

PAW PRINT GAIT ANALYSIS IN RATS WITH SPINAL CORD INJURY

APPROVED BY SUPERVISORY COMMITTEE

Committee Chairperson's Name MALU TANSEY

Malu Tansey, Ph.D.

Committee Member's Name KEITH TANSEY

Keith Tansey, M.D., Ph.D.

Committee Member's Name SHUXIN LI

Shuxin Li, M.D., Ph.D.

Committee Member's Name MARIO ROMERO-ORTEGA

Mario Romero-Ortega, Ph.D.

PAW PRINT GAIT ANALYSIS IN RATS WITH SPINAL CORD INJURY

by

GEORGE ZINKHAN

DISSERTATION

Presented to the Faculty of the Medical School

The University of Texas Southwestern Medical Center at Dallas

In Partial Fulfillment of the Requirements

For the Degree of

DOCTOR OF MEDICINE WITH DISTINCTION IN RESEARCH

The University of Texas Southwestern Medical Center at Dallas

Dallas, Texas

MAY, 2007

ACKNOWLEDGMENTS

I thank my mentor, Dr. Keith Tansey, for his willingness to teach and his infectious passion for testing new ideas.

I thank my committee chairperson, Dr. Malu Tansey, and my committee members, Dr. Shuxin Li and Dr. Mario Romero-Ortega, for their time and constructive feedback.

I thank Dr. McPhaul for his dedication to facilitating medical student research through the Summer Medical Student Research and the Distinction in Research Programs.

I thank my collaborators Dr. Barry Botterman and Dr. Herbert Hagler for their advice in developing software for paw print analysis.

I thank Myresa Hurst for keeping the lab running smoothly and for her help in caring for the animals.

I thank Paula Huntington for her work on analysis of results from the DigiGait study.

I thank Binh Nguyen for building the CatWalk during his summer research project.

I thank my wife, Erin Zinkhan, and the rest of my family for their constant encouragement and constructive feedback.

Copyright
by
George Zinkhan

ABSTRACT

PAW PRINT GAIT ANALYSIS IN RATS WITH SPINAL CORD INJURY

GEORGE ZINKHAN

The University of Texas Southwestern Medical Center at Dallas, 2007

Supervising Professor: Keith Tansey

Recovery of locomotion following spinal cord injury (SCI) in the rat may be tracked by different approaches to gait analysis, such as open field locomotor scoring and automated paw print analysis. Until recently, both of these approaches allowed the animal to walk at a self-selected speed. To further characterize gait recovery, the DigiGait system was developed to obtain video of paw prints through a clear plastic treadmill that may be set at various speeds. The DigiGait allowed the variability in gait parameters due to changes in speed to be characterized and controlled to mitigate effects of speed on these parameters. The purpose of the first study presented in this paper was to conduct a small preliminary study that uses the DigiGait with the rat model and to further characterize gait at multiple standardized treadmill speeds.

A minimum treadmill speed of 10 cm/s was determined for use with the Long-Evans rat because at this speed, the rats were able to run without frequent acceleration or deceleration and it was attainable by all of the subjects by two weeks after SCI.

Statistical analysis of the data was ongoing during the writing of this paper, and as a result the focus in this paper was to examine trends in the data. With increasing treadmill speed from 10 cm/s to 25 cm/s, hindlimb stride length and stride frequency were increased in injured animals by 1.3-fold and 1.9-fold, respectively; whereas, stance width was unaffected, and stance/swing ratio and hindlimb shared stance time were decreased by 1.6-fold and 2.4-fold, respectively. This study showed that certain gait parameters vary with a change speed. This variability may be reduced in future studies of gait parameters by using standardized speeds.

The second study was performed three years prior to the DigiGait study. At that time, results from the DigiGait system had not yet been published. A device called the CatWalk was available for acquiring video of rats crossing a clear walkway. Analysis software for the CatWalk existed but was not available commercially. The purpose of the second study was to design software that partly automates paw print analysis of video of rats using the CatWalk. The goal of this software was to measure the same gait measures as the CatWalk analysis software. These parameters included stride length, stance width, paw print area, swing duration, and interlimb coordination. The Java computer programming language was used to incorporate open-source image analysis tools available through the National Institution of Health's (NIH) ImageJ, which is a set of Java-based plug-ins. The software was able to analyze sequential series of frames that were extracted from video clips by third party software. A proof of concept study showed that this paw print analysis software was able to identify each paw in an image as left hindpaw, right hindpaw, left forepaw, or right forepaw. The software was then able to automatically measure and save the contact area of paws and the distance between

paws. Additional gait parameters may be added to this software in the future. Both of these studies contributed to the overall goal of better characterizing the efficacy of treatments that lead to improved locomotor function after SCI in the rodent model and that eventually may be used for human subjects with gait dysfunction due to SCI.

TABLE OF CONTENTS

LIST OF FIGURES 9

LIST OF TABLES 10

LIST OF DEFINITIONS 11

CHAPTER ONE—INTRODUCTION 12

CHAPTER TWO—THE FIRST SET OF STUDIES 17

CHAPTER THREE— THE SECOND SET OF STUDIES 30

CHAPTER FOUR—CONCLUSIONS AND RECOMMENDATIONS 39

BIBLIOGRAPHY 42

APPENDIX 46

VITAE 63

LIST OF FIGURES

FIGURE ONE.....	14
FIGURE TWO	15
FIGURE THREE	24
FIGURE FOUR	25
FIGURE FIVE	26
FIGURE SIX.....	27
FIGURE SEVEN	28
FIGURE EIGHT	29
FIGURE NINE.....	29
FIGURE TEN	33
FIGURE ELEVEN.....	34
FIGURE TWELVE.....	36
FIGURE THIRTEEN.....	37
FIGURE FOURTEEN	37

LIST OF TABLES

TABLE ONE	23
TABLE TWO.....	23

LIST OF DEFINITIONS

SCI – Spinal Cord Injury

BBB – Basso, Beattie, and Bresnahan scale

PBS - Phosphate-buffered saline

PFA - Paraformaldehyde

ROI – Region of interest

`/**` or `//` – Beginning of comments in Java computer language

`*\` – End of comments in Java computer language

CHAPTER ONE

Introduction

PAW PRINT GAIT ANALYSIS IN RATS WITH SPINAL CORD INJURY

Spinal cord injury (SCI) affects an estimated 250,000 people in the United States with an annual incidence of 11,000. It affects every age group with an average age at injury of 38 years old.¹ Morbidity and mortality depends on the severity and location of injury. One important factor in determining morbidity is gait disturbance through partial or complete paraplegia or quadriplegia. To assess the effect of locomotor training, the recovery of gait after SCI has been studied in human and rodent models and has been found to be amenable to treatment with body weight support treadmill training using the Lokomat and other training methods that encourage activity-dependent neural plasticity.^{2,3,4}

To characterize such treatments, the approaches to gait assessment may be grouped into three main areas: task performance measures, scoring of open field locomotion, and paw print analysis. Task performance measures, which involve tasks, such as walking along a rope, remaining on top of rotating rod, and crossing a horizontal ladder, provided objective results about the general severity of injury but do not allow for analysis of specific parameters like joint movement, weight support, limb coordination, paw clearance, and paw angle.^{2,4,6,7} Functional deficits also may not allow animals to perform these tasks.⁸ Scores of the open field locomotion like the Basso, Beattie, and Bresnahan scale (BBB), ranged from 0 to 21. A score of 0 denoted no observable hindlimb movement; whereas, a BBB score of 21 described the stepping pattern of an

uninjured rat. One major crossover point in the scores occurred at a score of ten at which occasional weight supported plantar steps are first observed. At scores lower than nine, measurement of movement at the hip, knee, and ankle was classified as none, slight, or extensive, which correspond to 0%, $\leq 50\%$, and $> 50\%$, respectively. Another major crossover point in the BBB scale occurred at a score of 12 at which forelimb-hindlimb coordination is first scored as occasional. The BBB score classified frequency of coordination as never, occasional, frequent, and consistent, which correspond to values of 0%, $\leq 50\%$, 51-94%, and 95-100% of normal, respectively. Paw position at initial contact and lift off was also assessed as inverted, everted, and parallel to the long axis of the rat. The advantage of the BBB was the short amount of time required to score each rat – four minutes. The disadvantages of the BBB included requiring trained, careful observers, lacking measurement of specific gait parameters, and having a qualitative, non-linear scale.⁹

Paw print analysis included non-automated systems or automated systems, like the CatWalk and DigiGait.^{10,11,12} These systems produced objective results and are able to examine more gait parameters that allow the gait disturbance to be better broken down into stride length, shared hindlimb stance, and its other components. These systems may be time-consuming and costly, and they were not able to analyze motor deficits of severely impaired animals that are unable to ambulate.¹³

Despite its drawbacks, paw print analysis provided the greatest amount of objective information. One method used for paw print analysis was inking the animal's hind paws and having the animal run on a strip of paper. Gait parameters, such as limb rotation, stride length, and base of support, were then measured on the paper by hand.⁶

Another approach involved filming animals from below with a digital camera and then examining the stepping either with a computer frame by frame or with software that automatically identifies paw contact and uses this information to compute specific gait parameters. The CatWalk method developed by Hamers et al. involved acquiring video of rats walking on a surface that illuminated paw-floor contact points (Figure 1). The CatWalk analysis software measured stride length, stance width, paw print area, swing duration, and interlimb coordination. It was also reported to have the potential of measuring force of contact.

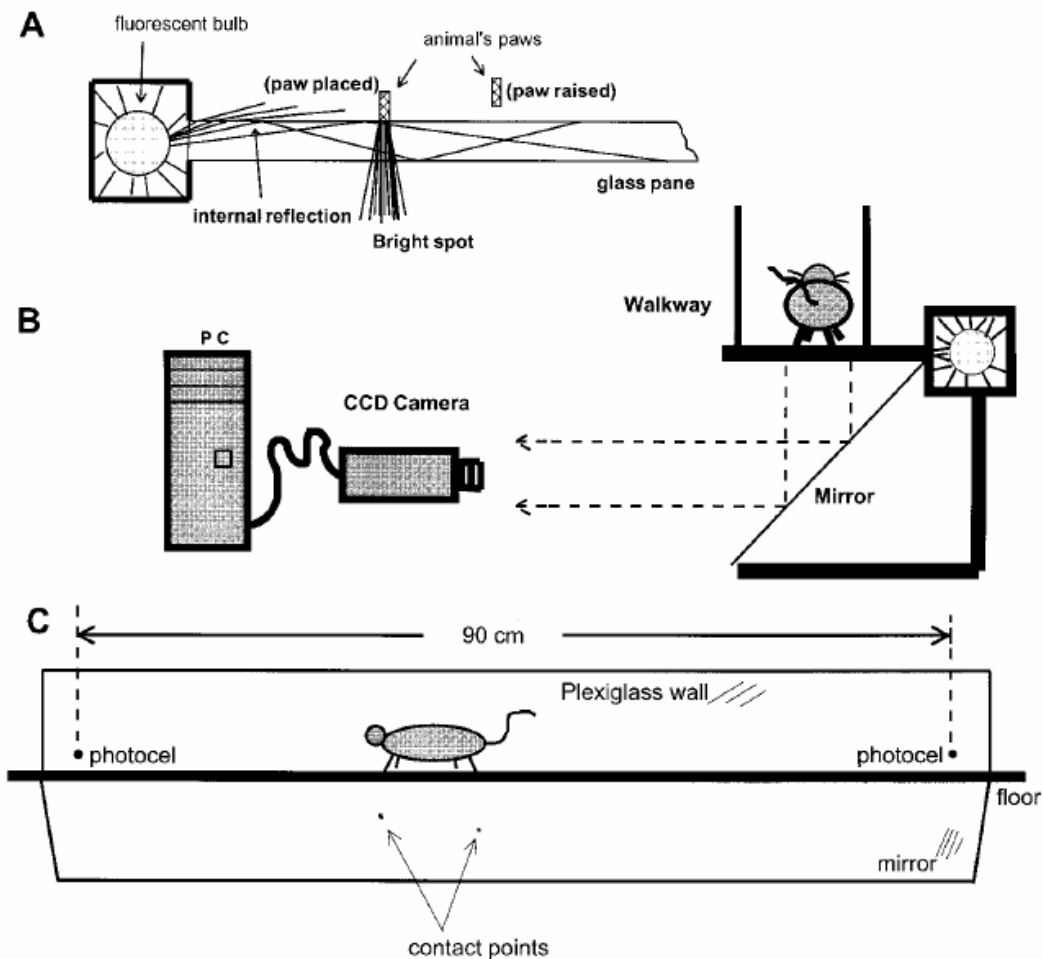


Figure 1. Diagram A shows how a paw making contact causes light to escape the walkway. Diagram B shows how the video camera is able to capture video of the rat walking on the surface. Diagram C shows how the contact points are reflected in the mirror so that video may be recorded. Reprinted from Hamers et al.

