

# Tracking Multiple Mice

by

Stav Braun

Submitted to the Department of Electrical Engineering and Computer  
Science

in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2012

© Massachusetts Institute of Technology 2012. All rights reserved.

Author .....  
Department of Electrical Engineering and Computer Science  
June 14, 2012

Certified by .....  
Tomaso Poggio  
Eugene McDermott Professor  
Thesis Supervisor

Accepted by .....  
Dr. Dennis M. Freeman  
Chairman, Masters of Engineering Thesis Committee

# Tracking Multiple Mice

by

Stav Braun

Submitted to the Department of Electrical Engineering and Computer Science  
on June 14, 2012, in partial fulfillment of the  
requirements for the degree of  
Master of Engineering in Electrical Engineering and Computer Science

## Abstract

Monitoring mouse social behaviors over long periods of time is essential for neuro-behavioral analysis of social mouse phenotypes. Currently, the primary method of social behavioral phenotyping utilizes human labelers, which is slow and costly. In order to achieve the high throughput desired for scientific studies, social behavioral phenotyping must be automated. The problem of automation can be divided into two tasks; tracking and phenotyping. First, individual body parts of mice must be accurately tracked. This is achieved using shape context descriptors to obtain precise point to point correspondences between templates and mice in any frame of a video. This method provides for greater precision and accuracy than current state of the art techniques. We propose a means by which this tracking information can be used to classify social behaviors between mice.

Thesis Supervisor: Tomaso Poggio  
Title: Eugene McDermott Professor

## Acknowledgments

First and foremost, I must thank my advisor, Professor Tomaso Poggio, better known as Tommy. Being in Tommy's lab has been a remarkable experience. Tommy has made research fun and exciting. His encouragement, support, and guidance have helped me throughout my master thesis. I am extremely grateful for the opportunity he has given me, and have learned an incredible amount from him.

In addition, I want to thank all my family, friends and my boyfriend for their endless support throughout this process. Specifically, I want to thank my sister Danielle, boyfriend John, and colleague Youssef, for their significant help in proofreading my thesis and providing ideas.

I want to thank all my colleagues in CBCL. Being in CBCL has been a fun, enlightening, and one of a kind experience because of them. Specifically, I want to thank Gadi for extensive discussions and maintaining a warm and happy work environment. Kathleen, for her continuous support and availability. Hueihan and Nick for their introduction to the mouse project and extensive help and input.

# Contents

<b>1</b>	<b>Introduction</b>	<b>10</b>
<b>2</b>	<b>Background</b>	<b>12</b>
2.1	Automated Single Mouse Behavioral Phenotyping . . . . .	12
2.2	Multiple Mouse Automated Recognition . . . . .	13
2.3	Automated Behavioral Recognition of Other Animals . . . . .	16
2.4	Shape Matching Methods for Recognition . . . . .	16
<b>3</b>	<b>Methods</b>	<b>18</b>
3.1	Challenges of Multiple Mice Tracking . . . . .	18
3.2	Conditions . . . . .	20
3.3	System Overview . . . . .	21
3.4	Locating the Mice . . . . .	22
3.5	Detecting Occlusion Events . . . . .	22
3.6	Parts Based Tracking . . . . .	23
3.6.1	Background: Matching Contours . . . . .	23
3.6.2	Tracking During Non-Occlusion Events . . . . .	26
3.6.3	Tracking During Occlusion Events . . . . .	28
3.6.4	Keeping Track of Mice Identities . . . . .	31
3.7	Post-Processing . . . . .	32

<b>4 Experiments</b>	<b>34</b>
4.1 Datasets . . . . .	34
4.2 Evaluation Techniques . . . . .	35
4.2.1 Frame by Frame Evaluation . . . . .	35
4.2.2 Automated Evaluation . . . . .	37
<b>5 Results</b>	<b>39</b>
5.1 Frame by Frame Results . . . . .	39
5.2 Comparison with Other Methods . . . . .	40
5.2.1 Ellipse Tracking . . . . .	40
5.2.2 Physics-Based Tracking . . . . .	40
5.3 Automated Evaluation Results . . . . .	41
<b>6 Conclusion</b>	<b>43</b>
<b>A Using Tracking to Automate Behavioral Phenotyping</b>	<b>44</b>
<b>B Evaluation of Human Labeling</b>	<b>47</b>
<b>C Extensions to Other Domains</b>	<b>49</b>
C.1 Side View . . . . .	49
C.2 Semi-Side View . . . . .	51
C.3 Tracking Three Mice . . . . .	52
<b>D Future Work</b>	<b>54</b>
D.1 Improving the Tracker . . . . .	54
D.2 Using Part-Based Tracking for Social Behavioral Phenotyping . . . . .	56
D.3 Fine-Grained Social and Single Mouse Behavioral Phenotyping . . . . .	56
D.4 Applications . . . . .	57
<b>E Algorithms</b>	<b>58</b>

# List of Figures

2-1	Example head and tail classification from Edelman and de Chaumont.	14
3-1	An example of the featureless boundary.	20
3-2	Importance of precision.	21
3-3	Tracker overview.	22
3-4	Edelman's method of foreground extraction.	22
3-5	Belongie's shape context.	24
3-6	Examples of inner-distances.	25
3-7	Examples of indistinguishable figures.	26
3-8	Training template with labeled head and tail locations.	27
3-9	Point to point correspondence between the training template mouse and the current mouse.	27
3-10	Leniency of matching toward rotation and noise.	28
3-11	Virtual examples created for occluding frames.	29
3-12	Training template and corresponding matching.	30
3-13	Rematching an individual mouse.	30
3-14	Mouse identification during occlusions.	32
3-15	Easy and hard to distinguish head and tail locations.	32
3-16	Plot of head and tail coordinates before and after SVM correction.	33
4-1	Examples of orientation scores.	36

4-2	Examples of tracking scores. . . . .	37
5-1	Empirical CDF of distance between true and predicted location. . . . .	42
A-1	Hard to distinguish behaviors from top view. . . . .	45
B-1	Confusion matrix between two labelers. . . . .	48
C-1	Mice occlude from side view much more than from top view. . . . .	49
C-2	Mice can take on many contour forms from the side view. . . . .	50
C-3	The templates used for side view matching. . . . .	50
C-4	Example of semi-side view frame. . . . .	52
C-5	Various example of three mice shown from the top view. . . . .	53

# List of Tables

4.1	Orientation score definition. . . . .	35
4.2	Tracking score definition. . . . .	36
5.1	Orientation score results on Datasets A and B. . . . .	39
5.2	Tracking score results on Datasets A and B. . . . .	40
5.3	Orientation and tracking score results on Dataset A, using the ellipse tracking method. . . . .	40
5.4	Orientation and tracking score results on Dataset B, using the physics-based tracking. . . . .	41
5.5	Summary of automated tracking results. . . . .	42
C.1	Top and side view orientation and tracking results. . . . .	51
C.2	Semi-side view orientation and tracking results. . . . .	52



# List of Algorithms

1	An algorithm used to classify a frame as occluding or non-occluding. .	23
2	Overview of tracking algorithm. . . . .	58

# Chapter 1

## Introduction

The extensive use of mice in basic science and disease modeling has created a need for reliable, high throughput, and high information content behavioral analytic tools. New mouse lines and experiments are rapidly growing in numbers, and most commonly mouse behavior in these experiments is assessed by humans. Unfortunately though, this requires a large amount of human time, and thus progress is hindered by the slow nature of human behavioral assessment [26]. Due to these time constraint, most mouse studies are designed to be short - only a few minutes long - instead of a more informative multi-day design.

Automating behavioral analysis of mice addresses many of the practical issues involved in human behavioral assessment. Mice can be studied for longer periods of time, at significantly lower cost, and the results can be easily reproduced across different labs. Much progress has been made in automating single mouse behavioral phenotyping. This includes automating behavioral recognition such as eating, drinking and grooming. In fact, many commercial systems such as Noldus' Ethovision, Cleversys' HomeCageScan, as well as some open source systems such as the one developed by Jhuang et al. [11] have been created and are actively being used. These automated systems work extremely well on singly-housed mice and have reached hu-

man level performance [11]. Unfortunately, much less progress has been made in automating behavioral phenotyping of mice in the social context.

Automating social behavior recognition of mice is quite challenging. Numerous complications arise that do not occur when studying a single mouse; mice interact in complex ways, frequently occlude one another, and appear very nearly identical.

Many diseases and conditions, (i.e. depression, autism, and Parkinson’s), have a strong social component. Researchers study these diseases in mice by performing social behavioral phenotyping. For instance, a decreased frequency of some social interactions is observed in BTBR mice, an inbred mouse strain that exhibits many autism-like behaviors [3]. This BTBR study was analyzed by humans which labeled the social behaviors, and thus would have greatly benefited from a more streamlined automated behavioral phenotyping. Similarly, Shank 3 mice - another mouse model for autism - were found to have reduced reciprocal interaction as well as reduced nose-to-nose and anogenital sniffing behavior when compared to wild-type mice [19]. Again, these observations were made using a human labeler. Automating the behavioral labeling process will enable scientists engaged in future studies to more quickly and accurately gage social behaviors amongst mice, saving time, and greatly benefiting the research community.

We decide to take a two part approach to the problem. First, we must be able to accurately track individual mice and understand where each part of each particular mouse, such as the head or tail, is located at every point in time. That is, we must create an accurate part-based tracking system. Second, we can use the head and tail positions from the part-based tracker to accurately train a phenotyping system that will classify mouse behavior at all times. In this project, we propose a method that makes great progress towards solving the first part. Additionally, we explore and begin to develop approaches to solving the second.

# Chapter 2

## Background

An integral step in the study of human disease is the development and analysis of analogous mouse models. This analysis can be done through behavioral phenotyping. The controlled environment of mouse studies makes this phenotyping an ideal problem for computer automation. In this section we will discuss successful prior work in the area of automated single mouse behavioral phenotyping, as well as ongoing work in social behavioral phenotyping of mice and other animals. We conclude the section by discussing computer vision approaches that have been applied to solve similar problems in various fields, which lay the ground work for the novel analytical methods proposed later in this thesis.

### **2.1 Automated Single Mouse Behavioral Phenotyping**

Early automation techniques used for single mouse behavioral phenotyping employed sensors capable of assessing position, coarse movements, and instances of specific actions including eating and drinking [9, 25]. These techniques are limited to the analysis of relatively simple pre-programmed behaviors.

