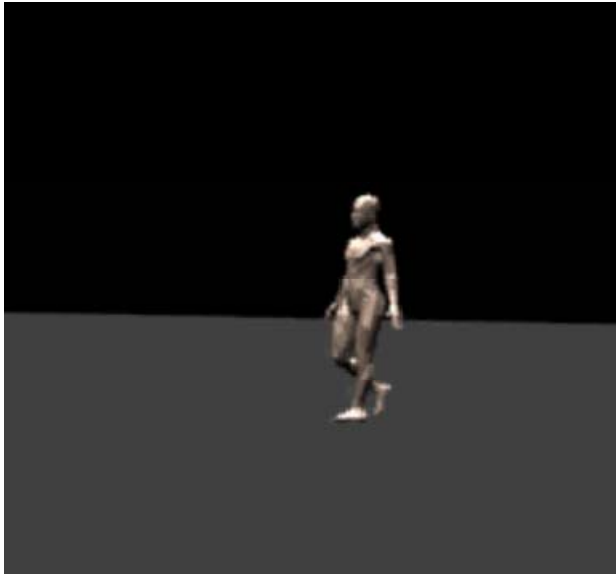


- In order to explore the effects on perceived gait due to changes in perspective, we will consider the same motion from different viewpoints

- Five different camera positions were considered relative to a simulated linear path.



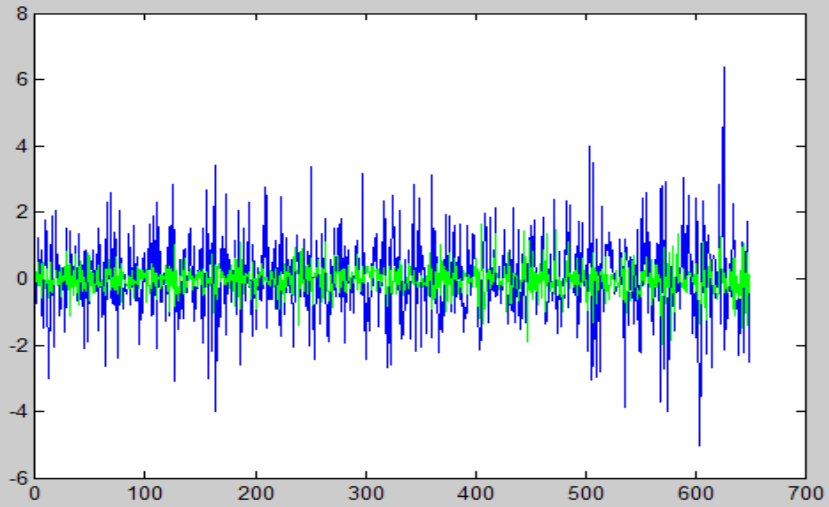
Different Parts



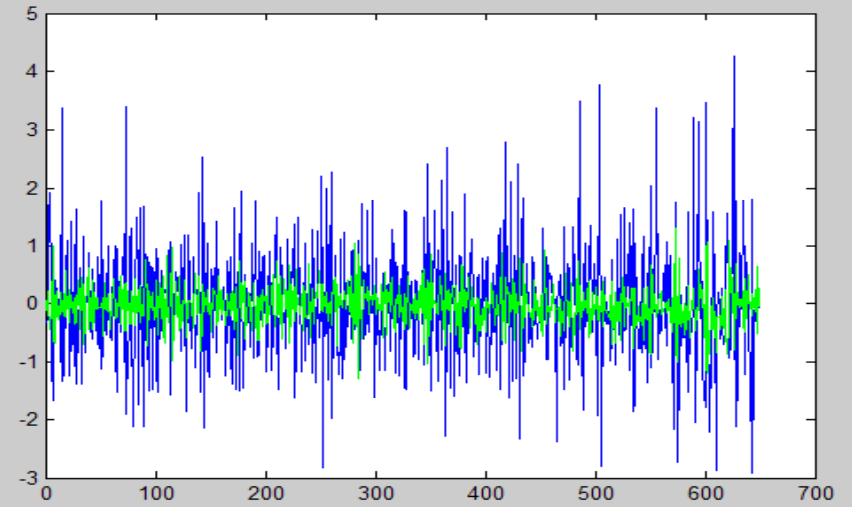
- The perceived motion of the arms legs and torso were considered in addition to the full body motion.