The CatWalk approach and many of the other methods for examining gait in animals allowed the animals to self-select a speed to walk. To more fully characterize gait, the DigiGait system (Mouse Specifics, Inc.) varied treadmill speed. It used a clear plastic treadmill with a video recorder underneath (Figure 2). The treadmill speed was varied from 0 cm/s to 99.9 cm/s in increments of 0.1 cm/s. The DigiGait also used software that automatically determined which paw is stepping as the animal walks and then determined many different gait parameters.

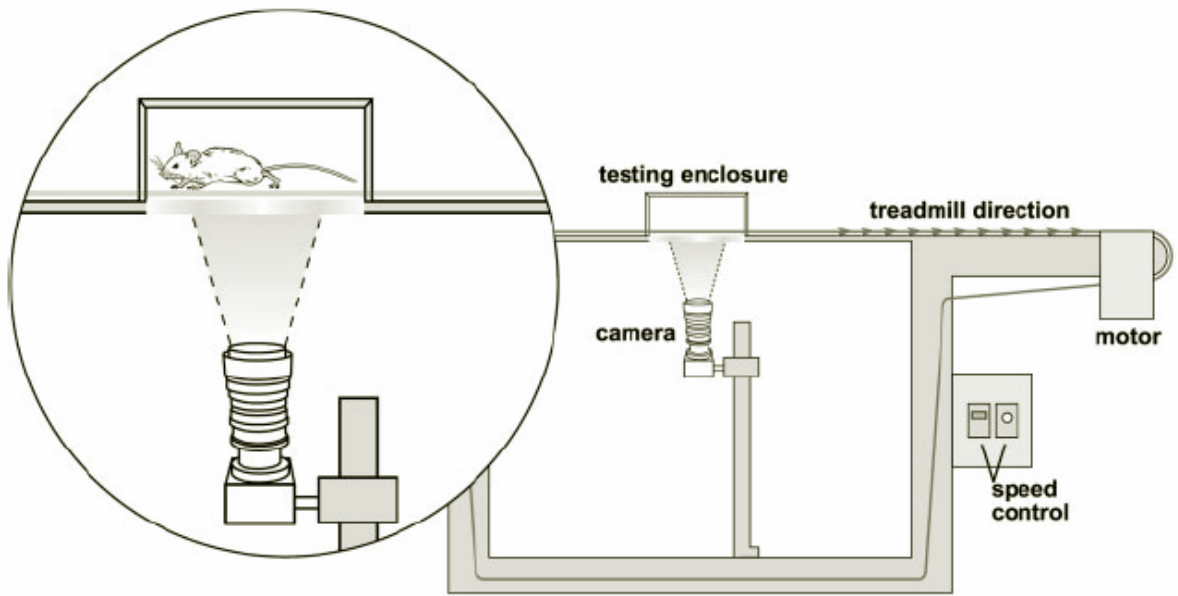


Figure 2. video of the rat is obtained from below a clear treadmill. The animal is enclosed so that he cannot wander off of the treadmill and to encourage the animal to walk in a straight line. The treadmill speed is varied by a control console. Reprinted from Wooley et al.

The current project aimed to develop a protocol for using the DigiGait in rats with SCI and to examine the inter- and intra- rat variability in hindlimb stride length and its coefficient of variability, hindlimb stance width, stride frequency, hindlimb shared stance time, and stance-to-swing ratio. The preliminary analysis examined the relationship of these gait parameters over recovery time and at different BBB scores. The change in BBB scores during recovery time was also observed. One major purpose of this small

preliminary study was to use its results to perform power analysis to determine sample size and the number of runs necessary for future studies.

The goal of the second study presented in this paper was to develop software that analyzes the digital video of rats walking on the CatWalk walkway. This software aimed to identify each paw as left, right, hind, and fore automatically. Once the paws were recognized by the software, gait parameters are measured, and the data was saved to a tab-delimited text file. One of the most important abilities of this software was to allow the user to confirm the results of its analysis. During development and testing, multiple videos of a Long-Evans rat crossing the CatWalk runway were analyzed.

CHAPTER TWO

Automated Paw Print Gait Analysis with the DigiGait

BACKGROUND

The DigiGait was a relatively new system at the time of this study. It had been used in mice models for Down syndrome, motoneuron disease, ethanol sensitivity, and SCI.^{12,15,16,17} Stride, stance, and swing time decreased in relation to increasing treadmill speed using the DigiGait in mice with motoneuron disease.¹⁵ Mice with transgenically inhibited Nogo-66 receptor, which was hypothesized to limit axonal growth in SCI, were shown to have an increase in stride length and a decrease in stride length variability measured by the DigiGait when compared to wild-type mice four weeks after dorsal over-hemisection spinal cord injury.¹⁷

Previous studies have shown variability in the self-selected speed used by different strains of uninjured rats. Sprague-Dawley rats were reported to walk at 10 cm/s, Lewis rats at 17.5 cm/s, and Wistar rats at 46 cm/s.^{14,7,18} Normal Long-Evans rats have been demonstrated to run at speeds from 72 to 186 cm/s when trained to seek food across as fast as possible across a central 60-70 cm of runway. In the same study, rats with contusion of the spinal cord had a reduced speed of 27 to 129 cm/s.¹⁰ This study showed the wide variability that may occur when rats self-select their speeds. One of the advantages of the DigiGait was its ability to determine what gait parameters changed with changing speed and its ability to control the variability in the parameters dependent on speed. The current study sought to determine speeds that Long-Evans rats may consistently obtain on the DigiGait so that standardized speeds may be used in future studies.

METHODS

Animals:

Young adult female Long-Evans rats (203-216 g, n=10) were used for this study. All procedures relating to animals in this study were approved by the Institutional Animal Care and Use Committee at the University of Texas Southwestern.

Surgical procedure and contusion:

All surgeries were performed by Dr. Keith Tansey, while anesthesia and assistance were provided by this author. The rats (n=10) were anesthetized with ketamine (75 mg/kg) and medetomidine (0.5 mg/kg). After a bilateral laminectomy of the T9 vertebra, the exposed vertebral column was stabilized by clamping the T8 and T10 vertebral bodies with Adson forceps on a plastic stage. Contusion of the exposed T9 spinal cord was performed using a commercially available Infinite Horizon spinal cord injury device (Precision Systems and Instrumentation, Lexington, KY). The actual contusion force varied from 182 to 227 kdyn. These forces were selected so that the animals would functionally recover enough to record them stepping on the treadmill.¹⁹ After surgery, animals were given the reversal agent, Antisedan (5.0 mg/mL). Sterile saline 10 mL subcutaneously was given immediately after surgery, and 3.0 mL was given as needed during the post-operative course. Buprenex (0.3 mg/mL) was given on post-operative day 1, and Baytril (2.27% solution) was given on post-operative days 1, 2, and 3. Postoperative care involved manual expression of bladders twice a day until bladder function returned.

Open field testing:

The rats were assessed at postoperative day 1 and then weekly with the Basso, Beattie, Bresnahan (BBB) locomotor rating scale.⁹ To assess the score, rats were observed in a large stainless steel tub for four minutes.

DigiGait-automated gait analysis:

Digital video from underneath rats walking on a clear motorized treadmill was obtained at 80 Hz with a high-speed video camera (DigiGait Imaging System, Mouse Specifics, Boston, MA). Afterwards, the DigiGait imaging software was used to select a clip during which the rat was moving at a relatively constant speed (e.g. no acceleration or deceleration to move faster or slower than the treadmill) and in a straight line. A minimum of five consecutive strides was obtained for each clip, and at two clips were obtained for each rat at each treadmill speed. Collection of at least five strides has been shown to be sufficient for analysis with fewer than five strides increasing variability and more than five strides not greatly reducing variability.¹⁵ Rats were recorded on the treadmill before contusion (n=10). All of these rats underwent contusion injury as described previously. Three rats were lost to perioperative complications. One rat was uninjured with a BBB score of 21 on the first post-operative day and therefore the animal was euthanized. As a result, six rats were recorded weekly from two to six weeks postoperatively. The treadmill was started at 0.0 cm/s, and the speed was increased by increments of 5 cm/s every thirty seconds until the animal was unable to maintain the speed due to tripping or turning in the opposite direction while the treadmill was running.

Increments of 5 cm/s were used partly because the rats were usually able to manage acceleration and deceleration changes of 5 cm/s without slipping. This increment was chosen partly to reduce the time required for acquiring video so that the rats did not become fatigued. Increments of 5 cm/s were also chosen because the rats were able to maintain balance and keep up with the treadmill as its speed was increased by these increments. The animals were given 30-second to one-minute rests in between trials. Testing was continued for no longer than fifteen minutes to limit inconsistent stepping due to fatigue. Through this method, each rat attained different maximum treadmill speeds; however every rat tested was able to run at 10 cm/s pre-operatively and at every postoperative time point. This speed was the minimum possible treadmill speed at which the rats were able to consistently maintain a constant speed. At lower treadmill speeds, the rats frequently stopped walking, which made it difficult to obtain video of sequential steps. The maximum treadmill speed for analysis at each time point was selected post hoc so that the largest possible sample size could be used to compare gait parameters during recovery time and for each BBB score (Table 2). By selecting the maximum speeds in this way, variability in gait parameters due to differences in speed was also reduced. The maximum speeds selected for analysis were 15.0 cm/s at postoperative week 2; 20 cm/s at postoperative weeks 3, 4, and 5; and 25.0 cm/s at week 6.

DigiGait analysis software was used to extract images of the video and identify each paw in the images. Gait parameters were exported to an Excel spreadsheet. The DigiGait measured many parameters including: swing duration, stance duration, braking duration, propulsion duration, stride duration, swing-to-stance ratio, stride frequency, paw area, hind limb shared stance, and stance factor. The DigiGait also measured stride

length, stance width, paw angle, and stance and their variability. For this study, the following parameters were assessed:

1. *Stance-to-swing ratio.* Swing duration is defined as the time during which a paw is not in contact with the surface. Stance duration is the time during which the paw is in contact with the surface. Stance duration is composed of braking and propulsion phases.
2. *Hindlimb stride length.* This length is the distance that a paw traverses during a given stride.
3. *Hindlimb stance width.* Hindlimb stance width in rats is important in rats for maintaining a base of support. Normally, the stance width of the forelimbs is less than the hindlimbs in rats.¹⁰
4. *Hindlimb stride length coefficient of variability:* This parameters are calculated from the equation: $(100 \times \text{standard deviation})/\text{mean value}$.
5. *Hindlimb stride frequency.* This parameter is the number of times per second that a paw takes a complete stride.
6. *Hindlimb shared stance.* This parameter is the time both hind paws contact the surface.