More recently, computer vision systems have been developed to support more complex behavior analysis, most of which are based on visual tracking of the mouse [23]. This allows the system to recognize position dependent behaviors, such as hanging and drinking, but lacks the ability to correctly analyze fine-grained behaviors such as grooming. Dollar et al. use spatio-temporal interest point descriptors in order to analyze fine-grained behavior [6]. Jhuang et al. use a computer vision system based on a hierarchical model [11]. This model is motivated by the processing of information in the dorsal stream in the primate brain. This system incorporates both position and motion information of the mouse, thus allowing it to accurately recognize the fine-grained behaviors that previous systems failed to recognize. This hierarchical method performs on par with human performances.

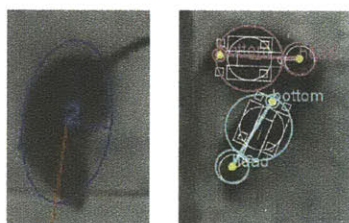
## **2.2 Multiple Mouse Automated Recognition**

Recently, much interest has formed in extending the automation of behavioral phenotyping to social behaviors, as this is needed to study diseases with a strong social component. Studying the social behavior of mice is much more challenging, since mice can interact in complex ways.

There is interest in studying many different social behavioral phenotypes. One common experiment is the Noldus three-chamber social test which is used to characterize the sociability of mice. In this experiment, the subject mouse is placed in the middle chamber of a three chamber box. An unfamiliar mouse is then placed in a cage on one of the side chambers, and an empty cage is placed on the other side. The subject mouse is then allowed to freely explore all three chambers. The social interaction of the mice is studied by analyzing time spent in each chamber and time spent in proximity to the unfamiliar mouse. Using this three chamber setup, only simple behavioral phenotypes can be studied.

Another common method of phenotyping is behavioral categorization, where the mouse’s behavior is characterized at each time point. For example, Peca et al. characterized nose-to-nose interactions, as well as nose-to-anogenital region interactions in two interacting mice [19]. Research into automating this type of phenotyping is still ongoing, and multiple approaches have been developed.

Edelman developed a method based on ellipse tracking and used it to create a trainable system to characterize social behavior of two mice [7]. Each mouse is tracked by an ellipse. The ellipses are then used to compute distance features - such as center to center distance - allowing us to train a machine learning system to recognize social behaviors of mice. Since our interest mainly lies in accurate head and tail detection, ellipses do not model the mouse well, as can be seen in Figure 2-1(a) and the performance of this method is far from human level.



(a) Edelman (b) de Chaumont

Figure 2-1: Example head and tail classification from Edelman and de Chaumont. Both methods often lack the ability to get the precise location of the head and tail.<sup>2</sup>

Additionally, the orientation detector, which is used to distinguish the head from the tail, is based on the mouse’s motion. Thus, when the mouse moves backwards, which happens when a mouse comes down from a rearing, the orientation detector fails.

A recent method proposed by de Chaumont is physics-based and tracks the mice by using its skeleton, joints, and physical limits imposed by the joints [5]. Using absolute rules to hard-code events, the system classifies mouse behavior. For example, to define a oral-genital contact the method uses the following rule: the distance from

---

<sup>2</sup>Figure 2-1(b) obtained from [5]

the center of one mouse's head to the start of the other mouse's tail is less than or equal to 1.5 cm. This method produces interpretable results and allows new behaviors to easily be defined and analyzed. However the system does not perform well under many conditions, including prolonged mouse interactions [5]. Under these conditions, the system needs to be corrected by humans - making it impractical for longer studies. Furthermore, as can be seen in Figure 2-1(b), precise locations of the head and tail region are not obtained, decreasing the accuracy of the behavioral labels. Lastly, it is unclear whether hard-coded rules are accurate representations of behaviors. For example, a mouse passing by another mouse's anogenital region will be classified as head-anogenital region contact, despite no form of social interaction occurring between the two mice.

Commercial systems, such as Noldus' Ethovision, recognize behaviors such as sniffing, but provide no evaluation of their results or detailed algorithms. During non-occlusion, these systems tend to use the extremities of the mouse to classify the head and tail regions [5]. The direction of movement between consecutive frames is used to distinguish between the head and tail [5]. This method fails during rearing, where the mouse may appear to go backwards, causing the head and tail to be reversed. Lastly, one of the problems with many commercial systems, is that they are not easily extendible to tracking when the mice occlude one another. Most software cannot handle this. For example, ViewPoint can only track mice if they are not occluding one another. Other programs, such as Cleverysys' SocialScan and Noldus' Ethovision can only track the mice during interactions if additional distinguishing traits are given. For example, these programs work when the mice are marked with colors or mice of different size or appearance are used. This is not practical, since numerous experiments have shown that marking mice with colors changes their behavior. In addition, we cannot always control the size or appearance of the mouse, reducing the usefulness of these methods.

## 2.3 Automated Behavioral Recognition of Other Animals

Much progress has been completed in automating social behavior of other animals. Recent accomplishments range from studying zebra fish using a vision based system [20] to monitoring a honey bee cluster using stereo vision [16]. Additionally, ant social behavior has been successfully studied, where researchers classify three behaviors: head-to-head, head-to-body, and body-to-body contact [1]. This system works quite well for ants, but relies on the fact that ants are rigid, which does not apply to mice. In addition, scientists have studied *Drosophila* (a type of small fly) social behavior such as aggressive tussling, courtship, lunging, circling, and copulation [4]. The behaviors were defined in terms of *Drosophila* spatial features (e.g. velocity, acceleration) and rules were hard-coded by an expert, in much the same fashion as [5]. Work in this field is extremely promising, but unfortunately relies on the rigidity of flies, and thus cannot be extended to mouse social behavior.

## 2.4 Shape Matching Methods for Recognition

The controlled setting of the laboratory allows for high quality videos of mice to be obtained. We can then use computer vision algorithms to get high accuracy segmentation of the mouse from the background. During mouse occlusions, little feature and texture information can be obtained to help distinguish two mice (Figure 3.1). A lot of the information which allows humans to recognize a mouse and its parts is encoded in the mouse shape (e.g. its contour). Ideally, we want a method that allows us to find the exact location of the head and tail on the mouse contour.

One approach is to first define a training template - an example mouse's contour with the head and tail locations labeled. We can then find a point to point correspon-



dence from every point on the training template to every new point on our current mouse. This allows us to find the head and tail points by simply seeing to where the head and tail points of the template are mapped. This matching can be achieved using various shape matching methods (see [27]).

The approach above relies on point to point correspondences. Thus, methods based on statistical moments, Fourier descriptors, Hausdorff distance, and medial axis transforms, do not fit our framework as they describe the shape based on its global properties. On the other hand, methods such as spin image descriptors [12, 13] and shape context [17, 2] can be used to find the point to point correspondences. In these methods, a descriptor that encodes the position relative to all other points on the shape is used. These methods perform well when similar points on the shapes being matched have similar distribution of relative positions. A drawback of these methods is that they are not invariant to non-rigid transformations.

Curvature maps can also be used as shape descriptors. In this approach, the shape descriptor is found based on sampling points and recording the curvature at each sample [8]. This method is not invariant to non-rigid transformation, and additionally is experimentally not robust to noise and bending in the shape [10].

The inner-distance can be applied as an extension to shape context matching [15]. Instead of using standard Euclidean distance to measure the relative positions, inner-distance (the distance between two points within a shape) is used. The inner-distance is insensitive to articulation and empirically found to capture part structures. For these reasons, we choose to use this approach, which is described in more detail in Chapter 3.

# Chapter 3

## Methods

In this section we describe a method for tracking the head and tail of the mouse with high accuracy, both during non-occluding and occluding frames. We begin by discussing the difficulty of tracking. We then describe shape context [17] and inner-distance shape context [15], both of which are essential for our head and tail detector. We conclude this chapter by describing our part-based tracking algorithm, and a post-processing technique that can be used to reduce head and tail confusions.

### 3.1 Challenges of Multiple Mice Tracking

The extension from single mouse automated systems to multiple mouse systems is non-trivial. Single mouse phenotyping systems perform well because of the ease of tracking an individual mouse. Determining the location of a single mouse is trivial, and allows for highly accurate features such as mouse width and height to be calculated. In addition feature-based systems look for features only where the mouse is located. Extending this to multiple mice, we encounter multiple problems - especially when the mice occlude one another. Several problems are listed below:

1. **Complex Occlusions**

The mice can crawl over and under one another, fight and roll around, and generally interact in very complicated and fast ways. This can pose difficulty for automated tracking algorithms.

## 2. **Featureless**

During occlusions and close interactions our interest lies in distinguishing the mice. Unfortunately, standard segmentation and contour detection algorithms cannot detect the boundaries between the mice due to lack of local identifying features, such as luminance, chrominance and texture differences. Figure 3.1 shows a scenario in which even a human cannot detect the boundary locally when only a small patch is shown. It is not until the patch is large enough to contain contextual information that a human can detect the boundary.

## 3. **Highly Deformable**

Mice can deform into many shapes, sizes, and orientations. Hence, trying to fit them using simple models, such as ellipses, does not produce optimal results.

## 4. **Identical Appearance**

Mice can vary in shape, color, and size depending on their age and species. In order to create a reliable, general purpose tracker, it needs to handle two mice of identical color and size. Keeping track of identity is extremely important. For example, when studying one knock-out mouse and one wild-type mouse, scientists may want to know the corresponding behavior of each mouse. This can be quite challenging, especially after a complex interaction where occlusion has occurred.

## 5. **Unpredictable Motion**

Mice move and change direction quickly and even move backwards. This makes it difficult to use motion models to predict future location or even just to predict

head and tail locations.

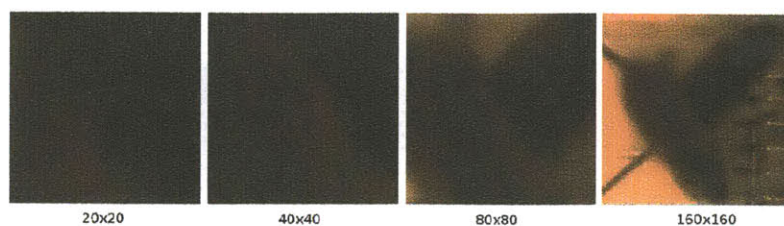


Figure 3-1: An example of the featureless boundary. From left to right, each image zooms out on the center of the patch. In the most zoomed in image,  $20 \times 20$  pixels, the boundary is not detectable to a human observer. While when we zoom out to  $80 \times 80$  pixels, the boundary becomes detectable to a human.<sup>2</sup>

## 3.2 Conditions

We aim to develop a system that will work under the following conditions:

1. **No modifications to mice**

When creating an automated system for studying mice, we want to observe the mice in their natural state without affecting their behavior. This includes coloring, shaving, or marking in any other way that would potentially aid with tracking and identification, as well as less obvious changes, such as using bright lights, which bother mice.