0 Degrees

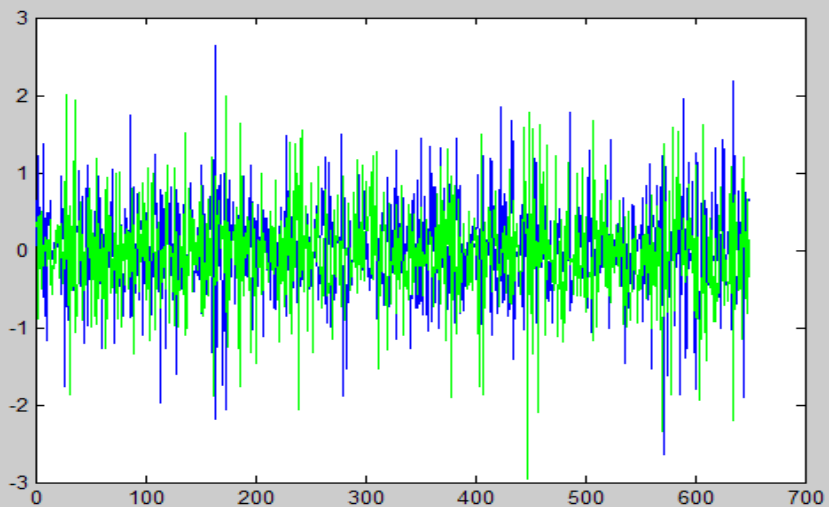
•Full Body



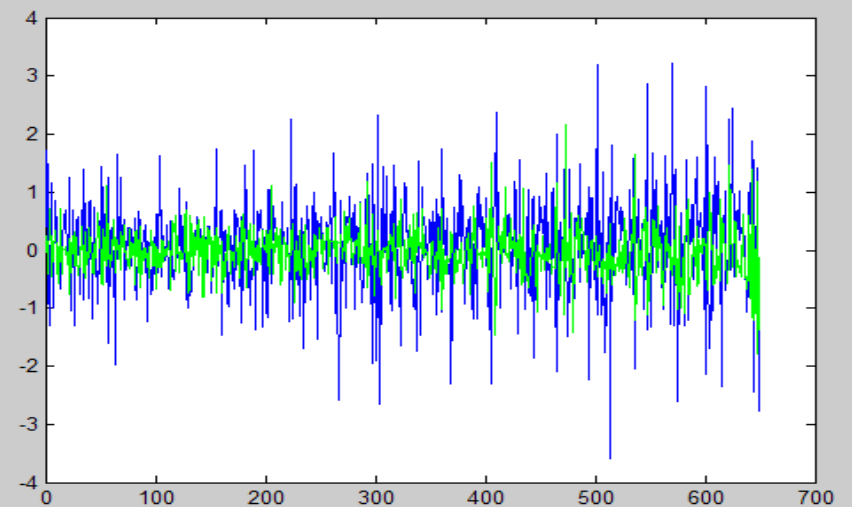
•Torso



•Arms

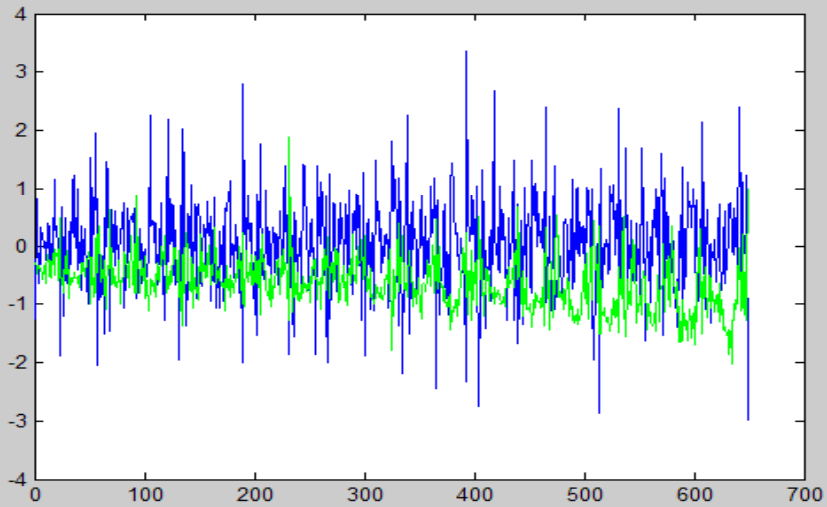


•Legs

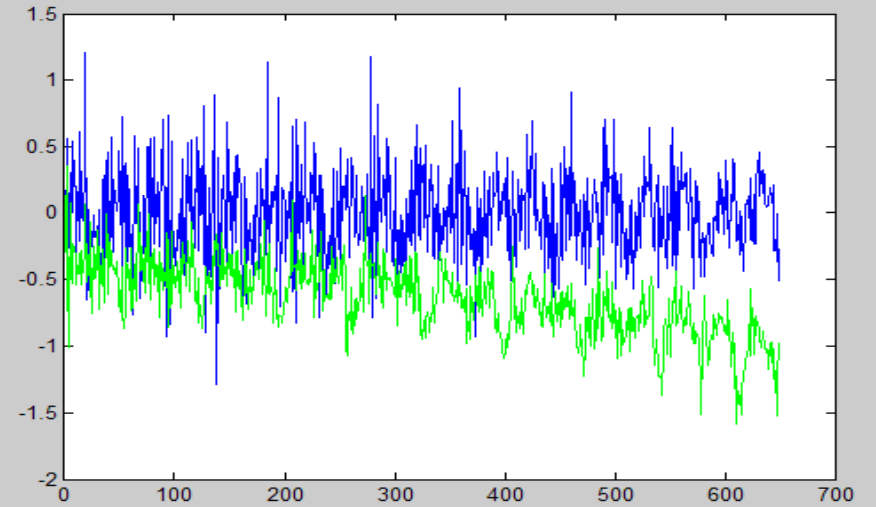


45 Degrees

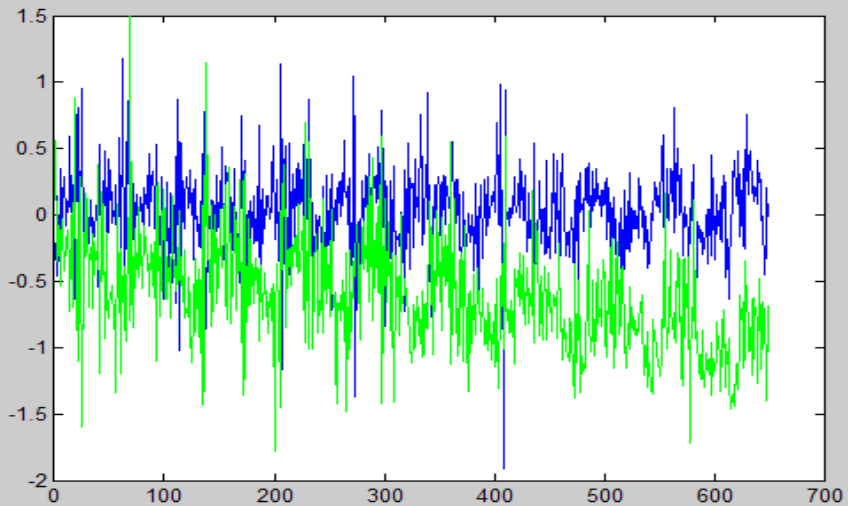
•Full Body



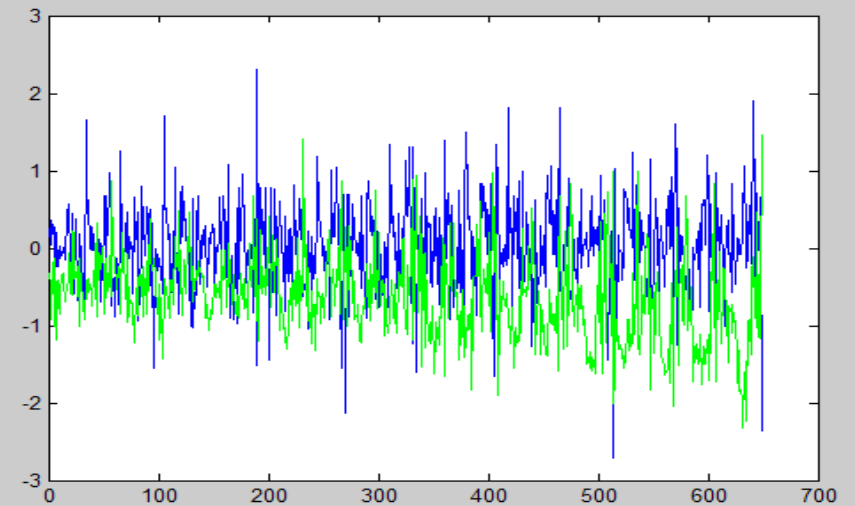
•Torso



•Arms

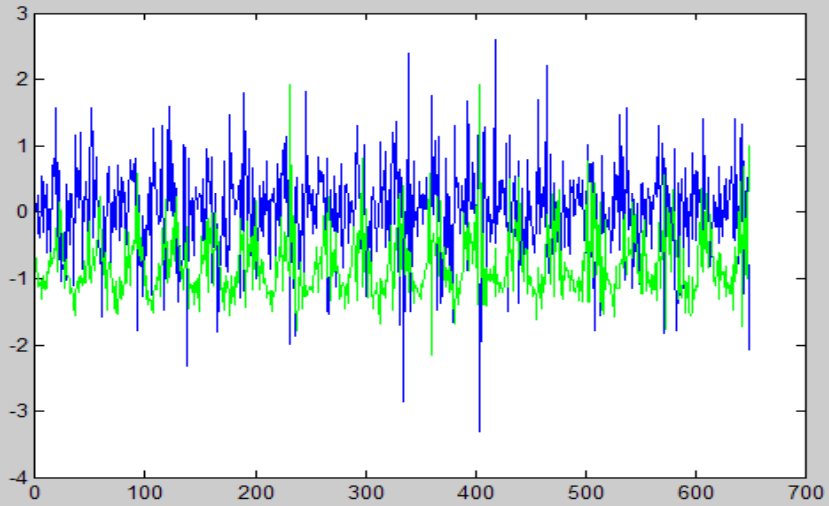


•Legs

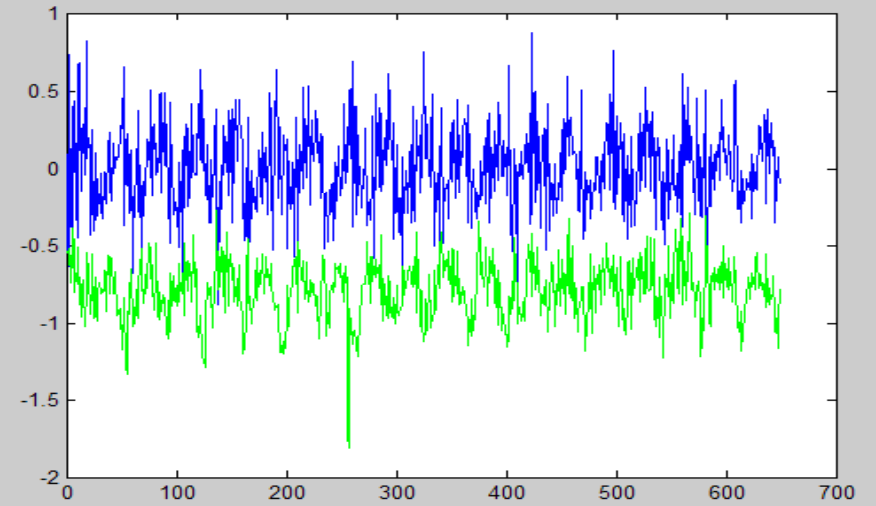


90 Degrees

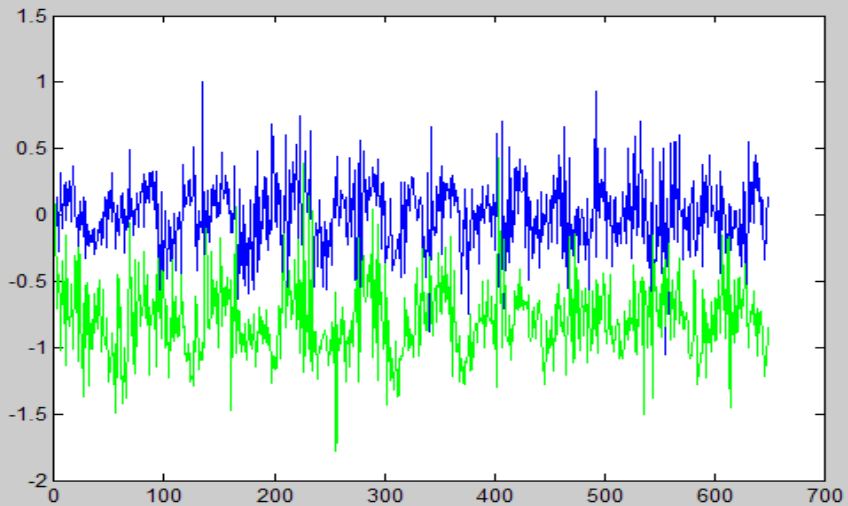
•Full Body



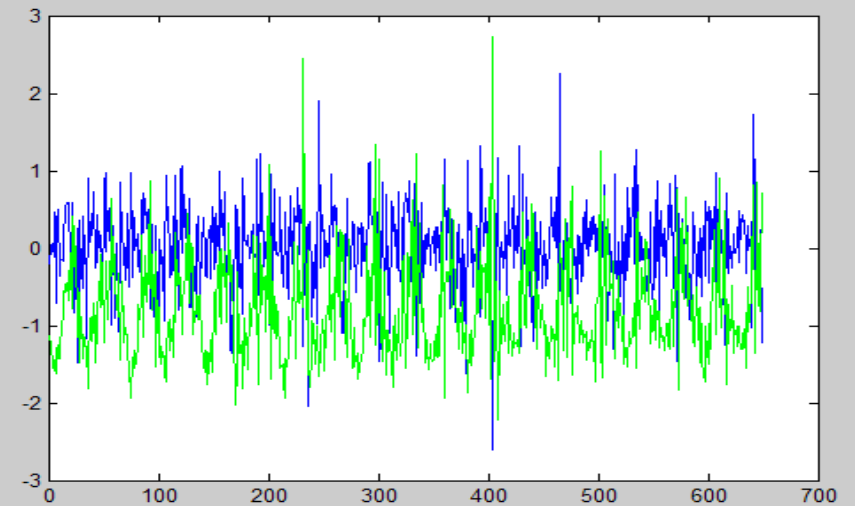
•Torso



•Arms

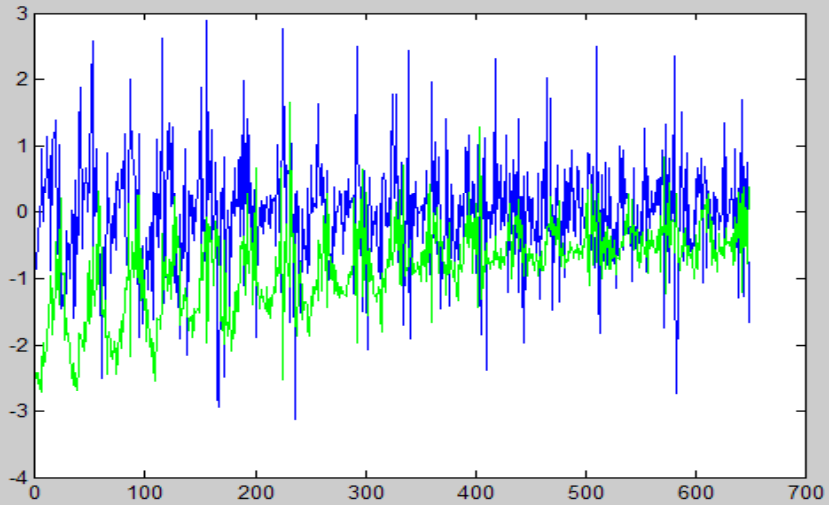


•Legs

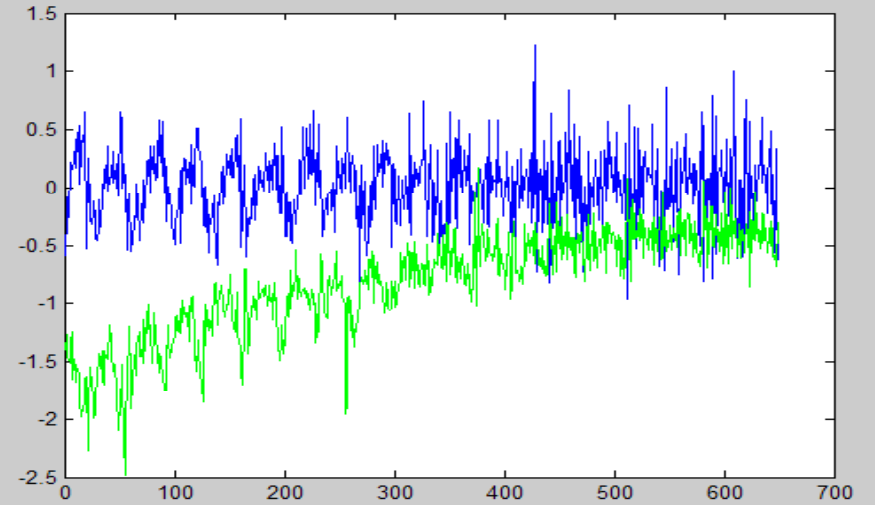


135 Degrees

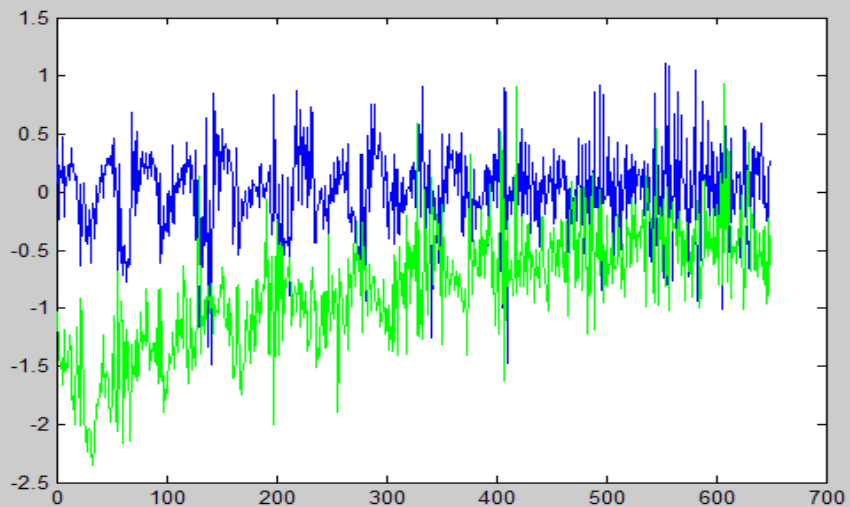
•Full Body



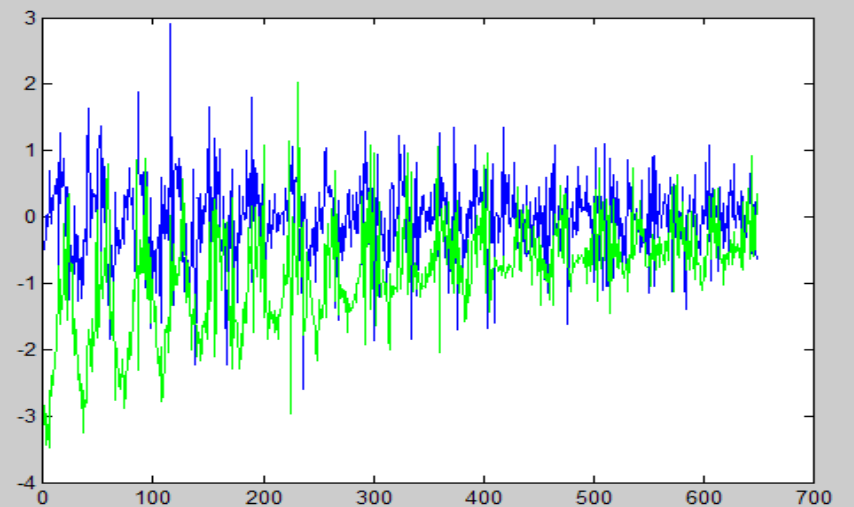
•Torso



•Arms



•Legs



Observations

- **Observing the walkers motion from the perspective of a camera introduces additional frequencies that are not present in the original 3D motion. This is essentially an aliasing phenomena introduced by dimensionality reduction and geometric masking issues. Furthermore, the way that this aliasing presents itself is perspective dependent**
- **The full body motion illustrated in the plots is a complex summation of the motions of the various body components. Plots are included of the “arms”, “legs”, and “torso” to illustrate this fact.**
- **From the 90° aspect one can see that these three components show significant high frequency behavior which becomes smoothed out when combined into the full body motion.**

Observations

- **The predominant frequency in the full body motion seems to reflect the frequency of the “legs” modified by the periodic motion of the other body parts.**
- **Thus, the high frequency behavior of the full body appears to be a weighted sum of the other components.**
- **While the average horizontal displacement will be related to the full body speed, variations about this average (i.e., the high frequency behavior) will be related to the individual’s specific body motion characteristics (e.g., arm swing frequency, foot motion, leg position, etc).**

Observations

- For the backside view the high frequency behavior of the horizontal motion is indicative of the rotational sway (i.e., yaw) in the body, primarily the torso as the “torso” data show, while the high frequency variations in the vertical direction result from the “bobbing” motion of the individual.
- Deviations of the velocity components from a constant average for nonorthogonal views result from the perspective size changes and obscuration in the body image. The degree of this deviation can be related to the view angle and corrections applied to the data.

First Principles Based Vehicle Dynamics Model

Keith Prussing

Electro-Optical Systems Laboratory

Georgia Tech Research Institute

School of Physics

Keith.Prussing@gtri.gatech.edu

September 14, 2010

Vehicle Dynamics Model

- **Due to the prevalence of untracked passenger vehicles in urban environments, an accurate model of vehicle motion was developed.**
- **Motion had to be constrained by physical properties of the vehicle and by the driver interaction.**
- **Routine had to access the Scene-Simulation Database (SSDB) to get scene information and write the results.**

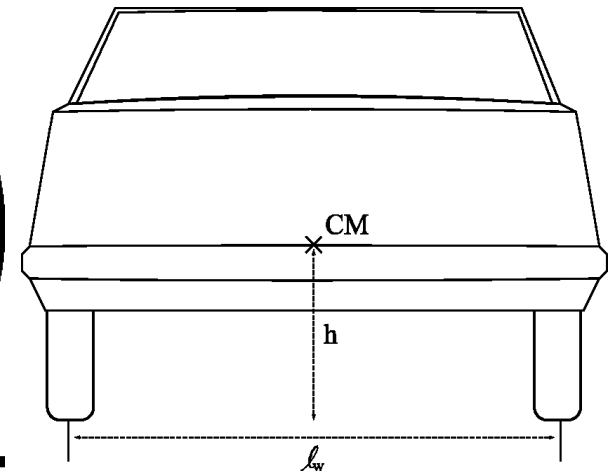
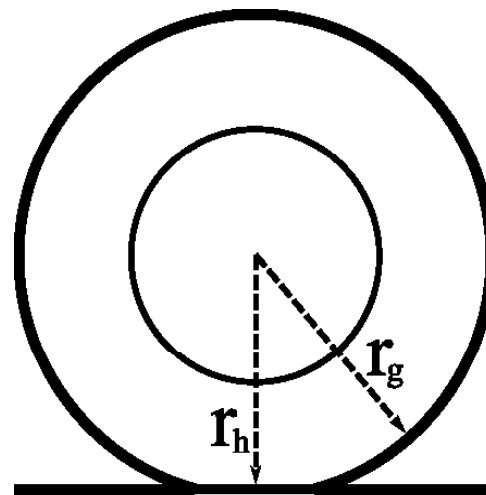
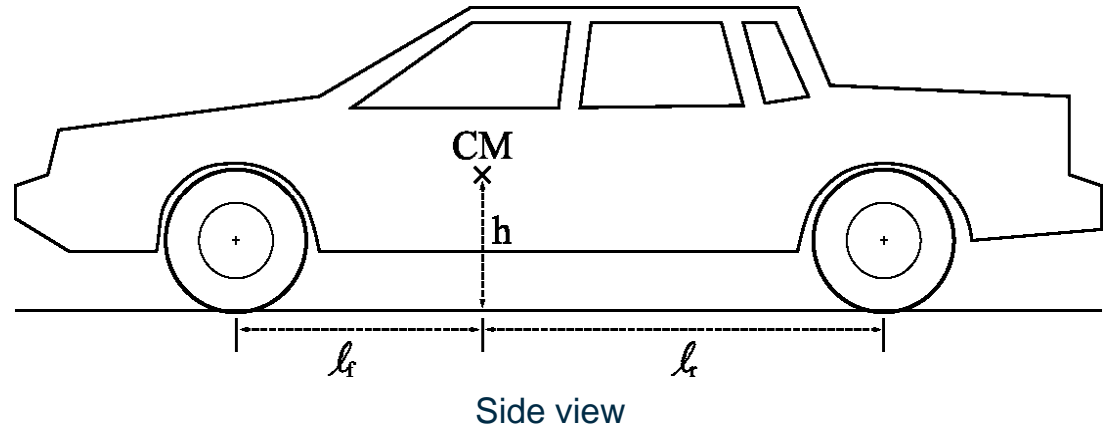
Vehicle Dynamics Modeling

- Modeled as five rigid bodies.
 - Body and four wheels.
- Rigid body equations are used to model motion.
 - Center of mass motion.
 - Rotations about center of mass.
- Physical properties influencing motion:
 - Mass,
 - Track width,
 - Wheel base.