Spinal cord preservation:

Myresa Hurst performed the spinal cord preservation. Each animal that underwent gait analysis was anesthetized with sodium pentobarbital and transcatheterially perfused with 0.1 M phosphate-buffered saline (PBS, pH 7.4), followed by 4% paraformaldehyde (PFA) in PBS. A spinal cord section spanning from one spinal level

above to one spinal level below the contusion was removed and fixed in 4% PFA for 4 hours.¹⁹ The spinal cord was then placed in 20% Sucrose in a 4 °C refrigerator for 24 hours. The cord was removed from the sucrose and placed in a freezing Petri dish with dry ice and 2-methylbutane (ReagentPlus \geq 99%). The cord was stored in a -20 °C freezer for future slicing with the cryostat.

Statistical analysis:

Statistical analysis for this study was ongoing at the time this paper was written. Power analysis of the data was planned at a later time to ensure that when future studies are designed they have sufficient sample size and number of runs on the treadmill. The main focus of analyzing the data at this stage was to observe trends in gait parameters over time, at each BBB score, and at various treadmill speeds. Only limited, preliminary analysis of the data with ANOVA in Microsoft Excel 2003 was done to establish whether the null hypothesis was true for each comparison group. Significant differences between groups have not yet been established but such analysis was planned for the future.

RESULTS

BBB scores of every contused rat at every postoperative time point are recorded in Table 1. Preliminary analysis of the data showed average BBB scores of 13.3, 13.5, and 13.5 at 4, 5, and 6 weeks. The maximum speed was decreased postoperatively for all rats at the second week, and every rat either improved or maintained the same maximum speed from the second to third weeks (Table 2). However, after the third week, time after injury did not seem to be related to maximum speed. The maximum speed attained by

each rat preoperatively was obtained postoperatively by only one animal. Although maximum speed seemed to vary postoperatively, all rats were able to obtain at least 15 cm/s by postoperative week 2, at least 20 cm/s by the third postoperative week, and at least 25 cm/s by the sixth week.

Animal #	Post Op Day 1	Post Op Week 1	Post Op Week 2	Post Op Week 3	Post Op Week 4	Post Op Week 5	Post Op Week 6	
1	0	6	13	13	14	14	14	
2	Died of perioperative complications							
3	Died of perioperative complications							
4	0	9	13	14	14	14	14	
5	9	12	13	13	13	14	14	
6	21	Animal was euthanized due to lack of injury						
7	1	6	10	12	12	12	12	
8	3	7	14	14	14	14	14	
9	1	9	13	13	13	13	13	
10	Died of perioperative complications							

Table 1. BBB scores reached a plateau by post op week 5 for all rats.

Animal #	Uninjured	Post op Week 2	Post op Week 3	Post op Week 4	Post op Week 5	Post op Week 6
1	35	35	35	30	30	35
2	35	25	25	25	25	25
3	45	30	30	20	20	35
4	45	20	45	40	40	50
5	45	35	45	40	25	25
6	35	15	20	20	20	25

Table 2. Maximum speed attained varied between rats and for each individual rat.

Stride length increased 1.2-fold from the second to third week and then remained relatively constant at each time point. The stride length coefficient of variability remained relatively constant at each time point. Stride length increased an increase in BBB score. Hindlimb stride length and frequency increased 1.3-fold and 1.9-fold with an increase in treadmill speed from 10 cm/s to 25 cm/s. The stride frequency increased from 1.1 +/- 0.3 stride/s at 10 cm/s to 2.1 +/- 0.2 stride/s at 25 cm/s. Stance width also did not change significantly during the postoperative period and did not vary significantly with treadmill speed. Stance width shows a relationship to BBB score with a width of 1.9 +/-

0.3 cm at a BBB score of 10 and a width of 3.2 +/- 0.5 cm at a BBB score of 14 (Figure 3). Stance/swing ratio increased with increasing BBB score but decreased 1.6-fold with treadmill speed increasing from 10 cm/s to 25 cm/s (Figures 4). Hindlimb shared stance time was decreased 2.4-fold with increasing treadmill speed from 10 cm/s to 25 cm/s and did not change significantly with recovery time.

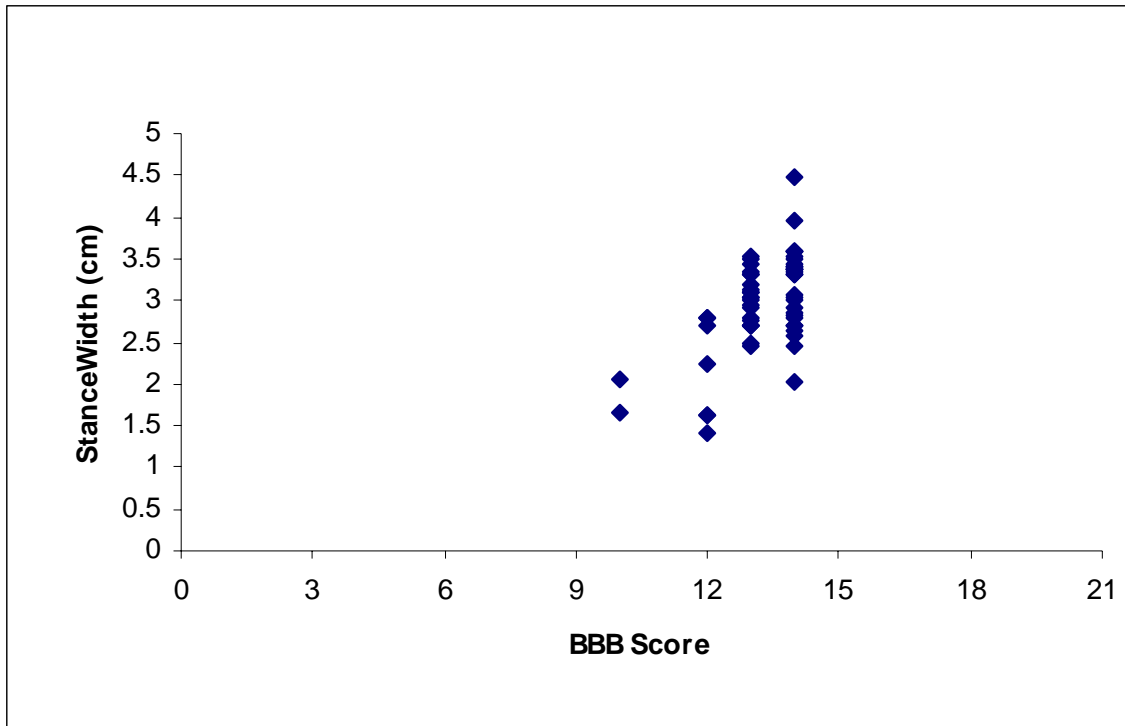


Figure 3. Stance width increases with BBB score.

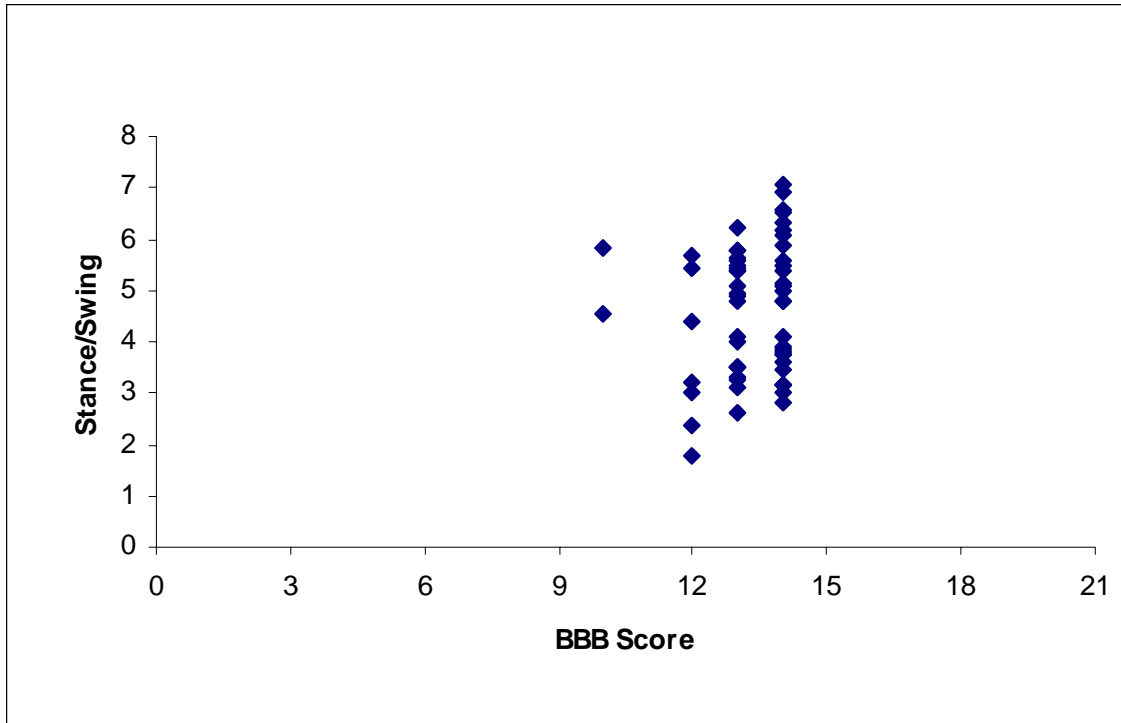


Figure 4. Stance/swing ratio increases with BBB score.

DISCUSSION

The BBB scores were consistent with scores previously reported for rats with intermediate contusion injury (199 to 379 kdyn) that had average BBB scores of 13.1, 13.5, and 13.7 at 4, 5, and 6 weeks, respectively, post-injury.²⁰ The maximum treadmill speed was decreased for all animals at the second postoperative week. The maximum speeds varied over time and did not show stepwise improvement with time for most of the animals. Maximum velocity alone did not seem to change in relation to recovery time. This finding suggested that it may not be necessary to run rats until they reach their maximum speed. Instead, it may be more efficient to select speeds that every rat is able to obtain so that the sample size is larger at each given speed analyzed and so that speed may be controlled for when analyzing gait parameters.

While some studies have reported changes in stride length after thoracic contusion of the spinal cord,^{10,11,21} other studies have reported no significant change in stride length after SCI.^{7,20} The current study suggested an initial increase in stride length from postoperative week 2 to postoperative week 3 followed by a plateau in stride length (Figure 5). The increase in stride length and stride frequency with increasing speed suggested that the rats both move their limbs more frequently and take longer steps as treadmill speed was increased from 10 cm/s to 25 cm/s (Figure 6). Stride length and frequency have been shown to increase in rodents as their speed increased.²²



Figure 5. Stride length increases with increasing treadmill speed.

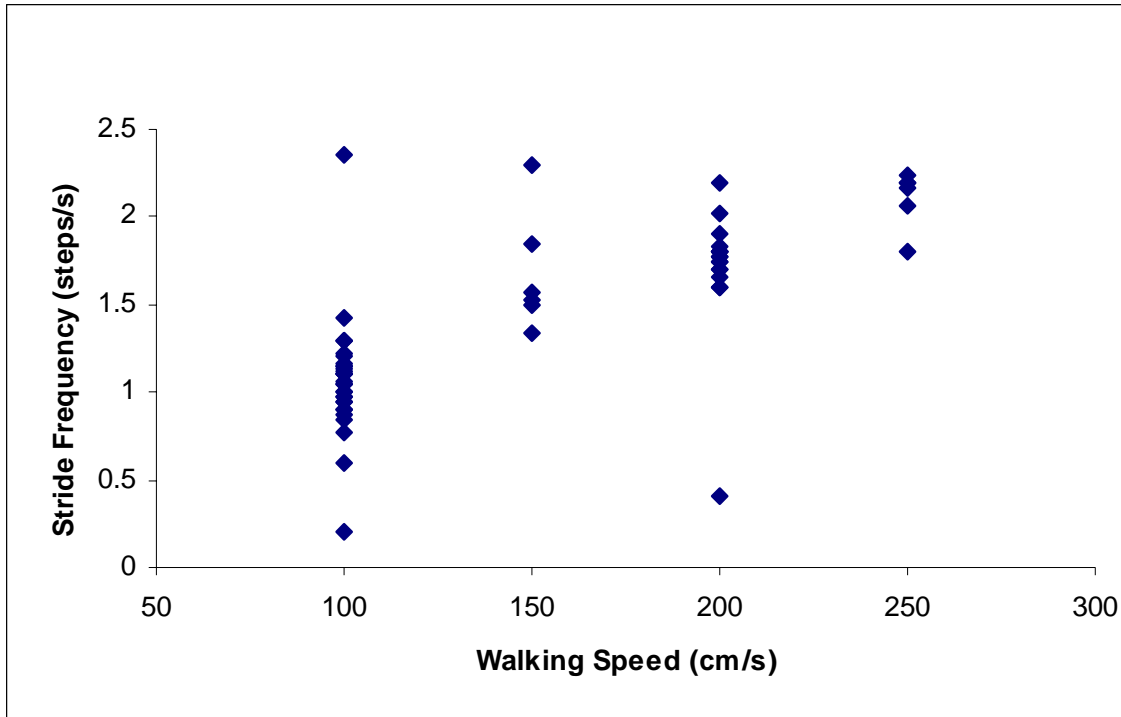


Figure 6. Stride frequency increases as treadmill speed increases.