2. **High quality video**

We assume that the camera is stationary and lighting has been set up to minimize glare, shadows, and reflection. In addition, we assume a high quality foreground extraction that gives a good quality contour of the mouse both during occlusions and non-occlusions.

---

<sup>2</sup>Figure obtained from [7]

### 3. Minimal human intervention

Our goal is to have a self correcting system that can operate with minimal human intervention.

### 4. Requirement for precise head and tail locator

Previous methods have focused on getting a general fit for the location of each mouse. While this is important, it is also extremely important to get a precise head and tail location - that is the nose location and the base of the tail location on the mouse. This allows for much more meaningful features and interactions to be computed between the mice. See Figure 3-2 for an example of why precision is important.

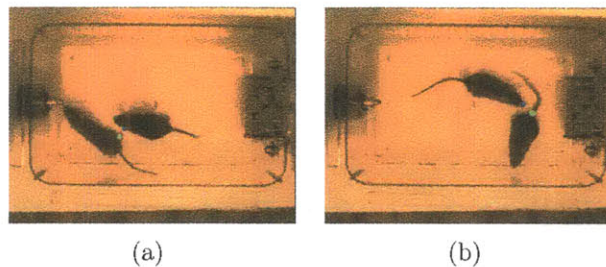


Figure 3-2: Importance of precision. The figure on the left shows slightly mislabeled head and tail locations, while the figure on the right shows correctly labeled head and tail. While the distance between the labeled head and tail in the two figures is similar, on the left the mice are not interacting at all, while on the right, a nose-to-anogenital interaction is occurring.

## 3.3 System Overview

Figure 3-3 shows an overview of the components of the system and the sections in which they are described. For more detailed pseudocode see Appendix E.

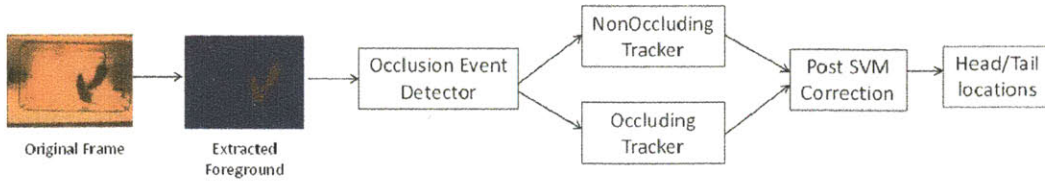


Figure 3-3: Tracker overview. We take as input a foreground frame from our video. We then extract the foreground (Section 3.4), and determine whether we have an occlusion event (Section 3.5). We then track the head and tail of the mouse both during non-occluding (Section 3.6.2) and occluding (Section 3.6.3) frames. Lastly, we do a post-processing SVM correction (Section 3.7). Finally, we output the head and tail locations of the mice.

### 3.4 Locating the Mice

To locate the mice we use the method developed by Edelman [7], which combines the usage of background subtraction and color clustering to find a good foreground, as is shown in Figure 3-4.

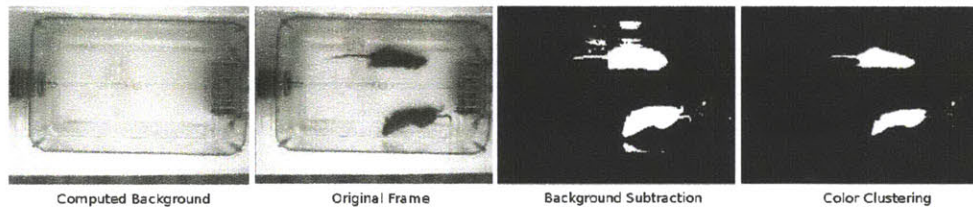


Figure 3-4: Edelman’s method of foreground extraction.<sup>3</sup>

### 3.5 Detecting Occlusion Events

After the foreground is extracted and the mice are located, detecting occlusion events is done using Algorithm 1. More generally, this algorithm looks at the largest two components in the foreground. If they are roughly of equal size, it classifies the frame

<sup>3</sup>Figure obtained from [7]

as a non-occluding event, otherwise, it classifies it as an occlusion event. For simplicity and efficiency we use this reliable algorithm when studying two mice. Extending this to more than two mice could be done using a Gaussian Mixture Model to track the mice and using the distance between mouse cluster centers to detect occlusion events [7].

---

**Algorithm 1** An algorithm used to classify a frame as occluding or non-occluding.

---

```

function COMPUTEOCCLUSION(fg)
  [fg1, fg2] ← computeTwoLargestFgComponents(fg)
  isOcclusion ← area(i1) > threshold * area(i2)
  return isOcclusion
end function

```

---

## 3.6 Parts Based Tracking

Our part-based tracking method uses shape context matching to find a point to point correspondence between a predefined template and our current mouse. We use an extension of shape context - the inner-distance shape context - as described in Section 3.6.1. Sections 3.6.2 and 3.6.3 describe our tracker for non-occlusion and occlusion frames. Finally, Sections 3.6.4 and 3.7 describe a method to keep track of mice identities, as well as a post-processing step that can be used to reduce head and tail confusions.

### 3.6.1 Background: Matching Contours

#### Matching with Shape Context and Transformation Estimation

Belongie et al. present the idea of shape context [17]. Shape context provides a way to quantitatively measure shape similarities as well as find point correspondences. Pick  $n$  sample points,  $p_1, p_2, \dots, p_n$ , on a shape's contour. For each point  $p_i$ , looking at the relative distance and orientation distribution to the remaining points gives a

rich descriptor of that point. A compact distribution for each point  $p_i$  is obtained by defining the histogram  $h_i$  - with log-polar uniform bins - of relative coordinates to the remaining  $n - 1$  points:

$$h_i(k) = \#\{p_j : j \neq i, p_j - p_i \in \text{bin}(k)\}.$$

The histogram defines the shape context of  $p_i$  (for a concrete example see Figure 3-5).

A benefit to shape context is the obvious translational invariance. Scale invariance can easily be implemented by normalizing distances, and rotational invariance can be obtained by measuring angles relative to the tangent at the point. It has been empirically shown that shape context is robust to deformations, noise, and outliers [2], making it applicable to matching mice contours to templates.

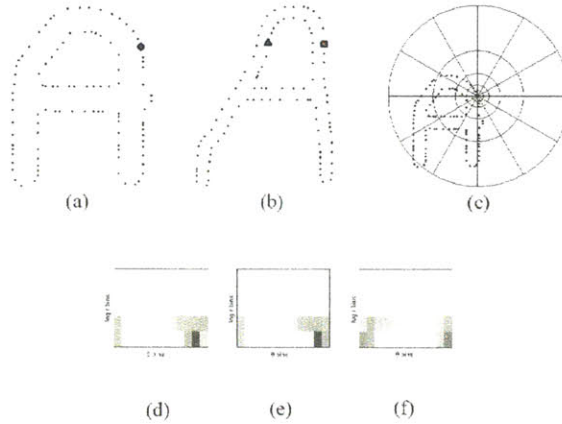


Figure 3-5: Belongie's shape context. (a) sampled points along the contour of the first version of the letter A. (b) sampled points along the contour of the second version of the letter A. (c) example of the log-polar bins used to obtain the shape context. (d) shape context for the highlighted point in (a). (e) the shape context for the square point in (b). (f) the shape context for the triangle point in (b). The shape context of (d) and (e) is similar since they are closely related points (same location on different versions of the letter A), while point (f), has a very different shape context. <sup>5</sup>

To match two shape context histograms, the  $\chi^2$  statistic is used. Belongie et al. use a combination of thin plate splines transformations and shape context to do

<sup>5</sup>Figure obtained from [28]



shape comparisons. Given the points on the two shapes, a point correspondence is found using weighted bipartite matching, with the weights equaling the shape context between a point on the first shape and a point on the second shape. Thin plate splines are used to find a transformation between the two shapes. The transformation is then applied to the points in the original shape, and the process of finding a point to point correspondence is repeated. The steps of finding correspondences and estimating transformation are iterated to reduce error. This method gives us a transformation between the two shapes and a similarity cost which is composed of a weighted combination of the shape context distance, appearance difference, and bending energy. This iterative method may converge only to a local minima and finding a transformation that can handle non-rigid shape deformations is difficult. Thus the applicability to non-rigidly deformed mice is limited.

### **Our Method: Matching with Inner-Distance Shape Context and Continuity Constraint**

The standard shape context uses Euclidean distance to measure the spatial relation between points on the contour. Using inner-distance instead of Euclidean distance provides more discriminability for complex shapes [15]. The inner-distance is defined as the length of the shortest path between two points within the shape (see Figure 3-6).

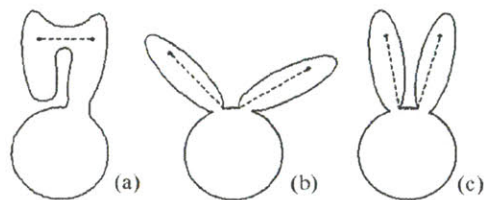


Figure 3-6: Examples of inner-distances shown as a dashed lines. <sup>6</sup>

Figure 3-7 shows some examples of instances for which Euclidean distance would fail to distinguish shapes, while inner-distance can easily do so. We follow the method described by Ling et al. using inner-distance and inner-angle as our shape descriptors with shape context, and utilize the code provided in [14]. In addition, we include a continuity constraint for shape context matching, and use dynamic programming to efficiently solve the matching problem (see [15] for details). This allows us to skip the transformation step used with the classic shape context, and is empirically found to perform better for mice shape matching.



Figure 3-7: Examples of indistinguishable figures. These figures cannot be distinguished using shape context with Euclidean distance, but can easily be distinguished using the inner-distance.<sup>8</sup>

### 3.6.2 Tracking During Non-Occlusion Events

Using the inner-distance shape context described in Section 3.6.1, we obtain a matching cost and point to point correspondence between any two shapes. Inner-distance shape context is invariant to translation, rotation, and scale, and has some natural leniency towards deformations, noise, and outliers. Thus, we only use one training template, shown in Figure 3-8, to match our non-occluding frames. In this template, we label the head and tail points.