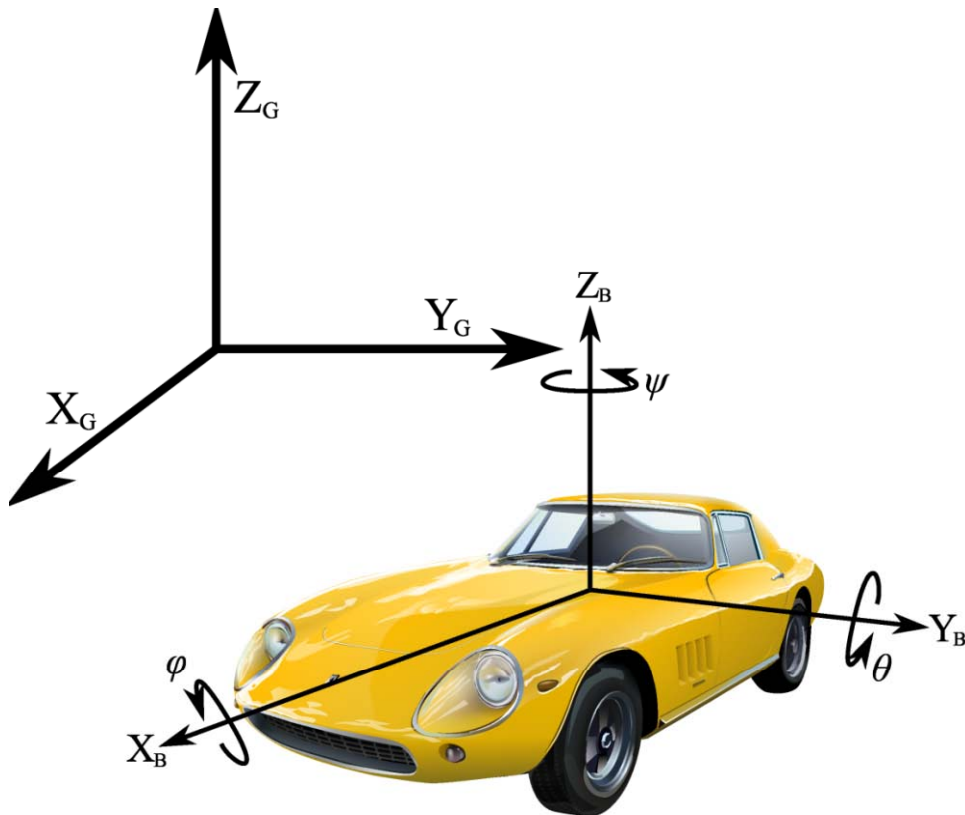


Geometric Parameters of Vehicle Motion

Description	Symbol
Distance to front axle	l_f
Distance to rear axle	l_r
Track width	l_w
Center of mass height	h
Geometric wheel radius	r_g
Loaded wheel radius	r_h
Effective wheel radius	r_{eff}



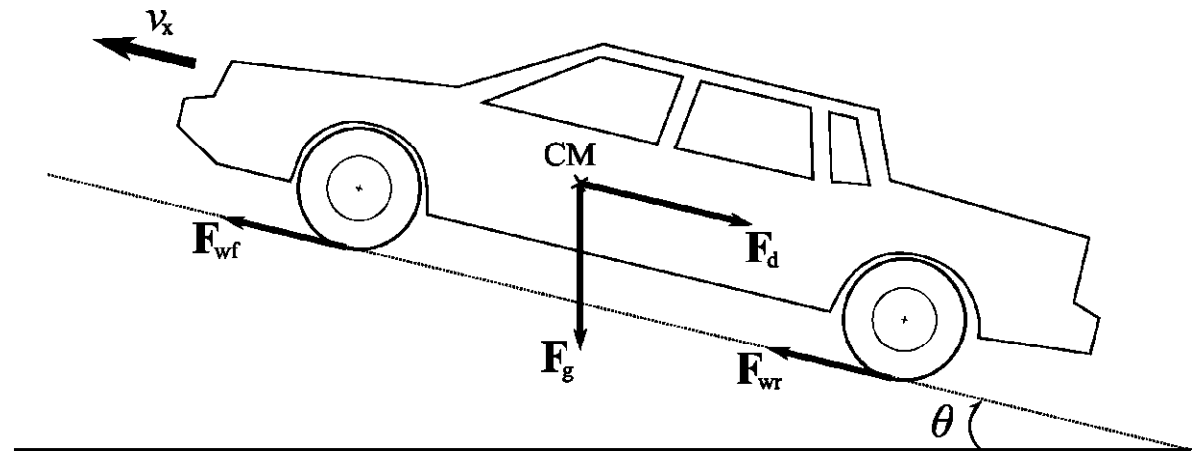
Coordinate Systems



- **Three coordinate systems are needed**
 - Global , Body, and Wheel.
 - All right handed with x forward.
- **Two standards for the z axis:**
 - SAE: z-down
 - ISO: z-up
- **ISO Standard used for greater intuition.**

Equations of Motion

- Forces initially resolved in the body coordinate system then rotated to plane of motion for computation.
- Simplest forms of forces used (Rajamani, 2006):
 - Elementary gravity,
 - Quadratic Drag,
 - Tire Rolling Resistance,
 - Dugoff Tire Model.

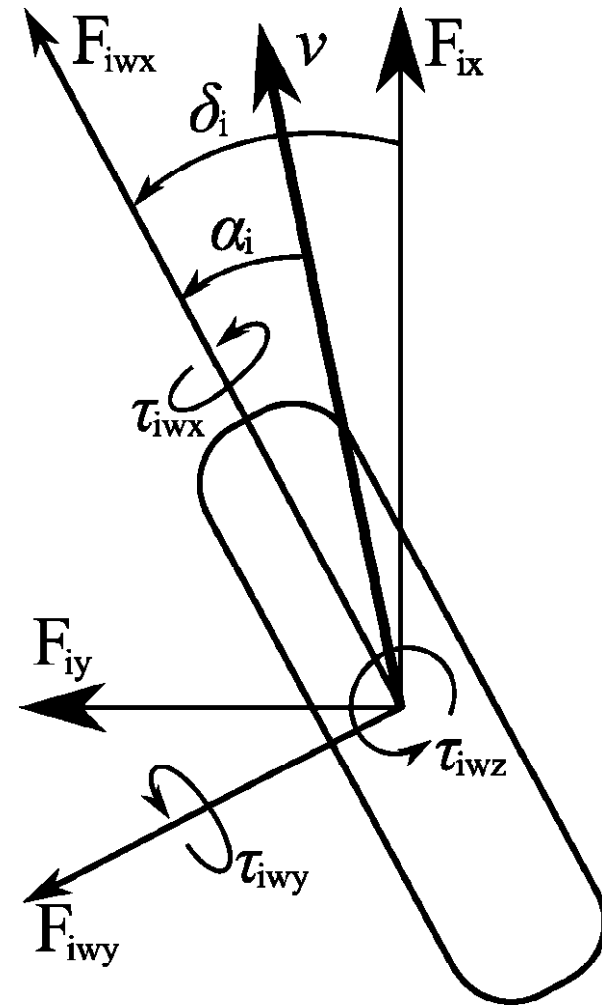


Forces acting upon a vehicle

Force	Symbol
Gravity	F_g
Aerodynamics Drag	F_d
Front Wheel	F_{wf}
Rear Wheel	F_{wr}

Wheel Forces

- This is the force that generates forward motion of a vehicle
- It is the sum of the rolling resistance and the Dugoff Tire Model (Rajamani, 2006).
- Depends on the load, F_{z_i} , steering angle, δ_i , and side slip angle, α_i , (Jazar, 2008).
- Total force and torque is the vector sum of forces and torques generated by each wheel (Jazar, 2008).



Dugoff (1969) Tire Model

- Analytic model of longitudinal and lateral forces produced by one tire.
- Derived assuming uniform pressure in tire contact patch.
- Valid for normal driving conditions.
- Breaks down at large slip values.

$$F_x = C_\sigma \frac{\sigma}{1 + \sigma} f(\lambda)$$

$$F_y = C_\alpha \frac{\tan \alpha}{1 + \sigma} f(\lambda)$$

$$\lambda = \frac{\mu F_z (1 + \sigma)}{2 \left((C_\sigma \sigma)^2 + (C_\alpha \tan \alpha)^2 \right)^{1/2}}$$

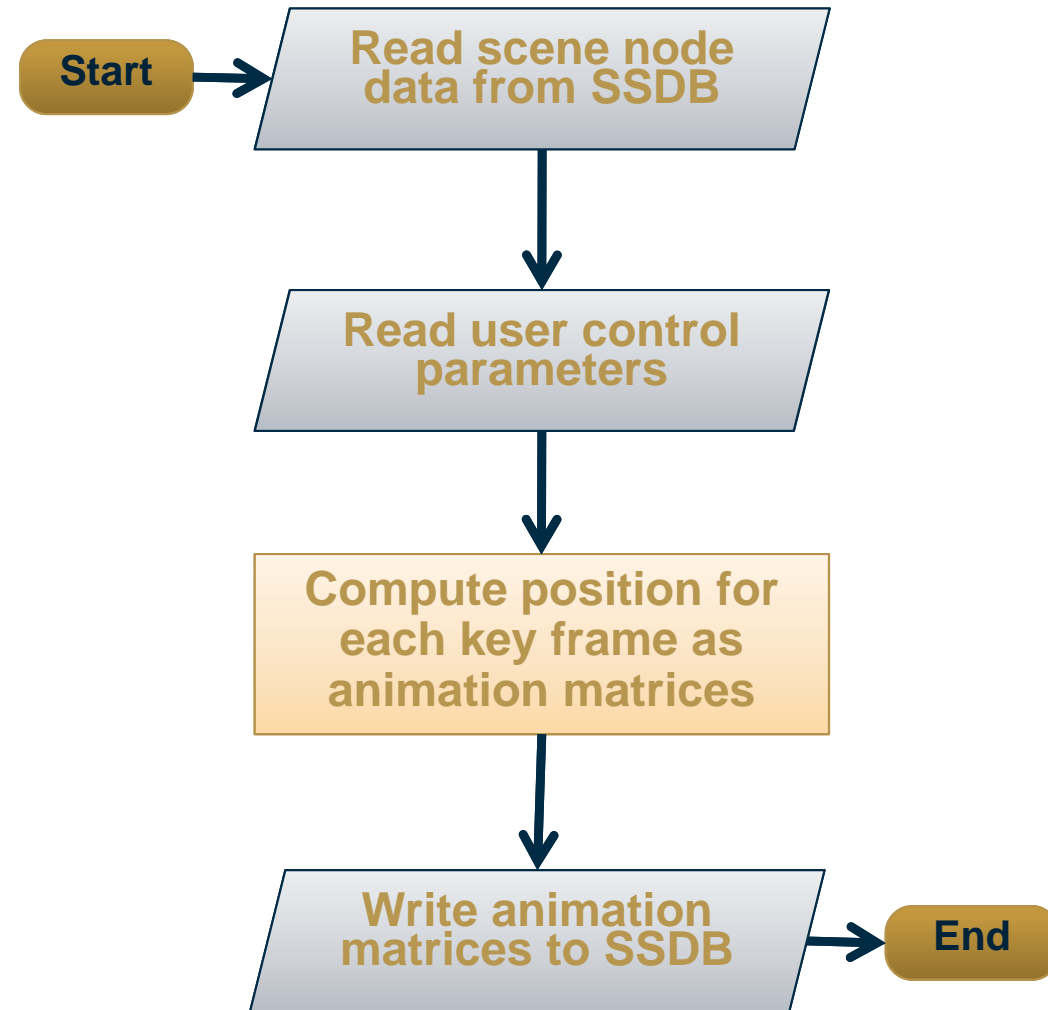
$$f(\lambda) = \begin{cases} (2 - \lambda)\lambda & \lambda \leq 1 \\ 1 & \lambda > 1 \end{cases}$$

$$\sigma = \begin{cases} \frac{r_g \omega_w - v_x}{v_x} & r_g \omega_w < v_x \\ \frac{r_g \omega_w - v_x}{r_g \omega_w} & r_g \omega_w > v_x \end{cases}$$

Description	Symbol
Longitudinal Tire Stiffness	C_σ
Lateral Tire Stiffness	C_α
Rotational velocity	ω_w

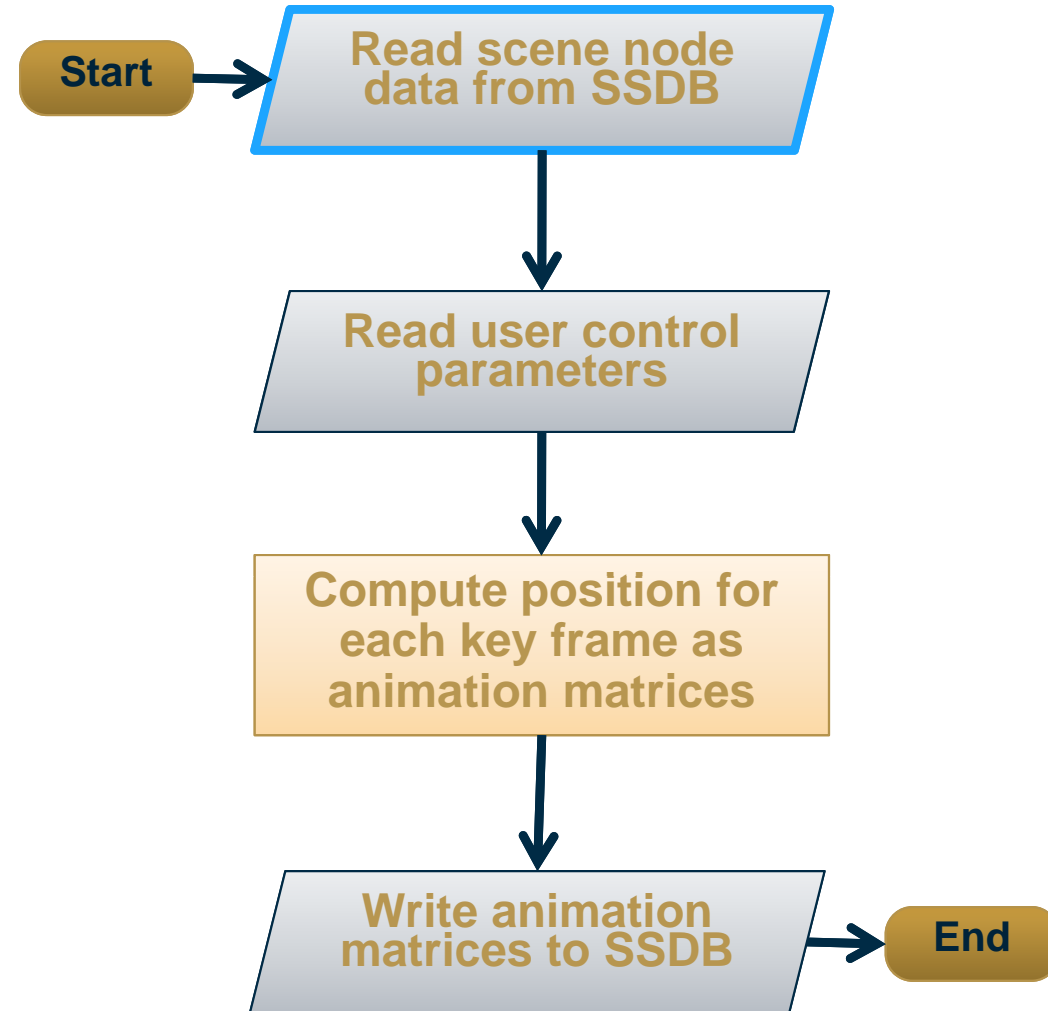
Computation Routine

- Routine can be broken into four steps:
 - Query the SSDB.
 - Acquire user control parameters.
 - Numerically integrate the equations of motion.
 - Write the results to the SSDB.



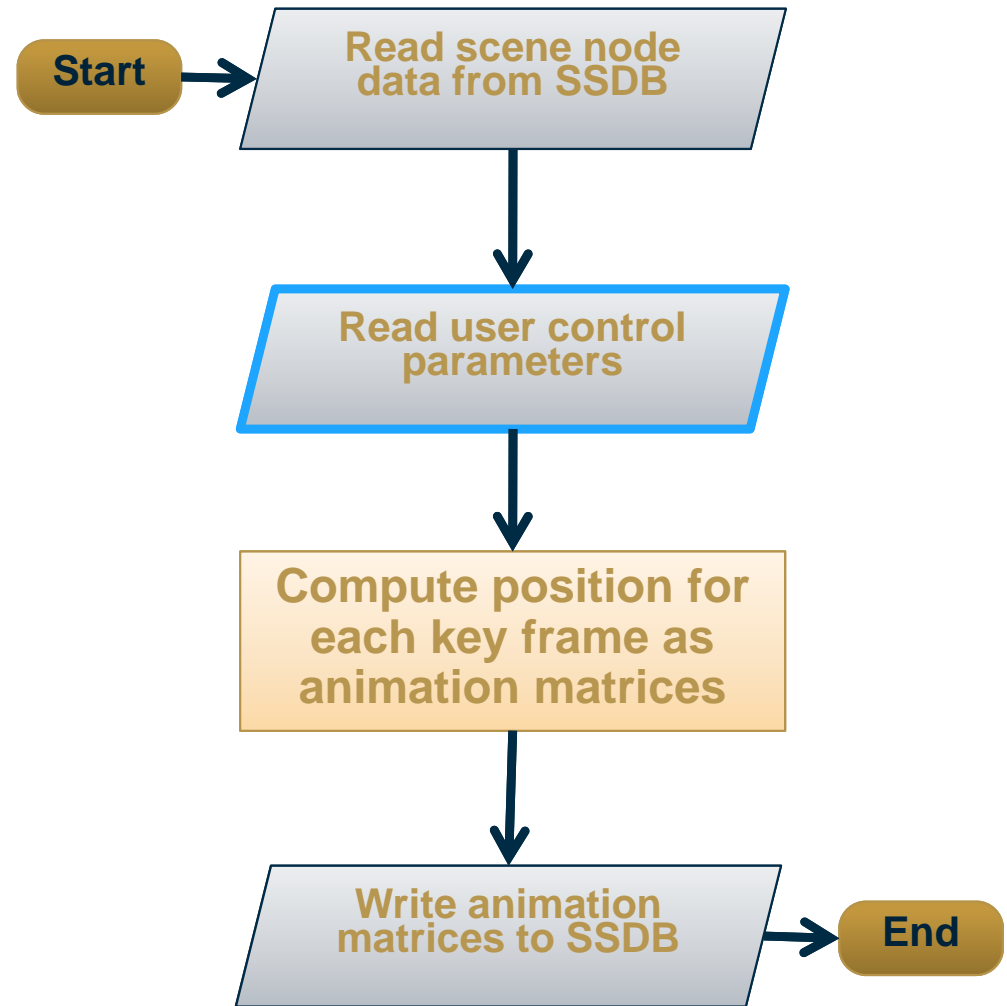
Scene Node Data

- Scene orientation is taken to be global coordinate system.
- A specified ground scene node is taken as plane of motion.
- Vehicle mesh is taken from scene node assignment.
- Geometric parameters are taken from vehicle mesh.