The stance width did not seem to change over time in this study; however, the stance width did seem to be related to BBB score. The rapid plateau in BBB scores by the third week may explain this discrepancy. The relationship of stance width and BBB score was consistent with studies that showed stance width of the hindlimbs increased following contusion.^{7,11,20,21} Since stance width did not change with increasing treadmill speed, it was possible that the speeds analyzed thus far were not at a sufficient pace to disturb stability and caused a compensatory increase in stance width. In corroboration of this hypothesis, the stance width of the preoperative group was not significantly different from that of the post-operative group at 25 cm/s. Comparison of stance width at higher treadmill speeds obtained for both groups may show a broader stance.

The decrease in hindlimb shared stance time and stance/swing ratio with increasing treadmill speed suggested that the hindlimb was in the stance phase for a

relatively shorter period of time as the rat was required to move at a greater speed (Figures 7, 8). This result was congruent with the increase in stride frequency that was observed with increasing speed. In order for the rat to maintain a higher speed, it had to spend more time propelling itself forward rather than standing. As in the current study, the hindlimb shared stance time has been shown not to change significantly with intermediate contusion injury (Figure 9). In severe contusion injury (422-437 kdyn), however, hindlimb shared stance time has been shown to increase. One hypothesis for this result was that with more severe injury, less white matter was spared in the spinal cord and the reticulospinal tract, which is important in controlling swing and stance phases, may be affected.²⁰ Anatomical assessment of spinal cord injury will allow further exploration of this hypothesis.

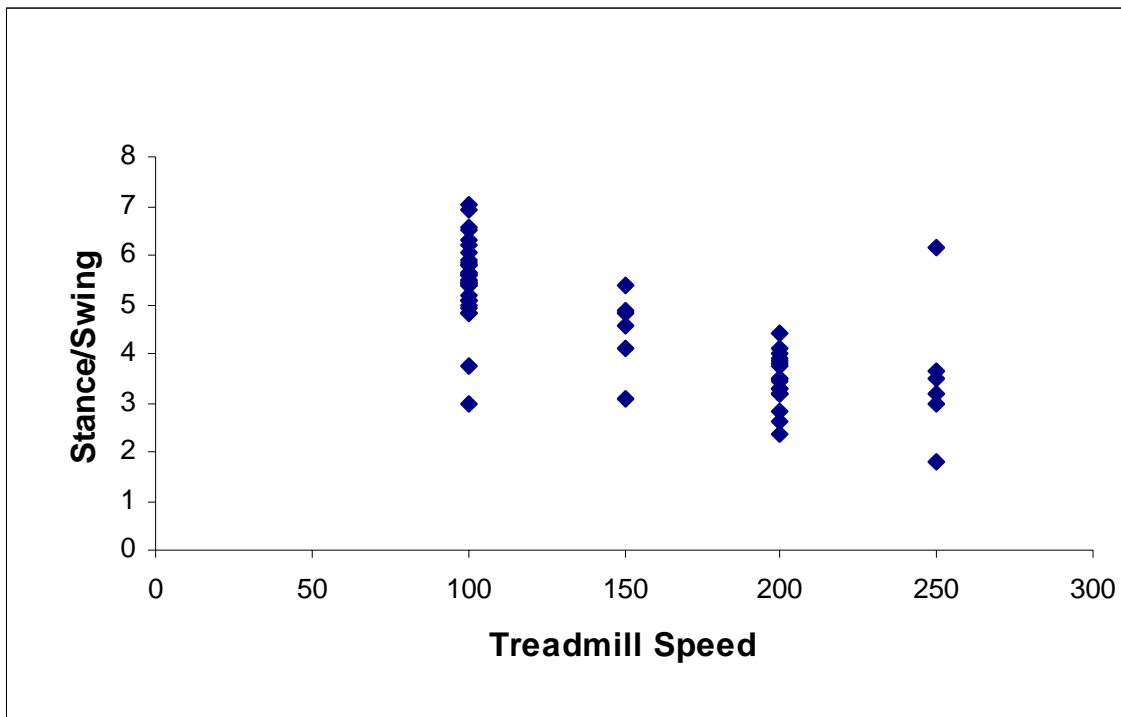


Figure 7. Stance/swing ratio decreases with treadmill speed.



Figure 8. Hind limb shared stance time decreased with increasing treadmill speed.

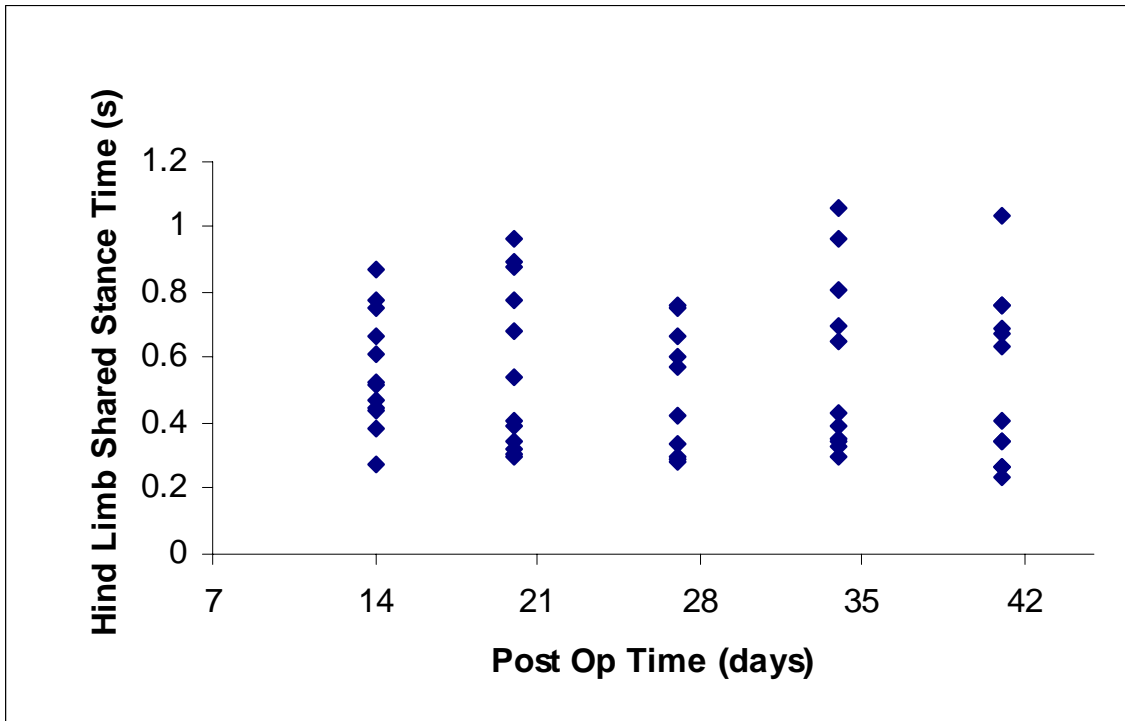


Figure 9. Hind limb shared stance time remained constant over time.

CHAPTER THREE

Semi-Automated Paw Print Gait Analysis Software

BACKGROUND

A variety of locomotor tests has been developed for studying deficits caused by SCI in rodents. The overall goal of these tests was to develop a fast and efficient way of tracking recovery. As discussed in the introduction, paw print analysis was a quantitative method for measuring gait parameters. With the development of inexpensive personal computers and digital video cameras, gait analysis has become more automated with software to replace the former method of measuring gait parameters either on a strip of paper over which a rat has run with paws dipped in ink or frame by frame on a computer. The CatWalk was one method of obtaining video of a rat stepping on an illuminated walkway (Figure 1).¹¹ The main goal of this study was to develop semi-automated software that analyzed video of rats walking across the CatWalk. Important qualities of such software included the ability to distinguish which paw is stepping, the ability to measure desired gait parameters, and the ability for a user to verify the program's output.

Identifying each paw from the others (e.g. left hind paw from the right forepaw) required comparing the relative position of each paw to one another. Identification was further complicated when only one or two paws are making contact with the surface. One solution to this problem was to define all footfalls on one side of the runway as left and all footfalls on the other side as right.

Gait parameters such as paw area, stride length, stance width, and paw rotation were important to assess in spinal cord injury recovery.^{9,11,23} Paw area may increase due to a paralyzed paw making complete contact with the walkway; whereas, an uninjured

paw usually contacts the floor with the front part only. Paw and toe dragging may also contribute to an increase in paw area.¹¹ To measure these parameters, the software must be able to identify the center of each paw. One reason the Java programming language was selected for writing this software was the National Institute of Health (NIH) hosts an open-source community that creates and shares Java plug-ins and macros for image processing and analysis. These tools are incorporated into the NIH ImageJ Software and may be accessed on the internet.²⁴

Since any automated system may not correctly identify a paw or measure a parameter perfectly every time, it was important that one step of the analysis involved a user checking the software. Such a method is used by the DigiGait system, in which the user reviewed the footfalls for each paw and deleted or connected any areas that were improperly identified as a paw stepping (Mouse Specifics Inc.).

METHODS

CatWalk method:

As described previously by Hamers et al., the CatWalk used a walkway with Plexiglas walls 8 cm apart and a floor, which measured 109 x 15 x 0.6 cm. Light from an encased white fluorescent bulb entered the glass floor at the long edge of the glass. The angle of the light was set up so that the light was internally reflected except at points where an object made contact with the floor. The contact caused the light to exit the floor and illuminate the contact area (Figure 1A). During use of the walkway, the room was kept dark. Underneath the walkway, a mirror placed at an angle of 45° under the floor

allowed a digital video camera setup on a tripod a few feet from the walkway to capture the light emitted by contact with the floor (Figures 1B, 1C).

The walkway was video recorded (Panasonic PV-GS120) at a frame capture rate of 60 Hz. In mice, capture rates of 30 Hz, 50 Hz, and 80 Hz have been used effectively to study treadmill speeds of 20 cm/s, 43 cm/s, and 25 cm/s, respectively.^{16,6,26} The video recorder was placed to maximize the length of walkway that was recorded, and with this video recorder, a 25 cm length of walkway was recorded. The video was downloaded to a personal computer (Intel Pentium 4 2.26 GHz and 516 MB RAM) by FireWire connection. The video was processed into individual frames, which are saved as bitmap files by third party software (HT Video Editor 6.1, Honest Technology).

Paw print analysis software:

Image analysis was begun by collecting user input. The start and end frame, the name of the text file in which the gait parameters were saved, and the identity of the initial paw (e.g. left hindlimb, etc.) were inputted (Appendix). If the first frame in a video clip had only one paw, the program would not have any frame of reference to identify the paw automatically. To solve this problem, the user was asked at the onset to input which of the four paws is making contact with the surface. The user was able to identify the paw because the outline of the rat was visible in the images. The edges of each paw and its toes were identified by using the ImageJ plug-in, Wand and PolygonRoi (Figure 10). Wand automatically outlined any bright areas. PolygonRoi gathered the x and y coordinates of the area outlined, called the region of interest (ROI). The x coordinates and y coordinates were saved in separate arrays. The value of the pixel at

each coordinate was set to 0 (pixels may have values between 0, which is black, and 255, which is white). All other pixel values were set to 255. The resulting black and white image was easier for the user to examine and determine whether paws are correctly being identified.