---

<sup>6</sup>Figure obtained from [15]

<sup>8</sup>Figure obtained from [15]

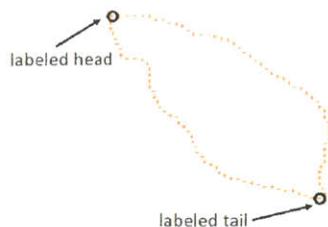


Figure 3-8: Training template with labeled head and tail locations.

During non-occlusion, the mice do not obstruct each other, therefore after extracting each mouse's foreground we can easily obtain its contour. Each mouse is then matched to the training template. The matching, as discussed in Section 3.6.1, gives us a cost and point to point correspondence. Using the point to point correspondence we obtain the corresponding head and tail locations of each mouse. Figure 3-9 shows an example of the point to point correspondence obtained by this algorithm. Figure 3-10 shows the leniency of the matching towards noise and mouse deformations.

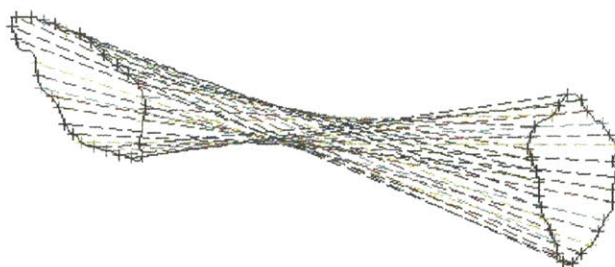
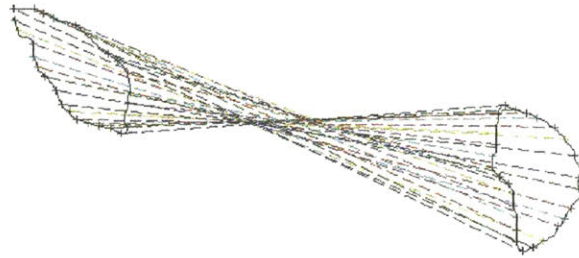
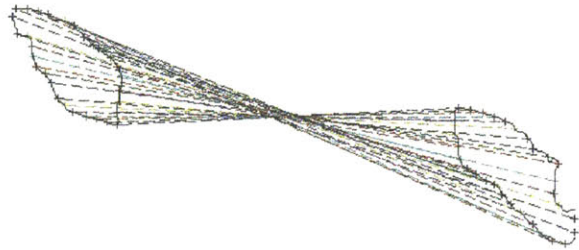


Figure 3-9: Point to point correspondence between the training template mouse (left) and the current mouse (right).



(a)



(b)

Figure 3-10: Leniency of matching toward rotation and noise. (a) despite the mouse's body being deformed in rotation, heads and tails are still successfully matched. (b) despite some noise disrupting the contour, very good matching is still obtained.

### 3.6.3 Tracking During Occlusion Events

Finding the head and tail location during frames in which the mice occlude one another is a more difficult task, thus we can no longer use the simple one-template technique used for non-occluding frames. Mice interactions are not generalizable, and one example of two mice interacting will not easily extend to other examples. One possible approach would be to label hundreds of occluding mouse contours with the head and tail locations for the two mice. Based on these labels, we can calculate the matching cost between an example frame and each of the templates, and choose the template which minimizes cost. Unfortunately, this approach involves significant human involvement in labeling each of the many example templates with head and tail locations. In addition, it is not obvious how to select templates from real images

that would be exemplary of generalized interactions between two mice.

Instead we take a more automated approach to develop our training templates. Our approach requires a human to label the head and tail locations of only a single template. Using this template, we construct virtual examples using two versions of the template, and combining them by taking the union of the two overlaid one on top of another, with one template rotated and then shifted. More specifically, we use twelve rotations between 0 and 360 degrees, as well as shifts of 10 pixels at a time (Figure 3-11). The head and tail location of the original template are known, thus we can easily find the head and tail of the rotated and shifted template. Using this approach we can generate many templates by labeling just one.



Figure 3-11: Virtual examples created for occluding frames.

Using the method described we obtain roughly 300 templates. For each template we know the head and tail locations based on how the template was generated, as well as which mouse every point in the template comes from. We calculate the match cost between the contour and every template and select the template that minimizes cost. One drawback to this approach is that it does not always most accurately describe the mouse, often the head and tail are switched. To resolve this issue, we use the template matching as an alignment step. We select the best matched template, and use it to split the contour points into two groups, one for each mouse, see Figure 3-12. We then rematch the individual points to a single mouse template, which works extremely well in accurately distinguishing the head and tail, see Figure 3-13. In order to save time, we perform the alignment step at low resolution - using only 30 contour points, while performing the matching for the alignment at higher resolution

with 200 points.

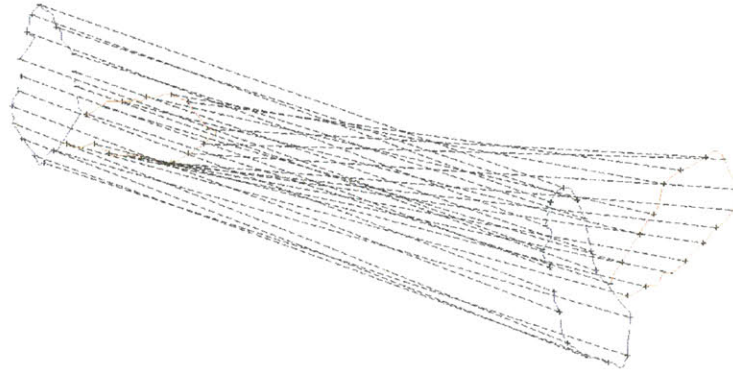


Figure 3-12: Training template and corresponding matching. Point to point correspondences are shown between the lowest cost matching template (left), and the current frame (right). These are used to separate the contour into the points from the two mice (shown in red and blue).

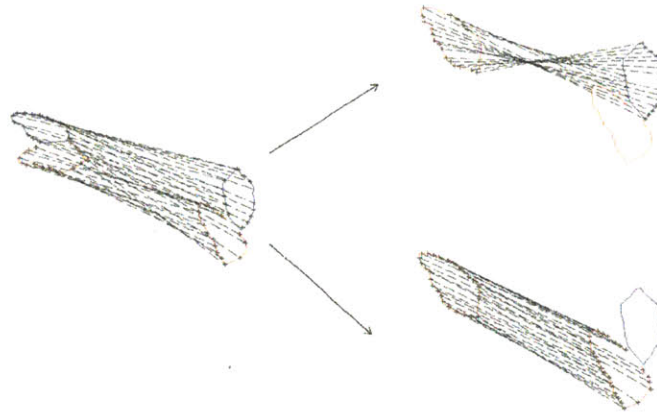


Figure 3-13: Rematching an individual mouse. Mice are rematched to a single template to improve precision. Originally the head and tail location of the blue mouse are switched (left). Rematching corrects this (right).

Below is a summary of the process we perform for each frame:

1. Match the frame and calculate matching cost to each of the 300 low resolution templates with 30 points per contour.

2. Choose the matching which minimizes cost.
3. Rematch the original frame with a high resolution version (200 points per contour) of the template that minimizes cost.
4. Use this matching to distinguish points on the contour that come from each mouse,  $C_{m_1}$ , and  $C_{m_2}$ .
5. Match each  $C_{m_1}$  and  $C_{m_2}$  to a single mouse template to find head and tail location, and use these as the final head and tail predictions.

### 3.6.4 Keeping Track of Mice Identities

Keeping track of mice identities is extremely important, as discussed previously. To do so, we use the information obtained from the original matching to determine which points come from which mouse as described in the previous section. Using these points we create a polygon for each mouse  $m_1$ ,  $m_2$  for our current frame, and  $p_1$ ,  $p_2$  for the previous frame. If,

$$area(m_1 \cap p_1) + area(m_2 \cap p_2) > area(m_2 \cap p_1) + area(m_1 \cap p_2)$$

then the current mouse 1 corresponds to the mouse labeled 1 in the previous frame and the current mouse 2 corresponds to the mouse labeled 2 in the previous frame. And vice versa if not. This is based on the idea that if mice in two consecutive frames are one and the same, their area of intersection should be large, as seen in Figure 3-14.

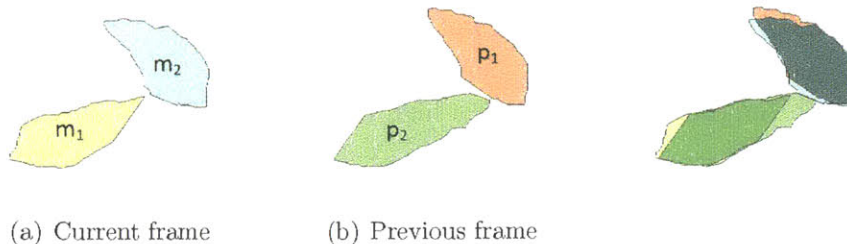


Figure 3-14: Mouse identification during occlusions. The intersection of  $m_1$  with  $p_2$ , and  $m_2$  with  $p_1$  is the best fit. Hence, mouse 1 in the current frame corresponds to the mouse 2 in the previous frame, and mouse 2 in the current frame corresponds to mouse 1 in the previous frame.

### 3.7 Post-Processing

Occasionally the mouse loses identifying contour information. For instance, Figure 3-15(a) shows mouse shape where the head and tail are obvious. Figure 3-15(b) shows one in which even a human may struggle distinguishing the head and the tail.

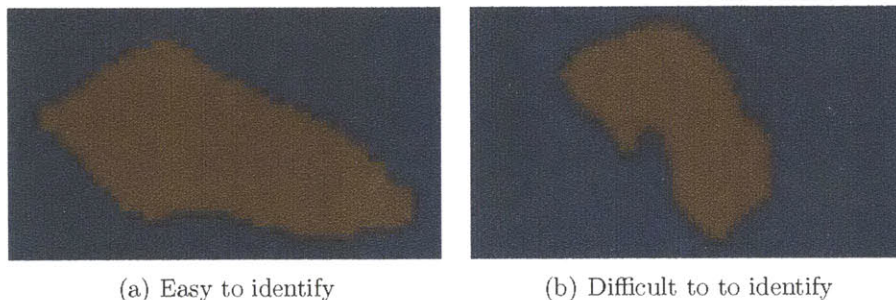


Figure 3-15: Easy (left) and difficult (right) to distinguish head and tail locations.