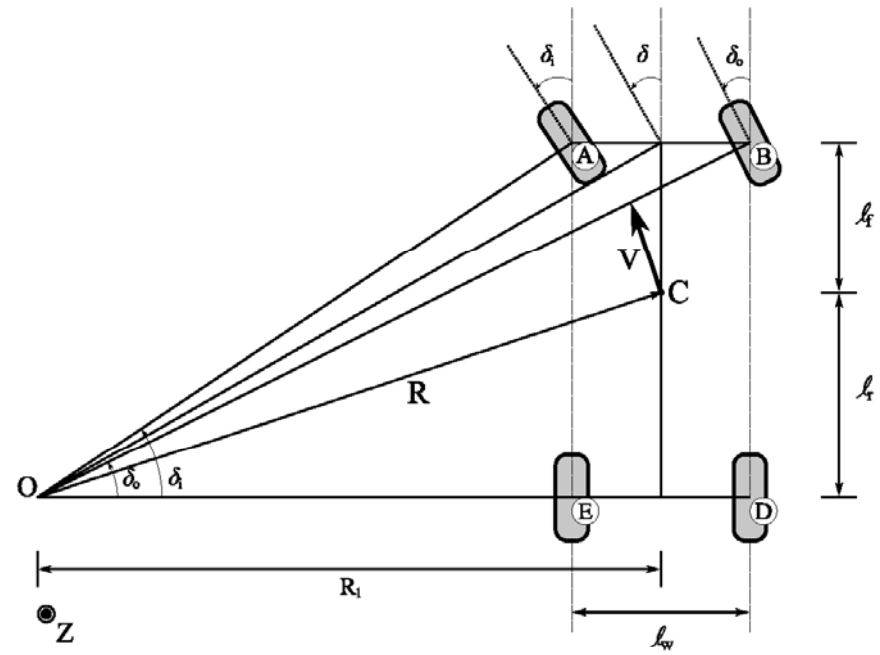
User Control Parameters

- Initial position and heading are taken as inputs.
- Engine speed as a function of time simulates driver adjusting gas pedal.
- Equivalent steering angle as a function of time simulates driver turning the steering wheel.



Equivalent Steering Angle

- Defined as cotangent average of inner and outer steering angles (Jazar, 2008).
- Can be decomposed into left and right steering angles for a front wheel drive vehicle using Ackerman steering condition (Jazar, 2008).
- Steering angles are used to determine wheel forces and the final orientation of each wheel independently.



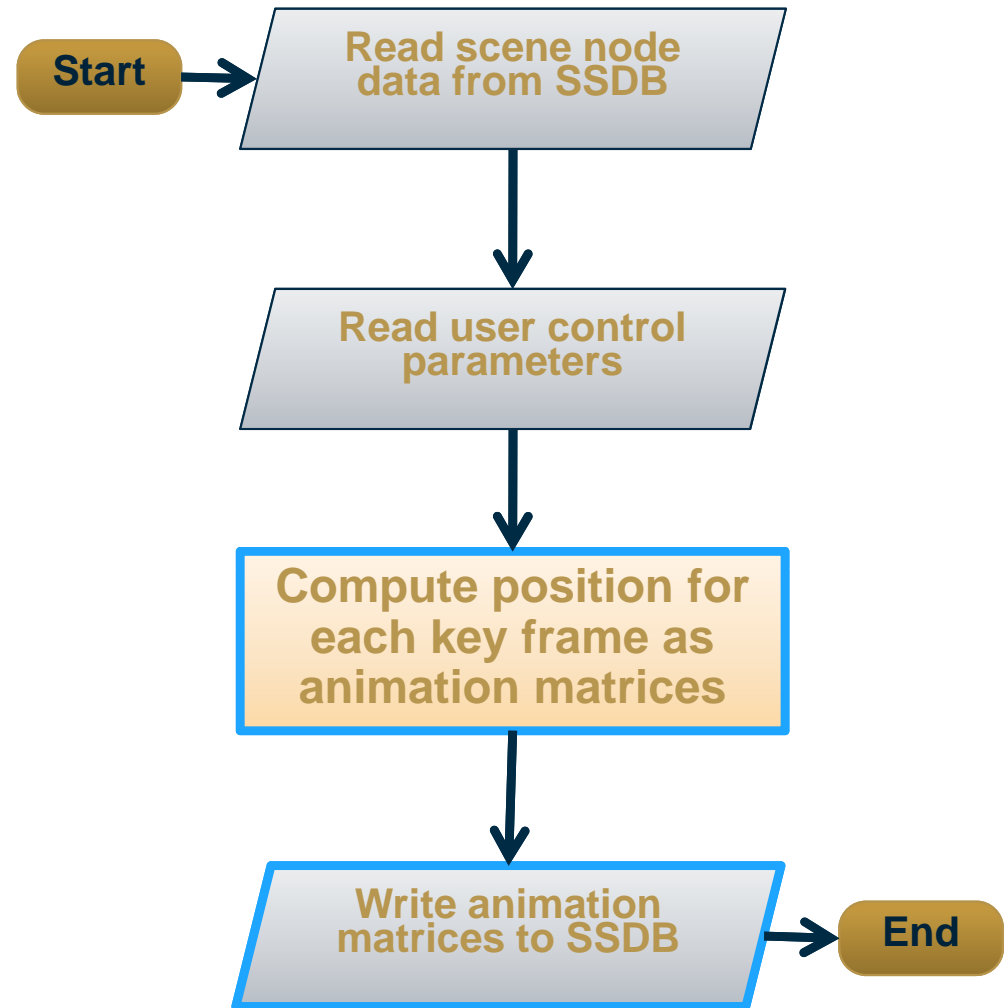
Ackerman Steering Geometry

$$\delta_{fl} = \arctan \left(\frac{\tan \delta}{1 - \frac{\ell_w}{2(\ell_f + \ell_r)} \tan \delta} \right)$$

$$\delta_{fr} = \arctan \left(\frac{\tan \delta}{1 + \frac{\ell_w}{2(\ell_f + \ell_r)} \tan \delta} \right)$$

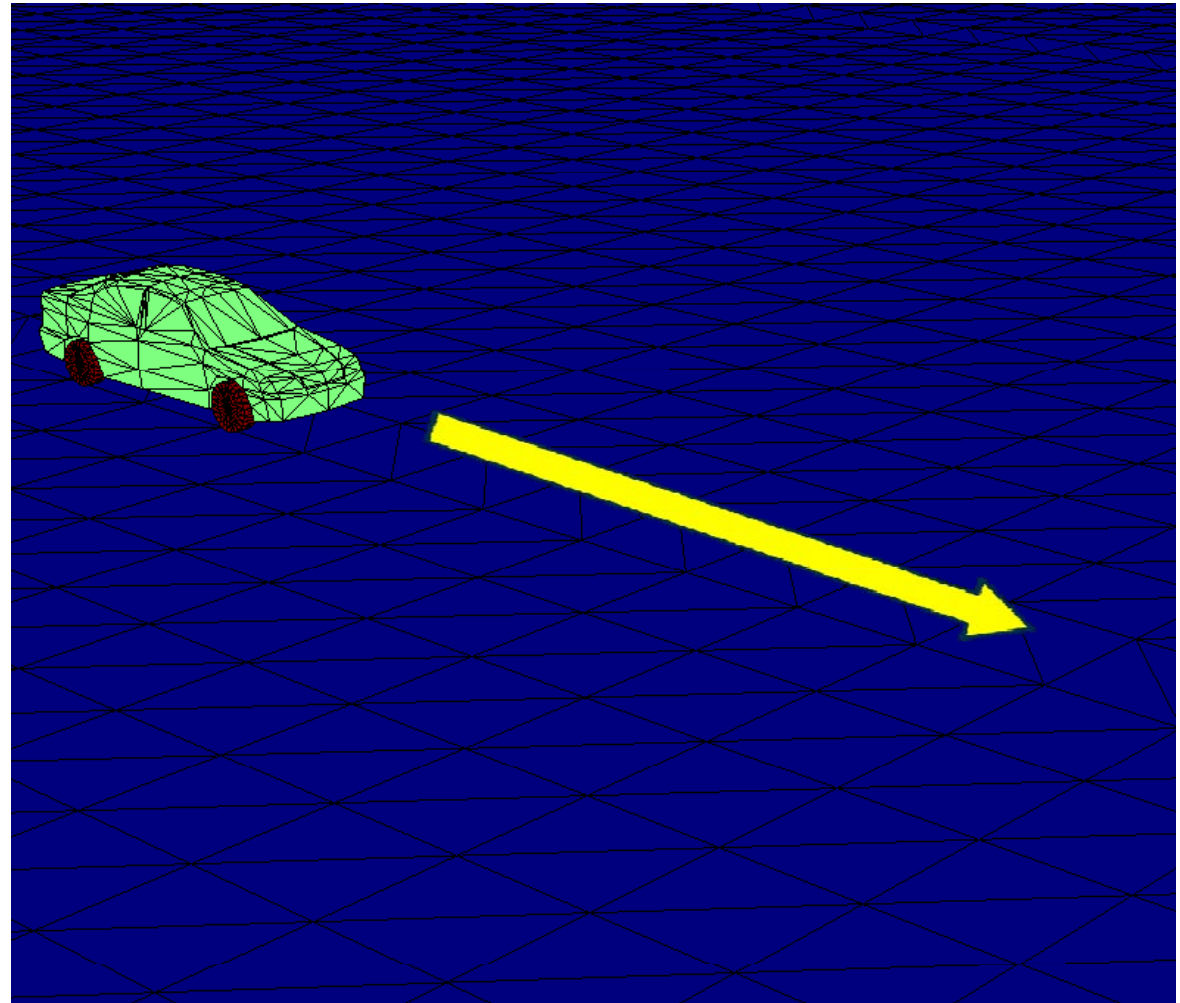
Compute Position and Store the Data

- Position and orientation computed numerically using second order Runge-Kutta method.
- Coordinates are used to compute the animation matrix for the vehicle body and each wheel as a separate child scene node.
- Animation matrices are written directly to the SSDB.



One Degree of Freedom Model

- **Vehicle constrained to move on a line.**
- **Body cannot rotate about any axis.**
- **Wheels are constrained by rolling condition.**



One Degree of Freedom Model

$$X_G(t) = X_{G0} + \cos \psi \cos \theta \frac{\nu}{b} \ln \left(\sqrt{1 - \left(\frac{v_0}{\nu}\right)^2} \cosh \left(bt + \operatorname{arctanh} \left(\frac{v_0}{\nu} \right) \right) \right)$$

$$Z_G(t) = Z_{G0} - \sin \theta \frac{\nu}{b} \ln \left(\sqrt{1 - \left(\frac{v_0}{\nu}\right)^2} \cosh \left(bt + \operatorname{arctanh} \left(\frac{v_0}{\nu} \right) \right) \right)$$

$$\nu = \sqrt{\frac{2(F_w + mg \sin \theta)}{\rho C_d A_F}}$$

$$b = \sqrt{\frac{(F_w + mg \sin \theta) \rho C_d A_F}{2m^2}}$$

$$F_w = -mg \sin \theta + \frac{1}{2} \rho C_d A_F v_{max}^2$$

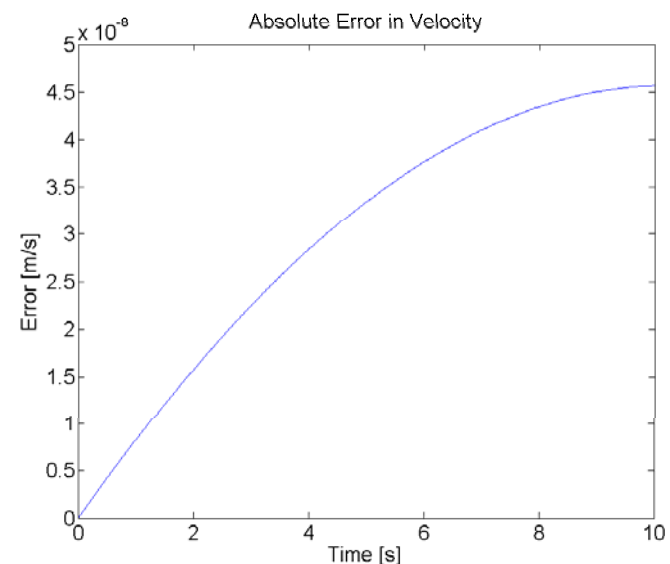
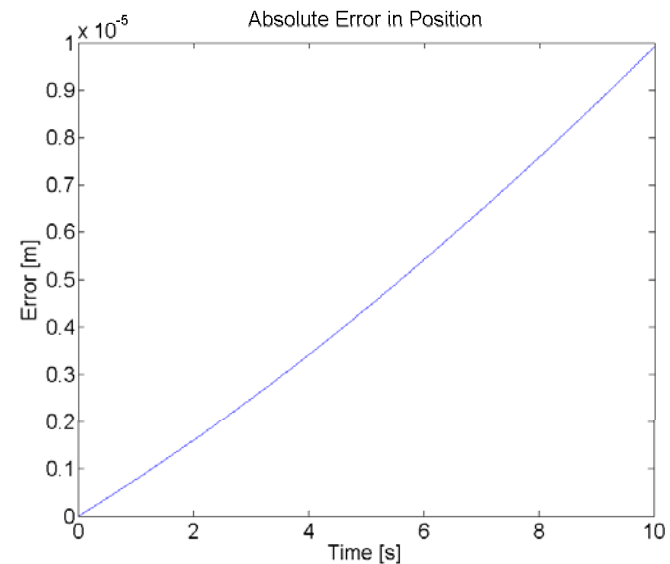
v_0	:	Initial velocity	ρ	:	Density of air
v_{max}	:	Maximum velocity	C_d	:	Drag coefficient
m	:	Vehicle mass	A_F	:	Frontal area

- **By assuming constant acceleration and maximum upper velocity, a closed form solution is possible.**

- **Simplest non-trivial motion used to verify the algorithm.**

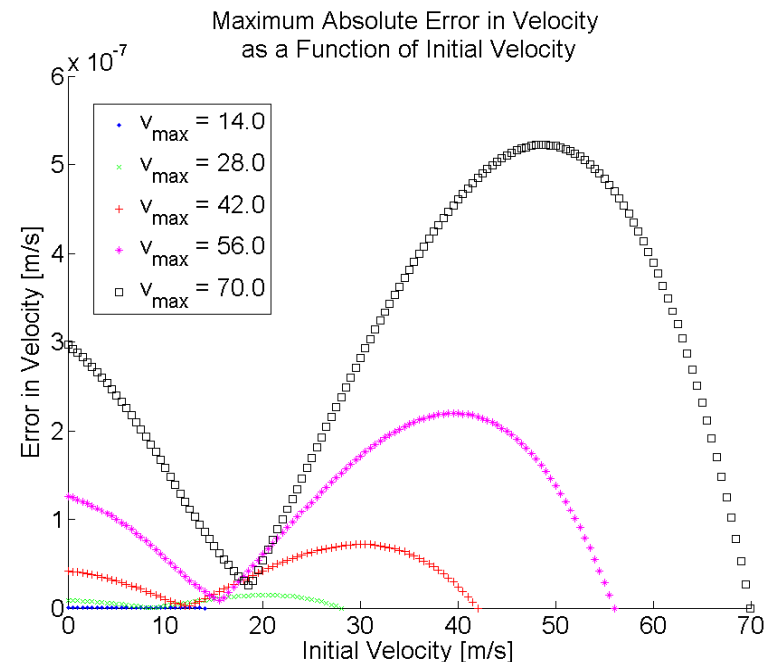
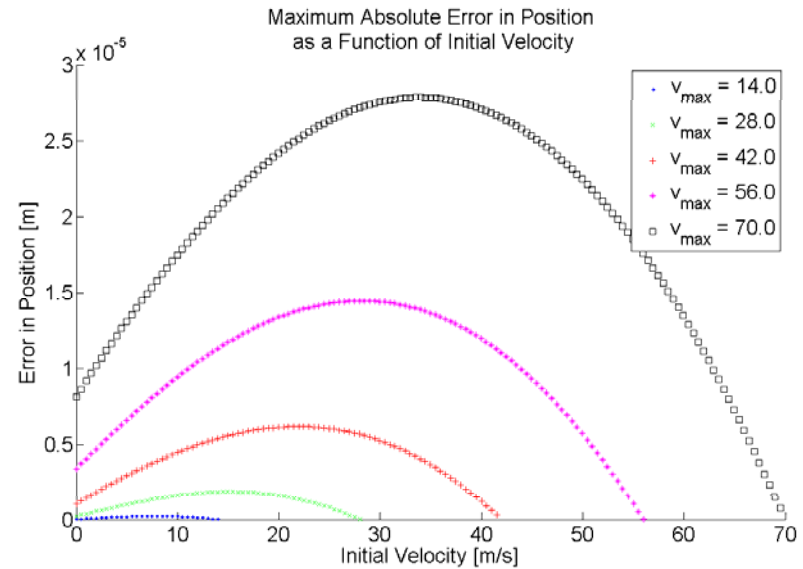
One Degree of Freedom Model

- The absolute error was taken to be the difference between the analytic and numerical results.
- The error in position was found to be on the order of 0.01% when the absolute error is divided by the radius of a wheel.