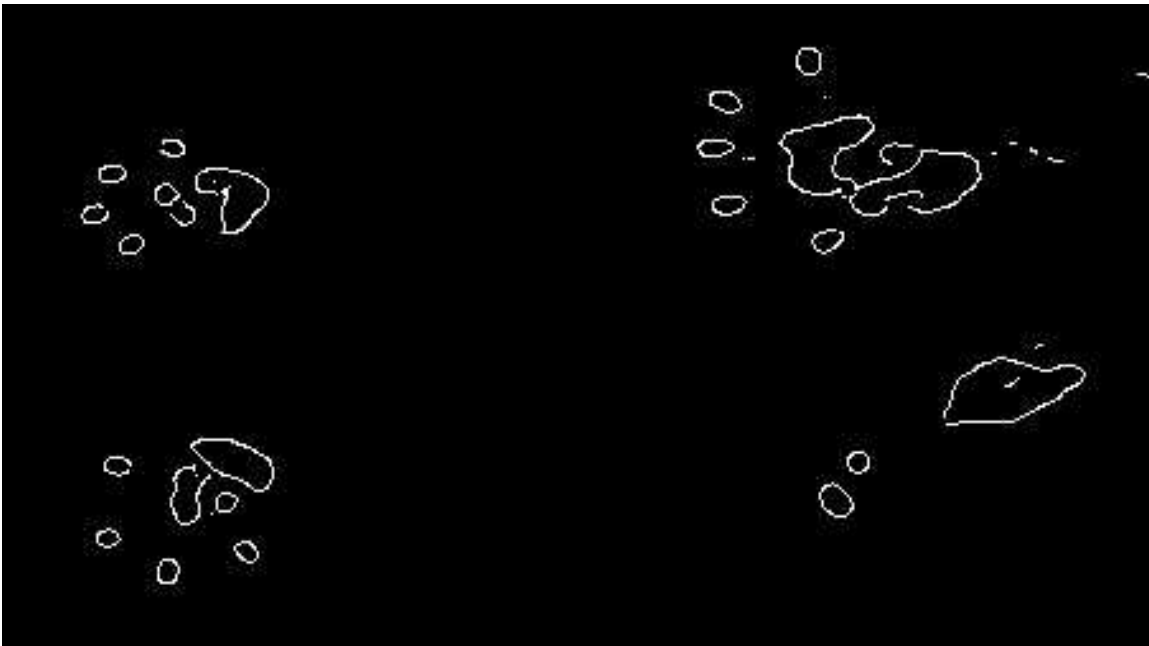


Figure 10. The ImageJ tools, Wand and PolygonRoi, are used to detect the edges of the paws and toes.

The next phase of the paw print analysis involved eliminating ROIs that were neither paws nor toes, such as the rat's body dragging on the walkway, the rat's abdomen that was not making contact with the runway but was bright enough to be recorded by the camera, and light dispersion from contact that caused the paw to seem larger (Figure 11). First, the average x and y values of each ROI were calculated to find its center. The shortest distance between two ROIs was calculated using the Pythagorean Theorem ($r = \sqrt{(x^2 + y^2)}$). If any ROI was within 75 pixels of another ROI, its x and y coordinates were saved in an array. All other ROIs were discarded because any ROI that was alone was unlikely to be a paw and its toes. A distance of 75 pixels was a generous estimate and

was unlikely to eliminate any valid data. Analysis of video showed that most toes were within 30 pixels of the paw.



Figure 11. The rat's body is making contact with the surface and must be correctly identified by the paw print software as an artifact.

The paw print analysis now must determine whether a ROI was a paw or toe. The area of each ROI was calculated. A paw was identified as a ROI that was more than 100 pixels in area. Once a paw was identified, the other ROIs in its array were identified as a toe if it was less than 50 pixels in distance from the paw and if its area was between 10 and 80 pixels. The coordinates for the paws and toes were saved in an array for each paw. All other coordinates were considered artifact and were discarded.

Once the paws and their toes have been defined, the program must identify the paw as left hind paw, right hind paw, left forepaw, or right forepaw. To accomplish this identification, first the center of mass for each paw with its associated toes was

determined by using the Java center of mass function, which determines a weighted average of x and y coordinates. Next, the relative position of each paw was determined by comparing its center of mass to the other paws. The distance between the paws and the area of each paw was both displayed and saved to a text file.

Animals:

A single young adult female Long-Evans rat from the Animal Resource Center at the University of Texas Southwestern was studied. All procedures relating to animals in this study were approved by the Institutional Animal Care and Use Committee.

Proof of concept:

Video clips of a rat traversing the CatWalk were recorded, and images were extracted as described. A series of 120 frames (about two seconds of walking) were selected from the images for each video. Series with one, two, three, and four paws were used.

RESULTS

The video analysis software was able to import the images, correctly identify the paws, and record the paw contact area and the distance between paws to a text file (Figure 12). The software also saved the analyzed images so that the results could be checked by hand (Figures 13, 14).

Figure 12. The contact area, coordinates for the center of mass for each paw, and the distances between paws is saved in text file by paw print analysis software.

Front Right Paw

Area: 2067.0

X Cm: 230

Y Cm: 182

Front Left Paw

Area: 1053.0

X Cm: 243

Y Cm: 303

Back Right Paw

Area: 1212.0

X Cm: 487

Y Cm: 165

Back Left Paw

Area: 1052.0

X Cm: 545

Y Cm: 256

FR Paw to FL 21

FR Paw to BR 272

FR Paw to BL 343

FL Paw to BR 254

FL Paw to BL 325

BL Paw to BR 74

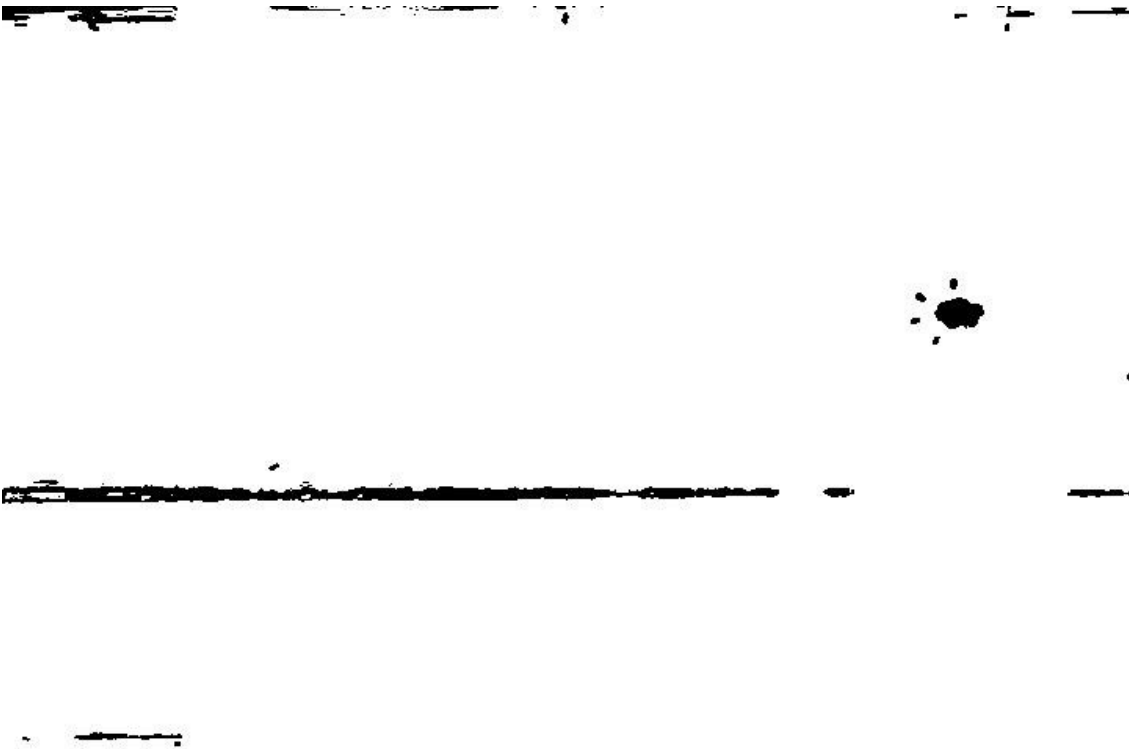


Figure 13. The paw print image was saved after it had been analyzed by the software. The lines across the bottom and top of the image are artifact created by the walkway's walls. This artifact is ignored by the software because it lies outside of the area analyzed for paw prints.



Figure 14. The paw print image before it is analyzed by the software

DISCUSSION

New software for image analysis of rats walking on the CatWalk has been developed. A brief proof of concept study demonstrated that gait parameters may be efficiently measured with the software described in this paper. This software was able to measure contact area and stance width. However, the original CatWalk analysis program and the DigiGait analysis programs were able to calculate more parameters. Since the paw identification portion of the paw print software was finalized, it will be feasible to add new gait parameter measurements. One approach to adding more parameters would involve writing a new program in Java or Microsoft Excel that analyzes data saved from the analysis of multiple sequential sets of frames. Such a program would be able to measure the stride length by the equation: $\text{stride length} = \text{distance between the centers of mass for a given paw}$. Stance phase could also be calculated by the equation: $\text{Stance phase} = \# \text{ frames a given paw is in contact with surface} / \text{sampling frequency (50 Hz)}$. A disadvantage of the CatWalk was that it was limited to self-selected speed; whereas, the DigiGait may eliminate variations in gait analysis due to changes in speed.

CHAPTER FOUR

Conclusions and Recommendations

Automated Paw Print Gait Analysis with the DigiGait

Conclusions:

Preliminary results from this study which used the DigiGait system elucidated how gait parameters change with treadmill speed during recovery. Maximum speed did not seem to vary with post-operative time after the third post-operative week. Hindlimb stride length and frequency increased in rats with contusion injury as treadmill increased. Stance width did not change with the speeds analyze, while hindlimb shared stance time decreased with increasing treadmill speed. Stance width increased as BBB score improved for each animal. Stance/swing ratio increased with BBB but decreased with increasing treadmill speed. This observation demonstrated the utility of DigiGait for controlling variability in gait parameters that change with speed. As was discussed previously, when rats were allowed to self-select speed there may be a wide range of speeds tested that contribute to variability in gait parameters. Stride length increased between 2 and 3 weeks after injury, but stabilized afterwards. These preliminary results suggested that 10 cm/s was a feasible minimum speed for Long-Evans in future studies because every rat in this study was able to attain this speed consistently both before and after injury. This work has contributed to an overall goal of better characterizing the efficacy of treatments that lead to improved locomotor function after SCI in the rodent model and that may be used eventually for human subjects with gait dysfunction

following SCI. The gait parameter studied may also contribute to understanding gait recovery in humans with systems such as the Lokomat.

Recommendations:

Using data that has been gathered, comparison of gait parameters at higher speeds may demonstrate increased variability in gait parameters of injured animals.

Examination of gait parameters at each animal's maximum recorded speed may further elucidate the relationship between speed and gait. Further analysis of the preoperative animals will allow more complete comparison of the changes in gait after rat injury.

Additionally, by analyzing data at the maximum speeds, the speeds at which significant changes occurred may be determined, and future studies may be designed with fewer speeds tested. Since a plateau was seen in many gait parameters after three weeks, future studies could be limited to four weeks. Future studies may also study gait with greater injury since different gait changes have been reported at various contusion forces.