Since the majority of the time contour information allows us to identify head and tail without difficulty, our algorithm rarely switches the two. We devise a post-processing technique which attempts to correct frames in which the head and tail are swapped.

Assume we have a list of head coordinates and tail coordinates for a mouse,  $l_{head}$  and  $l_{tail}$  respectively, where:



$l_{head} = (x, y, t) : (x, y)$  is the coordinate of the predicted mouse's head at time  $t$ .

$l_{tail} = (x, y, t) : (x, y)$  is the coordinate of the predicted mouse's tail at time  $t$ .

A plot of the two sets of data,  $l_{head}$  and  $l_{tail}$ , is shown in Figure 3-16(a).

We then locally train an SVM classifier on  $l_{head}$  and  $l_{tail}$  to distinguish the two categories. We change the head and tail labels for time points for which the SVM classifier is not in agreement with the original label (ignoring time points where the SVM predicts both coordinates as either head or tail). This corrects many of the misclassified head and tail cases, as shown in Figure 3-16(b).

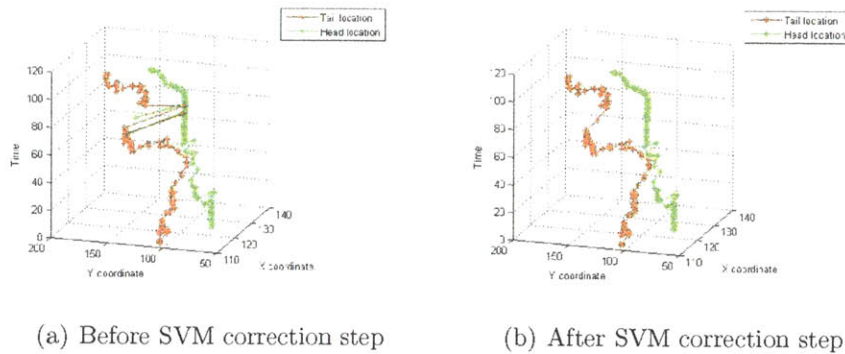


Figure 3-16: Plot of head and tail coordinates before and after SVM correction.

# Chapter 4

## Experiments

### 4.1 Datasets

Tracking performance is evaluated on two different types of videos:

1. **Dataset A:** The first dataset we use for evaluation is from Edelman [7]. This dataset contains both top and side view footage of two mice interacting in a social context. The dataset contains 22 recordings of C57BL/10J background mice with nearly identical brown coats. Each recording lasts at least 10 minutes, and consists of side view and top view recordings, each taken at 30fps and resolution  $640 \times 480$ . Of these 22 recordings, 12 have been annotated with mice social behaviors. We use these 12 videos to evaluate our system, similarly to Edelman [7].
2. **Dataset B:** The second dataset consists of the demo video provided by de Chaumont’s with his system, the Mice Profiler [5], taken at resolution  $320 \times 240$ . This short video (roughly 30 seconds) consists of two mice interacting in a social context. We use it to confirm our approach can be extended beyond videos recorded under our own lighting and mouse conditions (Dataset A).

## 4.2 Evaluation Techniques

We use two methods to evaluate our results. The first technique is a frame by frame evaluation described in detail in Section 4.2.1. The second technique, described in detail in Section 4.2.2, is automated. This provides for less human bias and is more thorough end to end.

### 4.2.1 Frame by Frame Evaluation

We evaluate the results on the two datasets described earlier. The first dataset, from [7], consists of 12 social behavior videos each approximately 10 minutes long, corresponding to more than 200,000 frames. Evaluating every frame is impractical, so we use a sampling approach, as described by Edelman [7]. For each video, we randomly sample 50 frames during occlusion, and 50 frames during non-occlusion, for a total of 1200 frames across the twelve videos. For the second dataset, we evaluate 100 random non-occluding frames and 100 random occluding frames.

To evaluate the results, we overlay the computed head and tail locations on the frame, and manually assign a score for each mouse. Each mouse receives an orientation score which reflects whether or not the orientation of the mouse is correct, defined in Table 4.1. Each frame receives two orientation scores, one for each mouse, an example of which is shown in Figure 4-1.

<b>1</b>	Head and tail are both on correct “half” of the mouse.
<b>2</b>	One head or tail is on the wrong “half” of the mouse.
<b>3</b>	Head and tail are swapped - i.e. both head and tail are on wrong “half” of the mouse.

Table 4.1: Orientation score definition.

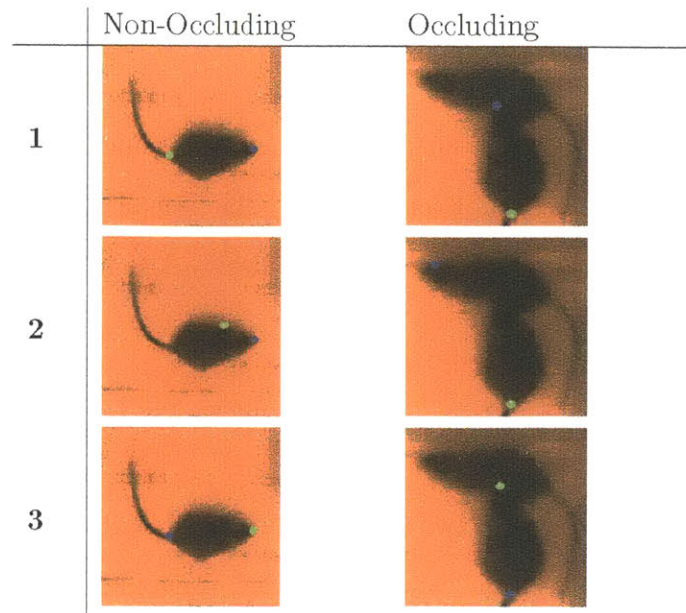


Figure 4-1: Examples of orientation scores. The colored circles indicate the predicted location of the head and tail, shown in blue and green respectively.

Additionally, a more detailed head and tail tracking score is assigned to each mouse. Each mouse receives one score for the head and one score for the tail, for a combined total of four tracking scores per frame, as defined in Table 4.2. Examples of the tracking scores are shown in Figure 4-2.

<b>1</b>	Head or tail are completely correct: exact match of head or tail location.
<b>2</b>	Head or tail are generally correct: tail is in tail “area” and head is in head “area” of mouse (i.e. tail placed where a protruding leg is, or head placed where ear is).
<b>3</b>	Head or tail are incorrect.

Table 4.2: Tracking score definition.

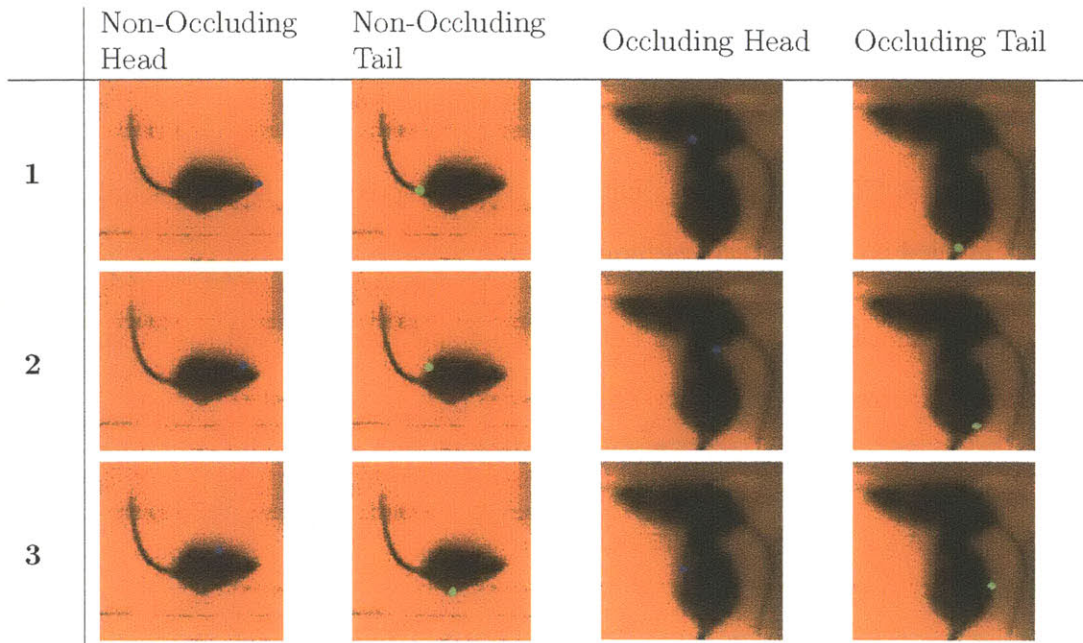


Figure 4-2: Examples of tracking scores. The colored circles indicate the predicted location of the head and tail, shown in blue and green respectively.

## 4.2.2 Automated Evaluation

To automate the process, we manually create a new database with annotated head and tail locations. A human then labels the true head and tail locations of the mouse. The head is labeled at the location of the nose, or in situations in which the nose cannot be directly seen (e.g. it is occluded by the second mouse), the location where it would be is interpolated. Similarly, the tail is labeled as the base of the tail, and interpolated if it cannot be seen.

We create a database with a continuous labeling of 5000 frames taken from a randomly selected video from Edelman’s dataset [7]. This is chosen to allow us to test properties such as number of mouse identity switches occurring, which we would not be able to test in a non-continuous evaluation. In addition, this dataset reduces human bias and vagueness involved in frame by frame labeling, and thus allows for easier and accurate evaluation of performance across multiple methods, using the

following measures:

1. Percentage of frames in which head and tail are swapped.
2. Empirical CDF of distances between true and predicted head (assuming head is labeled on correct side of mouse).
3. Empirical CDF of distances between true and predicted tail (assuming tail is labeled correct side of mouse).
4. Number of mouse identity switches.

# Chapter 5

## Results

### 5.1 Frame by Frame Results

We evaluate the orientation and tracking scores for the two datasets described in Chapter 4. Orientation scores are consistent across datasets (Table 5.1). The post-correction step provides an increase in performance. More specifically, for non-occluding frames, the increase in performance is small. However, for occluding frames, the increase is almost 5%. This is extremely promising and important, since the accuracy of the orientation is necessary for many applications of our tracker, such as distinguishing between nose to nose behavior and nose-to-anogential behavior.