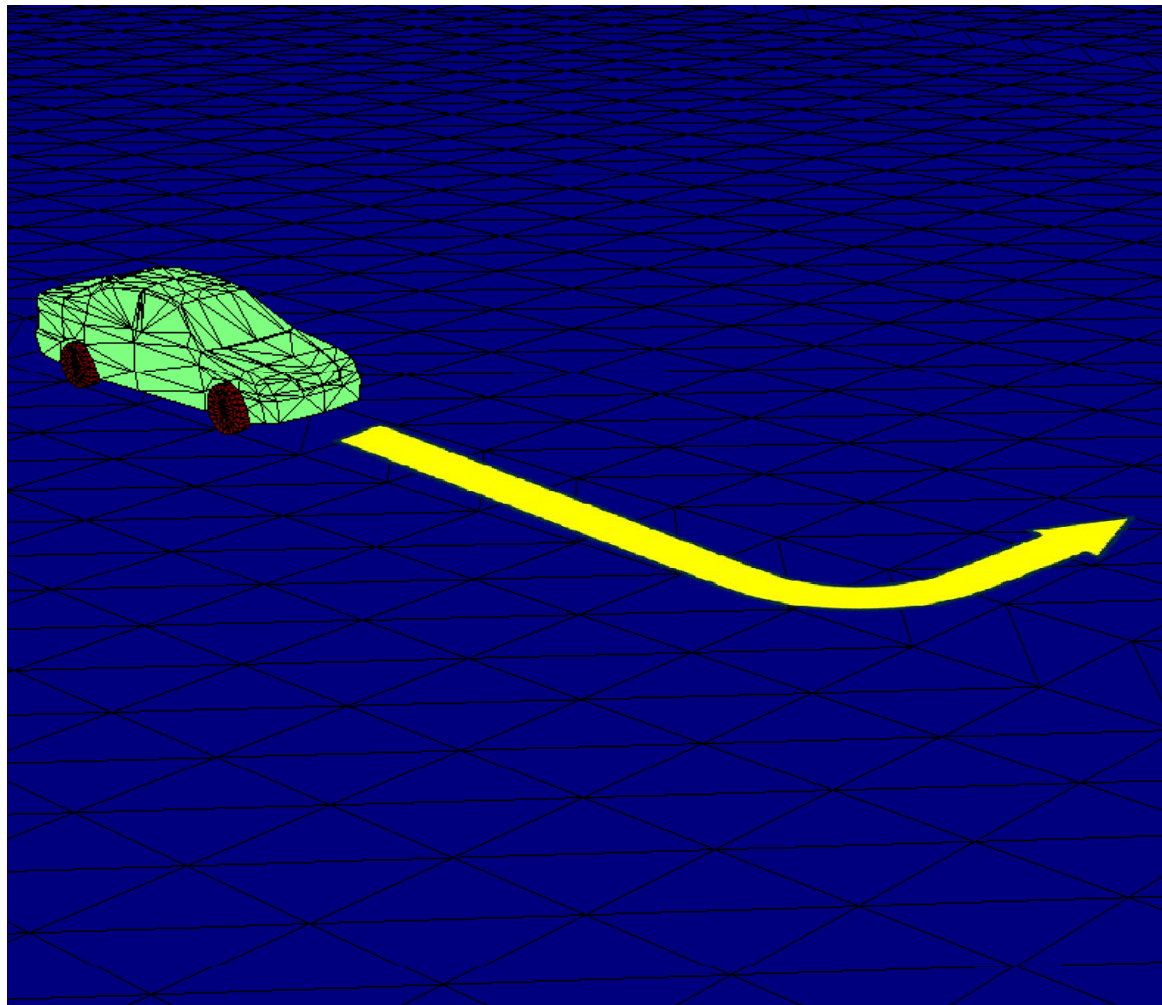


One Degree of Freedom Model

- By varying the maximum velocity, it was found that the error does increase with the maximum velocity, but the error remains below 0.1%.
- Variation of time step found that a time step of 1/30 s was adequate.



Three Degree of Freedom Model



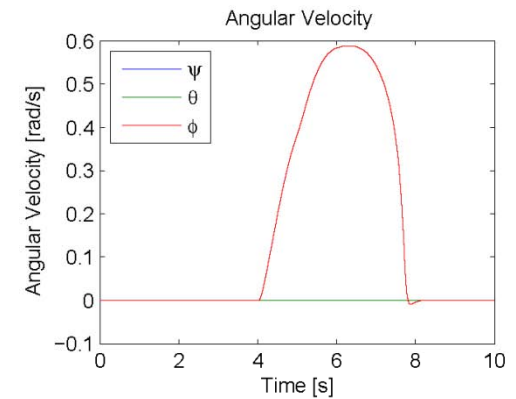
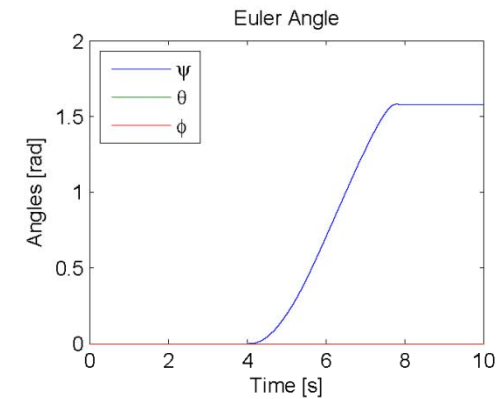
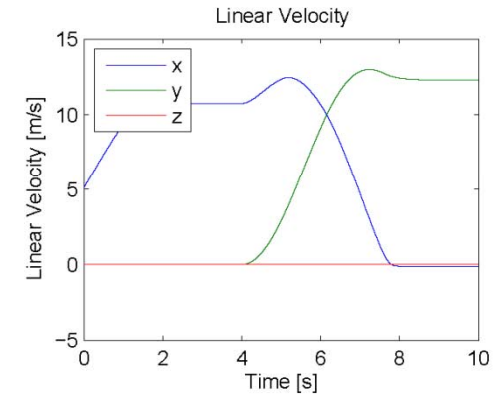
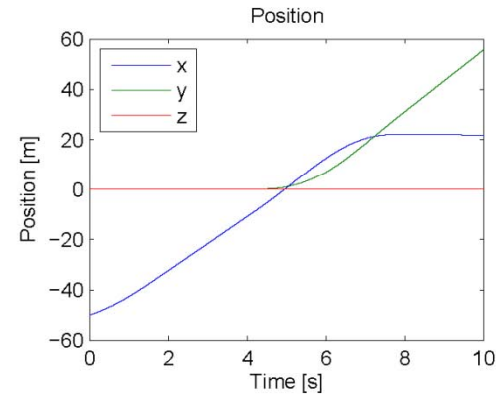
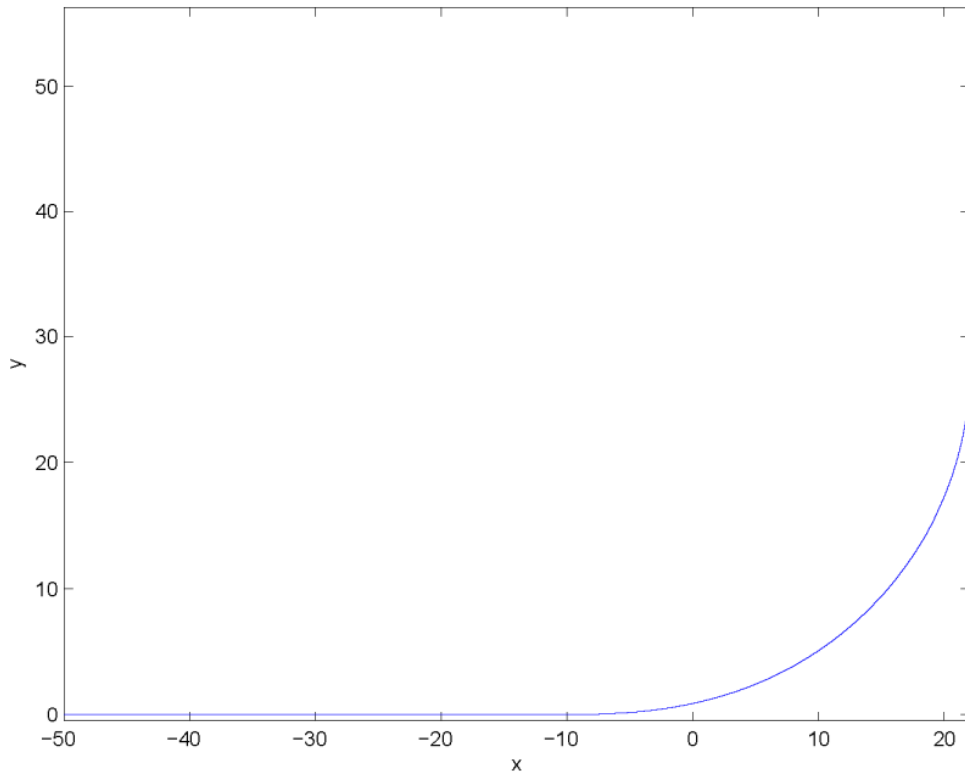
- **Vehicle is constrained to move in a plane.**
- **Straight line motion compared to one DOF model and found to be within numerical error of one DOF model.**

Three Degree of Freedom Model

- To examine the qualitative features of the model, a variety of maneuvers were simulated.
- Parametric plots of the motion and plots of the phase space coordinates as a function of time are provided on the next three charts for a vehicle:
 - Moving forward and making a left turn.
 - Performing a lane change.
 - Moving forward and making a u-turn.
- In all motions, we see that:
 - The vehicle is indeed constrained to move in the plane.
 - The coordinates change in the physically expected manner at a reasonable rate.

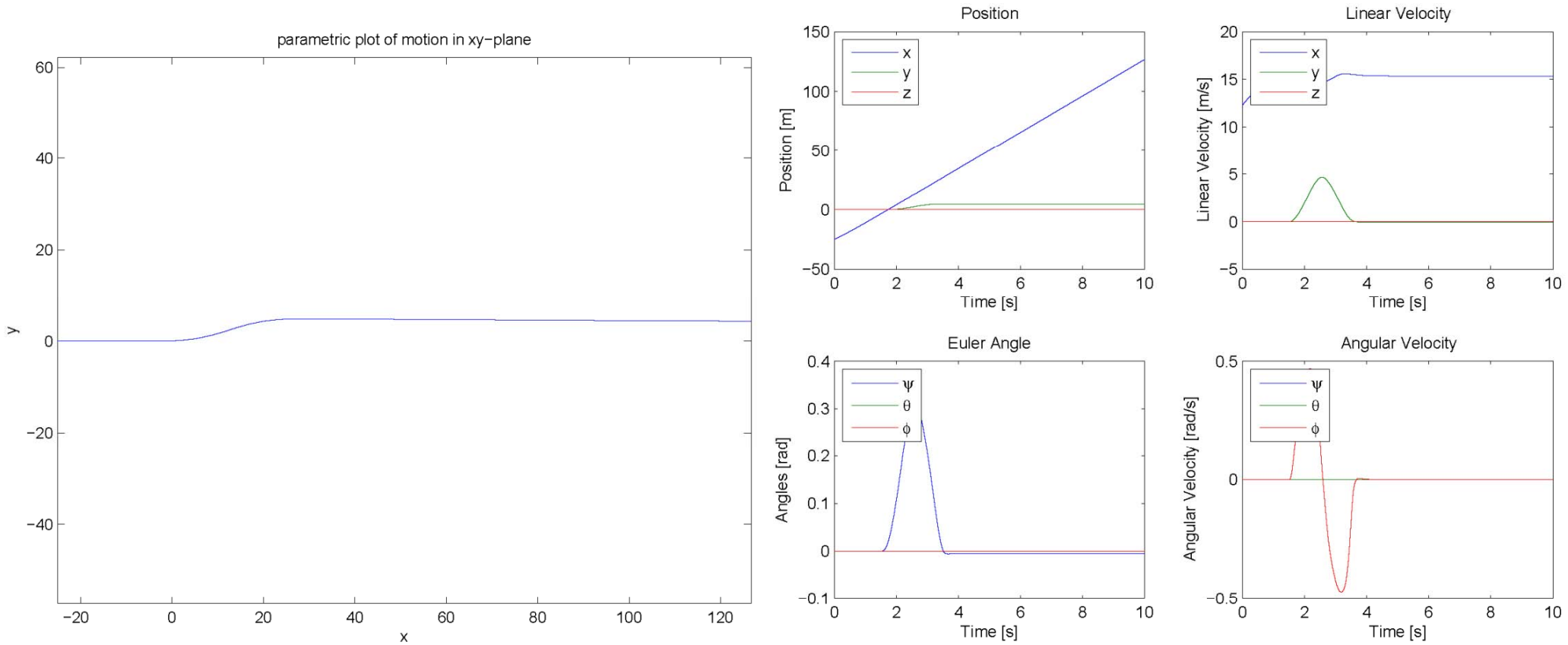
Three Degree of Freedom Model

parametric plot of motion in xy-plane



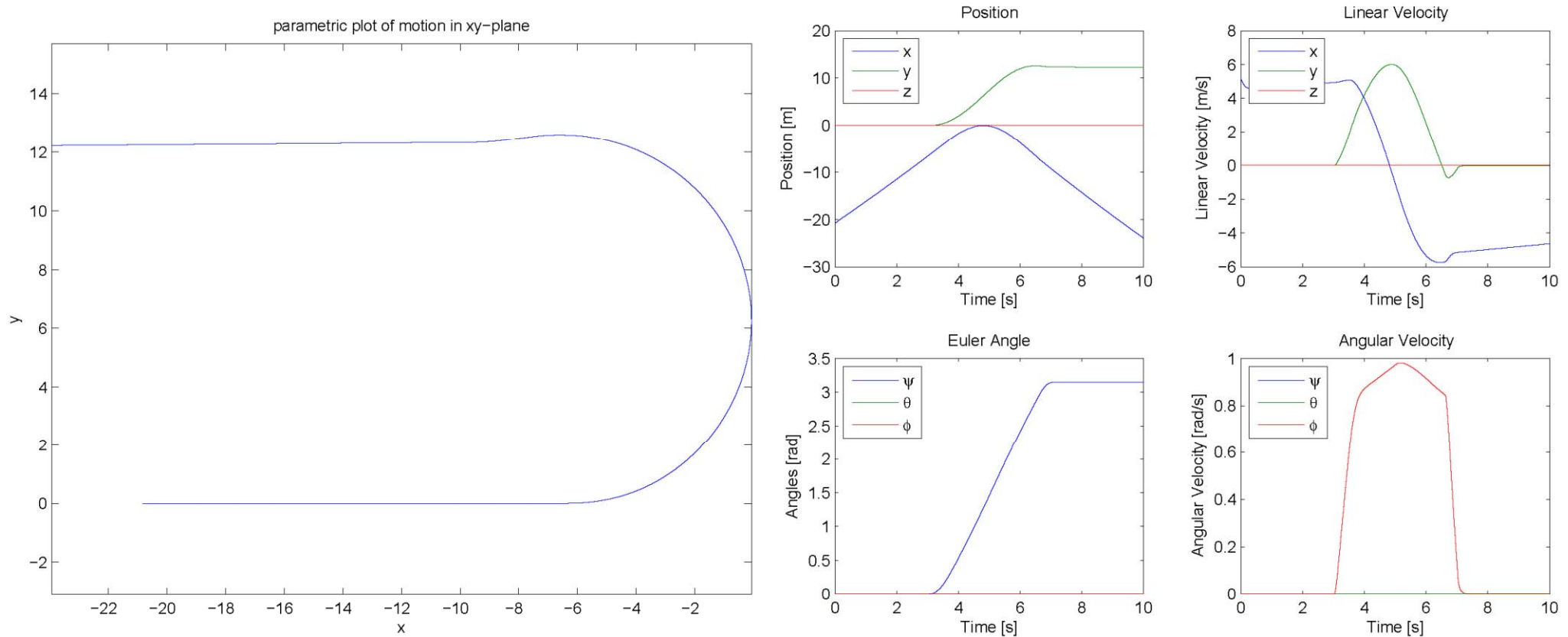
Example output of vehicle making a moving left turn