Anatomical correlation to gait recovery may be performed through stereology and histology of the spinal cords that were preserved at the end of the study. Once variability in gait parameters has been characterized using the DigiGait system, future studies using the system will be able to compare recovery between treatment groups and control groups. SCI is a debilitating disease that affects many people in the United States.

Improved treatments may one day be developed in part by gait analysis, using systems like the DigiGait, to characterize recovery from injury.

Semi-Automated Paw Print Gait Analysis Software

Conclusions:

Paw print analysis software has been developed to examine gait characteristics in video of rats traversing the CatWalk. Specifically, paw contact area and distance between the paws (e.g. stance width) may be measured by the software. Additionally, the software was able to identify paws semi-automatically after initial characterization by the user.

Recommendations:

By adding more parameters to the analysis software, its efficiency and usefulness will be increased. Time dependent parameters like stance/swing ratio and hindlimb shared stance may also be calculated by using the sampling frequency of the video recorder as discussed previously. The more parameters that are added to the software, the greater the efficiency in paw print analysis. Such analysis may be used in the future to better study changes in gait recovery with new treatments for SCI in rats that may eventually applied to injury in humans. Better understanding of the recovery of gait after SCI in rats and humans has also contributed to developing locomotor training with systems like the Locomat and other methods that encourage neural plasticity.⁴

BIBLIOGRAPHY

1. National Spinal Cord Injury Statistical Center (NSCISC). Accessed May 1, 2007.
<http://www.spinalcord.uab.edu/show.asp?durki=21446>
2. Nessler JA, De Leon RD, Sharp K, Kwak E, Minakata K, Reinkensmeyer DJ
2006 Robotic gait analysis of bipedal treadmill stepping by spinal contused rats:
characterization of intrinsic recovery and comparison with BBB. Journal of
Neurotrauma 23:882-896
3. Petruska JC, Ichiyama RM, Jindrich DL, Crown ED, Tansey KE, Roy RR,
Edgerton VR, Mendell LM 2007 Changes in motoneuron properties and synaptic
inputs related to step training after spinal cord transection in rats. Journal of
Neuroscience: The Official Journal of the Society for Neuroscience 27:4460-4471
4. Lunenburger L, Colombo G, Riener R, Dietz V 2004 Biofeedback in gait training
with the robotic orthosis Lokomat. Annual International Conference of the
Institute of Electrical and Electronics Engineers Engineering in Medicine and
Biology Society 7:4888-4891
5. Hsu JY, McKeon R, Goussev S, Werb Z, Lee JU, Trivedi A, Noble-Haeusslein LJ
2006 Matrix metalloproteinase-2 facilitates wound healing events that promote
functional recovery after spinal cord injury. Journal of Neuroscience: The Official
Journal of the Society for Neuroscience 26:9841-985
6. Goldberger ME, Bregman BS, Vierck CJ Jr, Brown M 1990 Criteria for assessing
recovery of function after spinal cord injury: behavioral methods. Experimental
Neurology 107:113-7

7. Metz G, Merkler D, Dietz V, Schwab ME, Fouad K 2000 Efficient testing of motor function in spinal cord injured rats. *Brain Research* 883:165-17
8. Muir GD, Webb AA 2000 Mini-review: assessment of behavioural recovery following spinal cord injury in rats. *The European Journal of Neuroscience* 12:3079-86
9. Basso DM, Beattie MS, Bresnahan JC, Anderson DK, Faden AI, Gruner JA, Holford TR, Hsu CY, Noble, LJ, Nockels R, Perot PL, Salzman, SK, Young W 1996 MASCIS evaluation of open field locomotor scores: effects of experience and teamwork on reliability. Multicenter animal spinal cord injury study. *Journal of Neurotrauma* 13:343–359
10. McEwen ML, Springer JE 2006 Quantification of locomotor recovery following spinal cord contusion in adult rats. *Journal of Neurotrauma* 23:1632-1653
11. Hamers FP, Lankhorst AJ, van Laar TJ, Veldhuis WB, Gispen WH 2001 Automated quantitative gait analysis during overground locomotion in the rat: its application to spinal cord contusion and transection injuries. *Journal of Neurotrauma* 18:187-201
12. Hampton, TG, Stasko MR, Kale A, Amende I, Costa AC 2004 Gait dynamics in trisomic mice: quantitative neurological traits of Down syndrome. *Physiology & Behavior* 82:381–389
13. Vrinten DH, Hamers FF 2003 'CatWalk' automated quantitative gait analysis as a novel method to assess mechanical allodynia in the rat; a comparison with von Frey testing. *Pain* 102:203-9

14. Cheng H, Almstrom S, Gimenez LL 1997 Gait analysis of adult paraplegic rats after spinal cord repair *Experimental Neurology* 148:544-557
15. Wooley CM, Sher RB, Kale A, Frankel WN, Cox GA, Seburn KL 2005 Gait analysis detects early changes in transgenic SOD1(G93A) mice. *Muscle & Nerve* 32:43-50
16. Kale A, Amende I, Meyer GP, Crabbe JC, Hampton TG 2004 Ethanol's effects on gait dynamics in mice investigated by ventral plane videography. *Alcoholism Clinical & Experimental Research* 28:1839-48
17. Li S, Kim JE, Budel S, Hampton TG, Strittmatter SM 2005 Transgenic inhibition of Nogo-66 receptor function allows axonal sprouting and improved locomotion after spinal injury. *Molecular and Cellular Neuroscience* 29:26-39
18. Collazos-Castro JE, López-Dolado E, Nieto-Sampedro M 2006 Locomotor deficits and adaptive mechanisms after thoracic spinal cord contusion in the adult rat. *Journal of Neurotrauma*. 23:1-17
19. Scheff SW, Rabchevsky AG, Fugaccia I, Main JA, Lumpp JE 2003 Experimental modeling of spinal cord injury: characterization of a force-defined injury device. *Journal of Neurotrauma* 20:179-93
20. Kloos, AD, Fisher LC, Detloff MR, Hassenzahl DL, and Basso DM 2005 Stepwise motor and all-or-none sensory recovery is associated with nonlinear sparing after incremental spinal cord injury in rats. *Experimental Neurology* 191:251–265
21. Keirstead HS, Nistor G, Bernal G, Totoiu M, Cloutier F, Sharp K, Steward O 2005 Human embryonic stem cell-derived oligodendrocyte progenitor cell

- transplants remyelinate and restore locomotion after spinal cord injury. *The Journal of Neuroscience: The Official Journal of the Society for Neuroscience* 25:4694-4705
22. Herbin M, Gasc JP, Renous S 2004 Symmetrical and asymmetrical gaits in the mouse: patterns to increase velocity. *Journal of Comparative Physiology. A, Neuroethology, Sensory, Neural, and Behavioral Physiology* 190:895-906
23. Hendriks WT, Eggers R, Ruitenber MJ, Blits B, Hamers FP, Verhaagen J, Boe GJ 2006 Profound differences in spontaneous long-term functional recovery after defined spinal tract lesions in the rat. *Journal of Neurotrauma* 23:18-35
24. ImageJ. Accessed May 14, 2007. <http://rsb.info.nih.gov/ij/>
25. Clarke KA, Still J 1999 Gait analysis in the mouse. *Physiology & Behavior* 66:723–729
26. Koopmans GC, Deumens R, Honig WM, Hamers FP, Steinbusch HW, Joosten EA 2005 The assessment of locomotor function in spinal cord injured rats: the importance of objective analysis of coordination. *Journal of Neurotrauma*. 22:214-225

APPENDIX

The software for the paw print analysis was written in Java.

//PawPrintAnalysis writes area, centers of mass, with each paw position.

```
package Demos;
import java.awt.*;
import java.awt.Polygon;
import java.awt.Shape;
import java.awt.image.IndexColorModel;
import java.util.Properties;
import java.util.Arrays;
import ij.process.PolygonFiller;
import ij.*;
import ij.gui.*;
import ij.process.*;
import ij.measure.*;
import ij.text.*;
import ij.plugin.filter.Analyzer;
import ij.plugin.frame.Recorder;
import ij.plugin.filter.PlugInFilter;
import ij.plugin.PlugIn;
import ij.gui.PolygonRoi;
import ij.plugin.filter.*;
import ij.plugin.filter.ParticleAnalyzer;
import ij.plugin.TextWriter;
import ij.gui.Line;
import java.awt.Graphics;

public class PawPrintAnalysis implements PlugInFilter {
    private ImageProcessor drawIP;
    private String arg;
    private Roi saveRoi;
    private Wand wand;
    private Wand wand2;
    private Wand wand3;
    private Wand wand4;
    protected int slice;
    protected boolean processStack;
    private ImageProcessor redirectIP;
    private boolean canceled;
    protected Analyzer analyzer;
    private int sizeBins;
    static final String BINS = "ap.bins";
    private static int staticBins = Prefs.getInt(BINS,20);
    int AREA;
```

```

public static final int SHOW_OUTLINES = 4;

//Parameters for GUI
ImagePlus    imp;
boolean      bAbort;
int xlow=0;
int ylow=13;
int xhigh=720;
int yhigh=285;
static boolean frpu;
static boolean flpu;
static boolean brpu;
static boolean blpu;

/**
 * Called by ImageJ when the filter is loaded
 */
public int setup(String arg, ImagePlus imp) {
    this.imp = imp;
    if (arg.equals("about"))
        {showAbout(); return DONE;}
    return DOES_ALL+NO_CHANGES;
}

/**
 * Called by ImageJ to process the image
 */
public void run(ImageProcessor ip) {

    // Parameters for PawAreaPosn
    int width = ip.getWidth();
    int height = ip.getHeight();
    int wh=(width)*(height);
    int [][] picmx = new int [2][wh];
    int count=0;
    int count2=0;
    int count3=0;
    int [] rpts = new int [100];
    int [][] oldrois = new int [1000][1000];
    int [][] oldroisavex = new int [1000][1000];
    int [][] oldroisavey = new int [1000][1000];
    int [] roilength = new int [1000];
    double [] areamx = new double [1000];
    boolean roifalse;
    boolean areafalse=true;

```

```

byte[] pixels = (byte[])ip.getPixels();

// A dialog box for user input appears
GenericDialog gd = new GenericDialog("Paw Print Analysis", IJ.getInstance());
gd.addCheckbox("Front Right Paw", frpu);
gd.addCheckbox("Front Left Paw", flpu);
gd.addCheckbox("Back Right Paw", brpu);
gd.addCheckbox("Back Left Paw", blpu);
gd.showDialog();
if (gd.wasCanceled()) {
    bAbort = true;
    return;
}

frpu = (boolean)gd.getNextBoolean();
flpu = (boolean)gd.getNextBoolean();
brpu = (boolean)gd.getNextBoolean();
blpu = (boolean)gd.getNextBoolean();

// Trace roi with WAND and PolygonRoi
for (int s=0;s<height;s++){
    for(int t=0;t<width;t++){
        if(pixels[(s*width)+t] != -1 ){
            wand = new Wand(ip);
            wand.autoOutline(t,s,0,127);
            PolygonRoi roi = new PolygonRoi(wand.xpoints, wand.ypoints, wand.npoints,
Roi.TRACED_ROI);

// Obtain x points and y coordinates of roi polygon
            Polygon p = roi.getPolygon();
            int [] xpts = p.xpoints;
            int [] ypts = p.ypoints;
            int [] xptspresort = new int [xpts.length];
            for(int i=0;i<xpts.length;i++){
                xptspresort[i]=xpts[i];
            }
            Arrays.sort(xpts);

// If roi found, set to 0 all points in roi
            roifalse=true;
            if (xpts.length!=0){
                for(int z=0;z<count3;z++){
                    if (oldrois[z][0]==xpts[0]){
                        roifalse=false;}}
                if(roifalse==true){
                    imp.setRoi( roi );

```



```

int measurements = Analyzer.getMeasurements();
ImageStatistics stats = imp.getStatistics(measurements);
areamx[count3]=stats.area;

for (int n=0;n<xptspresort.length;n++){
    oldrois[count3][n] = xpts[n];
    oldroisavex[count3][n] = xptspresort[n];
    oldroisavey[count3][n] = ypts[n];
}
roilength[count3]=xptspresort.length;
count3=count3+1;
}
}
}
}
}
}
}