	Dataset A			Dataset B		
	<b>1</b>	<b>2</b>	<b>3</b>	<b>1</b>	<b>2</b>	<b>3</b>
<b>Non-Occluding</b>	94.33%	.08%	5.58%	95%	0%	5%
<b>Occluding</b>	84.42%	1.42%	14.17%	84.5%	0%	15.5%
<b>Non-Occluding with Correction</b>	96.92%	.08%	3%	100%	0%	0%
<b>Occluding with Correction</b>	89.5%	1.42%	9.08%	100%	0%	0%

Table 5.1: Orientation score results on Datasets A and B.

Overall, tracking performance is higher on Dataset A compared to Dataset B (Table 5.2). This is likely explained by the fact that we use the training templates

from Dataset A, creating a bias. This could also be due to the fact that Dataset A has higher resolution, providing more detailed contour information. Despite this, we obtain precise results, especially for non-occluding frames.

	Dataset A			Dataset B		
	<b>1</b>	<b>2</b>	<b>3</b>	<b>1</b>	<b>2</b>	<b>3</b>
<b>Non-Occluding</b>	93.42%	5.38%	1.21%	85.5%	12.5%	2%
<b>Occluding</b>	73.88%	1.84%	7.75%	62.75%	34.5%	2.75%

Table 5.2: Tracking score results on Datasets A and B.

## 5.2 Comparison with Other Methods

### 5.2.1 Ellipse Tracking

We use our own scoring definitions from 4.2.1 to evaluate the ellipse tracking method, Table 5.3.

	Dataset A		
	<b>1</b>	<b>2</b>	<b>3</b>
<b>Non-Occluding Orientation</b>	77.25%	.25%	22.5%
<b>Occluding Orientation</b>	72.5%	1.17%	2.63%
<b>Non-Occluding Tracking</b>	68.3%	29.4%	1.7%
<b>Occluding Tracking</b>	39.54%	43.54%	16.91%

Table 5.3: Orientation and tracking score results on Dataset A, using the ellipse tracking method.

We obtain more precise results using our method than the ellipse tracker, especially during non-occlusion frames. The ellipse tracking method gives reasonable results (i.e. tracking score of 1 or 2) the majority of the time, but lacks the precision to locate the head and tail exactly.

### 5.2.2 Physics-Based Tracking

We compare our own method to the physics-based tracking method [5]. Due to variations in video conditions, the physics-based tracking method does not perform



well on Dataset A. Therefore, we make the comparison only on Dataset B. We sample 100 random non-occlusion frames and 100 random occlusion frames and compare results for these samples (Table 5.4). The physics-based tracking method locates a head and tail circle, thus we adjust our scoring to define the precise location within the mouse, instead of at the precise location on the contour. The physics-based tracking method has high orientation score, but lower tracking scores, meaning it lacks the ability to find the precise head and tail location, which is crucial in determining many behaviors, such as nose-to-nose contact.

	Dataset B		
	<b>1</b>	<b>2</b>	<b>3</b>
<b>Non-Occluding Orientation Score</b>	100%	0%	0%
<b>Occluding Orientation Score</b>	100%	0%	0%
<b>Non-Occluding Tracking Score</b>	75.5%	24.25%	.25%
<b>Occluding Tracking Score</b>	46%	51%	3%

Table 5.4: Orientation and tracking score results on Dataset B, using the physics-based tracking.

### 5.3 Automated Evaluation Results

We compare the automated performance of our method to the ellipse tracker [7]. Our method produces less head/tail swaps than the ellipse tracker (Table 5.5), likely explained by the fact that our method is based on the shape of the mouse, rather than the primary direction of motion. When orientation is swapped by our method, it is brief, and occurs only in frames where mouse shape is lost. With the ellipse tracker, however, orientation is swapped whenever the mouse’s primary direction of motion is not forward. These swaps can last for multiple seconds, resulting in orientation reversal for hundreds of frames.

Our method total number head/tail swaps	12.26%
Ellipse tracker total number head/tail swaps	19.02%
Our method total number identity swaps	0
Ellipse tracker total number identity swaps	2

Table 5.5: Summary of automated tracking results.

Our method more frequently gives precise matching of head and tail location of the mouse (Figure 5-1). In fact, in more than 80% of frames, the distance between the true and predicted head/tail location is less than five pixels. Such precision is seen in only around 30% of the frames using the ellipse tracker. This emphasizes the advantage of our method’s precision results, over alternative methods, such as ellipse tracking.

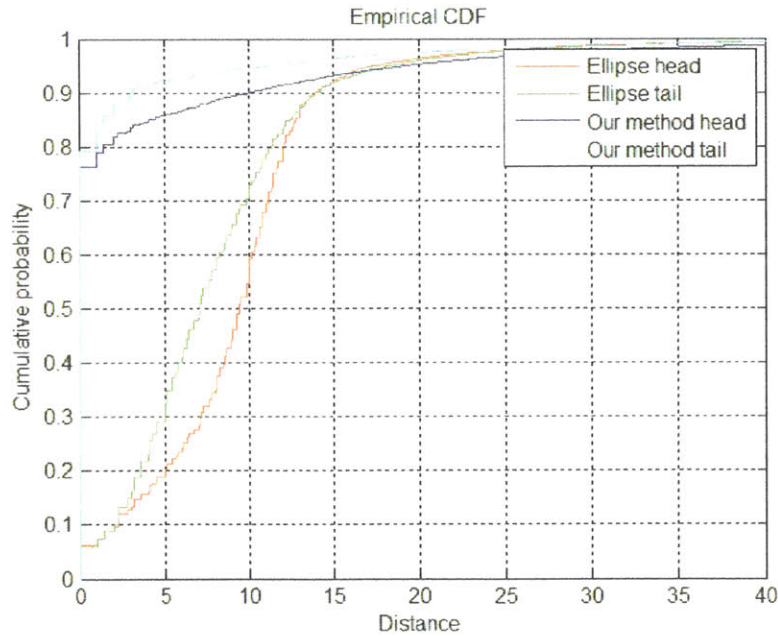


Figure 5-1: Empirical CDF of distance between true and predicted location.

# Chapter 6

## Conclusion

We have made large contributions towards automating social behavioral phenotyping. Our automated tracker can accurately track the head and tail locations of the mice by utilizing the extensive amount of information encoded in mouse contours. The tracker is based on finding a point to point correspondence between mouse templates with predefined head and tail location and the mouse to be classified.

Our part-based tracker is more precise than existing methods. Additional advantages of our tracker include self-correction, avoidance of mistakes associated with motion based methods, and easy extension to new setups.

The tracker can be used to obtain useful statistics about mice and with some modifications can be extended to create an automated end-to-end phenotyping system.

# Appendix A

## Using Tracking to Automate Behavioral Phenotyping

As discussed in Chapter 5, our part-based tracking technique tracks the head and tail location of the mouse more accurately than other leading techniques. In this section, we explore automating social behavioral phenotyping.

We approach the task of automating social behavioral phenotyping by using position based features to train a classifier [7]. We propose to use three main points in the mouse: the head, tail, and center. Thus we are limited to features defined based on these positions, for instance distance between the heads of the mice or the velocity of a mouse. Using this approach we face the following challenges:

1. **Vaguely defined events:** Many social events are difficult to define in terms of position features due to two factors. First, there may not be enough information from the top view. For instance, Figure A-1 shows two frames from the top view, one of which is a nose-to-head social behavior and the other is a background frame. Without additional contextual information - such as side view - there is not enough information engrained in the head and tail location to indicate that the mice are actually interacting.

2. **Limitations of using tracking data:** We train our system using the tracking method from Chapter 3. We use this information to obtain position features of the mouse - i.e. head to head distance. This implies that whenever a mouse's head or tail is misplaced, our classifier receives inaccurate position features as input, making it difficult to obtain an accurate classifier. As an alternate approach, we could hand label head, center, and tail locations for our training set, to allow accurate features when training the classifier.
3. **Limitations of training behavioral data:** As we see in Appendix B, the behavioral labeling task is potentially vague even for humans. This means that the classifier we train to distinguish between social behaviors is limited by the accuracy of our training data.

In order to create an automated social phenotyping system, we need to address these limitations.

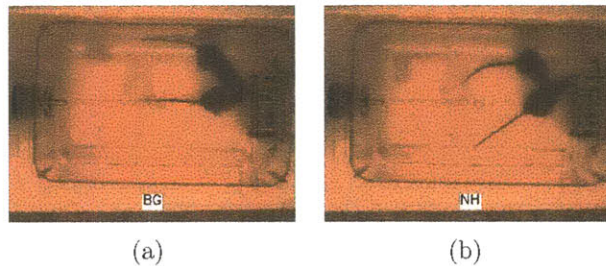


Figure A-1: Hard to distinguish behaviors from top view. The left figure shows non-interacting mice. The right figure shows interacting mice (nose-to-head interaction). Using just top view of one frame, the differences are very subtle, making it hard to distinguish the mice's behavior.



# Appendix B

## Evaluation of Human Labeling

Nick Edelman’s database has twelve videos labeled with the following social behaviors: nose-to-nose contact, nose-to-anogenital contact, nose-to-head contact, crawl over crawl under, and upright. These twelve videos were labeled by an expert in the field, who has much experience with mice and their behavior. This was compared to performance by a non-expert. The non-expert was given definitions of the behaviors and observed four ten minute videos of labeled examples in order to understand how the labels were assigned. The non-expert was asked to use both the top and side view. Overall agreement was 87.16% between the expert and non-expert. The majority of the time the two mice are not interacting, thus, a more meaningful measure of agreement, can be examined by looking at the confusion matrix, Figure B-1. Agreement is high for nose-to-head and nose-to-nose behavior, but much lower for other behaviors (note that crawl over crawl under and upright had very few example frames, making it difficult to obtain meaningful measurements of agreement). The expert observed 27 social contact events (defined as a continuous labeling of a social behavior), while the non-expert observed 79. More investigation is necessary in order to understand these differences. If the differences are due to the inherent ambiguity, then we have a ballpark figure of what is acceptable performance from an automated system. On

the other hand, if the non-expert has lower quality labeling, more investigation is needed to evaluate why the problem is difficult for humans. This might help improve our algorithms for automated behavior recognition. Similarly, if labels by the expert are found to be low quality, investing more time in obtaining higher quality labeling might be necessary.