Three Degree of Freedom Model



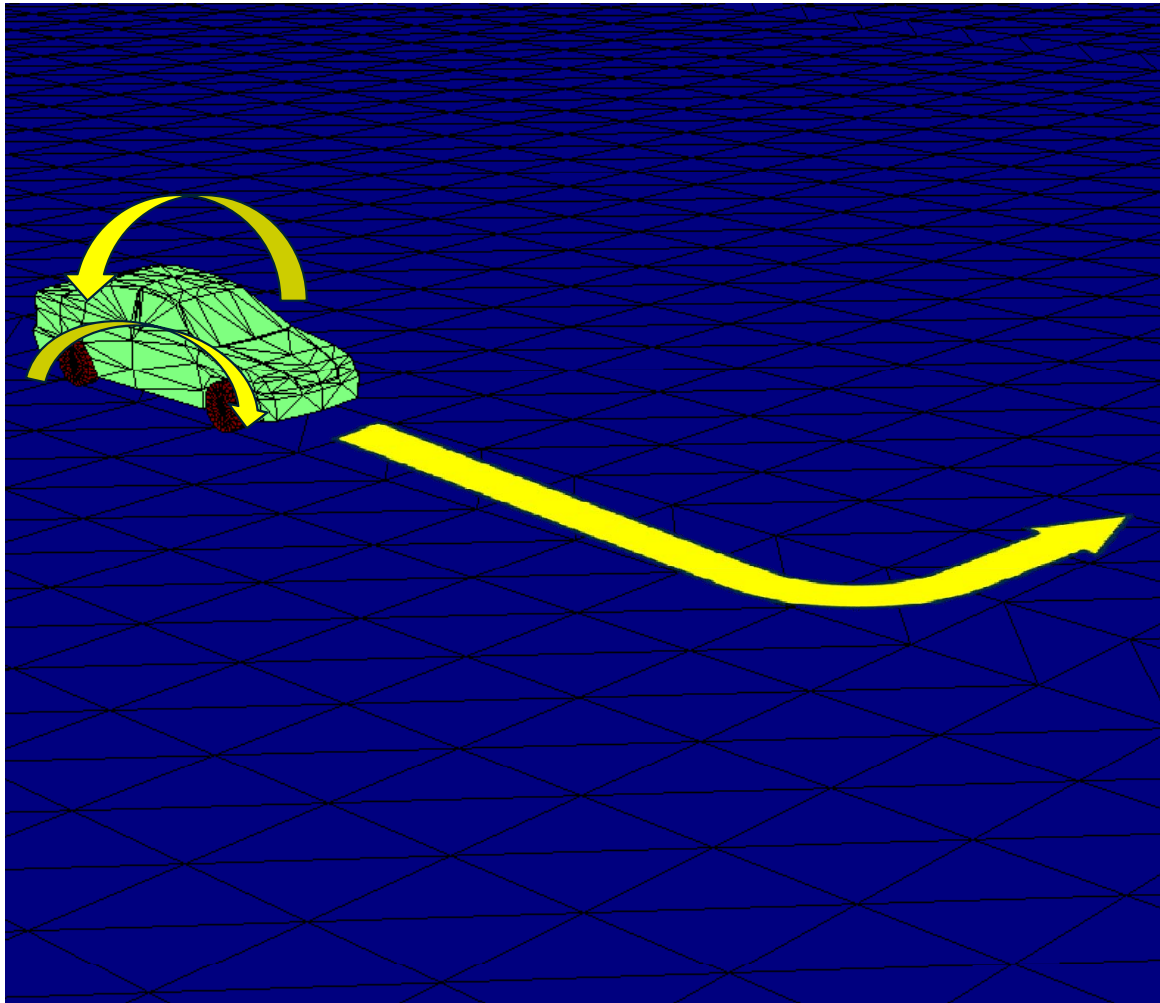
Example output of vehicle making a lane change at 25 MPH

Three Degree of Freedom Model



Example output of vehicle making a u-turn at speed

Five Degree of Freedom Model

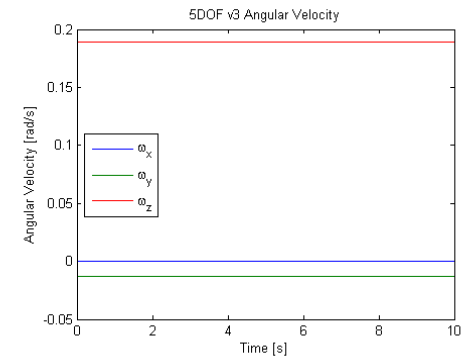
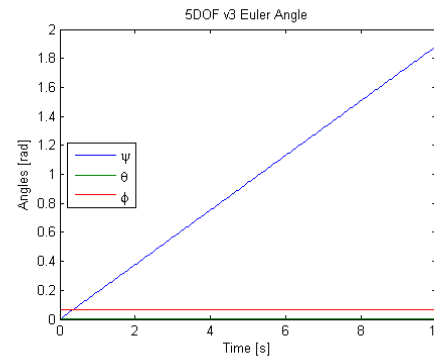
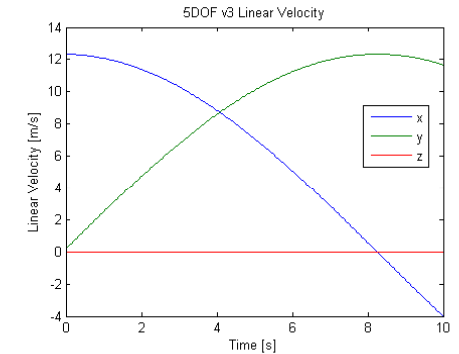
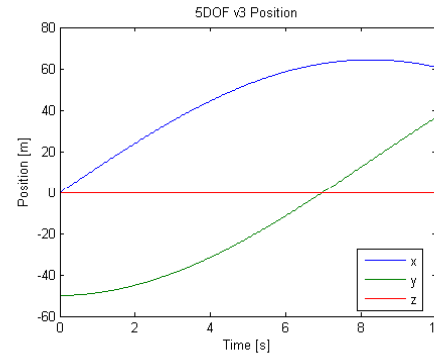
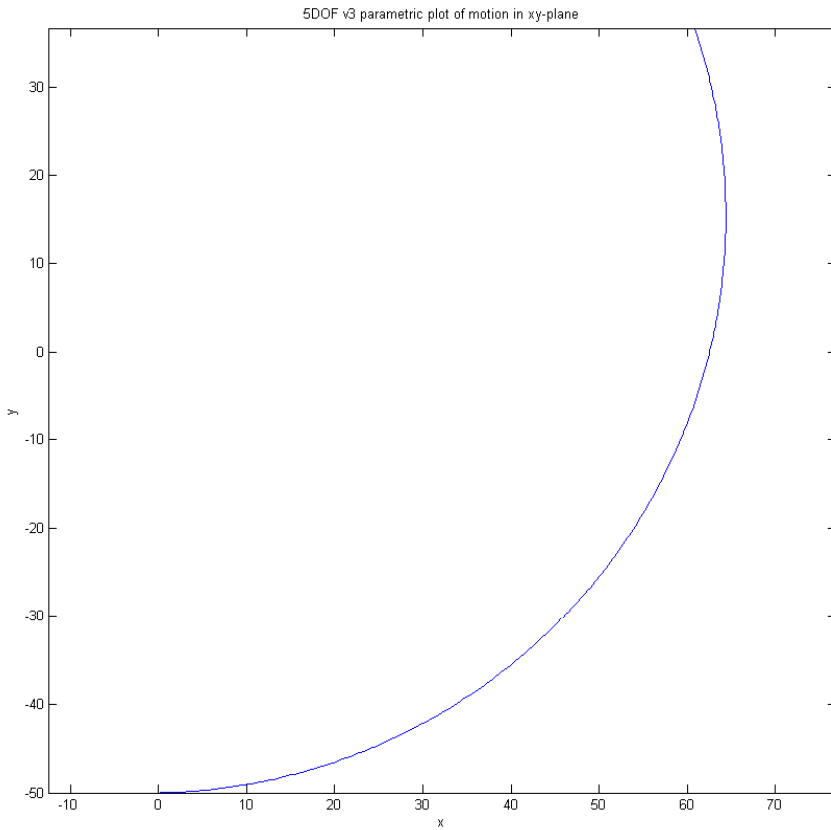


- Vehicle is constrained to move in a plane.
- Vehicle may rotate about the body x and y axes.

Five Degree of Freedom Model

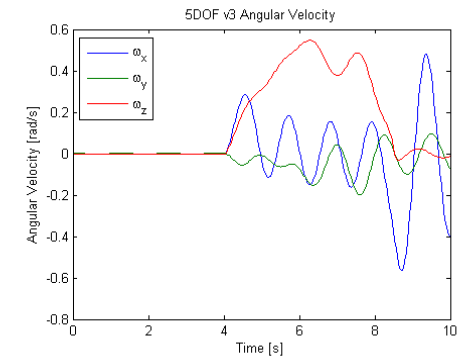
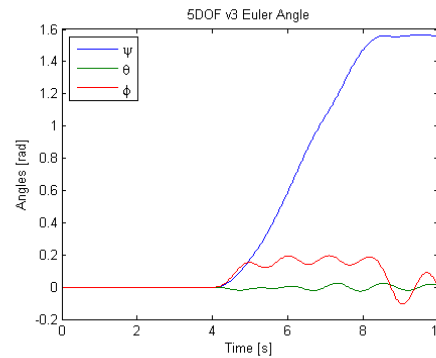
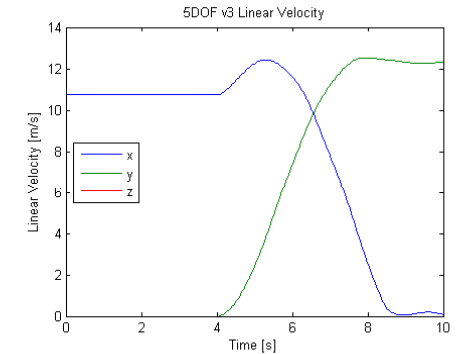
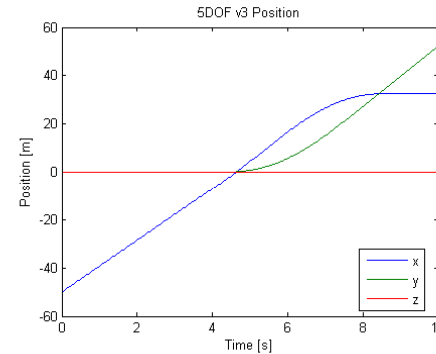
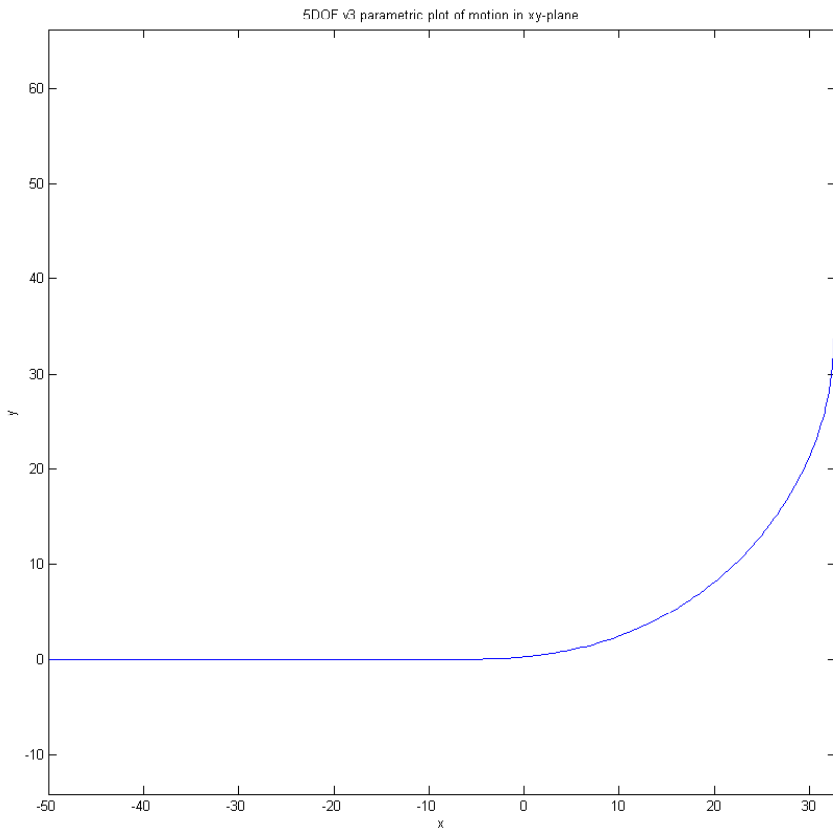
- Again, a the following maneuvers were simulated to examine the results of the model and the results are displayed on the following three charts.
 - **Circular motion, left turn, and lane change**
- The qualitative features of the motion are seen to agree with the three degree of freedom model and physical expectations.
 - We see that the when the vehicle is turning at a constant rate, the vehicle rolls as would be expected.
 - When the vehicle begins to turn, it will also rotate about the body x and y axes at a small angle.
- Exact results for motion were not available from the literature for comparison.