```

```

// Find avg. x and y of all areas
int [] avgx = new int [count3];
int [] avgy = new int [count3];
for (int b=0;b<count3;b++){
    int sumxtemp=0;
    int sumytemp=0;
    for (int c=0;c<999;c++){
        if (oldroisavex[b][c]!=0){
            sumxtemp=oldroisavex[b][c]+sumxtemp;
            sumytemp=oldroisavey[b][c]+sumytemp;
        }
        else if (oldroisavex[b][c]==0){
            avgx[b]=sumxtemp/(c);
            avgy[b]=sumytemp/(c);
            c=999;
        }
    }
}
}
}

```

```

// Eliminate outlier rois
double distance;
int x2;
int y2;
int count5=0;

//double dist2;
int [] finavgx = new int [avgx.length];
int [] finavgy = new int [avgy.length];

```

```

for (int d=0;d<avgx.length;d++){
    int count4 = 0;
    for (int e=0;e<avgx.length;e++){
        // Ensure not comparing point to itself
        if (d!=e){

// Use Pythagorean Theorem to determine distance
            x2= Math.abs(avgx[e]-avgx[d]);
            y2= Math.abs(avgy[e]-avgy[d]);
            distance = Math.sqrt( (x2*x2) + (y2*y2) );
            // Put all avg x and y of one paw into an array
            if(distance<75.0 && count4==0){
                finavgx[d]=avgx[d];
                finavgy[d]=avgy[d];
                count4 = count4+1;
                count5=count5+1;
            }
        }
    }
}

// Cut down arrays of avg values and final areas to right size
int count6 = 0;
int [] finalavgx = new int [count5];
int [] finalavgy = new int [count5];
double[] finalarea = new double[count5];
for (int u=0;u<finavgx.length;u++){
    if (finavgx[u]!=0){
        finalavgx[count6]=finavgx[u];
        finalavgy[count6]=finavgy[u];
        finalarea[count6]=areamx[u];
        count6=count6+1;
    }
}

// Find Each Paw
int count8 = 0;
int count7 = 0;
int toecount = 0;
int [][] pawsx = new int [4][15];
int [][] pawsy = new int [4][15];
int [][] toesx = new int [4][14];
int [][] toesy = new int [4][14];
//int [][] toeara = new int [4][15];
int [] hook = new int [finalavgx.length];
double [] pawsarea = new double [4];

```

```

for (int y=0;y<finalavgx.length;y++){
hook[y]=100;
}

for (int d=0;d<finalavgx.length;d++){
if(finalarea[d]>100 && hook[d]!=d){

for (int e=0;e<finalavgx.length;e++){
if (d!=e){
x2= Math.abs(finalavgx[d]-finalavgx[e]);
y2= Math.abs(finalavgx[d]-finalavgx[e]);
distance = Math.sqrt( (x2*x2) + (y2*y2) );
if(distance<50.0){
if(finalarea[e]<100 && finalarea[e]>10){
pawsx[count8][count7]=finalavgx[e];
pawsy[count8][count7]=finalavgx[e];
pawsarea[count8]=pawsarea[count8]+finalarea[e];
count7=count7+1;

if (finalarea[e]<80 && finalarea[e]>10){
toesx[count8][toecount]=finalavgx[e];
toesy[count8][toecount]=finalavgx[e];
toecount=toecount+1;

if (e==finalavgx.length-1)
{count7=0;
toecount=0;}
}

else if (e==finalavgx.length-1)
{count7=0;
toecount=0;}
}

else if(finalarea[e]>100){
pawsx[count8][count7]=finalavgx[e];
pawsy[count8][count7]=finalavgx[e];
pawsarea[count8]=pawsarea[count8]+finalarea[e];
hook[e]=e;
count7=count7+1;
}}

else if(e==finalavgx.length-1)
{count7=0;
toecount=0;}
}
}
}

```

```

else if(d==e && finalarea[d]>100 && hook[d]!=e){
    pawsx[count8][count7]=finalavgx[e];
    pawsy[count8][count7]=finalavgx[e];
    pawsarea[count8]=pawsarea[count8]+finalarea[e];
    count7=count7+1;
    }}
count8=count8+1;
}}

// Construct overall paw
// Find Cm of each paw
int [] pawscmx = new int [count8]; // Use count8 so that True # of paws is used
int [] pawscmy = new int [count8];
for (int b=0;b<count8;b++){
    int sumxtemp=0;
    int sumytemp=0;
    for (int c=0;c<15;c++){
        if (pawsx[b][c]!=0){
            sumxtemp=pawsx[b][c]+sumxtemp;
            sumytemp=pawsy[b][c]+sumytemp;
        }
        else if (c!=0 && pawsx[b][c]==0){
            pawscmx[b]=sumxtemp/(c);
            pawscmy[b]=sumytemp/(c);
            c=15;
        }
    }
}

// Determine Distance between each toe
int [][] toedistsi = new int [count8][14];
for (int i=0;i<count8;i++){
    for (int j=0;j<13;j++){
        if (toesx[i][j+1]!=0){
            Line toeline = new Line(toesx[i][j], toesy[i][j], toesx[i][j+1], toesy[i][j+1]);
            toedistsi[i][j] = new Double(toeline.getRawLength()).intValue();
            //IJ.write("toedistsi["+i+""]["+j+""]: "+toedistsi[i][j]);
        }
        else if (toesx[i][j+1]==0){
            j=13;
        }
    }
}

// Avg. Toe Distance for each paw
int [] toesavg = new int [count8];
for (int i=0;i<count8;i++){
    int temp=0;
    for (int j=0;j<14;j++){
        if(toedistsi[i][j]!=0){

```

```

        temp=toedists[i][j]+temp;
    }
    else if (toedists[i][j]==0 && j!=0){
        toesavg[i]=temp/j;
        //IJ.write("toe avg "+toesavg[i]);
        j=14;
    }
}

// Determine Center of Mass of just toes
int [] toescmx = new int [count8]; // Use count8 so that True # of paws is used
int [] toescmy = new int [count8];
for (int b=0;b<count8;b++){
    int sumxtemp=0;
    int sumytemp=0;
    for (int c=0;c<14;c++){
        if (toesx[b][c]!=0){
            sumxtemp=toesx[b][c]+sumxtemp;
            sumytemp=toesy[b][c]+sumytemp;
            //IJ.write("sumxtemp "+sumxtemp);
        }
        else if (c!=0 && pawsx[b][c]==0){
            toescmx[b]=sumxtemp/(c-1);
            toescmy[b]=sumytemp/(c-1);
            c=14;
        }
    }
}

// Determine Max values for Each Paw and toe
int [] tempxsort = new int [14];
int [] toesxsort = new int [4];
int [] toesysort = new int [4];
for (int i=0;i<count8;i++){
    for (int j=0;j<13;j++){
        tempxsort[j]=toesx[i][j];
    }
    Arrays.sort(tempxsort);
    for (int p=0;p<13;p++){
        if(tempxsort[p]!=0){
            toesxsort[i]=tempxsort[p];
            p=13;
        }
    }
}

for (int y=0;y<14;y++){
    if (toesxsort[i]==toesx[i][y]){
        toesysort[i]=toesy[i][y];
    }
}

```

```

    }
}

// Determine Angle
int [] toesangle = new int [count8];
for (int i=0;i<count8;i++){
    Roi roi = new Roi(pawscmx[i], pawscmy[i], imp);
    toesangle[i] = new Double(roi.getAngle(pawscmx[0], pawscmy[0], toescmx[0],
toescmy[0])).intValue();
    //IJ.write("Paw Angle " +toesangle[i]);
}

// Determine Paw Relative Position
int [] pawscmx = new int[count8];
int [] pawscmy = new int[count8];
double [] pawsareaf = new double [count8];
int [] toesavgf = new int [count8];
int [] toesanglef = new int [count8];
int frp=0;
int flp=0;
int brp=0;
int blp=0;
int condition=0;

//For 1 Paw:
if (count8==1){
    if (frpu==true){
        IJ.write("");
        IJ.write("Front Right Paw");
        IJ.write("Area: "+ pawsarea[0]);
        IJ.write("X Cm: "+pawscmx[0]);
        IJ.write("Y Cm: "+pawscmy[0]);
        IJ.write("Avg Dist. btw Toes: "+toesavg[0]);
        IJ.write("Angle rel. to Horizontal: "+toesangle[0]);
        IJ.write("");
    }
    else if(flpu==true){
        IJ.write("Front Left Paw");
        IJ.write("Area: "+pawsarea[0]);
        IJ.write("X Cm: "+pawscmx[0]);
        IJ.write("Y Cm: "+pawscmy[0]);
        IJ.write("Avg Dist. btw Toes: "+toesavg[0]);
        IJ.write("Angle rel. to Horizontal: "+toesangle[0]);
        IJ.write("");
    }
    else if(brpu==true){

```

```

IJ.write("Back Right Paw");
IJ.write("Area: "+pawsarea[0]);
IJ.write("X Cm: "+pawscmx[0]);
IJ.write("Y Cm: "+pawscmy[0]);
IJ.write("Avg Dist. btw Toes: "+toesavg[0]);
IJ.write("Angle rel. to Horizontal: "+toesangle[0]);
IJ.write("");
}
else if(blpu==true){
IJ.write("Back Left Paw");
IJ.write("Area: "+pawsarea[0]);
IJ.write("X Cm: "+pawscmx[0]);
IJ.write("Y Cm: "+pawscmy[0]);
IJ.write("Avg Dist. btw Toes: "+toesavg[0]);
IJ.write("Angle rel. to Horizontal: "+toesangle[0]);
IJ.write("");
}
}

// For 2 or 3 paws
else if(count8==2 || count8==3){

// For 2 paws
if (count8==2){
for (int k=0;k<count8;k++){
pawscmxf[k]=pawscmx[k];
}

Arrays.sort(pawscmx);
for (int i=0;i<count8;i++){
for (int j=0;j<count8;j++){
if (pawscmx[i]==pawscmxf[j]){
pawscmyf[i]=pawscmy[j];
pawsareaf[i]=pawsarea[j];
toesavgf[i]=toesavg[j];
toesanglef[i]=toesangle[j];
//IJ.write("pawscmx: "+pawscmx[i]);
//IJ.write("pawscmyf: "+pawscmyf[i]);
j=count8;
}}
}

if (frpu==true && flpu==true)
{condition=1;
}

```

```

else if (frpu==true && brpu==true)
{condition=2;
 //IJ.write("condition="+condition);
}
else if (frpu==true && blpu==true)
{condition=3;
}
else if (flpu==true && brpu==true)
{condition=4;
}
else if (flpu==true && blpu==true)
{condition=5;
}
else if (brpu==true && blpu==true)
{condition=6;
}

// Determine distance between each center of mass
int [][] pawdists = new int [4][4];
for (int i=0;i<count8;i++){
for (int j=0;j<count8;j++){
if (i!=j){
x2= Math.abs(pawscmx[i]-pawscmx[j]);
y2= Math.abs(pawscopy[i]-pawscopy[j]);
distance = Math.sqrt( (x2*x2) + (y2*y2) );
int disttemp = new Double(distance).intValue();
pawdists[i][j]=disttemp;
}}}
for (int m=0;m<1;m++){
IJ.write("Distances between Paws:");
if (condition==1){
IJ.write("FR Paw to FL " +pawdists[frp][flp]);
}
else if (condition==2){
IJ.write("FR Paw to BR " +pawdists[frp][brp]);
}
else if (condition==3){
IJ.write("FR Paw to BL " +pawdists[frp][blp]);
}
else if (condition==4){
IJ.write("FL Paw to BR " +pawdists[flp][brp]);
}
else if (condition==5){
IJ.write("FL Paw to BL " +pawdists[flp][blp]);
}
else if (condition==6){