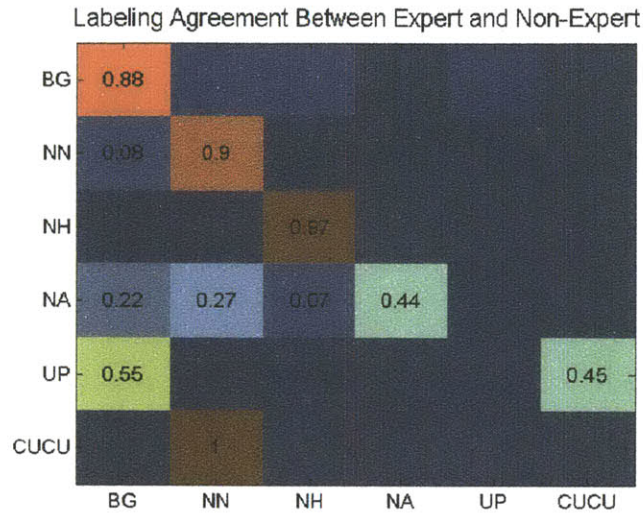


Figure B-1: Confusion matrix between two labelers. Each entry  $(x, y)$  in the confusion matrix is the probability with which an instance of a behavior  $x$  (along rows) is classified as type  $y$  (along column), and which is computed as  $(\text{number of frames annotated as type } x \text{ and classified as type } y) / (\text{number of frames annotated as type } x)$ . As a result, values sum to a value of 1 in each row. Entries with values less than 0.05 are not shown. The color bar indicates the percentage agreement.



# Appendix C

## Extensions to Other Domains

In this section we discuss the extension of our system to three different domains. We consider the extension of our part-based tracker to both side and semi-side views. We also discuss extending our tracker to three mice in a cage.

### C.1 Side View

The side view is significantly harder to study than the top view due to the fact that mice occlude each other much more often from the side view than the top view, as can be seen in Figure C-1.

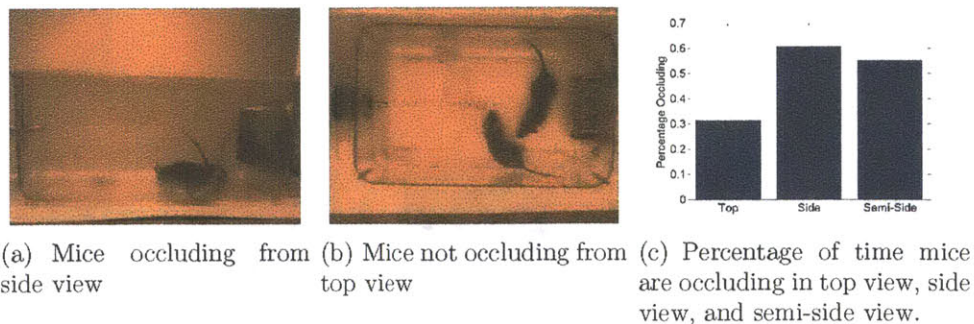


Figure C-1: Mice occlude from side view much more than from top view.

To analyze performance on side view, we will use the same dataset and sampling

technique from Chapter 4, sampling fifty frames from each of the twelve videos in the dataset. We focus on performance of non-occluding frames, since the system cannot be easily extended for side-view occlusions. Side view occlusions are far more complex than top-view occlusions. Unlike for top view, the combined shape of two mice occluding from the side view cannot be easily defined as a combination of two side view non-occluding mice.

The mice can acquire many more shapes from the side view, as seen in Figure C-2. For this reason, increasing the number of templates from one to ten, is necessary - see Figure C-3 for templates. These templates are selected manually. To find the head and tail, we match each frame with each of the ten templates, and choose the template minimizing the matching cost. We then apply the technique of Section 3.6 to find the head and tail locations of the mice.

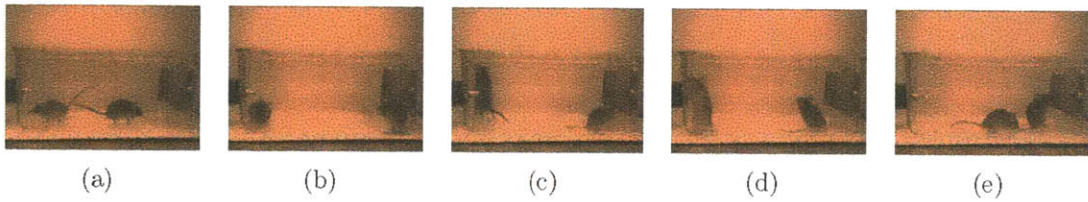


Figure C-2: Mice can take on many contour forms from the side view (a-e).

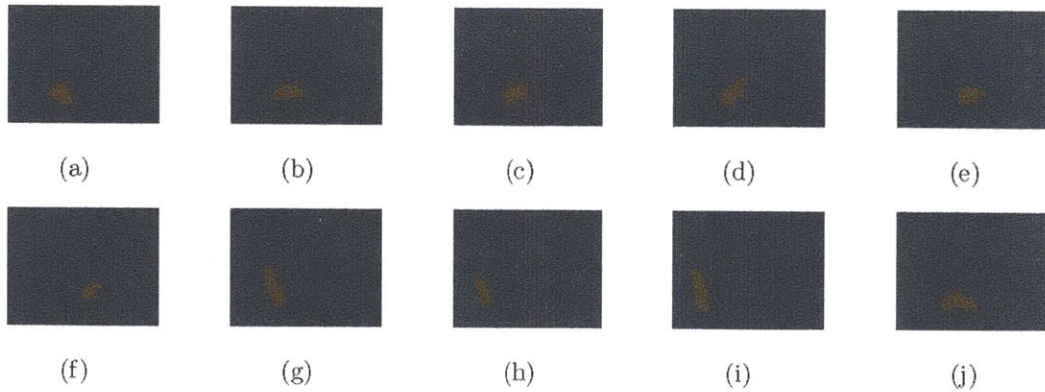


Figure C-3: The templates used for side view matching (a-j).

Tracking and orientation scores for top and side view in Dataset A are shown in Table C.1. We see that performance decreases as compared with top view. One possible reason for this decrease is that in the side view it is much harder to obtain a good contour of the mice. Additionally, the mice are occluded by the feeder and drinking tube much more often, and it is much harder to interpret ways. Lastly, since the mouse’s underside is light colored, when the underside is visible - such as when the mouse rears towards the camera - the foreground extracted is of bad quality. This partly explains the decrease in performance, while the other part is explained by the fact that the problem is significantly harder to solve in the side view. For example, when the mouse faces the camera, the tail is not visible, therefore its location is unclear. Despite performing poorer than the top view tracker, these results are useful in some scenarios. For example, we can use the good matches to aid in phenotyping.

	Top View			Side View		
	<b>1</b>	<b>2</b>	<b>3</b>	<b>1</b>	<b>2</b>	<b>3</b>
<b>Non-Occluding Orientation</b>	94.33%	.08%	5.58%	54.5%	1.45%	44.0%
<b>Non-Occluding Tracking</b>	93.42%	5.38%	1.21%	45.22%	40.09%	14.68%

Table C.1: Top and side view orientation and tracking results.

## C.2 Semi-Side View

Videos taken from a semi-side view (between a top and side view, Figure C-4) provide more fine-grained information than the top view, while having less occlusions and complex interferences than the side view (Figure C-1(c)).

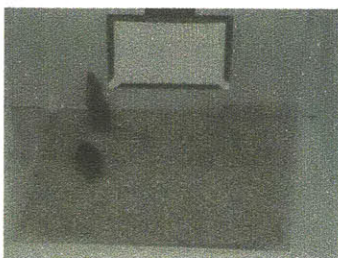


Figure C-4: Example of semi-side view frame.

We analyze our tracker performance on one semi-side view video which is longer than fifteen minutes. The video is taken in non-ideal conditions, thus we evaluate only frames for which the mice are selected as foreground objects, and not those frames in which only noise is selected. In addition, we use the side templates as they describe the mouse shape well. Results are based on a randomly sampled 200 non-occluding frames and are summarized in Table C.2.

	Semi-Side Video		
	<b>1</b>	<b>2</b>	<b>3</b>
<b>Non-Occluding Orientation Score</b>	78.75%	1.25%	20.00%
<b>Non-Occluding Tracking Score</b>	65.75%	31.00%	32.5%

Table C.2: Semi-side view orientation and tracking results.

Results from semi-side view are significantly better than from side view, but not as good as top view results. We propose using semi-side view as a basis for filming the mouse from multiple angles, and later combining the different view points intelligently. Using our part-based tracker, we can analyze semi-side view videos reasonably well.

### C.3 Tracking Three Mice

The extension to three mice is nontrivial. The template generation technique does not easily extend to more than two mice, thus a different template generation method would have to be used. Furthermore, when three mice interact, compared to two, the

contour does not provide as much information about each individual mouse. Figure C-5 shows examples of three mice interacting simultaneously.

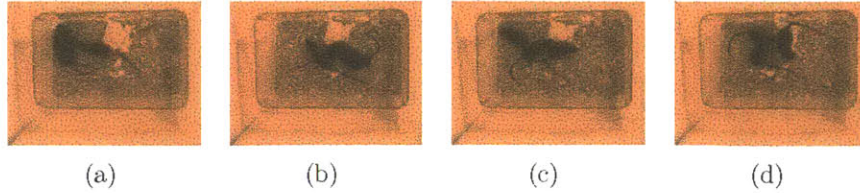


Figure C-5: Various example of three mice shown from the top view.

The complexity of the interaction makes it difficult to interpret the scene just based on the combined contour. However, when three mice are housed together in a single cage, most of the social interactions occur between only two of the mice. When only two mice are interacting, our tracker can be used to find the head and tail location of the interacting mice.

# Appendix D

## Future Work

We have described the first steps in automating social behavioral phenotyping of mice. Possible extensions to this work include: extending and improving our tracker, creating an end to end mouse behavioral phenotyping system, and applying our current work to new applications.

### D.1 Improving the Tracker

The part-based mouse tracker works extremely well during non-occlusion frames. It does not perform as well during more complicated occlusions, and can misidentify mice, leaving room for improvement. Many applications of mouse tracking require tracking mouse identity with high accuracy. For example, in studying how a knockout mouse - such as a Shank3 mouse - behaves towards a wild-type mouse, the identification of the mice at all times is extremely important. Methods such as ear punches, tattooing, or tail marking are used to keep track of mouse identity. Unfortunately, these methods are either too fine to be easily captured by video and identified by a computer system (ear punches, and tail markings) or not extendable to all mice (tattooing, which works best on light colored mice). As we saw in Chapter 5, keeping track of mouse identity is occasionally a problem for our tracker. One possible solu-

tion, which has been used by Santoso et al., is to insert radio-frequency identification devices (RFID) transponders in each mouse, allowing individual identification of the mice [21]. This enables the system to reaffirm the mouse’s identity and self correct based on the transponder information.