Five Degree of Freedom Model



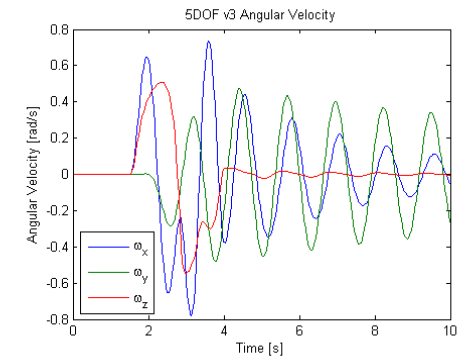
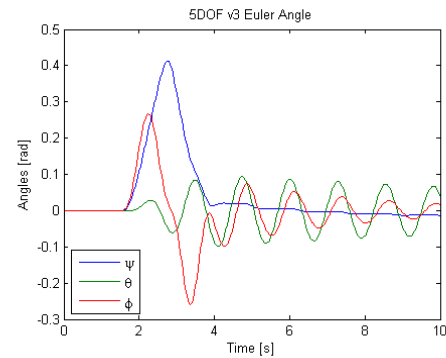
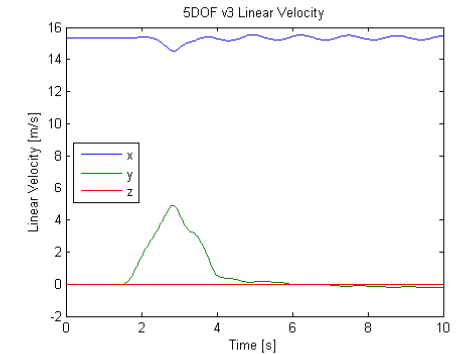
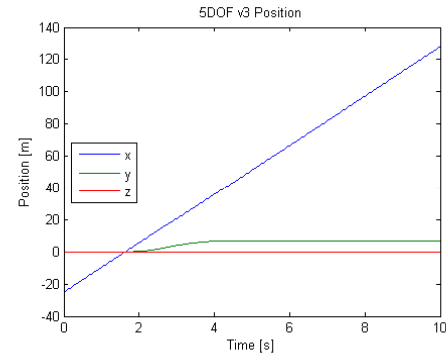
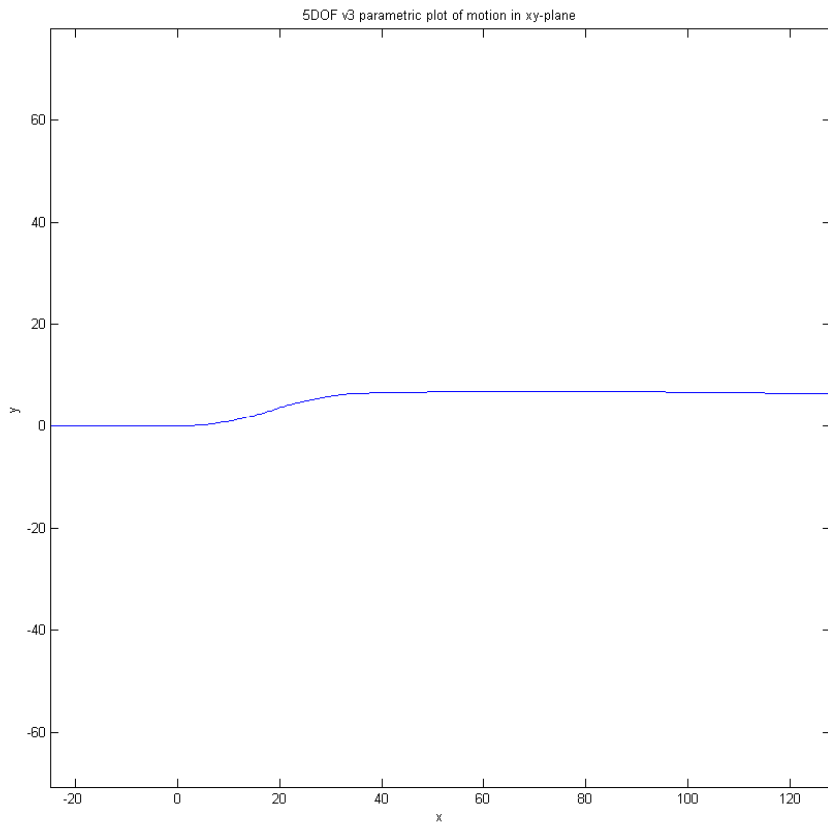
Example output of vehicle moving in a circle with 5 DOF

Five Degree of Freedom Model



Example output of vehicle making a moving left turn with 5 DOF

Five Degree of Freedom



Example output of vehicle making a lane change at 25 MPH with 5 DOF

References

1. **Dugoff, H., Fancher, P. S., & Segel, L. (1969). *Tire Performance Characteristics Affecting Vehicle Response to Steering and Braking Control Inputs* (Contract No. CST-460).**
2. **Jazar, R. N. (2008). *Vehicle Dynamics: Theory and Application*. New York: Springer.**
3. **Rajamani, R. (2006). *Vehicle Dynamics and Control*. New York: Springer Science.**

Signal Processing for Detection of Human Signals

Charles E Rohrs, MIT

1. Publications:

Peer reviewed conference publication:

M.B. Rudoy, C.E. Rohrs, J. Chen, "Signatures of Walking Humans from Passive and Active Acoustic Data using Time-Varying Vector Autoregressions," in the *Proceedings of the 41st Annual Asilomar Conference on Signals, Systems, and Computers*, (Asilomar, CA), November 4-7, 2007.

2. Scientific Personnel

Charles E Rohrs, Research Scientist, 50% time supported, 50% time worked
Alan Oppenheim, Professor, 5% time supported, more time worked as academic year salary is paid by MIT

Students:

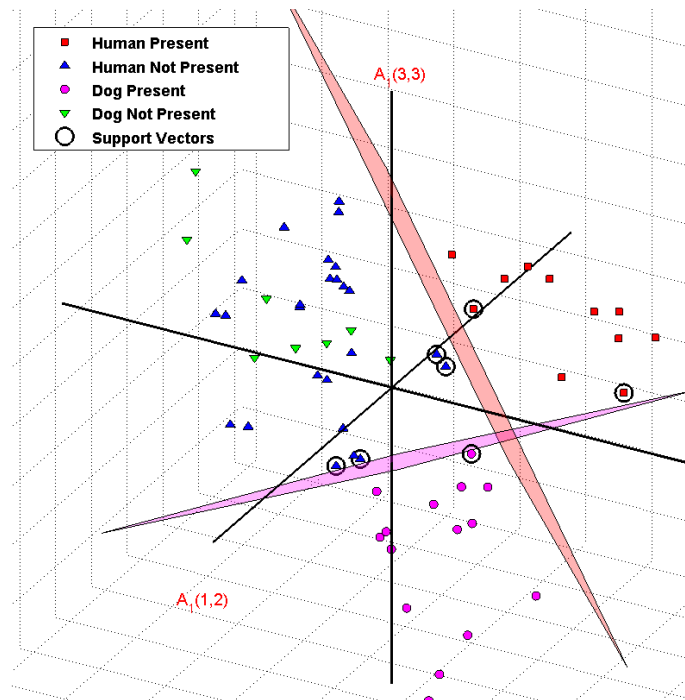
Melanie Shames, 50% time supported, 50% time worked
Tom Baran, 25% time supported, 25% time worked
Rajiv Divi, not supported, involved in discussions
Shay Maymon, not supported, significant contributions

3. Inventions Reported none

4. Research Summary and Accomplishments:

Processing to Produce Signature of Human Footsteps by Fusing of Active and Passive Ultrasound Signals. Signature can be used to Differentiate Human from Dog Footsteps.

The process involves treating active ultrasound spectrogram as image data to automatically extract two signals, one related to movement of limbs and the other related to movement of torso. A third signal from passive ultrasound is also used. The three signals are then used to identify parameters in a Vector Autoregressive (VAR) system. These parameters create the Signature. The Signature is used in a Support Vector Classifier to detect human footsteps in noise and to differentiate human footsteps from other animal footsteps, in particular, a test case of a dog. The results show clear separation of the three possibilities: noise, human, or dog.



Human, dog or no target decision regions in VAR parameter signature space.

Merging Data from N Sensors, Each Sampling at $1/N$ the Nyquist Rate. Estimating the Delay to Each Sensor and the Target's Position.

Consider a bandlimited signal that is captured simultaneously by N sensors. If each sensor samples the signal at a rate somewhat greater than the Nyquist rate divided by N and if the delay to each sensor is known, the original signal can be reconstructed using appropriate interpolation functions. The research accomplishment comes from recognizing that, if the interpolation is performed with other than the correct delays, energy is produced in the frequency band slightly above the highest frequency where energy is present in the original signal. Adjusting the delays in the reconstruction until this energy is minimized finds the correct delays and reconstructs the original signal. Efficient search algorithms were developed. Linearizing versus the delay makes it a Newton search that converges quickly. This is a significant development in multisensory fusion. Differing gains of different sensors are also easily found at the same time. Delay estimates are proportional to the difference in distance from the source to each sensor so information about the target's position becomes available.

5. Technology Transfer

Began initial discussions with group at BAE System, New Hampshire
 Found contact and established interest with group at MIT Lincoln Lab, expect significant handoff of technology that differentiates human vs. animal footprint signatures.

Human Signatures for Personnel Detection

**Multidisciplinary Research Program of the
University Research Initiative (MURI)**

James M. Sabatier

March 2010

Key Accomplishments

- Passive broadband footstep detection
- Active Doppler sonar
- Multi modal sensor performance
 - Combined passive and active ultrasonic sensors
 - Addition of radar
- Cadence frequency analysis
- Range analysis
- Light vehicle discrimination
- Future sensor concepts
- Technology transfer
- Student research

Footstep Measurement Summary

- Experimentally observed two components of human footsteps
 - Low frequency component (below 500 Hz-1 kHz) is generated by force normal to the surface.
 - High frequency component is generated by the tangential force. Frequency range depends on properties of the contacting surfaces and may be extend to the ultrasonic frequencies.
- Floor covering changes the footstep vibration signature.
- The low-frequency vibration component is reduced by walking “stealthily”.
- The high-frequency component increases the probability of footstep detection.
- Airborne signal attenuate less rapidly at higher frequencies than seismic signals leading to the use of microphones for footstep detection
- UM is patenting a high-frequency detection technology for footstep detection.

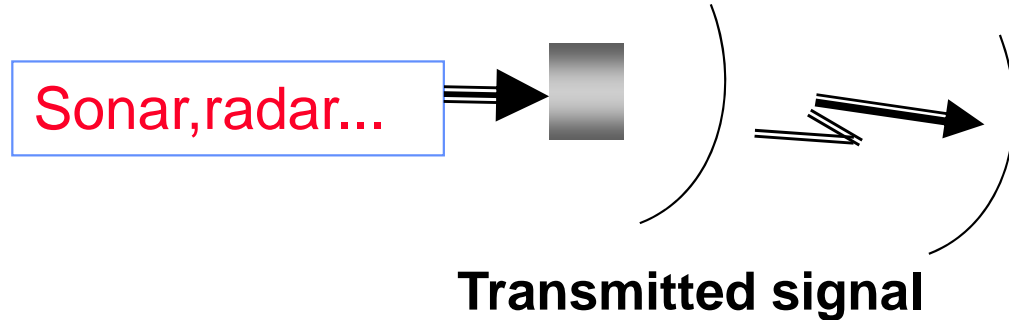
Ultrasonic Doppler SONAR

- Records motion of human body components (e.g. torso, head, arms, legs, etc)
- Signal is proportional to the cross section of the measured area (e.g. stronger signal for torso than arms/legs)
- May be useful for identifying a person by their whole body oscillations while walking.
- UM is patenting a high-frequency Doppler sonar detection technology.

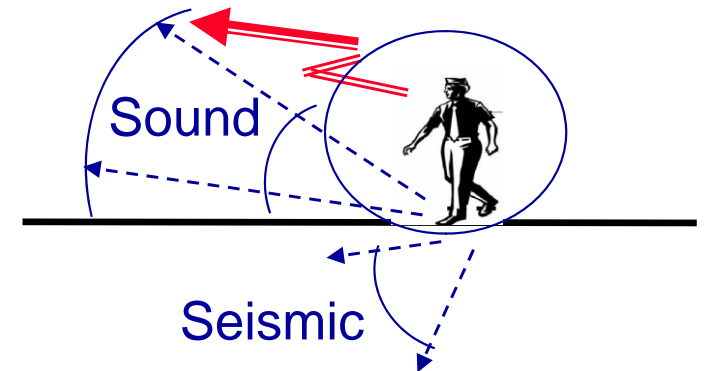
Human Passive and Active Signatures.

Active signatures

- Sonar and radar carrier frequency modulation by human motion (Doppler signature).



FM modulated reflected signal

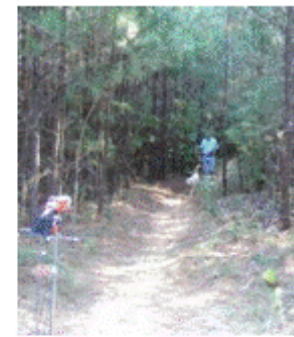


Passive signatures

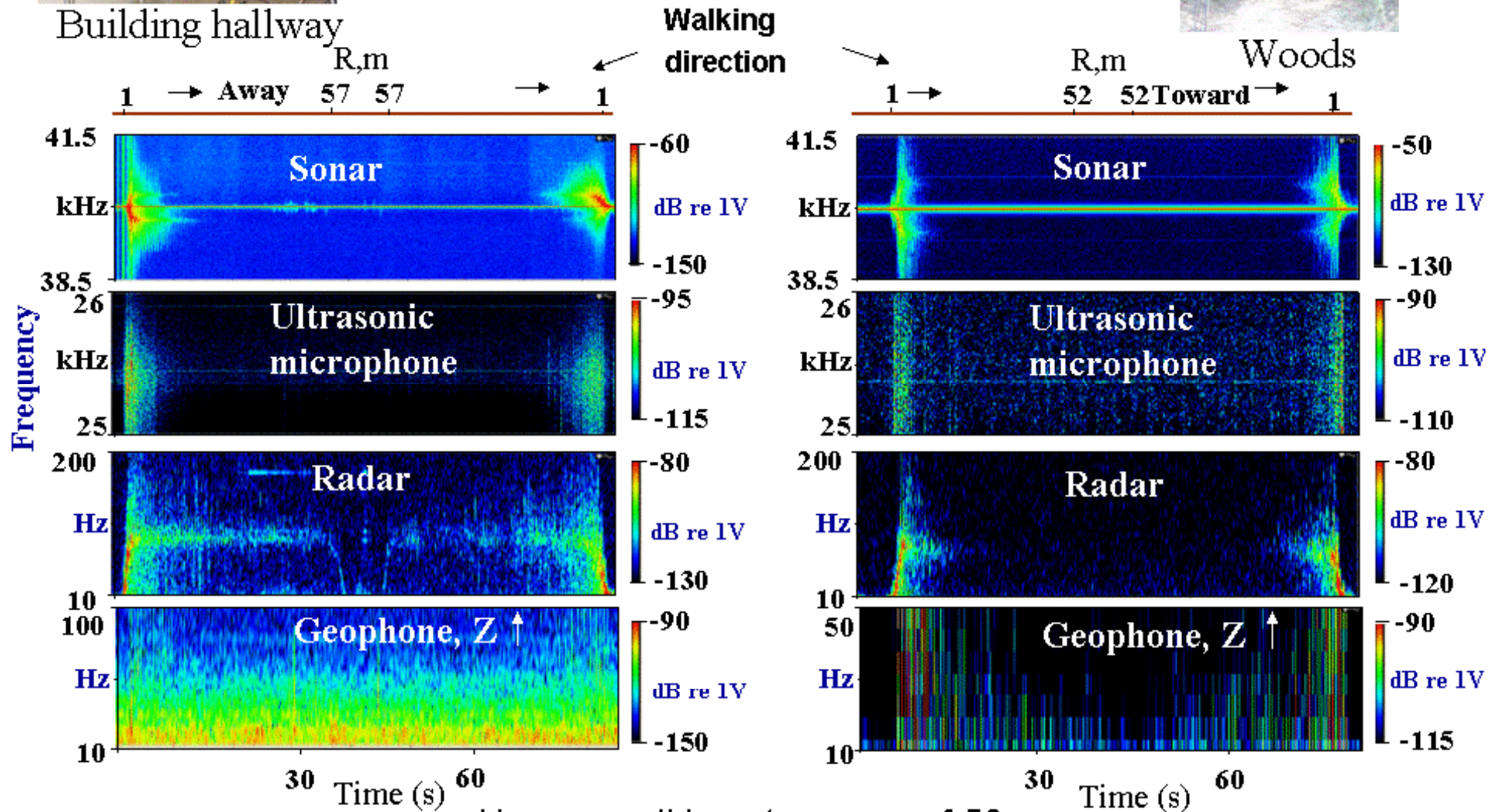
- Sound and seismic signals generated by human dynamic forces.
- Electromagnetic field modulation due to motion
- IR and video surveillance