```



```

        IJ.write("BL Paw to BR " +pawdists[blp][brp]);
    }}}

// For 3 Paws:
else if (count8==3){
    for (int k=0;k<count8;k++){
        pawscmf[k]=pawscmx[k];
    }

    Arrays.sort(pawscmx);
    for (int i=0;i<count8;i++){
        for (int j=0;j<count8;j++){
            if (pawscmx[i]==pawscmf[j]){
                pawscmyf[i]=pawscmy[j];
                pawsareaf[i]=pawsarea[j];
                toesavgf[i]=toesavg[j];
                toesanglef[i]=toesangle[j];
                j=count8;
            }
        }
    }

    for (int t=0;t<1;t++){
        if(frpu==true && flpu==true && brpu==true){
            if(pawscmyf[0]>pawscmyf[1]){
                frp=0;
                flp=1;
                brp=2;
                condition=1;
            }
            else if(pawscmyf[0]<pawscmyf[1]){
                frp=1;
                flp=0;
                brp=2;
                condition=1;
            }
        }

        else if(frpu==true && flpu==true && blpu==true){
            if(pawscmyf[0]>pawscmyf[1]){
                frp=0;
                flp=1;
                blp=2;
                condition=2;
            }
            else if(pawscmyf[0]<pawscmyf[1]){
                frp=1;
                flp=0;
                blp=2;
            }
        }
    }
}

```

```

        condition=2;
    }}

else if(frp==true && brp==true && blp==true){
    if(pawscmyf[1]>pawscmyf[2]){
        brp=1;
        blp=2;
        frp=0;
        condition=3;
    }
    else if(pawscmyf[1]<pawscmyf[2]){
        brp=2;
        blp=1;
        frp=0;
        condition=3;
    }}

else if(flpu==true && brpu==true && blpu==true){
    if(pawscmyf[1]>pawscmyf[2]){
        brp=1;
        blp=2;
        flp=0;
        condition=4;
    }

    else if(pawscmyf[1]<pawscmyf[2]){
        brp=2;
        blp=1;
        flp=0;
        condition=4;
    }}}

// Determine distance between each paw's center of mass
int [][] pawdists = new int [4][4];
for (int i=0;i<count8;i++){
    for (int j=0;j<count8;j++){
        if (i!=j){
            x2= Math.abs(pawscmx[i]-pawscmx[j]);
            y2= Math.abs(pawscmy[i]-pawscmy[j]);
            distance = Math.sqrt( (x2*x2) + (y2*y2) );
            int disttemp = new Double(distance).intValue();
            pawdists[i][j]=disttemp;
        }}
}

for (int p=0;p<1;p++){
    IJ.write("");
}

```

```

    IJ.write("Distances between Paws:");
    if (condition==1){
        IJ.write("FR Paw to FL " +pawdists[frp][flp]);
        IJ.write("FR Paw to BR " +pawdists[frp][brp]);
        IJ.write("FL Paw to BR " +pawdists[flp][brp]);
    }
    else if (condition==2){
        IJ.write("FR Paw to FL " +pawdists[frp][flp]);
        IJ.write("FR Paw to BL " +pawdists[frp][blp]);
        IJ.write("FL Paw to BL " +pawdists[flp][blp]);
    }
    else if (condition==3){
        IJ.write("FR Paw to BR " +pawdists[frp][brp]);
        IJ.write("FR Paw to BL " +pawdists[frp][blp]);
        IJ.write("BL Paw to BR " +pawdists[blp][brp]);
    }
    else if (condition==4){
        IJ.write("FL Paw to BR " +pawdists[flp][brp]);
        IJ.write("FL Paw to BL " +pawdists[flp][blp]);
        IJ.write("BL Paw to BR " +pawdists[blp][brp]);
    }
}

for (int s=0;s<1;s++){
    if (frpu==true){
        IJ.write("");
        IJ.write("Front Right Paw");
        IJ.write("Area: "+ pawsarea[frp]);
        IJ.write("X Cm: "+pawscmx[frp]);
        IJ.write("Y Cm: "+pawscmyf[frp]);
        IJ.write("Avg Dist. btw Toes: "+toesavgf[frp]);
        IJ.write("Angle rel. to Horizontal: "+toesanglef[frp]);
        IJ.write("");
    }
}

for (int s=0;s<1;s++){
    if(flpu==true){
        IJ.write("Front Left Paw");
        IJ.write("Area: "+pawsarea[flp]);
        IJ.write("X Cm: "+pawscmx[flp]);
        IJ.write("Y Cm: "+pawscmyf[flp]);
        IJ.write("Avg Dist. btw Toes: "+toesavgf[flp]);
        IJ.write("Angle rel. to Horizontal: "+toesanglef[flp]);
        IJ.write("");
    }
}

for (int s=0;s<1;s++){

```

```

if(brpu==true){
    IJ.write("Back Right Paw");
    IJ.write("Area: "+pawsarea[brp]);
    IJ.write("X Cm: "+pawscmx[brp]);
    IJ.write("Y Cm: "+pawscmyf[brp]);
    IJ.write("Avg Dist. btw Toes: "+toesavgf[brp]);
    IJ.write("Angle rel. to Horizontal: "+toesanglef[brp]);
    IJ.write("");
}
}

for (int s=0;s<1;s++){
    if(blpu==true){
        IJ.write("Back Left Paw");
        IJ.write("Area: "+pawsarea[blp]);
        IJ.write("X Cm: "+pawscmx[blp]);
        IJ.write("Y Cm: "+pawscmyf[blp]);
        IJ.write("Avg Dist. btw Toes: "+toesavgf[blp]);
        IJ.write("Angle rel. to Horizontal: "+toesanglef[blp]);
        IJ.write("");
    }
}

// for 4 Paws
else if(count8==4){
    for (int k=0;k<4;k++){
        pawscmf[k]=pawscmx[k];
    }

    Arrays.sort(pawscmx);
    for (int i=0;i<4;i++){
        for (int j=0;j<4;j++){
            if (pawscmx[i]==pawscmf[j]){
                pawscmyf[i]=pawscmy[j];
                pawsareaf[i]=pawsarea[j];
                toesavgf[i]=toesavg[j];
                toesanglef[i]=toesangle[j];
                //IJ.write("pawscmx: "+pawscmx[i]);
                //IJ.write("pawscmyf: "+pawscmyf[i]);
                j=4;
            }
        }
    }

    if(pawscmyf[0]>pawscmyf[1]){
        frp=0;
        flp=1;
    }

    else if(pawscmyf[0]<pawscmyf[1]){

```

```

    frp=1;
    flp=0;
}

if(pawscmyf[2]>pawscmyf[3]){
    brp=2;
    blp=3;
}

else if(pawscmyf[2]<pawscmyf[3]){
    brp=3;
    blp=2;
}

IJ.write("");
IJ.write("Front Right Paw");
IJ.write("Area: "+pawsarea[frp]);
IJ.write("X Cm: "+pawscmx[frp]);
IJ.write("Y Cm: "+pawscmyf[frp]);
IJ.write("Avg Dist. btw Toes: "+toesavgf[frp]);
IJ.write("Angle rel. to Horizontal: "+toesanglef[frp]);
IJ.write("");
IJ.write("Front Left Paw");
IJ.write("Area: "+pawsarea[flp]);
IJ.write("X Cm: "+pawscmx[flp]);
IJ.write("Y Cm: "+pawscmyf[flp]);
IJ.write("Avg Dist. btw Toes: "+toesavgf[flp]);
IJ.write("Angle rel. to Horizontal: "+toesanglef[flp]);
IJ.write("");
IJ.write("Back Right Paw");
IJ.write("Area: "+pawsarea[brp]);
IJ.write("X Cm: "+pawscmx[brp]);
IJ.write("Y Cm: "+pawscmyf[brp]);
IJ.write("Avg Dist. btw Toes: "+toesavgf[brp]);
IJ.write("Angle rel. to Horizontal: "+toesanglef[brp]);
IJ.write("");
IJ.write("Back Left Paw");
IJ.write("Area: "+pawsarea[blp]);
IJ.write("X Cm: "+pawscmx[blp]);
IJ.write("Y Cm: "+pawscmyf[blp]);
IJ.write("Avg Dist. btw Toes: "+toesavgf[blp]);
IJ.write("Angle rel. to Horizontal: "+toesanglef[blp]);
IJ.write("");

// Determine Distance btw each paw

```

```

int [][] pawdists = new int [4][4];
for (int i=0;i<4;i++){
  for (int j=0;j<4;j++){
    if (i!=j){
      x2= Math.abs(pawscmx[i]-pawscmx[j]);
      y2= Math.abs(pawscmy[i]-pawscmy[j]);
      distance = Math.sqrt( (x2*x2) + (y2*y2) );
      int disttemp = new Double(distance).intValue();
      pawdists[i][j]=disttemp;

      //IJ.write("Distance: Paw"+(i+1)+" to Paw"+(j+1)+" is "+pawdists[i][j]);

    }
  }

  IJ.write("");
  IJ.write("FR Paw to FL " +pawdists[frp][flp]);
  IJ.write("FR Paw to BR " +pawdists[frp][brp]);
  IJ.write("FR Paw to BL " +pawdists[frp][blp]);

  IJ.write("FL Paw to BR " +pawdists[flp][brp]);
  IJ.write("FL Paw to BL " +pawdists[flp][blp]);

  IJ.write("BL Paw to BR " +pawdists[blp][brp]);
}

// Write Text File
  TextWriter txtw = new TextWriter();
  void showAbout() {
    IJ.showMessage("About ROISelect_...",
      "This sample allows a ROI of a specific size to be specified.\n"); }}

```

Vitae

EDUCATION

University of Texas Southwestern Medical School, Dallas, TX 2003 - present
M.D. with Distinction in Research expected June 2007

Rice University, Houston, TX 1999 - 2003
B.S. in Biomedical Engineering

St. Thomas Episcopal High School, Houston, TX 1995 - 1999
Graduated Summa Cum Laude

RESEARCH EXPERIENCE

University of Texas Southwestern, Dallas, TX June 2004 – August 2004
Under Dr. Keith Tansey in the Spinal Cord Injury Rehabilitation Lab:
Project 1: Chose appropriate digital video recorder and used it to obtain video from underneath rats walking on the CatWalk; taught myself Java programming language; wrote video analysis software in Java that incorporated NIH's ImageJ image analysis tools to quantitate gait changes during recovery of rats with spinal cord injuries; trained spinal cord injured rats on a robot to improve gait; assisted with rat procedures - sedation, surgery, and maintenance.

Project 2: Used DigiGait to assess changes in gait parameters at various speeds of rats with spinal cord injury. Developed protocol for using the DigiGait with rats. Contused the animals' spines at T9. Assessed gait of these rats using the BBB locomotor scale and the DigiGait before injury and for six weeks after injury.

Rice University, Houston, TX May 2001 – May 2003
Under Dr. Kyriacos Athanasiou in the Musculoskeletal Bioengineering Lab:
Project 1: Designed a gravity-driven system for cell perfusion; designed parts for a direct compression machine and wrote Labview programs to run the direct compression machine; used the two systems described above to conduct an 8-week study of compression and perfusion effects on bovine articular cartilage seeded in polymer scaffolds; stiffness and strength analysis along with histological methods were used to determine these effects on the cartilage.

Project 2: Taught myself Labview programming language; wrote Labview programs to control and record data from a hydrostatic pressure chamber; developed protocol for loading and unloading bovine cartilage cells in a hydrostatic pressure chamber; conducted 6-week study to examine the effects of intermittent hydrostatic pressure on glycosaminoglycan and collagen production of bovine articular cartilage.

ACHIEVEMENTS AND AWARDS

Rice University Dean of Engineering Summer Research Grant, 5/02-8/02
Department of Bioengineering Award for "Outstanding Contributions in Bioengineering"
National Merit Finalist Scholarship
John F. Kennedy Library Leadership Essay Finalist Scholarship