One limitation of our training templates is that it assumes that the interaction between two mice can be defined as an overlay of two mice one on top of another. There are three major problems with this assumption. First, this assumes there are no shadows or other artifacts in our video. When shadows are present in the video apparent occluding interactions can occur. These interactions confuse the system and cannot be described by simply overlaying two mouse contours, as we do with our templates. This is due to the fact that the shadow between the two mice is considered to be part of the foreground, and thus part of the mouse contour. Second, mice are extremely deformable, meaning we cannot define them as a simple combination of two templates of un-deformed mice. Lastly, mice live in three dimensional space, meaning when they protrude up in the air, their contour changes around the head and tail, making a match with a training template difficult. Additionally, one mouse may appear smaller than the other, which does not exist in our occlusion training templates. All of these issues can be addressed by using training data derived from real data. However, one setback of this method is that it involves labeling the head and tail locations of real mice, which is time consuming. This direction needs to be explored more thoroughly.

We can improve our tracker by using multiple cameras, all from different angles. We can select the most reliable camera [18], or alternatively, we can combine the textureless mouse information from multiple cameras to create a three dimensional model [29].

A infrared camera, such as the Kinect, can also be used to obtain depth information. Unfortunately, the Kinect is only accurate to a few centimeters [22], but new

cameras are being developed that provide greater accuracy [24]. The cage walls would need to be covered, as the material of the cage walls may interfere with the infrared signal.

## **D.2 Using Part-Based Tracking for Social Behavioral Phenotyping**

Work needs to be done before we can create a reliable end-to-end system that takes in a video of the mouse and outputs behavioral labeling for every frame. To create this system two approaches can be taken. First, we can take a machine learning approach - as attempted in Appendix A, where the system takes as input labeled frames and learns rules from them. One setback is that our current training data is mostly labeled as background, making it very difficult for a system to learn meaningful rules. One possible solution is to label more categories in the data. For instance, instead of labeling just nose-to-tail behavior, we can add a follow behavior. Forming rules for identification of behaviors would be easier under this scenario. Alternatively, we can define our behaviors based on hard coded rules, as is done by [5]. This method has the advantage of making new behaviors extremely easy to study and eliminates the need for training data. However, further investigation is needed to confirm that these hard coded rules are good representations of mice phenotypes.

## **D.3 Fine-Grained Social and Single Mouse Behavioral Phenotyping**

Interest lies in studying either fine-grained social behavior, such as social grooming, or single mouse behaviors, such as eating, resting, or grooming. Our tracking system cannot easily extend to define fine-grained behaviors. We can potentially use the



tracking information to narrow the feature selection space. For example, for social grooming, looking at local features only near one of the mouse’s head can help define the behavior more accurately. Additionally, since the mice are not constantly occluding one another, creating a single mouse behavioral recognition system that can identify a single mouse’s behavior (eating, drinking, resting, walking, etc.) during non-occlusions frames is realistic.

## D.4 Applications

Scientists explore a variety of behavioral phenotypes when studying mice. Our non-occlusion highly accurate parts tracker can be applied beyond social behaviors. For example, Noldus’ Ethovision has an object exploration task, where the time exploring an object is measured. Our part-based tracker would perform accurately in this context.

Using our head and tail locator we can extend our original definition of head and tail from the single point of the nose location and the single point of the base of the tail location to a more general description of the head and tail areas. This is done by looking at the contour points before and after the precise head and tail points and connecting them in a polygon. We could then study features such as head and tail orientations, which might be of interest.

# Appendix E

## Algorithms

---

**Algorithm 2** Overview of tracking algorithm.

---

```
bg ← generateBg(videoFn);  
isOcclusion ← false;  
while frame = getNextFrame(videoFn) do  
    % Section 3.4  
    fg ← computeFg(frame, bg);  
    % Section 3.5  
    isOcclusion ← computeOcclusion(fg);  
    % Section 3.6  
    [head1, tail1, head2, tail2] ← matchHeadTail(fg, isOcclusion);  
end while  
    % Section 3.7  
    [head1_list, tail1_list, head2_list, tail2_list] ←  
    svmCorrection(head1_list, tail1_list, head2_list, tail2_list)
```

---

# Bibliography

- [1] T. Balch, F. Dellaert, A. Feldman, A. Guillory, Isbell C.L, Z. Khan, S.C. Pratt, A.N. Stein, and H. Wilde. How multirobot systems research will accelerate our understanding of social animal behavior. *Proceedings of the IEEE*, 94(7):1445–1463, july 2006.
- [2] S. Belongie, J. Malik, and J. Puzicha. Shape context: A new descriptor for shape matching and object recognition. In *In NIPS*, pages 831–837, 2000.
- [3] D. C. Blanchard, E. B. Defensor, K. Z. Meyza, R. L.H. Pobbe, B. L. Pearson, V. J. Bolivar, and R. J. Blanchard. Btbr t+tf/j mice: Autism-relevant behaviors and reduced fractone-associated heparan sulfate. *Neuroscience & Biobehavioral Reviews*, 36(1):285 – 296, 2012.
- [4] H. Dankert, L. Wang, E.D. Hoopfer, D.J. Anderson, and P. Perona. Automated monitoring and analysis of social behavior in drosophila. *Nature Methods*, 6:297–303, 2009.
- [5] F. de Chaumont, R. D-S. Coura, P. Serreau, A. Cressant, J. Chabout, S. Granon, and J.C. Olivo-Marin. Computerized video analysis of social interactions in mice. *Nature Methods*, 9:410–417, 2012.
- [6] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *Visual Surveillance and Performance Evalua-*

- tion of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*, pages 65 – 72, oct. 2005.
- [7] N. Edelman. Automating phenotyping of mouse social behavior. Master’s project, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, 2011.
- [8] T. Gatzke, C. Grimm, M. Garland, and S. Zelinka. Curvature maps for local shape comparison. In *Proceedings of the International Conference on Shape Modeling and Applications 2005, SMI ’05*, pages 246–255, Washington, DC, USA, 2005. IEEE Computer Society.
- [9] E. H. Goulding, K. A. Schenk, P. Juneja, A. W. MacKay, J. M. Wade, and L. H. Tecott. A robust automated system elucidates mouse home cage behavioral structure. *Proceedings of the National Academy of Sciences*, 105(52):20575–20582, 2008.
- [10] V. Jain and H. Zhang. Robust 2D shape correspondence using geodesic shape context. In *Proc. of Pacific Graphics*, page 121–124, 2005.
- [11] H. Jhuang, E. Garrote, X. Yu, V. Khilani, T. Poggio, A. Steele, and T. Serre. Automated home-cage behavioural phenotyping of mice. *Nature Communications*, 1:68, 2010.
- [12] A. E. Johnson and M. Hebert. Recognizing objects by matching oriented points. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (May):684–689, 1997.
- [13] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(5):433–449, May 1999.

- [14] H. Ling. Code and data. [http://www.dabi.temple.edu/~hbling/code\\_data.htm](http://www.dabi.temple.edu/~hbling/code_data.htm). [Online; accessed June-2012].
- [15] H. Ling and D. W. Jacobs. Using the inner-distance for classification of articulated shapes. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 719–726, 2005.
- [16] H. Bischof M. Maurer, M. Ruther and G. Kastberger. In *Rigid body reconstruction for motion analysis of giant honey bees using stereo vision. In Proc. Visual Observation and Analysis of Animal and Insect Behavior (VAIB'10), ICPR Workshop, 2010.*, 2010.
- [17] G. Mori, S. Belongie, and J. Malik. Shape contexts enable efficient retrieval of similar shapes. In *CVPR*, December 2001.
- [18] N. T. Nguyen, S. Venkatesh, G. West, and H. H. Bui. Multiple camera coordination in a surveillance system. *ACTA Automatica Sinica*, 29:408–422, 2003.
- [19] J. Peca, C. Feliciano, J. Ting, W. Wang, M. Wells, T. Venkatramann, C. Lascola, Z. Fu, and G. Feng. Shank3 mutant mice display autistic-like behaviours and striatal dysfunction. *Nature*, 472:437–442, 2010.
- [20] S. D. Pelkowski, M. Kapoor, H. A. Richendrfer, X. Wang, R.M. Colwill, and R. Creton. A novel high-throughput imaging system for automated analyses of avoidance behavior in zebrafish larvae. *Behavioural Brain Research*, 223(1):135–144, 2011.
- [21] A. Santoso, A. Kaiser, and Y. Winter. Individually dosed oral drug administration to socially-living transponder-tagged mice by a water dispenser under rfid control. *Journal of Neuroscience Methods*, 153(2):208 – 213, 2006.

- [22] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-Time Human Pose Recognition in Parts from Single Depth Images. June 2011.
- [23] A.J Spink, R.A.J Tegelenbosch, M.O.S Buma, and L.P.J.J Noldus. The ethovision video tracking systema tool for behavioral phenotyping of transgenic mice. *Physiology & Behavior*, 73(5):731 – 744, 2001.
- [24] B. Sterling. Augmented reality: Leap motion @ONLINE. [http://www.wired.com/beyond\\_the\\_beyond/2012/05/augmented-reality-leap-motion/](http://www.wired.com/beyond_the_beyond/2012/05/augmented-reality-leap-motion/), May 2012.
- [25] P. Tamborini, H. Sigg, and G. Zbinden. Quantitative analysis of rat activity in the home cage by infrared monitoring. application to the acute toxicity testing of acetanilide and phenylmercuric acetate. *Archives of Toxicology*, 63:85–96, 1989.
- [26] L. H. Tecott and E. J. Nestler. Neurobehavioral assement in the information age. *Nature Neuroscience*, 7(5):462–466, 2004. This is a full ARTICLE entry.
- [27] R.C. Veltkamp and M. Hagedoorn. State-of-the-art in shape matching. Technical Report UU-CS-1999-27, Department of Information and Computing Sciences, Utrecht University, 1999.
- [28] Wikipedia. Shape context — Wikipedia, the free encyclopedia. [http://en.wikipedia.org/wiki/Shape\\_context](http://en.wikipedia.org/wiki/Shape_context), 2012. [Online; accessed June-2012].
- [29] R. Yang, M. Pollefeys, and G. Welch. Dealing with textureless regions and specular highlights—a progressive space carving scheme using a novel photo-consistency measure. In *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2*, ICCV '03, pages 576–, Washington, DC, USA, 2003. IEEE Computer Society.