Multi Modal Sensor Performance

NCPA, ARDEC, ARL



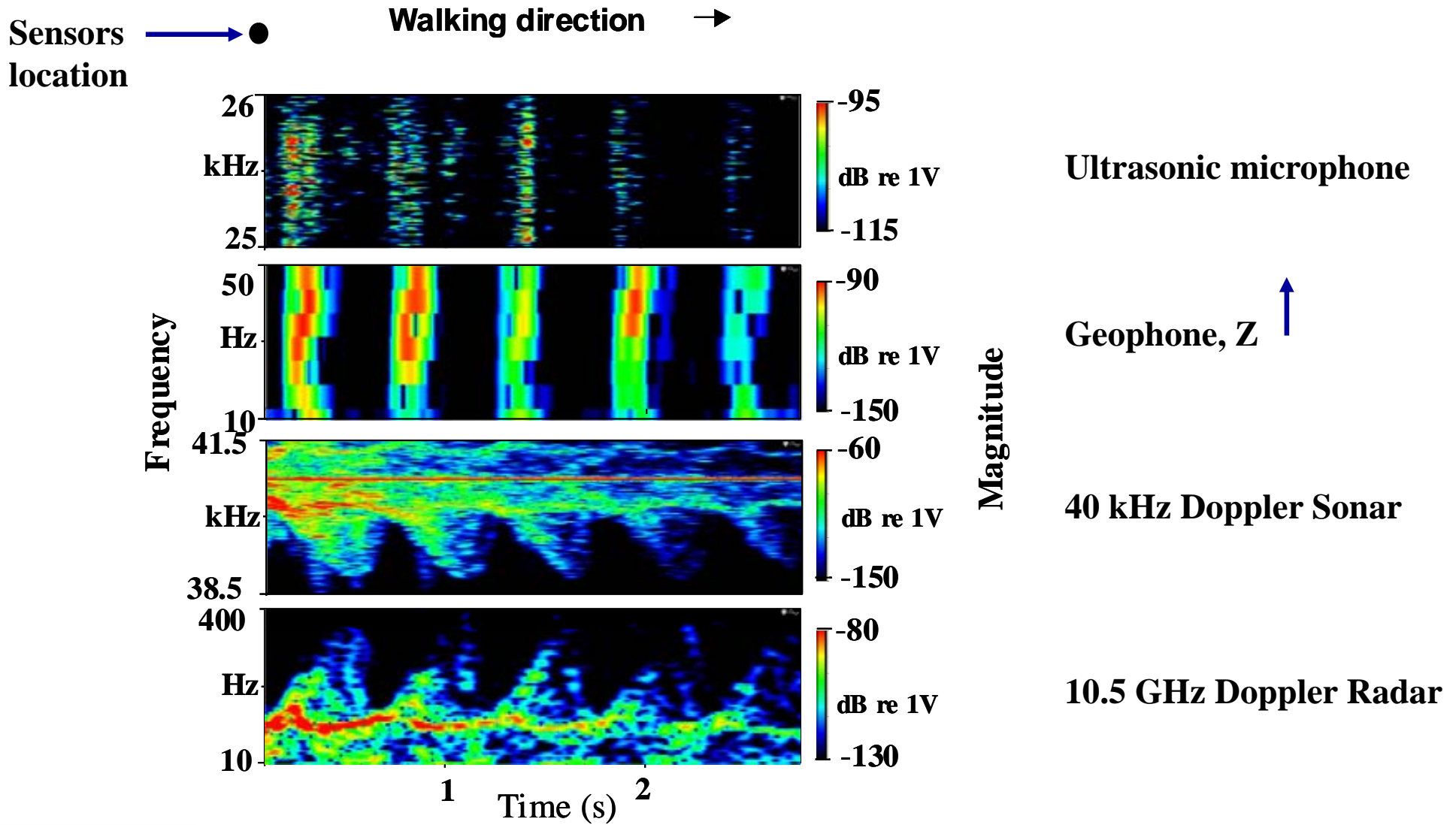
Building hallway



Human walking at a range of 50 m

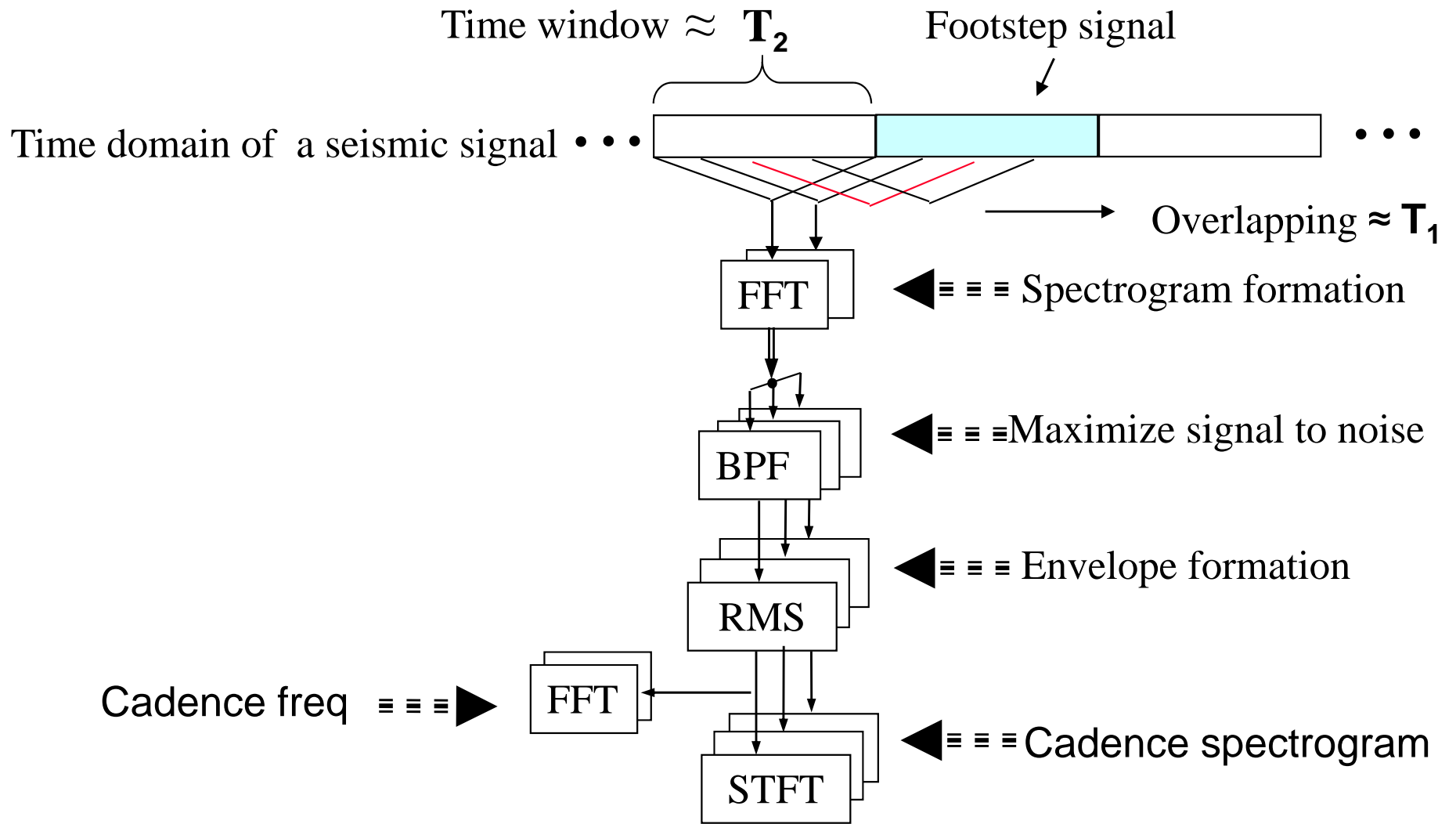
SR=96 kHz, FFT=16384

Synchronization between Human Passive and Doppler Signatures.



Walking on the ground

Processing Human Motion Signals



Technology Transfer

- **Technology transition to other Government agencies**
 - Dr James Sabatier served on ARL red team for Textron intruder detection systems
 - US Armament Research, Development, and Engineering Center (ARDEC) involved in signal processing research
 - Joint data collection exercises with ARDEC and ARL in October 2008 and with ARDEC in May 2009. These included tests at Yuma Proving Ground.
 - Proposals written to DHS and ONR – not funded
 - Related effort for light vehicle detection funded by ARL through ARO
 - Spin-off technology research program using “natural microphones” for obscured vehicle detection funded by US Army NVESD
 - UM-led Personnel Detection academic study group conducted first meeting at ARL in June 2009 on human, light vehicle, and tunnel detection. 49 participants from academia, industry, and government organizations. Included are 12 international participants. Future meetings are planned.
 - Dr Sabatier selected for an IPA assignment to ARL for research in human, light vehicle, and tunnel detection
 - Cadence frequency signal processing applied to acoustic signals from sperm whales for US Navy Space and Naval Warfare Systems Command
 - UM research program transitioned to US Army Armament Research, Development, and Engineering Center (Contract W15QKN-09-C-0163)
 - Formation of NATO Panel SET-158 “Disposable Multi-Sensor Unattended Ground Sensor Systems for Detecting Personnel,” Chair by James Sabatier, 2010-2012.

Technology Transfer

- Publications and Professional Meetings
 - Published in J. Acoust. Soc. Am.
 - Presented and published in proceedings
 - SPIE Defense and Security Symposium
 - Military Sensing Symposium on Battlefield Acoustics and Magnetic Sensing
 - IEEE International Conference on Technologies for Homeland Security
 - NATO Research and Technology Organisation Symposium on Battlefield Acoustic Sensing for ISR Applications.
 - Presented at the Acoustical Society of America
- Intellectual Property
 - Patent application filed for ultrasonic human detection technology and cadence frequency signal processing
 - Research disclosure filed on dynamic speckle sensing technology
 - UM spin-off company formed to develop ultrasonic human detection technology (SOAIR, LLC)

Publications

- **Peer-Reviewed**

- Alexander E. Ekimov and James M. Sabatier, “Vibration and sound signatures of human footsteps in buildings,” J. Acoust. Soc. Am., **120**(2), 762-768 (2006).
- Alexander E. Ekimov and James M. Sabatier, “Ultrasonic wave generation due to human footsteps on the ground,” J. Acoust. Soc. Am., 121(3), EL114-EL119 (2007).
- Alexander Ekimov and James M. Sabatier, “Human motion analyses using footstep ultrasound and Doppler ultrasound”, J. Acoust. Soc. Am., Vol.123, No 6, p. EL149 - EL154, (2008). Virtual Journal of Biological Physics Research - May 15, 2008, Volume 15, Issue 10
- Alexander E. Ekimov and James M. Sabatier, “Rhythmic Analysis of Human Motion,” Submitted Fall 2009, J. Acoust. Soc. Am.

Publications

- **Conference Proceedings**

- Alexander E. Ekimov and James M. Sabatier, “The velocity response to the human footstep force,” Proceedings of the Military Sensing Symposium on Battlefield Acoustic and Seismic Sensing, Magnetic and Electric Field Sensors, 9 pp. (2005).
- Alexander E. Ekimov and James M. Sabatier, “Broad frequency acoustic response of ground/floor to human footsteps,” Proc. SPIE, Vol. 6241, 202-209 (2006).
- Alexander E. Ekimov and James M. Sabatier, “Passive and active ultrasonic methods for human motion detection,” Proc. of the Military Sensing Symposium on Battlefield Acoustic and Seismic Sensing, Magnetic and Electric Field Sensors, 8 pp. (2006).
- Alexander E. Ekimov and James M. Sabatier, “Ultrasonic methods for human detection,” Proc. NATO Research and Technology Organisation Symposium on Battlefield Acoustic Sensing for ISR Applications, 8 pp. (2006).
- Alexander E. Ekimov and James M. Sabatier, “Passive ultrasonic method for human footstep detection,” Proc. SPIE, Vol. 6562, DOI: 10.1117/12.716899 (2007).

Publications

- **Conference Proceedings (Continued)**

- Alexander E. Ekimov and James M. Sabatier, “Evaluation of the range of human footstep detection,” Proc. Military Sensing Symposium on Battlefield Acoustics and Magnetic Sensing (2007).
- Alexander Ekimov and James M. Sabatier, “Human detection range by active Doppler and passive ultrasonic methods,” Proc. SPIE Defense and Security Symposium (2008).
- James M. Sabatier and Alexander Ekimov, “Range limitation for seismic footstep detection,” Proc. SPIE Defense and Security Symposium (2008).
- Alexander E. Ekimov and James M. Sabatier, “Detection and analysis of broadband acoustic signatures from walking humans in quiet and noisy environments,” Proc. Military Sensing Symposium on Battlespace Acoustics and Magnetic Sensing (2008).
- James M. Sabatier and Alexander Ekimov, “A Review of Human Signatures in Urban Environments Using Seismic and Acoustic Methods,” Proc. IEEE International Conference on Technologies for Homeland Security (2008).
- Alexander E. Ekimov and James M. Sabatier, “Orthogonal sensor suite and the signal-processing algorithm for human detection and discrimination,” Proc. SPIE Vol. 7303, DOI 10.1117/12.818823 (2009).

Conference Presentations

- **Acoustical Society of America**

- James M. Sabatier and Alexander Ekimov, “Vibration signature of human footsteps on the ground and in buildings,” J. Acoust. Soc. Am. 118, 2021 (2005).
- Alexander Ekimov and James M. Sabatier, “Adaptive mechanical model of human footsteps,” J. Acoust. Soc. Am. 119, 3390 (2006)
- Alexander Ekimov and James M. Sabatier, “Ultrasonic signatures of human motion,” J. Acoust. Soc. Am. 121, 3115 (2007)
- Alexander E. Ekimov and James M. Sabatier, “Directivity pattern of footstep sound at ultrasonic frequencies,” J. Acoust. Soc. Am. 122, 3061 (2007)
- Alexander E. Ekimov and James M. Sabatier, “Human Recognition by active and passive acoustic signatures,” J. Acoust. Soc. Am. 123, 3725 (2008).
- James M. Sabatier and Alexander E. Ekimov, “Orthogonal acoustic sensor package for human detection in quiet and noisy environments,” J. Acoust. Soc. Am., 124, 2508 (2008)

Conference Presentations

- **Acoustical Society of America**

- Asif Mehmood, Paul Goggans, and James Sabatier, “Instantaneous frequency analysis of ultrasound Doppler signal using Bayesian probability,” J. Acoust. Soc. Am. 125 2537 (2009).
- Alexander E. Ekimov and James M. Sabatier, “Human detection algorithm for seismic and ultrasonic detectors,” J. Acoust. Soc. Am., 124, 2499 (2008).
- Christopher L. Peters, Vyacheslav Aranchuk, James M. Sabatier, “Motion Analysis of an Oscillating Target Using Laser Speckles, Mid-South Chapter of the Acoustical Society of America Meeting, Conway, AR, March 6-7, 2009.
- Natalia Sidorovskaia, Philip Schnexnavder, Alexander Ekimov, James Sabatier, George E. Ioup, and Juliette W. Ioup, “Rhythmic analysis of sperm whale broadband acoustic signals,” J. Acoust. Soc. Am. 125, Issue 4, 2738 (2009).

- **ARL Study Group on Detection of Humans, Light Vehicles, and Tunnels**

- Alexander Ekimov and James M. Sabatier, “Human motion characterization,” Human, Light Vehicle and Tunnel Detection Study Group, Army Research Laboratory, June 16-17, 2009

Student Participation

- Five graduate students:
 - Asif Mehmood, PhD awarded in Electrical Engineering, dissertation completed entitled “Human Motion Detection using Ultrasound Doppler Vibrometer and Bayesian Model Selection”
 - Morris Mitchell, MS candidate in Physics
 - Christopher McNeil, MS awarded in Physics
 - Randy Ware, MS awarded in Electrical Engineering
 - Christopher Peters, MS awarded in Physics
- Three undergraduate students:
 - Celeste Sabatier completed a senior thesis in Physics entitled “Studying the Harmonic Motion of the Human Body via Ultrasonic Motion Detector and Ultrasonic Doppler Vibrometer.”
 - Tatsiana Aranchuk, BS in Electrical Engineering
 - Bradley Stroud, BS in Physics, University of Central Arkansas
- Two high school students (summer research projects):
 - Julia Chang (Mississippi School for Math and Science)
 - William Panlener (Mississippi School for Math and Science)

Summary

- Ultrasonic human detection technologies demonstrated under field conditions
- Rhythmic analysis of human motion signatures showed the equivalence of fundamental (cadence) frequency for signals from orthogonal sensors.
- Application of orthogonal sensors and common signal processing algorithms extended the distance of human detection.
- Strong technology transfer effort involves other research organizations
- Research has been transitioned to other Army agencies and has potential for the commercial market